Embedded Rate Scalable Wavelet-Based Image Coding Algorithm with RPSWS

Farag I. Y. Elnagahy
Telecommunications
Faculty of Electrical Engineering
Czech Technical University in Prague
16627, Praha 6, Czech Republic
faragelnagahy@hotmail.com

Boris Šimák
Telecommunications
Faculty of Electrical Engineering
Czech Technical University in Prague
16627, Praha 6, Czech Republic
simak@feld.cvut.cz

ABSTRACT

This paper proposes an embedded rate scalable wavelet-based image coding algorithm. We introduce a simple and efficient approach for coding the positions and the signs of the wavelet coefficients that will be transmitted as nonzero values in an embedded scheme. This algorithm is based on recursive partitioning of the significant wavelet sub-bands (RPSWS). The proposed algorithm produces a fully embedded bit stream and it encodes the image to exactly the desired bit rate (precise rate control). Some standard test images were used to verify the correctness of the proposed coding algorithm. Their results were compared to the two well-known embedded image coders: embedded zerotree wavelet coder (EZW) and set partitioning in hierarchal trees coder (SPIHT). These two techniques make use of "spatial orientation trees" (SOT). Spatial orientation trees are structures that use quad-tree representations of wavelet coefficients that belong to different sub-bands, but have the same spatial location.

Keywords

Image compression, Scalable, Embedded coding, Wavelet sub-band partitioning.

1. INTRODUCTION

Embedded coding is very similar to progressive transmission. The decoder receives the compressed data from the beginning of the bit stream up to the point where a selected data rate requirement is achieved. A decompressed image at that data rate can then be reconstructed and the visual quality corresponding to this data rate can be achieved. Thus, to achieve the best performance the bits that convey the most important information need to be embedded at the beginning of the compressed bit stream [She99].

Embedded image coding is important in many applications such as: progressive image transmission, Internet browsing, scalable image and video coding, digital cameras, and low delay image communication.

Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972
WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.

Wavelet transforms have proven to be very powerful tools for image coding. Many state-of-the-art image coders [And03; Sha93; Sai96; Sod99; Tau00; Xio97;] employ a wavelet transform in their algorithms. Ideally the image energy is compacted into a small fraction of the transform coefficients and compression can be archived by coding these coefficients. One advantage is the provision of both frequency and spatial localization of the image energy. Wavelet transform coefficients are defined by three parameters. These parameters are the magnitude, the position, and the sign. Hence the position information of the wavelet coefficients along with the magnitude and sign information must be encoded efficiently.

In order to perform embedded rate scalable image compression in the wavelet transform domain, an initial threshold is first determined. All wavelet coefficients (significant coefficients) that have an absolute value above or equal to the threshold are coded and refined to an additional bit of precision. To code the remaining coefficients (insignificant coefficients), the threshold is halved so that some of these coefficients become significant coefficients relative to the new threshold. The process continues until the desired bit rate is achieved.

The proposed coding algorithm is a modified version of the algorithms published in [Eln03a; Eln03b]. In [Eln03a], the wavelet sub-band is coded by partitioning the sub-band into four quarters and these quarters are further sub-divided into blocks. Where in [Eln03b], the same coding algorithm was used and was implemented with two different strategies. In the first strategy, the significance map pass for the sub-band is coded and is followed by the refinement pass. The second strategy is used in this paper and is presented in Section 2.

While in the proposed coding algorithm, the sub-band is divided into four blocks and each block is then subdivided into four sub-blocks, the block division process continues until the last block contains four coefficients or less, depending on the dimensions of the sub-band.

The paper is organized as follows; a short background about the Wavelet transform is presented in section 2. The general structure of the proposed coding algorithm will be presented in section 3. The significance map pass is presented in section 4, while the magnitude refinement pass is presented in section 5. A simple coding example is given in section 6. Section 7 discusses experimental results for various rates and for various standard test images. The paper concludes with Section 8.

