An Improved SPIHT Algorithm based on Double Significance Criteria

Chang-Mo Yang and Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)
1 Oryong-dong Puk-gu, Kwangju, 500-712, Korea
Phone: +82-62-970-2263, Fax: +82-62-970-2204
E-mail: cmyang@gogh.kjist.ac.kr, hoyo@kjist.ac.kr

Abstract: In this paper, we propose an improved SPIHT algorithm based on double significance criteria. According to the defined relationship between a threshold and a boundary rate-distortion slope, we choose significant coefficients and trees. The selected significant coefficients and trees are quantized and entropy-coded. Experimental results demonstrate that the boundary rate-distortion slope is well adapted and the proposed algorithm is quite competitive to and often outperforms the SPIHT algorithm.

1. Introduction

Discrete wavelet transform (DWT) provides a frame work for decomposing images into a hierarchy of frequency components, each represented by spatial resolution proportional to the frequency component. The resulting hierarchical image representation provides an attractive tradeoff between spatial and frequency resolutions for image compression applications [1].

However, early DWT image coding algorithms have shown only marginally better performance than standard image coding algorithms based on block transforms, until the introduction of the embedded zerotree wavelet (EZW) algorithm proposed by Shapiro [2]. The EZW algorithm exploits the correlation among wavelet coefficients to generate zerotrees and encode them using the significance map and residual data.

The work by Said and Pearlman [3] provides even better performance than the EZW algorithm by encoding insignificant blocks of wavelet transformed coefficients with a smaller number of bits. This algorithm, called the set partitioning in hierarchical trees (SPIHT), also exploits the correlation among the amplitudes of related coefficients at different scales. While the EZW algorithm scans the zerotree from subband to subband and labels the nodes as zerotree root (ZTR), isolated zero (IZ), positive significant (POS) or negative significant (NEG), the SPIHT algorithm uses the concept of set partitioning to encode the location of nonzero coefficients. The descendants of a given coefficient at a lower subband are considered as a set in the list of insignificant sets (LIS). Once a significant coefficient emerges from this set, the set is further decomposed. Although the SPIHT and EZW algorithms both try to exploit the inter-band correlation and partial ordering of the coefficients, the scanning order of the significant coefficients and the way of encoding their positions are quite different.

Although the EZW and SPIHT algorithms are very effective for progressive transmission due to the inherent embedded nature, they cannot claim that the distortion resulting from the quantization process is the minimum under a given rate constraint. In order to improve the coding efficiency of the SPIHT algorithm, we introduce two criteria for the significance map: a threshold and a boundary rate-distortion slope. In the proposed algorithm, significant coefficients and trees are selected with respect to the threshold and the boundary rate-distortion slope.

2. Improved SPIHT (ISPIHT) coder

2.1. Motivation and General Description

When we decompose the image into wavelet coefficients, as shown in Figure 1, most coefficients in high frequency bands have very small magnitudes and can be quantized to zero with unnoticeable distortion. Thus, the portion of quantized zero coefficients is high at very low bit-rates. Therefore, we only need to send the positions of nonzero coefficients in high frequency subbands.

