A FLEXIBLE ZEROTREE CODING WITH LOW ENTROPY

*Sang-hyun Joo, *Hisakazu Kikuchi, *Shigenobu Sasaki, **Jaeho Shin
* Department of Electrical Engineering, Faculty of Engineering, Niigata University, Japan
** Department of Electronics, Faculty of Engineering, Dongguk University, Korea

ABSTRACT We introduce a new zerotree scheme that effectively exploits the inter-scale self-similarities found in the octave decomposition by a wavelet transform. A zerotree is useful to code wavelet coefficients and its effectiveness was proved by Shapiro's EZW. In the coding scheme, wavelet coefficients are symbolized and then entropy-coded. The entropy per symbol is determined from the produced symbols and the final coded size is calculated by multiplying the entropy and the total number of symbols.

In this paper, we analyze symbols produced from the EZW and discuss the entropy per symbol. Since the entropy depends on the produced symbols, we modify the procedure of symbol generation. First, we extend the relation between a parent and children used in the EZW to raise the probability such that a significant parent has significant children. The proposed relation is flexibly extended according to the fact that a significant coefficient is likely to have significant coefficients in its neighborhood.

Our coding results are compared with the published results in paper [1] and improvements come from the use of lower entropy per symbol. We also give the comparison of the number of produced symbols.

KEYWORD: image compression, wavelet transform, zero-tree coding

I. INTRODUCTION

In the wavelet-based coding, dependencies among bands using quadtrees were exploited in EZW(Embedded Zerotree Wavelet)[1], SPIHT(Set Partitioning In Hierarchical Trees)[2], SFQ(Space Frequency Quantization)[3,4]; that is, one coefficient at a given band is related with four coefficients at the same spatial location at the next finer band in terms of a relation between parent and children and the relation is applied for all coefficients except for DC coefficients. The EZW coder was designed by Shapiro who first applied an embedded zerotree using a wavelet. The algorithm is based on three concepts; 1) prediction of the absence of significant coefficients across scales by exploiting the self-similarity inherent in images 2) successive approximation for decoded coefficients 3) adaptive arithmetic coding for the streamed out symbols. After that, Said and Pearlman published their great work – SPIHT – that gives nice performances and fast processing. They use three lists to find significant coefficients in bands; 1) an LIP for insignificant coefficients 2) an LSP for significant coefficients 3) an LIS for insignificant descendants. The LIS includes two kinds of information for the descendants at the forms of type A or type B. The three lists are identically duplicated in a decoder by the transmitted bit stream. In the EZW and SPIHT algorithms, their merit is on the termination ability at any point that an encoder/decoder wants to stop, and the decoder reconstructs an approximated image from the information he has received. This property is clearly desirable when we consider of our constrained communication channels. More recently, Z.Xiong et. al. published an SFQ algorithm that surpasses the EZW and SPIHT in performance and there are two versions according to wavelet decompositions. One of them uses the octave band wavelet and the other uses the wavelet-packet. They get a coding performance while pruning branches from trees in a rate-distortion sense and scalar-quantizing the coefficients at the survived nodes. The decision to prune a branch or not depends on the pre-assigned bit budget and comparing of costs between pruning and non-pruning.

Most of coding schemes have two common procedures; 1) symbol generation (model transformation) and 2) entropy coding of the symbol stream. The symbol stream is produced for the purpose of representation and then symbols are entropy-coded. In this paper, we introduce a new zerotree scheme that lead lower entropy and thus more compression. Since the entropy per symbol is determined by the probabilities of produced symbols, we thus modify the procedure of the symbol generation with flexible treeing. The tree is flexibly designed in view of entropy. In the EZW scheme, a node on a tree branches out into four nodes and this relation is referred to as a fixed relation in the sense that the relation is not changed. On the other hand, our proposed relation is referred to as a "flexible tree" in the sense that a node on a tree branches into basic four nodes and flexibly extends its branches to nodes in neighbor. The idea to the flexible tree comes from how to extend more branches.

II. ZEROTREE BASED COMPRESSION

1. Embedded Zerotree Wavelet coding

Jerome M. Shapiro [1] developed an algorithm that exploits a relation between subbands in image compression. In the algorithm, zerotrees have been combined with bit plane coding and demonstrate the effectiveness of wavelet based coding. The algorithm is based on the zerotrees that efficiently represent many insignificant coefficients. As wavelet coefficients are located having some dependencies in bands, the dependencies are well exploited with a quadtree structure. The compression has three step procedures: 1) wavelet decomposition 2) symbol generation 3) entropy coding. We briefly review the coding algorithm and discuss produced symbols and its entropy. To describe the compression scheme, we quote several definitions - like parent, child, ancestor, descendant, root etc. - from the reference [1].

