Reversible EZW-Based Image Compression Using Best-Transform Selection and Selective Partial Embedding

Michael D. Adams and Andreas Antoniou

Abstract

A best-transform selection method and a selective partial embedding technique are proposed for use in a reversible embedded image compression system based on the embedded zerotree wavelet (EZW) coding scheme. Together, these approaches are shown to yield significantly better compression performance than a baseline system employing fully embedded EZW coding in conjunction with a single fixed transform such as the S+P transform.

Keywords

Reversible embedded image compression, wavelet/subband coding.

I. Introduction

In recent years, there has been a growing interest in reversible embedded image compression systems [1–3]. One of the main advantages of such systems is their ability to efficiently handle progressive lossy-to-lossless transmission of images. In this paper, we describe two techniques for improving the lossless coding performance of a reversible embedded image compression system based on the embedded zerotree wavelet (EZW) coding scheme [4]. In Section II, we propose a best-transform selection scheme, its main benefits being simplicity and compatibility with efficient lossless compression. In Section III, we adapt the EZW scheme to handle reversible compression, and propose a selective partial embedding technique that improves performance by relaxing the embedding constraint at high bit rates. In Section IV, we combine these two techniques and analyze the performance of the resulting system. As will be seen, the performance of our modified coder is much better than that of a baseline system employing the fully embedded EZW coding scheme in conjunction with a single fixed transform.

II. Best-Transform Approach

Recently, Calderbank et al. [5] proposed a systematic lifting-based method for constructing reversible integer-to-integer versions of linear wavelet transforms. Using this technique, one can generate a large number of reversible transforms that are effective for image coding purposes. With many excellent transforms at our disposal, it is possible to consider a best-transform approach to reversible embedded image coding. By selecting a transform that is better suited to a given image, improved compression performance can be achieved. There are, however, two problems associated with such a scheme. The first problem is deciding upon a set of candidate transforms which can potentially be used during the coding process. The second problem, and also the more difficult one, is to find efficient and reliable methods for selecting an appropriate transform to use for a given image from a particular set of candidate transforms.

Manuscript received XXX; revised XXX. This paper was recommended by Associate Editor N. R. Shanbhag. This work was supported by Micronet, Networks of Centres of Excellence Program, and the Natural Sciences and Engineering Research Council of Canada.

M. D. Adams was with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada V8W 3P6. He is now with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4 (e-mail: mdadams@ece.ubc.ca).

A. Antoniou is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada V8W 3P6 (e-mail: andreas@ece.uvic.ca).

Publisher Item Identifier XXX.

In our best-transform approach, we initially considered the reversible transforms listed in Table I as potential members of the candidate set. These transforms were constructed using the lifting-based method mentioned earlier or a generalization thereof, and are amongst some of the best known for image compression. For these particular transforms, experimentation has shown that the smoothness and uniformity of an image largely determine which transform is most effective for lossless compression purposes. For images characterized by smooth intensity variations and a lack of very large regions of uniform intensity (e.g., natural imagery), the 13/7, S+P, 9/7, and 5/11 transforms tend to perform quite well, with the 13/7 transform being the most effective. For images with moderate amounts of nonsmooth variation (e.g., some computer generated images), or smoothly varying images with very large regions of uniform intensity (e.g., a photograph with a large proportion of sky), the 5/3 transform performs very well and the 9/3 transform yields reasonable, but less impressive, results. For images characterized by much nonsmooth variation and large regions of uniform intensity (e.g., images consisting of mixed text and graphics), the Haar transform is quite effective. Since most images fall into one of the above three categories, it is only natural to select a candidate set with three members, one for each image class. Choosing the best transform for each of these classes, we obtain a candidate set consisting of the Haar, 5/3, and 13/7 transforms.

