A SURVEY OF THE STATE-OF-THE-ART AND UTILIZATION OF EMBEDDED, TREE-BASED CODING

William A. Pearlman

Rensselaer Polytechnic Institute
ECSE Dept.
Troy, NY 12180 — pearlman@ecse.rpi.edu

Amir Said

Iterated Systems, Inc.
3525 Piedmont Road, Suite 7/600
Atlanta, GA 30025 — said@ipl.rpi.edu

ABSTRACT

We review first the development of tree-based, embedded image coding. We then explore their use with different image transformations, the reasons for their effectiveness and low complexity, their flexibility and incorporation into state-of-the-art compression systems.

1. INTRODUCTION

Six years after their introduction, tree-based algorithms for embedded image coding established a firm reputation as some of the most powerful, efficient and versatile methods known for image compression. They are now studied by a very active group of researchers, and this paper presents a short survey of the state-of-the-art in the field.

Interestingly, those compression methods initially faced a certain skepticism. They were able to support very desirable features, like embedded coding, fast compression and decompression, very precise distortion or rate control, support for lossless compression, great scalability, no need for training, etc. At the same time, they had a performance superior to even the methods painfully adapted to support only one (or a few) of those features. For some, it seemed definitely too good to be true.

One possible reason for this skepticism was that they use two techniques that had been used for some time, but which had been yielding mediocre results. The first technique is tree-based recursive partitioning schemes—as employed in quadtree image compression [26]. The second is bit-plane [13], which permits progressive image transmission with gradual quality improvement.

The origins of these methods can be traced back to when subband/wavelet [30, 31] image coding methods began to gather wider acceptance, as even very simple schemes could produce remarkably good visual quality, together with progressive image transmission. Lewis and Knowles [9] were the first to use trees to exploit the statistical properties found in the pyramidal decomposition of natural images. They proposed an efficient decomposition and coding scheme using tree structures that follow the same spatial location across different subbands, which we call spatial orientation trees.

Then, Shapiro [21, 22] proposed a very clever method to combine bit-plane coding, applied to the wavelet coefficients, with a tree-based partitioning similar to the one developed by Lewis and Knowles. This combination, called embedded zerotree of wavelets (EZW), is by no means a trivial juxtaposition of methods. It identified that efficient compression required separately coding the significance data (to identify if a pixel magnitude is larger than a threshold) and the rest of the bit planes. Further, it shows how much tree structures are efficient to code significance data of wavelet image pyramids.

Shortly after the EZW introduction, we presented a more general analysis of the algorithm [14, 15], relating it to some selection (sorting) and set-partitioning problems. We also proposed some changes that improved its performance, and have shown how to use it for lossless image compression [16, 18]. Later we studied how to improve the method to achieve better embedding, faster coding and decoding, simpler and more efficient entropy-coding, and less memory usage. The result was a new algorithm, called SPIHT, that indeed achieved all those goals [17].

In this paper we present a cross section of what happened in the field recently. Due to lack of space the survey is not supposed to be comprehensive, and we apologize for all missing references. Also, we had to exclude important work that is embedded but not tree-based, and tree-based but not embedded, extensions to video, etc.

2. USE WITH DIFFERENT IMAGE TRANSFORMATIONS

One common misconception about tree-based embedded methods is that they work only with wavelets. They were indeed designed to exploit special properties of the wavelet pyramid coefficients, under a particular tree structure. Nonetheless, a number of experiments show that they are quite effective for other transformations as well. For instance, they were sucessfuly employed together with 8×8 DCTs, and even better results are obtained with 16×16 DCTs, [32, 11].