2. WAVELET TRANSFORM

Discrete wavelet transform corresponds to two sets of analysis/synthesis digital filters, g/~g and h/~h, where h is a low pass filter (LPF) and g is a high pass filter (HPF). In two dimensions filtering is usually applied in both horizontally and vertically. Filtering in one direction, results in decomposing the image in two components. The total number of components we have after vertical and horizontal decompositions is four. We will refer to these components as image sub-bands, LL, HL, LH, and HH. This is the first level of the wavelet transform. One of the four sub-bands, the LL sub-band, will contain low pass information, which is essentially a low resolution version of the image. The operations can be repeated on the low-low (LL) sub-band for some number of levels, D, producing a total of 3D+1 sub-bands whose samples represent the original image.

Wavelet coefficients are usually real numbers, therefore they require a significant number of bits in their representation, and do not directly offer any form of compression. The aim of this paper is to introduce an algorithm to code these coefficients efficiently in an embedded way.

3. CODING ALGORITHM

The discrete wavelet transform is used to decompose the original image into sub-bands. Each sub-band is encoded and decoded separately in the proposed algorithm. The sub-bands scanning order is shown in Fig.1. For an N-scale transform, the scan begins at the lowest frequency sub-band, denoted as LL_N, and scans sub-bands HL_N, LH_N, and HH_N, at which point it moves on to scale N-1, etc. Two passes are used to encode and decode each sub-band. A significance map pass is used to code the position and the sign of the significant wavelet coefficients. Successive approximation quantization (SAQ) [Sha93] is used in the magnitude refinement pass to refine the magnitudes of the previously coded coefficients.

The general structure of the proposed coding algorithm is as follows:

The initial threshold T_0 is chosen so that $T_0=2^i$, where i is computed in such a way that $2^{i+1} > x > 2^i$ and x is the maximum absolute value of the wavelet coefficients.

```
T= T_0
While the desired bit rate does not achieved Do
    For each sub-band Do
        Significance map pass
    End For
    For each sub-band Do
        Magnitude refinement pass
    End For
    T=T/2
End while
```

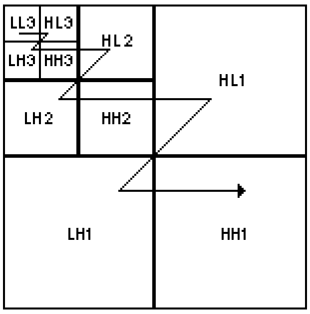

Figure 1 Sub-bands scanning order

4. SIGNIFICANCE MAP PASS

A wavelet coefficient that has an absolute value above or equal to the threshold is called a significant coefficient; otherwise it is an insignificant coefficient. The insignificant wavelet coefficient is coded with one bit "0" while the significant wavelet coefficient is coded with two bits. The first bit "1" indicates the significance of the wavelet coefficient and the second bit indicates its sign ("0" for positive and "1" for negative coefficient). A sub-band that has no significant wavelet coefficient is called an insignificant sub-band. The insignificant sub-band is entirely coded with one bit. This means that a large number of insignificant coefficients are coded with one symbol (one bit). Coding a large number of insignificant wavelet coefficients with one bit is the main idea of the proposed coding algorithm. The encoder sends one bit to inform the decoder about the significance of the sub-band. This bit is called the sub-band significance flag bit (SSF), for insignificance sub-band SSF is equal to zero. When the decoder receives SSF=0 it knows that the sub-band is insignificant and hence all coefficients in that band are insignificant. The decoder sets all coefficients reconstructed values in that sub-band to zero. In other words all wavelet coefficients in that band are quantized to zero.

For a significant sub-band, the encoder outputs SSF=1 and the decoder knows that this sub-band contains significant coefficient(s). In this case both the encoder and the decoder partition the sub-band into four blocks and the significance test is then applied to each block. An insignificant block is also coded with one bit “0”. This bit is called the block significance flag bit (BSF), while the significant block is coded with BSF=1 and this block is again sub-divided into four new blocks. This block division process continues until the last block contains four coefficients or less, depending on the dimensions of the sub-band.