There are two types of passes performed: 1) a dominant pass 2) and a subordinate pass. The dominant pass finds significant coefficients to a given threshold, and the subordinate pass refines all significant coefficients found in all previous dominant passes. We use four symbols to tell a dominant pass to a decoder. A ZTR symbol is used for a zerotree root that is insignificant and has no significant descendants. One more needed symbol is an Isolated Zero symbol (named IZ) used when a coefficient is insignificant but has some significant descendants. Besides the symbols, two symbols are used for a significant coefficient – POS and NEG

according to its sign. After all, the use of ZTR and IZ symbols is to inform locations of significant coefficients (POS and NEG) as efficiently as possible.

After a dominant pass, a subordinate pass is performed in order to refine the coefficients found to be significant in the previous dominant passes and these two passes are entropy-coded with an adaptive arithmetic coder[7].

2. The Shannon's entropy

As we reviewed in the previous section, a symbol stream is produced from the alternate passes and then the stream is entropy-coded for more compression. In this sub section, we briefly study the Shannon's entropy theorem to analyze the symbol stream.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n symbols. Given $D = \{d_1, d_2, \ldots, d_l\}$, a data set of l symbols in a sequence (the number l is also called the data length of D), the probability distribution of the symbol set S in the data D is the collection of positive numbers $P = \{p_1, p_2, \ldots, p_n\}$, one for each symbol, defined by

$$p_i = |\{d_k \in D \mid d_k = s_i\}| / l, \quad \text{for } i = 1,2,\cdots,n. \qquad (1)$$

If the probability distribution is the only assumed redundancy information, the pair (S,P) is called a zero-order Markov source. The data sequence D is called a zero-order Markov sequence.

Using the above notations, the (zero-order) entropy of the data sequence D is defined to be

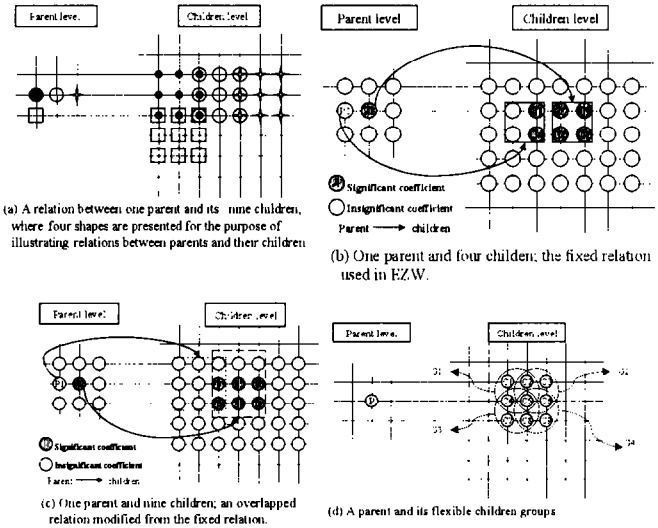
$$e(D) \quad -\Sigma p_i \cdot \log_2 p_i. \qquad (2)$$

3. A relation between the number of symbols and its entropy

With the Shannon's entropy, we consider a relation between the number of symbols and its entropy. The entropy per symbol largely depends upon occurrence probabilities of symbol alphabets and thus the final coded size is calculated by multiplying the average entropy and the number of symbols. Therefore, we can achieve more reduction in final coded size with two ways; one is to reduce the entropy per symbol and the other is to reduce the number of symbols.

We assume two particular source models for this discussion. Both models are composed of two symbols (S_1 and S_2) but the probability distributions and the numbers of symbols are different. Assume the first model has ten S_1 and ten S_2. In this case, the probabilities of symbols and its entropy are calculated by using equations 1 and 2. The model is assumed to be uniformly distributed and the entropy is 1 bit/symbol. Therefore, we should use 20 bits for the model. On the other hand, we assume the second model that has five S_1 symbols and 20-bit budget. In this case of the model, we consider how many S_2 symbols we can insert in the 20 bits. Using the equations 1 and 2, we can insert, at least, 20 S_2 symbols into the source. Therefore, the final output sizes are the same at 20 bits though they have different source length of symbols. It is important to compare the difference between the two distributions; in the second model, five S_1 has a worth of ten S_2 if we consider only the number of symbols. If the replacement is accomplished without any deformation of the contents, our attention will go to the number of symbol to be replaced. When we accomplish the replacement in smaller number than two, we do expect more compression with less entropy although the total number of symbols are increased. This consideration is discussed in the next section with more detailed example and we will apply to the EZW by using our flexible tree structure.