The observations made above immediately suggest two characteristics of an image that could be used for transform selection: 1) the smoothness of intensity variations, and 2) the uniformity of intensity (i.e., the proportion of an image constituted by regions of constant intensity). In order to use these two characteristics for transform selection, two simple statistics based on first-order pixel differences were used. Suppose that we are given an image where the pixels assume values from the set $\{0, 1, \ldots, R-1\}$. The differences between adjacent pixels in the horizontal and vertical directions are sampled as depicted in Figure 1. For convenience, denote these differences as d_i for $i = 0, 1, \ldots, N-1$. In terms of these differences, we define the smoothness measure $s \triangleq 100N_s/N$ where N_s is the number of d_i such that $|d_i| \geq R/2$. This metric can assume values on the interval $[0, 100]$ with smaller values corresponding to greater smoothness. Similarly, we define the uniformity measure $u \triangleq 100N_u/N$ where N_u is the number of d_i such that $d_i = 0$. This metric can assume values on the interval $[0, 100]$ with a larger value corresponding to greater uniformity. For a given image, we first calculate the smoothness metric s and uniformity metric u for the image as defined above. Then, we select the decorrelating transform to use as indicated in Table II. This selection process incurs a minimal amount of additional overhead, typically only a small fraction of the computation necessary for a single-level Haar wavelet decomposition.

The Haar transform often introduces undesirable blocking artifacts when used for lossy compression. If such artifacts are deemed unacceptable for a particular application, the 5/3 transform can be used in place of the Haar transform. In almost all observed cases, if the Haar transform performs better than the 13/7 transform, the 5/3 transform will also lead to better results than the 13/7 transform.

In order to evaluate the effectiveness of the proposed method, it was employed in a reversible embedded image compression system based on the EZW coding scheme (described later in this paper). For test data, a suite of 25 grayscale images was used (as listed in Table B.1 of [3]), covering images with a wide range of sizes and content.

Each of the 25 test images was coded in a lossless manner, and a representative subset of the results is given in Table III. The results for each individual transform (i.e., Haar, 5/3, 9/3, 9/7, 5/11, 13/7, S+P) are given along with the results obtained with our proposed scheme. Since our method can be used both with and without the Haar transform, two sets of results are provided for this method. For each image, the result associated with the transform selected by our method is highlighted. As is evident from these numbers, the proposed method yields better results than those obtained with any of the other permissible transforms individually.

III. Improving the EZW Coding Scheme

The EZW coding scheme [4] has proven to be an effective technique for coding the coefficients of a wavelet decomposition. Although this coding scheme was initially proposed for use in lossy image coders, it can also be used for lossless compression provided that the decorrelating transform employed by the codec is truly invertible. For the most part, our implementation of the EZW coding scheme is a faithful reproduction of that described in Shapiro's original paper with the following changes: 1) The coding scheme was extended to handle arbitrary-sized images. 2) In order to maintain better locality between successive coefficients, and thus, improve compression efficiency, the scan order for coefficients within a subband was changed from the raster-scan order suggested by Shapiro to the zig-zag order shown in Figure 2. 3) To accommodate both lossless coding and the use of reversible integer-to-integer wavelet transforms, additional changes were required as described below.

By applying the (forward) wavelet transform to an image, we obtain a transformed image with a subband structure like that shown in Figure 3. The LL_n, LH_n, HL_n, and HH_n labels are used to denote the various subbands. In order to obtain good lossy compression performance (in the mean-squared error sense) with the EZW coding scheme, the decorrelating transform needs to be near-orthogonal. Often, reversible transforms must have their coefficients weighted on a per-subband basis to approximate a near-orthogonal transform. Thus, in general, we must code a weighted version of the transform coefficients. For the purposes of our presentation here, we will assume that the weights are always powers of two. If this is not the case, one can approximate the true weights with such values.

After weighting, it is often the case that some transform coefficients cannot have nonzero bits in their least significant bit (LSB) positions. Obviously, the bits that are known to always be zero do not need to be coded. This leads to a further change in the EZW coding algorithm. That is, an entry should be removed from the subordinate list once all of its potentially nonzero bits have been coded.

In the context of reversible embedded image compression using the set partitioning in hierarchical trees (SPIHT) coding scheme, Said and Pearlman [6] suggest an approach whereby all but the three LSBs of the (weighted) transform coefficients are coded using an embedded scheme, and then the remaining bits are coded using a simple nonembedded method. Although they did not describe their nonembedded coding method in detail, they did make a software implementation available (see [6]). Switching to a nonembedded coding method at high bit rates was motivated mainly by the fact that the SPIHT coding scheme becomes less efficient in terms of memory and CPU usage at such rates. For some classes of images, this technique also leads to significantly better lossless compression performance. With improved lossless compression performance as a motivating factor, we applied the technique to the EZW coding scheme. Based on experimentation, we also made some minor changes to the technique which result in slightly better compression performance. We will now discuss this modified scheme.