The encoder creates a list to store information about the significant wavelet coefficients. This list is called the coefficients significance list (CSL). Each time the encoder finds a significant wavelet coefficient it appends the wavelet coefficient value to the coefficients significance list. The encoder then sets the value of the significant wavelet coefficient to zero in the wavelet coefficients matrix. The coefficients significance list is used during the magnitude refinement pass. The sub-band significance map coding process mentioned above is implemented in two steps. The fist step is called sub-band Or-tree significance map assignment and the second is sub-band Or-tree significance map coding. These two steps are presented in the following subsections.

4.1 Sub-band Or-Tree significance map assignment

Recursive partitioning of the significant wavelet sub-band can be represented by a significance map tree. The significance map tree consists of a root node that represents the significance of the sub-band. The root node is significant if it has SSF=1 and it is insignificant if it has SSF=0. The root node has four child nodes; these child nodes indicate the significance of the four blocks that construct the entire sub-band. Generally, each node in the tree has four nodes except the leaf nodes. Leaf nodes represent the significance of the wavelet coefficients. Leaf nodes may be four wavelet coefficients or less, depending on the dimensions of the image to be compressed. The sub-band significance tree is an OR-tree; this means that if one of the child nodes is significant then the parent node is significant, while the parent node is insignificant when all child nodes are insignificant. The node significance assignment process is begun from the bottom of the tree (wavelet coefficients) up to the root node. If all wavelet coefficients are insignificant with respect to the threshold T then the parent node of these wavelet coefficients is assigned a “0” value. The value of the node (significance value) is set to “1” if at least one of its child nodes has a value “1”.

4.2 Sub-band Or-Tree significance map coding

The coding of the sub-band Or-tree significance map starts from the root node in depth first search manner. First the significance flag of the root node is sent to the decoder. When the significance flag is equal to zero (SSF=0), there is no need to send additional bits to the decoder. If the root node is significant (SSF=1), the encoder outputs “1” and pushes its child nodes to the top of the stack. Then the stack’s top node is popped and its significance bit is sent to the decoder. The child nodes are pushed to the stack when the node significance bit is equal to “1”. If the top node is a leaf node, the significance bit is sent to the decoder in addition to the sign bit. The process continues as long as the stack is still contains nodes. The following algorithm summarizes the sub-band Or-tree significance map coding:

```
If root node is insignificant Then
  output “0”
Else
  output ”1”
  Push all children on S
  While S is not empty Do
    Z=Top(s)
    Pop Top(s)
    If Z is insignificant Then
      output “0”
    Else
      output “1”
      If Z is a node Then
        Push all children on S
      Else
        output Sign (“0” for positive, “1” for negative)
        add coefficient value to CSL
      End If
    End If
  End While
End If
```

Where S; is an auxiliary stack.

5. MAGNITUDE REFINEMENT PASS

In the magnitude refinement pass, the previously coded significant coefficients are refined to an additional bit of precision. The following algorithm summarizes the magnitude refinement pass:

```
For each coefficient in CSL Do
    If coefficient value >= 2T Then
        Ouput "(coefficient value And T/2)"
    End If
End For
```

An additional task done by the decoder is to update the reconstructed wavelet coefficients. For an insignificant coefficient, the constructed value is set to zero, while the reconstructed value is set to 1.5 T, -1.5T for significant positive and negative coefficient, respectively. Similarly, during the magnitude refinement pass the decoder adds 0.5T to the reconstructed value if the refine bit received is "1". The decoder adds -0.5T to the reconstructed value if the refine bit received is "0".

6. CODING EXAMPLE

In this section, a simple example will be used to highlight the order used in the proposed coding algorithm. Consider an 8x8 sub-band with the significance map as shown in Fig. 2. The coefficients labeled "0" are insignificant coefficients, while coefficients labeled a, b, c, and d are significant coefficients with respect to the threshold T. The significance map Or-tree for that sub-band is shown in Fig. 3. The node labeled "1" in Fig. 3 is the root node of the Or-tree significance map of the sub-band shown in Fig. 2, and it represents the significance of the sub-band (SSF=1). The nodes labeled 2, 3, 4, and 5 are the child nodes of root node 1. Nodes 2, 3, 4, and 5 represent the significance of the 4x4 left-top block, 4x4 right-top block, 4x4 left-bottom block, and 4x4 right-bottom block in Fig. 2, respectively.

