LOW COMPLEXITY RESOLUTION PROGRESSIVE IMAGE CODING ALGORITHM : PROGRES (PROGRESSIVE RESOLUTION DECOMPRESSION)

Yushin Cho and William A. Pearlman

Rensselaer Polytechnic Institute
Troy, NY, USA

Amir Said

Hewlett-Packard Labs
Palo Alto, CA, USA

ABSTRACT

A very fast, low complexity algorithm for resolution scalable and random access decoding is presented. The algorithm avoids the multiple passes of bit-plane coding for speed improvement. The decrease in dynamic ranges of wavelet coefficients magnitudes is efficiently coded. The hierarchical dynamic range coding naturally enables a resolution scalable representation of a wavelet transformed image.

1. INTRODUCTION

Modern image coding methods, like JPEG2000's EBCOT, are able to support simultaneously sub-image decompression (ROI), and also quality (SNR), resolution, and spectral scalability. Unfortunately, while the loss in compression incurred by supporting these features can be quite small, they may increase computational complexity significantly.

Quality scalability is commonly done via bit-plane coding, which also helps to improve compression, since neighboring bits provide convenient and powerful contexts for entropy coding. However, in many important applications, the images always need to have a pre-defined high quality, and any extra effort required for quality scalability is wasted.

In this paper we consider fast coding methods that support only resolution scalability and efficient decompression of sub-images. We focus on the entropy coding effort, which becomes significant on high-quality images since its complexity grows with bit rate. Our solution addresses the challenge of avoiding compression loss and at the same time reducing complexity by not using bit-plane coding (and its contexts), nor standard entropy coding.

The proposed algorithm, PROGRES (Progressive Resolution Decompression) is a method that exploits the same image properties as SPIHT, but adapted to support resolution scalability with great speed. For a pre-defined quality, it can very efficiently decompress any image region at several resolutions. It is an excellent choice for remote sensing and GIS applications, where rapid browsing of large images is necessary.

2. PREVIOUS WORK AND OVERVIEW

Speed improvements were observed in hybrid forms of bit-plane coding, where once an image transform coefficient is classified as significant during a bit-plane pass, its sign and all its less significant bits are encoded together, so that refinement passes are not needed [3]. Oliver and Malumbres [4] presented LTW (Lower-Tree Wavelet), which is another solution for resolution scalable wavelet image coding with low complexity, based on non-embedded coding.

Similar to other wavelet based image coding methods using intra and inter-band coding contexts, our method is based on two properties of natural images: (a) energy in each subband normally decreases with frequency; (b) statistics in a local neighborhood are similar. Thus, we also use the strategy of coding wavelet coefficients following the order of expected importance, i.e., from low to high-resolution subbands. To reduce the computational burden we do not follow a plane-by-plane scan. Each coefficient, represented by sign and magnitude, is processed only once.

Since we want to avoid using standard entropy coding methods like arithmetic or Huffman codes, we code only the sign bit and the bits below the most significant non-zero bit (MSNB), so the position of that MSNB (dynamic range) must be known in advance. We code that value by coding its difference from similar values at same position in the corresponding subband with lower resolution. Each spatial-orientation tree rooted at LL subband is coded independently. This way we sacrifice SNR scalability for faster coding, but preserve both resolution scalability and the ability to decode sub-images.

3. COEFFICIENT DYNAMIC RANGES

3.1. Representing the Dynamic Range of Coefficients

We use $c_{i,j}$ and $s_{i,j}$ to represent, respectively, a wavelet coefficient at location (i, j), and the spatial orientation tree (set of coefficients) with root at location (i, j).

As mentioned above, to represent the magnitude compactly, the number of required bits should be known in advance. When the dynamic range of a coefficient magnitude

Table 1. Dynamic range of coefficients

Dynamic range bits	Dynamic range of coefficient	# bits for symbol index	# bits for sign
0	[0]	0	0
1	[-1, 0, 1]	1	1
2	[-3, $\cdots$, 3]	2	1
$\vdots$	$\ldots$	$\vdots$	$\vdots$
10	[-1023, $\cdots$, 1023]	10	1

is represented by the number of bits, k, the magnitude varies in the range of $[0, 1, \cdots, 2^k - 1]$. Thus, we call the number of bits to represent the dynamic range as *dynamic range bits*. For example, if the dynamic range bits of a coefficient is 3, it can have the values varying from -7 to +7 with additional one bit for sign information.