This nonembedded residual coding scheme is based on the idea of coding the remaining bits of the transform coefficients as small integers. The resulting symbols are then entropy coded using context-based adaptive arithmetic coding [7]. Transform coefficient information for each subband is coded independently with the arithmetic coder being re-initialized at the start of each subband. The process starts with the lowest frequency subband, and proceeds towards the highest frequency subband as shown in Figure 4. When a subband is encountered that has no bits that need to be coded due to coefficient weighting, the subband is simply skipped without any output being generated.

Within a subband, coefficients are processed in the same zig-zag order as shown in Figure 2. Since the same weighting is always applied to all coefficients in a particular subband, all coefficients in a subband always have the same number of magnitude bits remaining to be coded. Suppose that there are n such bits where $0 < n \leq 3$. There are two possibilities to consider:

1. The coefficient has previously been found to be significant. In this case, the sign of the coefficient has already been coded on an earlier dominant pass. Thus, we only need to code the n magnitude bits. This leads to an alphabet of size 2^n, corresponding to the integers $0, 1, 2, \ldots, 2^n - 1$.
2. The coefficient has not previously been found to be significant. In this case, the coefficient's sign has not yet been coded. Thus, we need to code the n magnitude bits plus the sign of the coefficient. This leads to an alphabet of size $2^{n+1} - 1$, corresponding to the integers $0, \pm 1, \pm 2, \ldots, \pm(2^n - 1)$.

Five conditioning contexts are used for each of the above two cases, for a total of ten in all. In both cases, the choice of context is made in the same manner. Two coefficients in a causal neighborhood surrounding the coefficient currently being processed are examined. Normally, we choose the following two coefficients: 1) the coefficient that was processed immediately before the current one, and 2) the coefficient directly above the current one. In cases where one (or both) of these coefficients is not well-defined, an alternative choice is made. Several different scenarios are depicted in Figure 5. In each case, the box marked with an 'X' represents the current coefficient, the arrow indicates the direction of the scan, and the boxes labelled a_0 and a_1 represent the two coefficients used for conditioning. Let the values of the two coefficients used for conditioning (before weighting) be denoted as a_0 and a_1. The context is then chosen as $c = \min(\lfloor (b_0 + b_1)/2 \rfloor, 4)$ where

$$b_i = \begin{cases} \lfloor \log_2 |a_i| \rfloor + 1 & \text{if } a_i \neq 0 \\ 0 & \text{if } a_i = 0. \end{cases}$$

This always yields a context in the range 0 to 4 inclusive. Depending on whether or not the coefficient under consideration was already found to be significant, we select from either the first or second set of five contexts. After each new symbol is output, the probabilities associated with the selected context are updated.

In the process described above, no potentially time-consuming significance tests and manipulation of dominant/subordinate lists are required. Furthermore, only trivial calculations are required in order to determine the next symbol to entropy code. Therefore, this process has much lower computational requirements than EZW coding.

The EZW coding scheme proper, as discussed previously, always produces a fully embedded bit stream. Here, we compare fully embedded EZW coding with a partially embedded variant based on the nonembedded residual coding method described above. Note that the choice of full or partial embedding only affects the rate-distortion behavior at very high bit rates, typically at compression ratios less than 8. Since we are rarely interested in lossy compression at such low compression ratios, the difference in lossy results is of little concern and is not considered further.

In both the partially and fully embedded cases, the first part of the bit stream is generated using the EZW coding scheme. The difference in the two approaches is what happens once the third LSB of the (weighted) transform coefficients is reached. In the fully embedded case, nothing changes, and the EZW coding scheme is used to code the rest of the coefficient bits. With the partially embedded approach, the remaining bits of the transform coefficients are coded explicitly.

