A MULTIRESOLUTIONAL CODING METHOD BASED ON SPIHT

Keun-hyeong Park [*], Chul Soo Lee, and HyunWook Park

Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{charlie, hwpark}@athena.kaist.ac.kr

[*] Hanaro Telecom Inc., Seoul, Korea
khpark@hanaro.com

Abstract. In this work, we incorporate a multiresolutional coding functionality into the SPIHT algorithm [1]. The multiresolutional coding can be considered as a kind of the region of interest (ROI) coding with multiple regions of interest. Therefore, the ROI coding proposed by authors can be extended for the multiresolutional functionality without any cost in performance. The parent of ROI (PROI) and the multiple lists for insignificant sets and pixels, which were proposed for the ROI coding, are also used for the multiresolution coding.

1. Introduction

The image coder should provide a good rate-distortion performance. In addition, several functionalities become important when we consider various applications. One of the most important functionalities is the capability to encode various resolution images in a single bit stream. The functionality gives us the flexibility in using the network resources. While users with a narrow bandwidth can reconstruct a low resolution image by decoding a small fraction of the bit stream, other users with a wider bandwidth can reconstruct a high resolution image by decoding the whole bit stream. It would be desirable to incorporate such a feature into an image coding system without incurring heavy cost such as increased computational complexity or reduced rate-distortion performance. Xiong et al. proposed a multiresolutioal image decoding method [2]. They selected necessary coefficients to reconstruct various resolution images. It is difficult to use the multiresolution method in the low bit rate environment.

In this work, we adopt the scheme of the region of interest (ROI) coding and slightly extend it in order to incorporate multiresolutional coding functionality into the SPIHT algorithm. The multiresolutional coding can be considered as a kind of the ROI coding with multiple regions of interest.

This paper is organized as follows: The following section describes the SPIHT algorithm briefly. Section 3 presents the ROI coding and the proposed multiresolutional coding. Experimental results are given in section 4. Finally, we conclude this paper in section 5.

2. SPIHT

The SPIHT algorithm is one of the most efficient algorithms for still image compression. This algorithm was designed to transmit images progressively. Using wavelet transform, images are decomposed into subband images and the SPIHT algorithm arranges the wavelet coefficients in the magnitude order and transmits them from the most significant bit plane.

When a threshold is given, the algorithm partitions the coefficients or the sets of coefficients into significant ones and insignificant ones. Significant coefficients are added to the list of significant pixels (LSP) and insignificant coefficients are to the list of insignificant pixels (LIP) or the list of insignificant sets (LIS). While the LIP contains the insignificant pixels, the LIS contains the sets that include at least four insignificant pixels. An entry of the LIS, that is a set, is partitioned into several significant pixels, insignificant pixels, and insignificant sets when it contains one or more significant pixels. Whenever the algorithm determines whether the coefficient or the set is significant or not, it produces one bit for the information. The resultant bits from significant tests are generated as many as the number of entries in the LIP and the LIS, and the sign bits are produced as many as the number of entries that are added to the LSP. Once a pixel is added to the LIP, the pixel generates a bit in every bit plane to show whether it is significant or not. Therefore, it is impossible to reduce the number of bits caused by such pixels after they are added to the LIP. To reduce the bit generation, we must keep the LIS from partitioning. This property is essential in adding functionalities to the SPIHT algorithm.

3. Multiresolutional Coding

In the previous work [2], they decoded the whole bit stream and selected the necessary coefficients to reconstruct an image they wanted. It is difficult to use this function in the narrow bandwidth environment because the whole bit stream should be transmitted to reconstruct even a small image. If we rearrange coefficients at the encoder, several decoders with different bandwidths can receive enough coefficients to reconstruct images with the corresponding resolution. This is the purpose of multiresolutional coding.