The encoded bit stream of that sub-band is given in Table 1. The first step to encode that sub-band is to construct its significance map tree, as shown in Fig. 3. As mentioned before, the node significance assignment process is begun from the bottom of the tree (wavelet coefficients) up to the root node. Encoding the significance map tree is the second step.

The first bit to be sent to the decoder is the value of SSF. In case of SSF=0, there is no need to send additional bits to the decoder. In our example SSF=1, which means that the sub-band contains at least one significant coefficient. Additional bits are needed to encode the positions and the signs of these coefficients. These bits are the bit stream for node 2, node 3, node 4, and node 5, respectively.

The bit stream of node 2 consists of the value of the node (BSF=1 in this case) and the bit stream of the remaining sub-tree below it ([1 1S_a 0 0 0] 0 0 0). The first bit "1" indicates the significance of the first child, and it is followed by the bit stream of the four leaves. The two bits "1S_a" indicate the significance of the first coefficient "a". The three bits follow the sign bit (S_a) corresponding to the last three insignificant coefficients. The last three zeros of the node 2 bit stream indicate the insignificances of its remaining children.

The bit stream of node 3 consists of one bit only, because this node is an insignificant node. The bit stream for nodes 4 and 5 is shown in Table 1. The absolute magnitude values of coefficients a, b, c, and d are appended to the coefficients significance list (CSL) to be refined in the magnitude refinement pass.

a	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0
c	0	0	0	D	0	0	0

Figure 2 Example of an 8x8 sub-band.

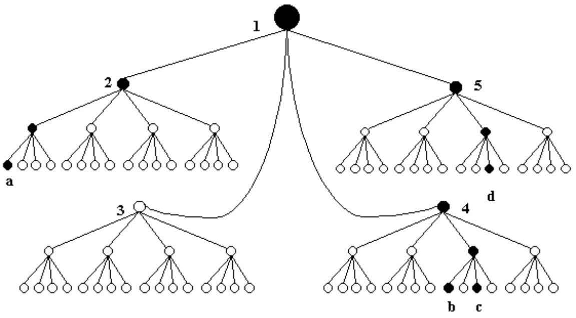

Figure 3 Significance map Or-tree for the sub-band in figure 2; black circle represents a significant node while white circle represents an insignificance node.

Node #	Encoded bit stream
1	1
2	1 [1 1S_a 0 0 0] 0 0 0
3	0
4	1 0 0 [1 1S_b 0 1 S_c 0] 0
5	1 0 0 [1 0 0 1 S_d 0] 0

Table 1 Sub-band encoded bit stream

7. EXPERIMENTAL RESULTS

All experiments were performed by encoding and decoding an actual bit stream to verify the correctness of the proposed algorithm. The proposed algorithm was applied to the standard gray 8 bits per pixel (bpp) test images, 512x512 Lena, Barbara, and Goldhill. These images were obtained from (http://www.icsl.ucla.edu/~ipl/psnr_images.htm). We used 6-level pyramids constructed with 9/7-tap filters of [Ant92], and using a reflection extension at the image edges. The bit stream of the proposed coding algorithm was not entropy coded. EZW [Sha93] results were obtained by applying the EZW software (http://pesona. mmu.edu.my/~msng/EZW.html) on these images with the same level and filters. The EZW entire bit stream is arithmetically encoded using a single arithmetic coder with an adaptive model [Wit89].