It is important to observe that there is one fundamental difference between the two approaches. When partial embedding is employed, all of the remaining (i.e., uncoded) transform coefficient bits must be coded explicitly. In the case of full embedding, however, not all of the remaining bits necessarily need to be coded explicitly. This is because, in a sense, the EZW coding scheme only codes nonzero coefficients. Thus, if a large number of the remaining bits correspond to zero-valued coefficients, very little additional information needs to be output. Also, the EZW coding scheme has the property that it can very efficiently encode trees of zero bits. Thus, even if the remaining (i.e., uncoded) bits correspond to nonzero coefficient values, so long as these bits are mostly zero, the EZW coding scheme can encode them extremely efficiently. Thus, one might speculate that full embedding would perform better in cases where the three LSBs of the (weighted) transform coefficients tend to be zero. Otherwise, the partially embedded approach would likely be more effective. Experimental results have confirmed this hypothesis, and furthermore it was found that full embedding typically yields better compression results in cases where more than one-third of the weighted transform coefficients with bits remaining to be coded have all zeros in their three LSB positions. This suggests a further improvement for our image codec. One could easily choose between full and partial embedding on the basis of how many weighted transform coefficients with bits remaining to be coded have all three of their LSBs equal to zero. If more than 33% of the coefficients satisfy this condition, full embedding would be selected; otherwise, partial embedding would be used. Although one additional bit in the output stream is required to differentiate between the partially and fully embedded cases, the potential savings are far more significant.

In order to evaluate the effectiveness of our technique for selecting between full and partial embedding, we again considered the 25 test images from Section II. For the purposes of this analysis, the Haar, 5/3, and 13/7 transforms were employed. Each image was compressed in a lossless manner using full embedding, partial embedding, and our selective partial embedding scheme. A representative subset of the results is given in Table IV. For each image/transform pair, the scheme selected by our method is highlighted. As is readily apparent, our scheme is quite effective at predicting the best coding technique. Considering the complete set of 25 test images, our scheme yields the best compression performance in almost all cases.

IV. Combined Results

Although the techniques described in Sections II and III were introduced in isolation, they can be used together to good effect. To demonstrate this, we employed both techniques in combination in order to compress the 25 test images in a lossless manner. The same images were also compressed using a baseline coder employing the fully embedded EZW coding scheme in conjunction with the S+P transform. Some representative results are given in Table V. In almost all cases, the modified codec utilizing our proposed techniques outperforms the baseline codec.

V. Conclusions

Two techniques were proposed for improving the compression performance of a reversible embedded image codec based on the EZW coding scheme. As our results demonstrate, by using both of these techniques, one can make significant gains in compression performance without incurring excessive computational overhead. The resulting system almost always outperforms a baseline system using fully embedded EZW coding in conjunction with a single fixed transform such as the S+P transform.

REFERENCES

[1] 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.

[2] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, "CREW: Compression with reversible embedded wavelets," in *Proc. of IEEE Data Compression Conference*, Snowbird, UT, USA, Mar. 1995, pp. 212–221.

[3] M. D. Adams, "Reversible wavelet transforms and their application to embedded image compression," M.A.Sc. thesis, Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, Jan. 1998, Available from `http://www.ece.ubc.ca/~mdadams`.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[5] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.

[6] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. on Image Processing*, vol. 5, no. 9, pp. 1303–1310, Sept. 1996.

[7] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.

List of Figures

List of Tables

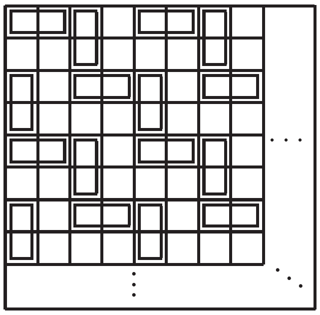

Fig. 1. Pairs of pixels used in difference calculations.

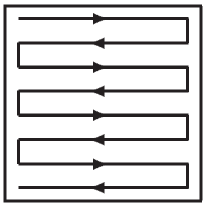

Fig. 2. Coefficient scan order.

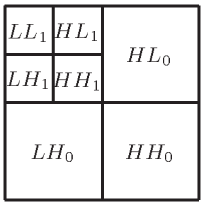

Fig. 3. Subbands of wavelet decomposition.