Each set (a spatial orientation tree) will contain different dynamic range for the magnitudes, based on the activity of its coefficients. We define the *dynamic range bits* $r_{i,j}$ of the set $s_{i,j}$ as:

$$r_{i,j} = \lceil \log_2 (\max_{c_{p,q} \in s_{i,j}} |c_{p,q}| + 1) \rceil \,,$$

which accounts for how many bits are required to represent every coefficient magnitude in the set.

Table 1 shows the dynamic range bits, their corresponding dynamic ranges, the number of bits for symbol index in each range, and the corresponding sign information.

3.2. Coding of Energy Ranges in a Partitioned Set

When a set is partitioned into its subsets, each subset will have a different dynamic range, probably a decreased one because the root coefficient of the set likely to have the largest magnitude in the set. Thus, every child set (i.e. subset) is likely to have smaller dynamic range than that of their parent set.

Therefore, it is a good idea to predict the dynamic range of energy in each subset based on the dynamic range of energy of a parent set, as shown in Fig. 1. Let $I(i,j) = \{(2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1)\}$ denote the set of position indices of the children of set $s_{i,j}$. Assuming that a parent set $s_{i,j}$ is partitioned into four subsets $s_{m,n}$, $(m,n) \in I(i,j)$, then a $r_{m,n}$, $(m,n) \in I(i,j)$ is the dynamic range for each subset, respectively.

Now, for representing the dynamic range bits of each subset $s_{m,n}$, we encode

$$d_{base} \quad = \quad r_{parent} - r_{children} \,,$$

where r_{parent} is the dynamic range $r_{i,j}$ of the parent set $s_{i,j}$ and $r_{children}$ is the dynamic range of the children sets

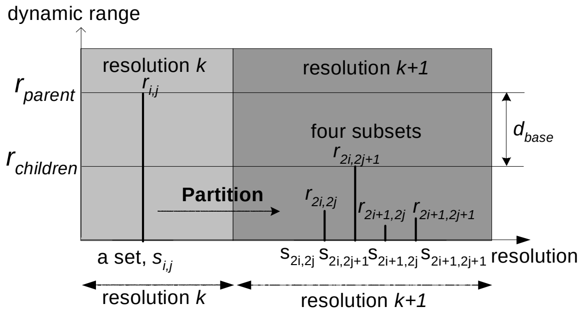

Fig. 1. Coding of Dynamic Ranges : the dynamic range bits for each subset $s_{m,n}$, is reconstructed by : $r_{children} = r_{parent} - d_{base}$, where the information of $r_{children}$ is common to every subset $s_{m,n}$.

of $s_{i,j}$, i.e.

$$r_{children} = \max_{(m,n) \in I(i,j)} (r_{m,n}) \,.$$

Note that one dynamic range bits, $r_{children}$, is used to represent the magnitudes in all children sets.

Then, in decoder side, given the information of r_{parent} and d_{base}, the $r_{children}$ can be reconstructed and we use this value as the dynamic range bits for the children sets $s_{m,n}$. Note that the information of $r_{parent} - d_{base}$ is common to every subset $s_{m,n}$.

Now, the coded information for the tree $s_{i,j}$ with two resolution scales will be:

$$r_{i,j}, c_{i,j}, d_{base}, c_{2i,2j}, c_{2i,2j+1}, c_{2i+1,2j}, c_{2i+1,2j+1},$$

where $c_{2i,2j}, c_{2i,2j+1}, c_{2i+1,2j}, c_{2i+1,2j+1}$ are root coefficients of each child subset. The $c_{i,j}$ and $c_{m,n}$, $(m,n) \in I(i,j)$ contain sign information.

There is a reason why we choose d_{base} rather than $r_{children}$. From our experience, it is more probable that $d_{base} \leq r_{children}$, i.e. $P(d_{base} \leq r_{children}) > 0.5$ in any wavelet transformed image. And $P(d_{base} \leq r_{children})$ is getting closer to 1 for lower bit rates. This explains that coding d_{base} will cost less bits than coding $r_{children}$.

The above coding scheme of dynamic range bits is applied to every two adjacent resolution scales, k and (k+1), $k = 0$ to $M - 1$, where M is the highest resolution. In this case, note that the number of parent-children relationships increases four times for each additional resolution scale.