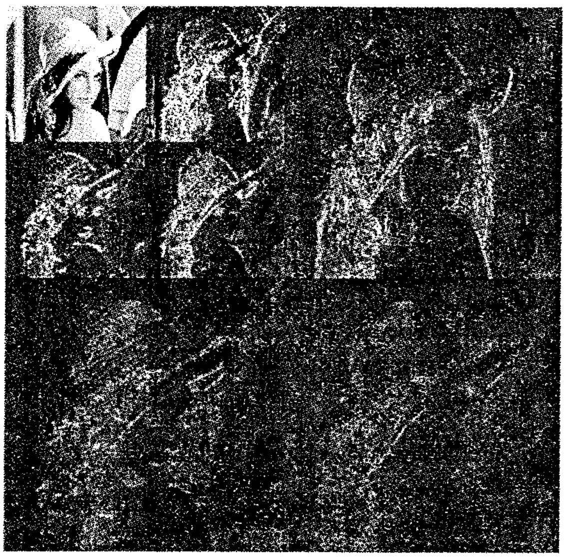


Figure 1. Two-level wavelet decomposition

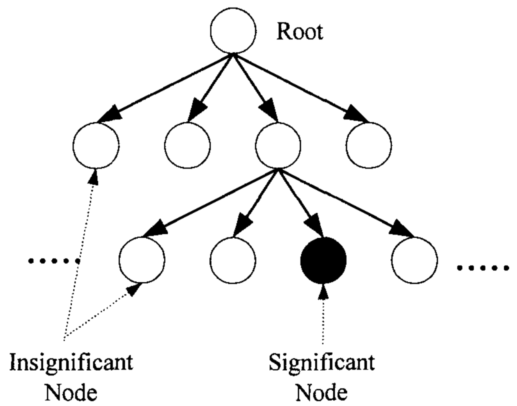
The most straightforward way is to send a map of nonzero coefficients for each subband to the decoder. However, inter-band correlation is not exploited in this case and some redundancy still exists in the data.

A zerotree structure of wavelet coefficients is generally employed to improve the compression efficiency for the significance map [2, 3]. The basic idea is that high-activity areas are identified in the highest level of the decomposition pyramid, and they are replicated in lower levels at the same spatial locations. Therefore, zerotree coding reduces the cost of encoding the significance map by using self-similarity.

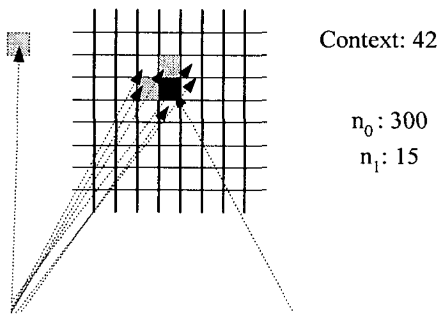
For zerotree encoding [2, 3], the positions of coefficients above a given threshold should be identified to be "significant" by the definition of trees or sets and the scanning order. At the decoder, the significant coefficients are reconstructed by the updated threshold and information of the refinement process.

However, some insignificant coefficients, which have significant neighbors or descendants, should also be identified to specify the positions of significant coefficients. In general, a large number of bits are wasted to represent these insignificant coefficients.

More specifically, let's consider a simple example. When there is only one significant node in the low level of the tree in Figure 2(a), several bits are needed to represent one significant coefficient.



(a) Tree



(b) Coefficient

Figure 2. Overused bits in a rate-distortion sense

From a rate-distortion sense, this is very inefficient. If we simply ignore the significant node as insignificant, all the nodes in the tree can be treated as an insignificant block and only one bit is needed to represent this tree. Therefore, if the cost of the increased distortion is not too high compared to the gain of the decreased rate, we can improve the coding efficiency by ignoring sparse significant nodes.

We can apply the same concept to the significance map coding of coefficients. In the QM-coder [4], we first calculate the rate-distortion slope of coefficients with a given threshold in Figure 2(b), where n_1 is the number of significant coefficients and n_0 is the number of insignificant coefficients with the same context of a given coefficient. If we encode this coefficient as significant, we need many bits for representing this significant coefficient. Therefore, we can improve the coding efficiency by ignoring the significant coefficient as insignificant when distortion is not so much increased by the reduced rate.

This is the main motivation of the proposed improved SPIHT (ISPIHT) algorithm, where the significance of trees and coefficients is determined by two criteria: threshold and boundary rate-distortion slope.

2.2. Boundary Rate-Distortion Slope

We performed extensive experiments with various test images to find a suitable boundary rate-distortion slope. Figure 3 shows one of the experimental results with BARBARA image.