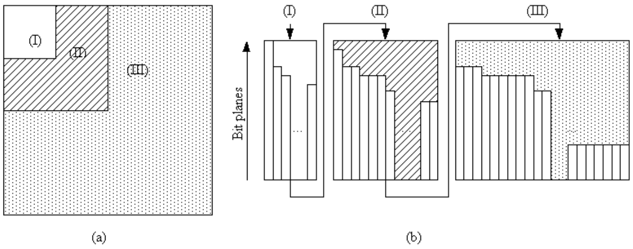

Fig. 1. (a) After wavelet transform, an image is decomposed into several subbands. (I) ROI for the quarter resolution image. (II) ROI for the half resolution image with (I). (III) ROI for the full size image with (I) and (II). (b) A coding sequence of the proposed multiresolutional coding.

If the encoder puts the bit stream for a low resolution image at the front part, a decoder can reconstruct the low resolution image with only the small fraction of the bit stream. If some decoders with wider bandwidth receive more bits, they can reconstruct a higher resolution image with them. Therefore, we can consider each part of coefficients as ROIs for reconstructing images. In Fig. 1 (a), we can think of the part (I) as an ROI for the quarter resolution image, and we can reconstruct the half resolution image from the coefficients of the part (I) and (II). Thus we can apply the ROI coding scheme into the multiresolutional coding. After describing the ROI coding in section 3.1, we present the multiresolutional coding in section 3.2.

3.1 ROI Coding

The ROI coding is a method to code images with an emphasis to regions of interest. Usually an encoder allocates more bits to coefficients of the ROI in order to emphasize the ROI. We proposed an algorithm that incorporated this functionality into the SPIHT algorithm [3].

```
a) Modified Sorting Pass:
      a.1) for each entry (i,j) in all the LIP(p) for $n \le p \le n_{max}$ do:
   if (i,j) is one of the ROI coefficients, do:
      a.1.1) output $S_n(i,j)$;
      a.1.2) if $S_n(i,j) = 1$, then move (i,j) to the LSP and output the sign of $c_{i,j}$;
   a.2) for each entry (i,j) in the LIS(p) for $n \le p \le n_{max}$ do:
   a.2.1) if the entry is of type A, then
         if (i,j) is one of the PROI coefficients, do:
            output $S_n(D(i,j))$;
            if $S_n(D(i,j)) = 1$, then
            for each $(k,l) \in O(i,j)$ and it belongs to the ROI, do:
                  output $S_n(k,l)$
                  if $S_n(k,l) = 1$, then add (k,l) to the LSP and output the sign of $c_{k,l}$;
                  if $S_n(k,l) = 0$, then add (k,l) to the end of the LIP(n);
                  if $L(i,j) \neq \varnothing$, then move (i,j) to the end of the LIS(n) as an entry of type B, and
                     goto Step a.2.2); otherwise, remove entry (i,j) from the current LIS(p);
                  if (k,l) doesn't belong to the ROI coefficients, do:
                     move (k,l) to the end of the LIP(n);
   a.2.2) if the entry is of type B, then
         if (i, j) is one of the PROI coefficients, do:
            output $S_n(L(i,j))$;
            if $S_n(L(i,j)) = 1$, then
               add each $(k,l) \in O(i,j)$ to the end of the LIS(n) as an entry of type A;
                remove (i,j) from the LIS(p);
b) Refinement Pass: for each entry (i,j) in the LSP,
     except those included in the last sorting pass (i.e., with same n),
          if (i,j) belongs to the ROI coefficients, do:
          - output the n-th most significant bit of $|c_{i,j}|$;
c) Quantization-Step Upgrade : decrement n by 1 and go to Step a) if $n \ge R$
```

Fig. 2. Modified procedure for the ROI

In the ROI coding algorithm, coders perform the node test only when the node is in the ROI mask, and also perform the descendant test only if the node is in the PROI mask [3]. Then the LIP contains three types of coefficients: significant and non-ROI coefficients; insignificant and non-ROI ones; insignificant and ROI ones. At the following bit plane, sorting pass checks only the last ones and non-ROI coefficients stay in the LIP unrelated to thresholds.