4. CODING ALGORITHM

The encoding algorithm of PROGRES is described here. For simplicity, we assume that LL subband has one wavelet coefficient. Thus, the algorithm works on size $2^M \times 2^M$

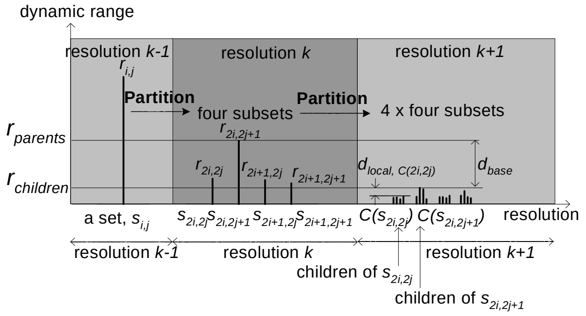

Fig. 3. The Extended Idea of Dynamic Range Coding : the dynamic range bits for each set $C(s_{m,n})$, is reconstructed by: $r_{parents} - d_{base} - d_{local,C(s_{m,n})}$, where the information of $r_{children}$ is common to every set $C(s_{m,n})$.

wavelet coefficients if M levels of wavelet decomposition is performed. The list L contains the sets to be coded.

The set $s_{0,0}$ rooted in LL subband has three subsets $s_{0,1}, s_{1,0}, s_{1,1}$, corresponding to subbands HL_M, LH_M , HH_M. Except root and leaf sets, every set $s_{i,j}$ has four subsets, $s_{2i,2j}$, $s_{2i,2j+1}$, $s_{2i+1,2j}, s_{2i+1,2j+1}$.

Fig. 2 shows the encoding algorithm. Note that '//' indicates the comments in corresponding statement.

As seen in Statement 5. in the algorithm, the PROGRES coder encodes the transformed wavelet information resolution by resolution, from lower to higher. This enables the progressive resolution decoding. Also, when blocks of wavelet coefficients corresponding to the same sub-image are coded together, each sub-image is both random access decodable and progressive resolution decodable. In this way, the target sub-image can be decoded by random access with progressive resolution.

If the LL subband has more than one coefficient, each of those coefficients becomes a root of a spatial orientation tree. Each tree is coded independently by this algorithm.

5. THE EXTENDED IDEA OF DYNAMIC RANGE CODING

The PROGRES image coder is built on the extended idea of dynamic range coding. Instead of sharing the d_{base} value among four children, it is shared by sixteen children, whose parent sets are rooted at the same tree level. In other words, these sixteen children have the same grand-parent, as shown in Fig. 3.

We assume $(m, n) \in I(i, j)$ as before. Then, in Fig. 3, $C(s_{m,n})$ at resolution k+1 indicates the children sets of each set $s_{m,n}$ at resolution k. Our goal here is to code the root coefficients in $C(s_{m,n})$ at resolution k+1, i.e. the grand children coefficients of the set $s_{i,j}$.

Table 2. The comparison of coding time between original RPI 2D-SPIHT, LTW, and the presented PROGRES (Lenna 8 bpp 512×512, Woman 8 bpp 2048×2048

Bitrate (bpp)	Encoding (cycles $\times 10^6$)			Decoding (cycles $\times 10^6$)		
	SPIHT	LTW	PROGRES	SPIHT	LTW	PROGRES
Lenna						
0.125	24.25	34.7	23.66	4.46	12.3	1.60
0.25	30.87	38.9	26.12	7.78	17.4	2.61
0.5	46.18	46.7	29.01	16.04	27.1	4.55
1.0	67.86	62.4	34.80	33.31	47.1	8.32
Woman						
0.125	400.66		378.43	73.90		24.09
0.25	524.05	N/A	404.34	150.11	N/A	41.92
0.5	788.30		450.13	307.33		74.71
1.0	1370.54		528.42	675.15		128.42

The information of $r_{parents}$ is available to every child $s_{m,n}$ at resolution k, since every root coefficient $c_{m,n}$ is coded using $r_{parents}$ bits. Now, the dynamic range bits for each $C(s_{m,n})$ at resolution k+1 can be predicted in two stages. First, the $r_{children}$ is predicted by d_{base}, and then, second, the $d_{local,C(s_{m,n})}$ is further used to predict the dynamic range bits for each $C(s_{m,n})$.

Thus, each set $C(s_{m,n})$ has the dynamic range bits, $r_{parents} - d_{base} - d_{local,C(s_{m,n})}$, where $r_{parents} - d_{base} = r_{children}$. As a result, the sixteen root coefficients from $C(s_{m,n})$ are sharing the information of d_{base}, which leads to efficient coding of dynamic ranges.

6. EXPERIMENTAL RESULTS

Tests were performed using a Intel 2.0 GHz Xeon processor, MS-Windows 2000, and Visual C++ 6.0 Compiler with speed optimization. The coding times for two 8 bpp gray scale images, 512×512 Lena and 2048×2048 Woman, at the rate of 0.125, 0.25, 0.5 and 1.0 bpp are shown in Table 2. The coding time is measured in CPU cycles of the Pentium processor.