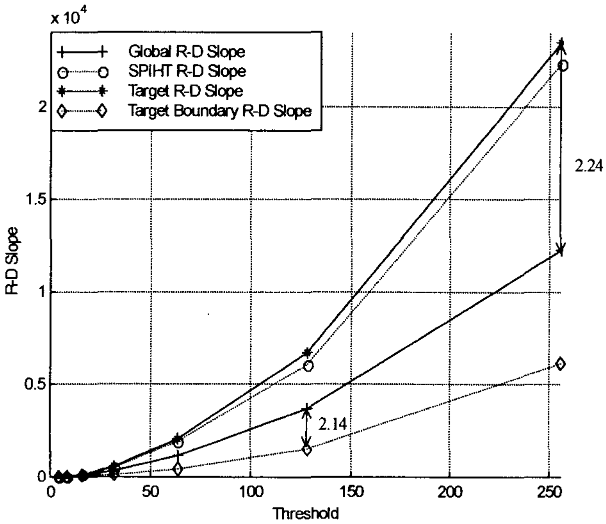


Figure 3. Boundary Rate-Distortion Slope

The target rate-distortion slope and the target boundary rate-distortion slope are determined by the rate-distortion slope where the compression performance is maximized. In Figure 3, we can observe that the ratio between the global rate-distortion slope and the target boundary rate-distortion slope in the current stage is similar to the ratio between the target rate-distortion slope and the global rate-distortion slope in the previous stage.

From this point of view, the boundary rate-distortion slope $\lambda_{T_n}^*$ is formulated by the global rate-distortion slope $\lambda_{T_n}^G$ and prediction factor ξ_{T_n}.

$$\lambda_{T_n}^{*} = \frac{\lambda_{T_n}^{G}}{\xi_{T_n}} \quad (1)$$

The global rate-distortion slope is determined by the ratio of the expected value of the decreasing distortion D_{Tn} and the increasing rate R_{Tn}, when encoding each significant map.

Using the mean squared error as a distortion measure, we can calculate the expected value of the decreasing distortion.

$$E[\Delta D_{T_n}] = p_{T_n}(1.5T_n)^2 \quad (2)$$

Where p_{Tn} is the probability of significant wavelet coefficients and T_n is a threshold value at stage n.

The expected value of the increasing rate can be calculated as

$$\begin{aligned} E[\Delta R_{T_n}] &= p_{T_n}(-\log_2 p_{T_n} + 1) - (1 - p_{T_n})\log_2(1 - p_{T_n}) \\ &= H(p_{T_n}) + p_{T_n} \end{aligned} \quad (3)$$

where $H(p_{Tn})$ is the first-order entropy of probability p_{Tn}. The additional bit in the significance map coding is for coding of sign bit.

The prediction factor ξ_{Tn} is determined by the ratio of the rate-distortion slope of the decoded data λ_{Tn-1} and the global rate-distortion slope λ_{Tn-1}^{G} in the previous stage.

$$\xi_{T_n} = \frac{\lambda_{T_{n-1}}}{\lambda_{T_{n-1}}^{G}} \quad (4)$$

When we calculate λ_{Tn-1}, we only consider the increasing rate and the decreasing distortion in the sorting pass. Finally, the boundary rate-distortion slope can be computed as

$$\lambda_{T_n}^{*} = \frac{\lambda_{T_n}^{G}}{\xi_{T_n}} = \frac{\lambda_{T_{n-1}}^{G} \cdot (1.5T_n)^2}{\lambda_{T_{n-1}} \cdot (1 + H(p_{T_n}) / p_{T_n})} \quad (5)$$

2.3. Rate-Distortion Slope of Coefficients and Trees

The result of significance identification is encoded by the QM-coder. The QM-coder encodes each significance identification using the probability that is calculated by monitoring the pattern of past zeros and ones under the same context. Therefore, we can calculate the rate-distortion slope of coefficients using the probability of the QM-coder.

If the probability that the current coefficient of coordinate (i,j) appears to be significant is $p_{i,j}$, the rate increase $d[\Delta R_{i,j}^{coeff}]$ for encoding this significance coefficient is calculated as

$$d[\Delta R_{i,j}^{coeff}] = 1 - \log_2 p_{i,j} \quad (6)$$

where the additional bit is for the sign bit. However, we do not consider to additional bit for an insignificant coefficient.