After coding the ROI, the coder must deal with coefficients that are excluded in the ROI coding procedure. Then, the coder performs the node test for the entries in the LIP from the large threshold. However, the LIP contains lots of entries that are added to the LIP with relatively small thresholds. To reduce the bit generation, we partition the LIP into multiple LIP(n), and the LIS into LIS(n).

When we divide the LIP into several LIP(n) during the ROI coding procedure, the LIP(n) is a list of pixels that enter the list at the threshold of 2^n. After the ROI coding procedure, coders perform the node tests only the entries in the LIP(k), where k is greater than n. From this division, we can save lots of significant tests of the nodes in the LIS or the LIP. The modified coding procedure of the ROI is shown in Fig. 2 and we underlined the modified parts of the algorithm. During the procedure of the non-ROI coefficients, the coder concatenates the divided lists into each list, i.e., LIS and LIP. We omit the procedure for the non-ROI because it can be easily deduced.

3.2 Multiresolutional Coding

In the previous section, we reviewed the ROI coding method. We apply the concept of the ROI coding to multi-resolutional coding. As mentioned before, the multiresolutional coding is an ROI coding with multiple ROIs. In addition, the degree of interest is different among the regions. In Fig. 1(a), the region (I) is the most interesting and the region (II) is the second interesting. Therefore, we first encode the region (I) with the ROI coding method. During the ROI coding, both the region (II) and (III) are unrelated to the ROI. After completing the coding at the user-defined bit plane (R in Fig. 2), the region (II) becomes the second ROI. After coding of the region (II), the region (III) becomes the final ROI. Figure 1(b) shows such coding order, and the narrow bars in each region are the sorted coefficients.

Let us consider the bit plane where coders finish the ROI coding procedure. As the value of R is smaller, more bits are allocated to the corresponding ROI. On the contrary, a higher value of R supports wider regions. If the bit budget remains after encoding all of the regions, it is possible to code all coefficients in the whole regions ((I), (II), and (III) in Fig. 1(a)) using the original SPIHT algorithm.

4. Experiment Results

The following results were obtained from the Lena image with monochrome and 512×512 with 8 bpp for fixed rate coding. We used the 9/7-tap biorthogonal wavelet filters [5] to decompose the image.

Figures 3 (a) and (b) show the layered bit streams generated by the previous multiresolutional encoding [2] and the proposed multiresolutional encoding at 1 bpp and 0.1 bpp, respectively. In the figures, we can see that the first layer of the bit stream generated by the previous multiresolutional encoding method is much longer than the first layer of our algorithm. This result shows that the number of coefficients for low resolution image is very small. In addition, 'R' indicates a user-defined threshold – how many bit planes the coders must deal with. With a high bit rate, encoder can deal with many bit planes (small R) as well as various resolution images. However, a low bit rate limits the available bit planes and resolutions of support. This is coincident with our intuition.

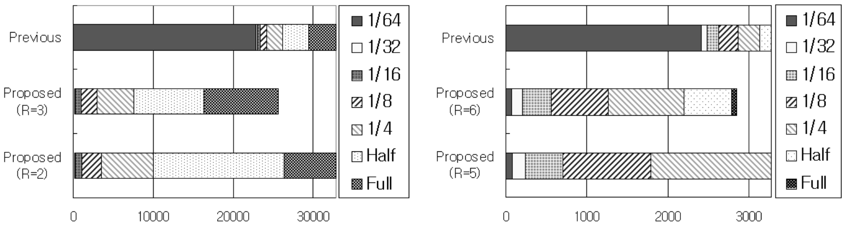

Fig. 3. (a) Left - Comparison of bit streams when the image is compressed with one bpp bit rate. (b) Right - Comparison of bit streams when the image is compressed with 0.1 bpp bit rate and with different thresholds.