III. A FLEXIBLE RELATION IN PARENT-CHILD



(a) A relation between one parent and its nine children, where four shapes are presented for the purpose of illustrating relations between parents and their children

(b) One parent and four children; the fixed relation used in EZW.

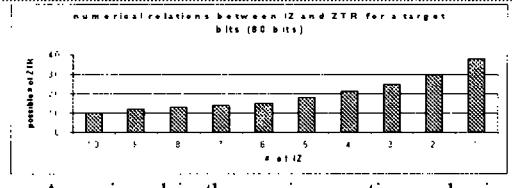
(c) One parent and nine children; an overlapped relation modified from the fixed relation.

(d) A parent and its flexible children groups

Figure III - 1. An example to explain a difference between the 1-4 relation and 1-9 relation.

Table III-1. The possible numbers of ZTR according to the variable numbers of IZ for a given budget (80 bits)

Cases	*# of POS*	*# of NEG*	*# of IZ*	*# of ZTR*	*Total # of symbols*	*Entropy (bits/sym.)*	*Coded size (bits)*
1	10	10	10	10	40	2.000	80
2	10	10	9	12	41	1.992	80
3	10	10	8	13	41	1.978	80
4	10	10	7	14	41	1.958	80
5	10	10	6	15	41	1.929	80
6	10	10	5	18	43	1.866	80
7	10	10	4	21	45	1.788	80
8	10	10	3	25	48	1.683	80
9	10	10	2	30	52	1.553	80
10	10	10	1	38	59	1.377	80



As reviewed in the previous section, a dominant pass in the EZW tells where significant coefficients with respect to a given threshold exist and which signs they have. In the pass, we use four symbols – ZTR, POS, NEG and IZ – to inform the locations and signs. Once an image is decomposed using a wavelet, the number of significant coefficients is decided. Therefore, it is the number of ZTR and IZ to decide length of a symbol stream and its entropy. We now consider the occurrence of these symbols. While a ZTR is produced when a coefficient and its descendants are insignificant, an IZ is produced when a coefficient is insignificant but some of descendants are significant.

Assumed that the numbers of POS and NEG are fixed as ten respectively, table III-1 shows relations in number between ZTR

and IZ for a targeted budget (80 bits) in the views of numerical and graphical charts. As we can see in the table, when the numbers of IZ are decreased at a rate of one symbol, see how many ZTR can be coded into a stream. For a clear comparison, we give our attention to only the cases 1 and 10. In the case 1, the number of IZ are ten and therefore ten ZTR can be coded for the target size (80 bits). On the other hand, there is only 1 IZ and thus 38 ZTR can be coded for the same target size in the case 10. Although their target sizes are the same, we can code the different number of symbols. Comparing with the case 1, the case 10 can be interpreted as the decreased nine IZ symbols are replaced with the increased 28 ZTR symbols. The ratio 28 / 9 means that one IZ has the worth of 3.1 ZTR symbols. Therefore, we conclude a fact that it is more effective for an entropy coding to use three ZTR rather than one IZ, if and only if possible. Our interests are then on a possibility to such a symbol replacement and a ratio in the replacement.

To implement the symbol replacement, we have two step procedures; one is to decrease the number of IZ symbols and the other is to replace them with several ZTR symbols in order to compensate the decrease. We first suggest a solution to decrease IZ symbols. An insignificant coefficient is coded as an IZ when some descendants are significant. In other words, the occurrence of IZ is caused from one reason that the significant descendants belong to the insignificant ancestor. Therefore, a simple solution is to suppress the occurrence; let the significant descendants belong to a significant ancestor. It is only possible that the descendants have a power to select their ancestors. In the EZW, the relation does not allow such a selection and always maintains one parent to four children; this is referred as a fixed relation. The relation can be modified with another form so that some children can select their parent. Selecting a parent means that there should be several candidates for the parent and we can imagine that a modified relation must have an overlapped form. It can be made in various forms. One of them is suggested in figure III-1 (a). Four parents are displayed in the parent level having their own shapes. These shapes help to understand the relations between parents and their children. Each parent has nine children respectively and some children are shared by several candidates to be their parent; that is, a child can belong to one or more parents. In other words, we can scan a child after a parent among all candidates. This is referred as a modified relation; one parent–nine children.