The distortion decrease $d[\Delta D_{i,j}^{coeff}]$ by encoding a significant coefficient of coordinate (i,j) is calculated by a real coefficient value.

$$d[\Delta D_{i,j}^{coeff}] = c_{i,j}^{2} \quad (7)$$

The distortion decrease by encoding an insignificant coefficient is zero because there is no distortion reduction by encoding insignificant coefficient.

Therefore, the rate-distortion slope of a significant coefficient $c_{i,j}$ is calculated as

$$\lambda_{i,j}^{coeff} = \frac{c_{i,j}^{2}}{1 - \log_2 p_{i,j}} \quad (8)$$

The rate-distortion slope of a tree can be calculated in the same fashion. The rate increase $d[\Delta R_{i,j}^{tree}]$ for encoding each tree is calculated as

$$d[\Delta R_{i,j}^{tree}] = \sum_{(i,j)\in O_{i,j}} [d[\Delta R_{i,j}^{coeff}] + d[\Delta R_{i,j}^{tree}]] \quad (9)$$

where (i,j) denotes the coordinate of the parent node of each tree.

The distortion decrease $d[\Delta D_{i,j}^{tree}]$ for encoding a tree is calculated as

$$d[\Delta D_{i,j}^{tree}] = \sum_{(i,j)\in O_{i,j}} [\varepsilon_{i,j} + d[\Delta D_{i,j}^{tree}]] \quad (10)$$

where $O_{i,j}$ means the children node of the root of the tree and $\varepsilon_{i,j}$ is defined as

$$\varepsilon_{i,j} = \begin{cases} (c_{i,j})^2 & \text{if } |c_{i,j}| \text{ is greater than threshold} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Therefore, the rate-distortion slope of a tree is calculated as

$$\lambda_{i,j}^{tree} = \frac{\sum_{(i,j)\in O_{i,j}} [\varepsilon_{i,j} + d[\Delta D_{i,j}^{tree}]]}{\sum_{(i,j)\in O_{i,j}} [d[\Delta R_{i,j}^{coeff}] + d[\Delta R_{i,j}^{tree}]]} \quad (12)$$

We note that two kinds of trees are used in the ISPIHT algorithm as in the SPIHT algorithm. When the current tree is type A, we calculate the rate and distortion using Eq. (9) and Eq. (10). In other case, we eliminate $\varepsilon_{i,j}$ and $d[\Delta R_{i,j}^{coeff}]$ in Eq. (9) and Eq. (10), and calculate the rate-distortion slope of the tree.

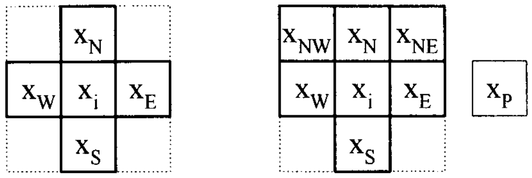
2.4. Entropy Coding

It is well known that the QM-coder is computationally efficient. The QM-coder estimates the probability of significance of coefficients with a state machine and then arithmetically encodes it. The minimum code-length l of the sequence in bits is given by

$$l = -\log_2 \prod_{i=1}^{n} p(x_i \mid x_{i-1}, x_{i-2}, \ldots, x_1) \quad (13)$$

where $p(x_i| x_{i-1}, x_{i-2},\ldots, x_1)$ is a conditional probability of x_i given $x_{i-1}, x_{i-2}, \ldots, x_1$. However, $p(x_i| x_{i-1}, x_{i-2},\ldots, x_1)$ is generally unknown in practice. Therefore, we have to estimate $p(x_i| x_{i-1}, x_{i-2},\ldots, x_1)$ based on the past observations in the coding process. A set of past observations on which the probability of the current symbol is conditioned is called the modeling context [5].

Figure 4 shows the context model for the proposed algorithm. We use 4 contexts for the tree map, and 7 contexts for the coefficient map. x_P is the parent of the current coefficient x_i, and the symbols x_{NW}, x_N, x_{NE}, x_W, x_E and x_S denote the coefficients located to the northwest, north, northeast, west, east and south of the current coefficient x_i respectively. These contexts are shared among different wavelet scales and orientations. Whenever a neighbor or parent coefficient is unavailable, the corresponding context bit is set to zero. The sign bit and the refinement bit are encoded with a fixed one context.