SPIHT [Sai96] results are obtained by applying the SPIHT software (http://www.cipr.rpi.edu/research/SPIHT/spiht3.html) on these images and using the adaptive arithmetic coding algorithm of Witten et al. [Wit89]. The coding results for the three coders are summarized in Table 2. Where the PSNR is obtained as: PSNR=10 $\log_{10}$ (255^2/MSE) dB, where MSE is the mean square error between the reconstructed and original images. The compression ratio in the experiment is chosen to be in the range from 8:1 (1.0 bpp) to 1024:1 (0.0078125 bpp). Since all three coders are embedded coders, the coding can stopped at the exact bit rate. From the results in Table 2 we see that the performance of SPIHT outperforms both the proposed coding algorithm (RPSWS) and EZW, while the performance of the proposed coding algorithm surpasses the EZW. Some reconstructed images at different bit rates for both SPIHT and RPSWS are shown in Fig. 4, Fig. 5, and Fig. 6.

Figure 4 shows a comparison of SPIHT and RPSWS operating on the "Lena" image. Figure 5 and Fig. 6 show a comparison of SPIHT and RPSWS operating on "Barbara" and "Goldhill", respectively. From these figures we note that there is no visual difference between the SPIHT and RPSWS reconstructed images.

8. CONCLUSION

In this paper, an embedded rate scalable wavelet-based image coding algorithm (RPSWS) is presented. This algorithm is based on recursive partitioning of the significant wavelet sub-bands. The proposed algorithm produces a fully embedded bit stream and it encodes the image to exactly the desired bit rate (precise rate control). The proposed algorithm can be used in the following applications: progressive image transmission, Internet browsing, scalable image and video coding, digital cameras, and low delay image communication The performance of the proposed coding algorithm can be improved by using an efficient sign coding and estimation of zero-quantized coefficients, as presented in [Dee03]. Also, the resulting bit stream should be entropy coded. RPSWS could be extended to handle color images.

<table>
<tr><th colspan="4">Goldhill (512 x 512)</th></tr>
<tr><th>Bit Rate</th><th>SPIHT</th><th>EWZ</th><th>RPSWS</th></tr>
<tr><td>1</td><td>36.55</td><td>35.59</td><td>35.68</td></tr>
<tr><td>0.5</td><td>33.13</td><td>32.59</td><td>32.46</td></tr>
<tr><td>0.25</td><td>30.56</td><td>30.28</td><td>30.20</td></tr>
<tr><td>0.125</td><td>28.48</td><td>28.04</td><td>28.15</td></tr>
<tr><td>0.0625</td><td>26.73</td><td>26.43</td><td>26.50</td></tr>
<tr><td>0.03125</td><td>25.27</td><td>25.12</td><td>25.12</td></tr>
<tr><td>0.015625</td><td>23.94</td><td>23.60</td><td>23.84</td></tr>
<tr><td>0.0078125</td><td>22.63</td><td>22.21</td><td>22.48</td></tr>
<tr><td>Average</td><td>28.41</td><td>27.98</td><td>28.05</td></tr>
<tr><th colspan="4">Lena (512 x 512)</th></tr>
<tr><th>Bit Rate</th><th>SPIHT</th><th>EWZ</th><th>RPSWS</th></tr>
<tr><td>1</td><td>40.41</td><td>39.58</td><td>39.68</td></tr>
<tr><td>0.5</td><td>37.21</td><td>36.42</td><td>36.55</td></tr>
<tr><td>0.25</td><td>34.11</td><td>33.34</td><td>33.43</td></tr>
<tr><td>0.125</td><td>31.10</td><td>30.41</td><td>30.52</td></tr>
<tr><td>0.0625</td><td>28.38</td><td>27.78</td><td>27.91</td></tr>
<tr><td>0.03125</td><td>25.97</td><td>25.71</td><td>25.67</td></tr>
<tr><td>0.015625</td><td>23.97</td><td>23.61</td><td>23.71</td></tr>
<tr><td>0.0078125</td><td>22.08</td><td>21.72</td><td>22.04</td></tr>
<tr><td>Average</td><td>30.40</td><td>29.82</td><td>29.94</td></tr>
<tr><th colspan="4">Barbara (512 x 512)</th></tr>
<tr><th>Bit Rate</th><th>SPIHT</th><th>EWZ</th><th>RPSWS</th></tr>
<tr><td>1</td><td>36.41</td><td>35.09</td><td>35.61</td></tr>
<tr><td>0.5</td><td>31.40</td><td>30.47</td><td>30.84</td></tr>
<tr><td>0.25</td><td>27.58</td><td>26.64</td><td>27.00</td></tr>
<tr><td>0.125</td><td>24.86</td><td>24.09</td><td>24.42</td></tr>
<tr><td>0.0625</td><td>23.35</td><td>23.09</td><td>23.17</td></tr>
<tr><td>0.03125</td><td>22.24</td><td>21.95</td><td>22.10</td></tr>
<tr><td>0.015625</td><td>21.03</td><td>20.88</td><td>20.87</td></tr>
<tr><td>0.0078125</td><td>19.80</td><td>19.42</td><td>19.64</td></tr>
<tr><td>Average</td><td>26.70</td><td>26.03</td><td>26.29</td></tr>
</table>