We give a simple example to explain the modified relation. Assume that there are two parents at parent level – one is significant and the other is insignificant – and six significant children (named as C1 to C6) in child level as shown in the figure III-1 (b) and (c). Our goal is to find all significant coefficients according to the scanning order that we do not scan any children before any parents. We will use two relations to find them; the fixed and the modified relations. We have seven coefficients – one parent and six children – to be found as significant coefficients. We first find them with the fixed relation as shown in figure III-1 (b). The significant parent P2 has four significant children and they are scanned under P2. However, the insignificant parent P1 has also two significant children; thus the parent should be symbolized with an IZ symbol to find these two significant children. Therefore, we make a symbol stream in this case – IS (at parent level) SZSZ, SSSS (at child level); where S,I,Z mean a significant coefficient, an isolated zero and a zerotree root respectively. The stream has seven S, three Z and one I symbols. On the other hand, when the modified relation is applied to the example as shown in figure III-1 (c), all significant children belong to one significant parent and thus we need no IZ symbol. In this case, the symbol stream is output as ZS (at parent level) ZZZSSSSSS (at child level); seven S and four Z symbols. As was shown in the above explanation, two symbol streams were obtained for the same example by using two different relations. We knew that a relation between a parent and its children plays an important role in producing a symbol stream. According to specific relations, the kind and the number of produced symbols are different and thus the resulting entropy is different. In the cases of (b) and (c), entropies are 1.157 bit/symbol and 0.946 bit/symbol respectively. Their entropy coded sizes are 11.57 bits and 10.41 bits. Comparing those two streams, we conclude that one I symbol in the case (b) was replaced with two Z symbols in the case (c). After all, when we change a relation with another, an important thing is how many Z symbols are increased instead of decreasing I symbols. The ratio between the increase and decrease will be an important factor for an entropy coding.

To decrease the ratio, we again change the modified 1-9 relation with a flexible relation. In the previous example, the 1-9 relation was more efficient than the fixed relation as no use of I symbols. However, that is only the special example to explain a relation between I and Z symbols in numbers. If the P1 were also significant in the example, the symbol stream of the case (b) would not have included any I symbol and thus only two Z symbols are needed for the case. This means that the modified relation is not always better than the fixed relation is. Therefore, we need a general relation to compromise these two relations.

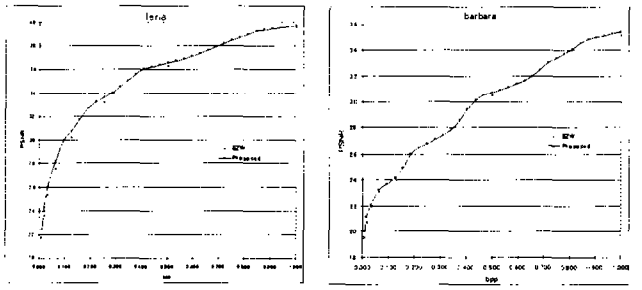
We exploit the dependencies in neighboring coefficients for that purpose. This can be realized by using a flexible relation; that is, the number of children a parent has is variable at a bound between four and nine. To define the flexible relation, we divide nine children into four groups that are named as G1,G2,G3 and G4 as shown in figure III-1 (d). A parent has the first group G1 and selectively has the rest groups of G2,G3 and G4; where each of rest groups – G2 ,G3 and G4 – is selected only when the first child in each group is significant. For example, G2 is selected when C2 is significant; in this case, the parent has G1 and G2 groups and six children C1 to C6 belong to the parent. Therefore, the G1 should be scanned before G2,G3 and G4. After all, selections of the rest groups G2,G3,G4 are determined by the significance of the children C2,C4,C5 in G1.

Back to the previous example, we apply the flexible relation. The first group G1 to the P1 has no significant children and thus P1 is coded as a zerotree root. The next parent P2 has two significant children C1 and C2 in G1; therefore, the children groups are G1,G3 and G4. In this case, the parent P2 has eight children except the second child of G2 among nine children. The resulting stream is ZS (at parent level) ZZSS (from G1) SS (from G3) SS (from G4); seven S and three Z symbols are included. Note that we do not need to scan a child twice. The flexible relation enables to decrease one more Z symbol than the modified relation.

IV. EXPERIMENTAL RESULTS

Our flexible tree is designed to reduce the number of IZ symbols and thus let the entropy lower. The decreased IZ symbols induce some increase of ZTR symbols in numbers by defining an extended relation. The ratio between the decrease and increase is efficiently exploited with the flexible treeing. We use two standard images – Lenna and Barbara (512 X 512 with a grey scaled level) – from the RPI site, ftp://ipl.rpi.edu/pub/image/still/usc. Our all results are based on 6-scaled octave wavelet transform and we use the 9/7 filter of [5] and mirror extensions at boundaries. Experimentally, the performances are compared with the published results at the reference [1] and they are plotted in fig IV-1 (a) and (b) for the two images respectively. The performances in PSNR are calculated over the ranges from 256 to 32768 Bytes. Our flexible coder shows 0.2~0.7 dB better performances than the EZW coder. The improvements are based on the symbol replacements by which we use a frequent symbol (ZTR) instead of infrequent symbol (IZ) as many as possible. We know the replacements are well accomplished with the flexible relation as shown in the performance curves.