The binary uncoded version of 2D-SPIHT from RPI is chosen for comparison. Note that wavelet transformation times are not included. Six and eight levels of wavelet decomposition with Daubechies 9/7 filters are used for Lena and Woman, respectively. The PROGRES scheme performs lossless coding of quantizer bin numbers on the pre-quantized wavelet transformed image. Note that both SPIHT and PROGRES do not use subsequent entropy coding of the code streams.

Table 2 shows that the encoding time of PROGRES increases very slowly along the increasing bit rate and reveals greater speed improvement over SPIHT for higher bit rate, two times at 1.0 bpp. The speed improvement in decoding is achieved over all bit ranges, four times on average. The loss of decoding quality (in PSNR) is almost ignorable as shown in Table 4. For 0.5 bpp Woman, the decoded

1. Find the maximum dynamic range bits r_{parent} and binary encode it;
2. **if** $r_{parent} = 0$ **return**; // no coefficients to encode ?
3. Initialize a list $L \longleftarrow$ a set in the lowest resolution (i.e. LL subband);
4. Binary encode a root coefficient in the list L using r_{parent} bits;
5. **for** each resolution level k(from the lowest to the highest)
   (a) **for** each set j in current resolution level k
      i. Enumerate subsets of the current set j;
      ii. $r_{parent} \longleftarrow$ maximum dynamic range bits of current set j;
      iii. $r_{children} \longleftarrow$ maximum dynamic range bits of subsets in current set j;
      iv. $d_{base} \longleftarrow r_{parent} - r_{children}$;
      v. Unary encode d_{base};
      vi. **if** $r_{children} = 0$, goto (a)
      vii. **for** each subset i
         A. Binary encode the the root coefficient of the subset i using $r_{children}$ bits and encode its sign information using one bit;
         B. **if** subset i has its descendants,
            then append subset i to the end of the list L for next resolution coding;
      viii. Remove the current set j from the list L;

Fig. 2. Encoding algorithm.

Table 3. Decoding time of progressive resolutions

Lena (512×512)		Woman (2048×2048)	
Resolution	Decoding time (cycles $\times 10^6$)	Resolution	Decoding time (cycles $\times 10^6$)
16×16	0.4997	64×64	2.1385
32×32	0.5851	128×128	2.9157
64×64	0.8160	256×256	5.5415
128×128	1.5392	512×512	13.2441
256×256	3.0343	1024×1024	35.2448
512×512	4.6403	2048×2048	75.2314

quality of PROGRES is slightly better than SPIHT. Also, PROGRES is significantly faster than LTW in [4], up to two times in encoding and up to seven times in decoding, LTW uses arithmetic coding of bit ranges of coefficients in subbands.

Table 3 shows that the decoding times of Lena and Woman at 0.5 bpp are increasing for progressively increasing resolutions. .

7. CONCLUSION

The low time complexity 2D-image coding algorithm, PROGRES (Progressive Resolution Decompression), is presented. Non bit-plane coding scheme is applied to reduce the coding time. The dynamic ranges of wavelet coefficients are efficiently coded by sharing information of decrease in energy along subbands with increasing frequencies. The proposed method is faster than original 2D SPIHT, two times in encoding and four times in decoding at 1.0 bpp. With only small loss of quality, this scheme achieves a very low time-complexity with resolution scalable and random access decodable features.

Table 4. Quality of decoded images by 2D-SPIHT and PROGRES in PSNR

Bitrate (bpp)	SPIHT (dB)	PROGRES (dB)
Lena		
0.125	30.7189	30.5900
0.25	33.7231	33.7118
0.5	36.8703	36.8496
1.0	40.0276	39.8907
Woman		
0.125	26.9272	26.8867
0.25	29.4227	29.4022
0.5	32.9156	33.0160
1.0	37.7515	37.7472

8. REFERENCES

[1] J. M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficient,” *IEEE Trans. on Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.

[2] A. Said and W. A. Pearlman, “A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6., pp. 243-250, June 1996.

[3] W. A. Pearlman, “Trends of Tree-Based, Set Partitioning Compression Techniques in Still and Moving Image Systems,” *Proceedings Picture Coding Symposium 2001*, Apr., 2001, pp. 1-8. (Invited, keynote paper)

[4] J. Oliver and M. P. Malumbres “Fast and Efficient Spatial Scalable Image Compression using Wavelet Lower Trees”, in *Proc. 2003 IEEE Data Compression Conference*, pp. 133-142, Mar. 2003.