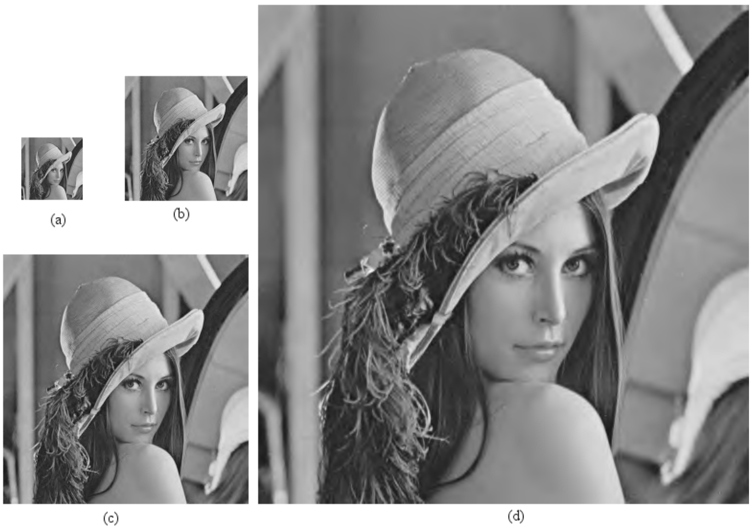

Fig. 4. Reconstructed images with the proposed multiresolutional coding method (bit rate = 1 bpp) from a single bit stream. (a) 1/8 resolution. (b) Quarter resolution. (c) Half resolution. (d) Full resolution.

Figure 4 shows the images reconstructed at different resolutions with the proposed multiresolutional coding method. These images are reconstructed from a single encoded bit stream with the bitrate being set at one bpp.

Table 1 shows the computation times for encoding and decoding processes with the original SPIHT algorithm and the proposed algorithm. We applied the proposed method to the Lena image with 256×256 resolution. We got the results under a Pentium III (550 MHz) with Windows 2000. The computation time of the proposed algorithm is almost similar to that of the original algorithm. This fact shows that there is a little overhead for the additional functionality.

bpp	SPIHT	Proposed multiresolutional coding method
1.0	0.19/0.13	0.20/0.13
0.9	0.18/0.12	0.19/0.12
0.7	0.16/0.10	0.17/0.10
0.5	0.14/0.09	0.14/0.09
0.3	0.12/0.07	0.13/0.07
0.1	0.10/0.05	0.10/0.05

Table 1. Comparison of computation times of the original SPIHT algorithm and the proposed algorithm for encoding/decoding. (unit: sec)

5. Conclusions

We extended the ROI coding algorithm to incorporate a multiresolutional coding functionality, which can decode multiresolution images from a single embedded bit stream. By moving the ROI from the highest tree level to the lowest level, we can obtain the multiresolutional coding functionality easily. The greatest advantage of the proposed multiresolutional coding algorithms is that we can incorporate a new function into the SPIHT algorithm without cost loss in the performance. This multiresolutional coding has many applications in network communications such as scalable multicast transmission in heterogeneous network, fast decoding, image database browsing, telemedicine, multipoint video conferencing, and distance learning, etc.

References

1. Said and Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, June, 1996.
2. Z. Xiong, B.J. Kim and W. A. Pearlman, “Multiresolutional encoding and decoding in embedded image and video coders,” *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 6, pp.3709-3712, 1998.
3. K. H. Park, C. S. Lee and H. W. Park, “Region-of-interest coding based on set partitioning in hierarchical trees,” to be presented in *IEEE International Conference on Image Processing (ICIP-01)*, Thessaloniki, Greece, Oct. 2001.
4. E. Atsumi and N. Farvardin, “Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees,” *International Conference on Image Processing (ICIP-98),* October 4-7, 1998 Chicago, Illinois, USA.
5. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Processing,* Vol. 1, pp. 205-220, April, 1992.