In addition, we compare the number of symbols between the EZW and our coder. To give an exact comparison, we stop to code right after a threshold becomes 16: that is, the coding is terminated when the dominant and subordinate passes are all coded with respect to the threshold 32. The same condition is applied to the Barbara image and the results are given in table IV-1 (b). As we can see in the table, the numbers of POS and NEG symbols are the same but the compressed sizes are different. In the flexible relation, some IZ symbols are disappeared instead of some increase of ZTR symbols. As we can see in the table IV-1, we get different symbol streams from those two coders respectively. Comparing with the number of produced symbols in the EZW, our coder produce 1575 less IZ symbols and 4704 more ZTR symbols for the Barbara image. Therefore, the ratio can be calculated by dividing the increase in ZTR by the decrease in IZ; 4704 / 1575 = 2.99. The value 2.99 means that one IZ symbol was replaced with 2.99 ZTR symbols. The decreased IZ symbols play a part in lowering an entropy and therefore the image can be compressed with a smaller size. We can calculate each entropy for symbol distribution; 1.274 bits/sym. and 1.195 bits/sym. for the EZW and the proposed coder respectively. By using the low entropy, we can compress more compactly, though total number of symbols is increased.



(a) Lenna (b) Barbara

Figure IV-1. performance curves for the test images.

Table IV-1. Comparisons of produced symbols in numbers for the same number of significant coefficients.

Used coder	Compressed size (Bytes)	PSNR (dB)	Compression ratio	# of POS	# of NEG	# of ZTR	# of IZ
EZW	6511	32.51	40.26 : 1	3876	3743	43249	1566
Proposed	6328	32.51	41.43 : 1	3876	3743	46829	1369

(a) Lenna (512 X 512, 8 bits grey image, original size = 262144 Bytes)

Used coder	Compressed size (Bytes)	PSNR (dB)	Compression ratio	# of POS	# of NEG	# of ZTR	# of IZ
EZW	15878	30.05	16.51 : 1	9741	9586	72003	7603
Proposed	14584	30.05	17.97 : 1	9741	9586	76707	6028

(b) Barbara (512 X 512, 8 bits grey image, original size = 262144 Bytes)

V. CONCLUSIONS

We described a new relation that a parent takes its children with a flexible and selectable method. We extend the fixed relation used in the EZW scheme in order to decrease entropy per symbol. The ways to lower the entropy are accomplished by using more symbols that are frequent and less symbols that are infrequent. The infrequent symbol is IZ in the EZW and we can avoid the use of the symbol by extending the relation in parent-child; a parent has nine children and some of them are shared with neighboring parents. With the extended relation, the number of IZ symbol is decreased.

We showed that a symbol stream is coded with less entropy using the flexible relation in parent-child. Experimentally, our flexible coder has 0.2 ~ 0.7 dB better performances than the EZW's. We suppose that the flexible coder can be improved by a more efficient relation in parent-child.

ACKNOWLEDGMENTS: This work was in part supported by the Grant-in-Aid for Scientific Research, No. 07650419, from the Ministry of Education, Science and Culture of Japan and the research grant by Telecommunications Advancement Organization of Japan.

REFERENCES

1. J.M.Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, 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, no. 3, pp. 243-250, June 1996.
3. Z.Xiong, K.Ramchandran and M.T.Orchard, "Space-Frequency Quantization for Wavelet Image Coding", *IEEE Trans. On Image Processing*, Vol. 6, No. 5, pp. 677-693, May 1997
4. Z.Xiong, K.Ramchandran and M.T.Orchard, "Wavelet packets image coding using space-frequency quantization," *IEEE Trans. on Image Processing*, January 1996.
5. I.H. Witten, R. Neal, and J.G. Cleary, "Arithmetic Coding for Data Compression", *Comm. ACM*, Vol. 30, No. 6, pp. 520 - 540, June 1987.
6. G. Strang, T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.
7. John W. Woods, *Subband image coding*, Kluwer Academic Publishers, Boston, MA, 1991.
8. S.H.Joo, H. Kikuchi, J.Shin, "A Flexible relationship in parent-child on embedded image coding", *Proceedings of IEICE Spring Conference*, vol. 1, no. 2, pp. 17, Mar. 1997.