Among several exciting developments we have schemes for wavelet packets [33] and general lapped block transforms [28]. The lifting scheme [27] allows the development of better transforms for lossless compression [1], and wavelets on manifolds [20]. Tree-based embedded coding methods are readily adapted anywhere the lifting scheme can be used. For example, an interesting application is efficient (and embedded) compression of functions defined on

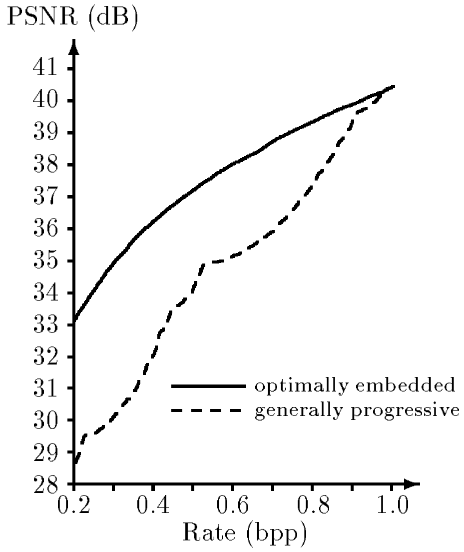

Figure 1: Performance of different preogressive transmission methods when used for embedded coding.

the surface of sphere [6].

While reading this paper the reader should keep in mind that we are discussing a coding algorithm, and as such, its performance (compression efficiency, memory usage, reconstruction visual quality, etc.) depends on the image transformation being used.

3. OPTIMIZED EMBEDDED CODING AND NON-EMBEDDED CODING

Shapiro [22] correctly defined his embedded coding with two objectives: "(1) obtaining the best image quality for a given bit rate; (2) all encodings of the same image at lower bit rates are embedded in the beginning of the bit stream for the target bit rate." Without the first condition, almost any coding method can be called embedded, but possibly with dismal performance at intermediary rates. For example, Figure 1 shows the performance[1] of SPIHT [17], compared to a method designed for progressive resolution [19], adapted for embedded coding.

Unfortunately, now the literature is filled with methods using the expression embedded coding incorrectly. In fact, image pyramids are sometimes called "embedded wavelets," so that any coding method using them is also called embedded.

It is important to observe that bit-plane embedded coding does not exclude the possibility of using other forms of progressive transmission. For instance, with the proper file format it is possible to recover subbands independently. An impressive application based on this fact is a program that allows real-time enhancement—up to lossless recovery—in an user-selected region [4].

[1] $\mathrm{PSNR} = 10\log_{10}(255^2/(\text{mean squared error}))$.

Embedded coding necessarily adds some extra complexity to the coding algorithm, as bit planes are sequentially visited, and it is true that embedded coding methods may not be as fast as the fastest non-embedded methods. However, embedded tree-based methods are not much worse either. First, it should be noted the computational effort spent in the first passes is insignificant compared to the effort in the last pass. So, it is not too different from a single-pass algorithm (non tree-based do not have this property). Second, it is not a fair comparison, as there are "shortcuts" that can be taken when embedded coding is not required.

4. THEORETICAL ANALYSIS

The development of tree-based embedded coding methods followed an unusual path: a rigorous theoretical basis is being discovered many years after the practical methods were devised from empirical analysis of distributions in wavelet pyramids formed from natural images. Little was known about which images would be "natural" enough to yield similar results. In the same token, little was known about how to use these methods with different image transformations, like DCTs.

Davis and Chawla [5] proposed a method based on dynamic programming to find the optimal tree structures (set-partitioning rules) for different transformations (like DCTs). Surprisingly, the variation found among different trees can be quite small, showing that tree-based algorithms are by nature quite robust.

New work [25] model the dependence that exist between the magnitudes of wavelet coefficients in different scales, and orientation. This phenomenon originated the use of the expression "self-similarity" to explain the efficacy of tree-based coding. However, it was never used in [21, 22, 15, 17] to mean fractal self-similarity: it meant similarity in a distribution sense. So, in contrast with theories based on conceptual analysis [8], the statistical models above provide not only an explanation to the success of tree-based coding with wavelets, but also to other transforms.

5. COMPUTATIONAL COMPLEXITY

The speed of bit-plane embedded coding methods is defined by the need to perform several passes on the image, comparing magnitudes each pass. The tree-based methods have an advantage over other methods because the necessary information is stored in linked lists, which are, for the most common bit-rates, much smaller than the total number of pixels. Experiments show that most of the CPU time is spent testing for significance, and a small fraction is used for pixel value refinement.