Table 2 PSNR results, measured in dB, for various images and bit-rates.

Coding delay is produced when coding the sub-bands with large size, especially the highest frequency sub-bands. Dividing these sub-bands into blocks and coding these blocks separately can reduce the coding delay. The complexity of the encoder and the decoder in term of computational cost is the same as SPIHT. Both use the same principle (bit-plane coding), only logical operations are needed to identify the significance of the wavelet coefficients and their refinements bits. Also, they used the same memory resources to hold both the transformed coefficients and the significant coefficients.

9. REFERENCES

[And03] Andra, K., Chakrabati, C., and Acharya, T., A high-performance JPEG2000 Architecture, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 3, pp. 209-218, 2003.

[Ant92] Antonini, M., Barlaud, M. Mathieu, P., and Daubechies, I. Image coding using wavelet transform, Transactions on Image Processing, Vol. 1, pp. 205-220, 1992.

[Dee03] Deever, A. T., and Hemami, S. S., Efficient sign coding and estimation of zero-quantized coefficients in embedded wavelet image codecs, IEEE Transactions on Image Processing, Vol. 12, No. 4, pp 420-430, 2003.

[Eln03a] Elnagahy, F., and Šimák, B., A new algorithm for embedded image coding**,** 26th International Conference Telecommunications and Signal Processing TSP 2003, Brno, Czech Republic, pp. 147-150, 2003.

[Eln03b] Elnagahy, F., and Šimák, B., An algorithm for wavelet-based rate scalable image compression," 10th International Workshop on Systems, Signals and Image Processing IWSSIP 2003, Prague, Czech Republic, pp. 24-27, 2003

[Sai96] Said A., and Pearlman, W. A., A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 243--250, 1996.

[Sha93] Shapiro, J. M., Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing, Vol. 41, pp. 3445--3462,1993.

[She99] Shen, K., and Delp, E. J., Wavelet based rate scalable video compression, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 1, pp. 109-122,1999.

[Sod99] Sodagar, I., Lee, H-J., Hatrack, P.,and Zhang, Y. -Q. ,Scalable wavelet coding for synthetic/natural hybrid images, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 2, pp. 244-254, 1999.

[Tau00] D. Taubman, High performance scalable image compression with EBCOT, IEEE Transactions on Image Processing, Vol. 9, No. 7, pp. 1158-1170, 2000.

[Xio97] Xiong, Z., Ramchandran, K., and Orchard, M. T., Space-frequency quantization for wavelet image coding, IEEE Transaction on Image Processing Vol. 6, No. 5, pp.677 –693, 1997.

[Wit89] Witten, H., Neal, R., and Cleary, J. G., Arithmetic coding for data compression, Comm. ACM, Vol. 30, pp. 520-540, 1987.

Figure 4 Reconstructed Lena images at bit rate 0.125 bits/pixel. SPIHT (left) and RPSWS (right)

Figure 5 Reconstructed Barbara images at bit rates 1, 0.125, and 0.015625 bits/pixel respectively. SPIHT (left) and RPSWS (right)

Figure 6 Reconstructed Goldhill images at bit rates 1, 0.125, and 0.015625 bits/pixel respectively. SPIHT (left) and RPSWS (right)