(a) Tree Map (b) Coefficient Map

Figure 4. Context Models of ISPIHT

3. Experimental Results and Conclusions

Performance comparisons between the ISPIHT and SPIHT algorithms are represented in Figure 5. The experiments are performed on two monochrome images, LENA and BARBARA, of size 512×512 pixels. These images are decomposed by 5-level Villasenor's 10/18 filter. We use the peak signal-to-noise ratio (PSNR) as a performance measure, which is defined as

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad dB \qquad (14)$$

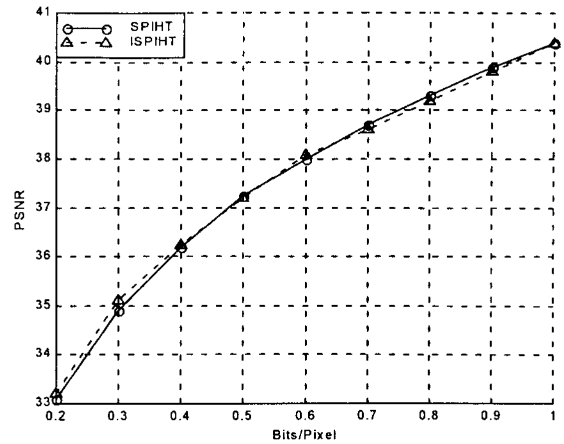
where MSE denotes the mean squared error between the original and reconstructed images.

As shown in Figure 5, the boundary rate-distortion slope proposed in this paper is well adapted and the proposed algorithm is quite competitive to and often outperforms the SPIHT algorithm.

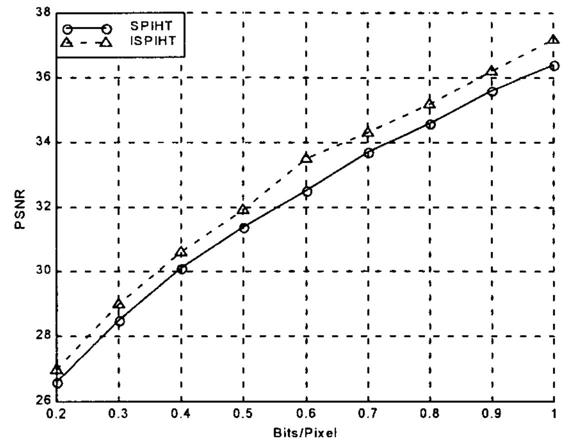
One of the strong points of the proposed algorithm is that the encoder does not send any information about the rate-distortion slope to the decoder. The decoder only needs the information about the threshold as in the SPIHT algorithm, and simply executes the reverse process of encoding with the information of the threshold. In addition, computational complexity of the proposed decoder is almost the same as that of the SPIHT algorithm, which is well known for good compression performance and computation simplicity. The proposed algorithm also retains the embedded nature that is very attractive for a number of applications.

Acknowledgments

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.



(a) LENA



(b) BARBARA

Figure 5. Comparison of ISPIHT and SPIHT

References

[1] J. Chen, T. Yu, and X. Wu, "Non-Embedded Wavelet Image Coding Scheme", *Internation Conf. on Image Processing*, vol. 1, pp. 584-587, October 1997.

[2] J. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.

[3] A. Said and W. Pearlman, "A New, Fast, and Efficient Image Codec based on Set Partitioning in Hierarchical Trees," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, no. 3, pp. 243-250, June 1996.

[4] W. Pennebarker and J. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold Publishers, New York, 1993.

[5] X. Wu and J. Chen, "Context Modeling and Entropy Coding of Wavelet Coefficients for Image Compression," *IEEE International Conf. on Acoustics, Speech and Signal Processing*, vol. 4, pp. 3097-3100, April 1997.

[6] C. Yang, *A Study on Embedded Image Compression using Wavelet Coefficients*, M.S. thesis, Kwangju Institute of Science of Technology (K-JIST), Korea, 2000.