The encoder needs to first compute the maximum magnitudes of the pixels in the trees, but this can be done with very efficiently with a single pass of binary-OR operations done during the image transformation [23]. So, this pass through the whole image is computationally equivalent to a single pass of other bit-plane algorithms.

Next, both the encoder and decoder have to maintain a dynamic list of coefficients. Shapiro's EZW algorithm [22] adds four descendants to the lists whenever a parent or descendant is significant. One improvement is to add descendants only when at least one descendant is significant [14, 15]. This yields a very significant reduction in the number of elements in the lists, with proportional reduction in CPU time and memory required for list processing.

The next speed bottleneck was the requirement of using arithmetic coding for effective compression. The following breakthrough was the proposal of a different—and even more compact—tree set partitioning [17], which would keep more insignificant descendants grouped together. This representation was efficient enough to yield excellent results, even without any form of entropy-coding. Furthermore, it allows dealing with groups of 2×2 pixels/trees, for another significant reduction of list management resources.

The first implementations had nearly equal encoding and decoding times, but this was due to the arithmetic code. With faster entropy-coding the decoder can be quite faster because it does not require computation of maximum magnitudes.

Those advanced new coding algorithms yielded such large reduction in the coding/decoding complexity, that, in a software implementation, coding/decoding became much faster than the image transformation. Even though it is frequently thought that list processing is a complex task, it is a well known programming problem, and thus has been extensively studied and optimized. Consequently, tree-based decoding algorithms can be faster than any other algorithm, because they process only a small fraction of the image pixels.

Significant improvements have been proposed for specialized architectures, and parallel processing [2]. A method to exploit embedded coding to reduce the complexity of the wavelet forward and reverse transformation is proposed by Paris et al. [12].

6. RESILIENCE TO TRANSMISSION ERRORS

Error control is done by adding redundant bits, or by asking for retransmission. This cause an overhead on the bandwith that depends on the error rates, and the efficiency of the protection method. Transmission errors can drastically reduce the effective bandwidth if improper error control procedures are used. For instance, assuring no data errors all the time is indispensable for text or database data, but it is not necessary for still images and video. This happens because we can still visually gather most of the information if only some parts are degraded. However, in all efficient compression methods an error in the compressed data can lead to a possibly long sequence of errors, or a catastrophic loss of all data after the error.

As explained above, embedded coding is, by definition, based on sorting the image components according to their importance. This can be extremely useful when designing error control systems, as error protection resources can be assigned according to the importance of the data.

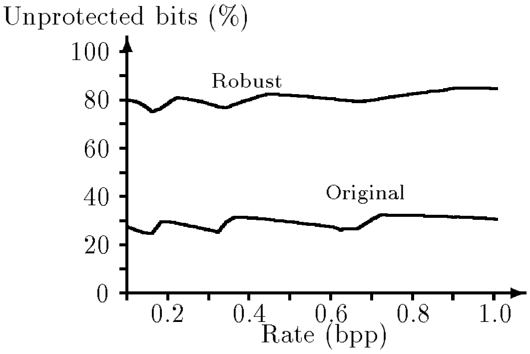

Figure 2: Fraction of bits that do not need error protection, according to embedded coding method.

Bit-plane coding helps in two ways. First, it guarantees that only the data below a certain plane is corrupted after an error. Second, the least significant bits do not require entropy coding. So, for those bits it is not necessary to use error protection because without variable-length coding there is no catastrophic error propagation.

On the other hand, all error in significance data do lead to catastrophic error propagation. The significance data can be protected with common error-correction codes. An interesting alternative it to slightly change the coding algorithm [10] to avoid error propagation. Figure 2 shows the fraction of bits that do not lead to error propagation in a bit-plane compressed file (assuming a previous wavelet decomposition), using the original and modified coding methods.