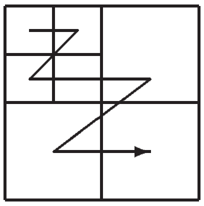

Fig. 4. Subband scan order.

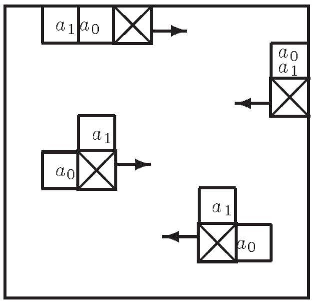

Fig. 5. Neighboring coefficients used for conditioning.

TABLE I
REVERSIBLE TRANSFORMS

Name	Description/References
Haar	Haar Transform $\leftrightarrow$ [3], [6], equation (3.3) in [5]
5/3	[3], equation (4.1) in [5]
9/3	[3], equation (4.3) in [5]
9/7	[3], equation (4.2) in [5]
13/7	[3], equation (4.5) in [5]
5/11	equation (4.6) in [5]
S+P	S+P Transform with Predictor B $\leftrightarrow$ [3], [6]

TABLE II
TRANSFORM SELECTION CRITERIA

Condition	Transform
$0 \leq u < 20$ and $0 \leq s < 0.25$ $20 \leq u < 40$ and $s < -0.01u + 0.45$ $40 \leq u < 75$ and $0 \leq s < 0.05$	13/7
$0 \leq u < 25$ and $5 \leq s \leq 100$ $25 \leq u < 50$ and $2 \leq s \leq 100$ $50 \leq u \leq 100$ and $1 \leq s \leq 100$	Haar
otherwise	5/3

TABLE III
LOSSLESS COMPRESSION RESULTS FOR BEST-TRANSFORM APPROACH

	Compression Ratio							
Image	Haar	5/3	9/3	9/7	5/11	13/7	S+P	Proposed with/without Haar
barb	1.541	1.669	1.675	1.706	1.705	**1.715**	1.693	1.715
chart_s	2.374	**2.449**	2.425	2.400	2.394	2.395	2.416	2.449
cmpnd1	**4.889**	3.817	3.486	3.146	3.074	3.074	3.559	4.889/3.817
finger	1.269	1.435	1.429	1.473	1.472	**1.472**	1.440	1.472
lena	1.721	1.858	1.853	1.873	1.870	**1.874**	1.859	1.874
molecule	4.549	**4.231**	4.193	4.094	4.078	4.097	4.148	4.231
Mean	2.481	2.316	2.251	2.203	2.188	2.193	2.253	2.573/2.333

TABLE IV
LOSSLESS COMPRESSION RESULTS FOR FULL VERSUS PARTIAL EMBEDDING

	Compression Ratio								
	Haar			5/3			13/7		
Image	Full	Partial	Selective	Full	Partial	Selective	Full	Partial	Selective
barb	1.541	**1.556**	1.556	1.669	**1.687**	1.687	1.715	**1.734**	1.734
chart_s	**2.374**	2.364	2.374	**2.449**	2.450	2.449	**2.395**	2.404	2.395
cmpnd1	**4.889**	4.689	4.889	**3.817**	3.701	3.817	**3.074**	3.047	3.074
finger	1.269	**1.272**	1.272	1.435	**1.439**	1.439	1.472	**1.475**	1.475
lena	1.721	**1.736**	1.736	1.858	**1.875**	1.875	1.874	**1.893**	1.893
molecule	**4.549**	4.441	4.549	**4.231**	4.106	4.231	**4.097**	4.000	4.097
Mean	2.481	2.436	2.484	2.316	2.293	2.320	2.193	2.189	2.197

TABLE V

LOSSLESS COMPRESSION RESULTS FOR COMBINED APPROACHES

Image	Compression Ratio	
	S+P and Full Embedding	Proposed with/without Haar
barb	1.693	1.734
chart_s	2.416	2.449
cmpnd1	3.559	4.889/3.817
finger	1.440	1.475
lena	1.859	1.893
molecule	4.148	4.231
Mean	2.253	2.577/2.337