Several other ingenious solutions have been proposed to improve resilience to transmission errors, like special transmission protocols, adapted algorithms, combination with error correction, etc. [3, 24, 29].

7. CONCLUSIONS

We have traced the development of tree-based, embedded coding techniques to Lewis and Knowles [9] and Shapiro [22] and have shown their current realization in SPIHT coding and several variants to accommodate more features, such as resilience to channel errors and region-of-interest coding. It is apparent that, due to the high performance, low complexity, and versatility of these techniques, they represent the current state-of-the-art in image coding.

8. REFERENCES

[1] R. Calderbank, I. Daubechies, W. Sweldens and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," Proc. IEEE Conf. Image Processing, 1997.

[2] C.D. Creusere, "Image coding using parallel implementations of the embedded zerotree wavelet algorithm," Proc. SPIE, Digital Video Compression: Algorithms and Technologies, vol. 2668, pp. 82–92, March 1996.

[3] C.D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," IEEE Trans. Image Processing, vol. 6, pp. 1436–1442, Oct. 1997.

[4] C.D. Creusere, "Spatially partitioned lossless image compression in an embedded framework," Proc. Asilomar Conf. Signals, Syst.,& Computers, Pacific Grove, CA, Nov. 1997.

[5] G.M. Davis and S. Chawla, "Image coding using optimized significance tree quantization," Proc. Data Compression Conf., pp. 387–396, Snowbird, Utah, March 1997.

[6] K. Kolarov and W. Lynch, "Compression of functions defined on surfaces of 3D objects," Proc. Data Compression Conf., pp. 251–260, Snowbird, Utah, March 1997.

[7] B.-J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," Proc. Data Compression Conf., pp. 251–260, Snowbird, Utah, March 1997.

[8] O. Kiselov and P. Fisher, "Image compression with iterated functions systems, finite automata and zerotrees: grand unification," Proc. Data Compression Conf., p. 443, Snowbird, Utah, March 1996.

[9] A.S. Lewis and G. Knowles, "A 64 Kb/s video codec using the 2-D wavelet transform," Proc. Data Compression Conf., pp. 196–201, Snowbird, Utah, 1991.

[10] H. Man, F. Kossentini, M.J.T. Smith, "Robust EZW Image Coding for Noisy Channels," IEEE Signal Processing Letters, vol. 4(9), pp. 227–229, Aug. 1997.

[11] D.M. Monro and G.J. Dickson, "Zerotree Coding of DCT Coefficients," Proc. IEEE Int. Conf. Image Processing, 1997.

[12] J. Paris, J. Alamanac, R. Mateu, X. Masvidal, and X. Ginesta, "Low bit rate software-only wavelet video coding," IEEE First Workshop Multimedia Signal Processing, pp. 169–174, Princeton, NJ, June 1997.

[13] M. Rabbani, and P.W. Jones, Digital Image Compression Techniques, SPIE Opt. Eng. Press, Bellingham, Washington, 1991.

[14] A. Said, An Improved Zero-Tree Algorithm for Image Compression, Image Processing Lab. Report IPL TR–122, Rensselaer Polytechnic Institute, Troy, NY, Nov. 1992.

[15] A. Said and W.A. Pearlman, "Image compression using the spatial-orientation tree," IEEE Int. Symp. Circuits & Systems, Chicago, IL, pp. 279–282, May, 1993.

[16] A. Said and W.A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding," SPIE Conf. Visual Commun. Image Processing, Proc. SPIE 2094, pp. 664–674, Cambridge, MA, Nov. 1993.

[17] A. Said and W.A. Pearlman, "A new fast and efficient codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 243–250, June 1996.

[18] A. Said and W.A. Pearlman, "An image multiresolution for lossless and lossy compression," IEEE Trans. Image Processing, vol. 5, pp. 1303-1310, Sept. 1996.

[19] A. Said and W.A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," Proc. SPIE Conf. Visual Commun. Image Processing, vol. 3024, pp. 25–37, San Jose, CA, Feb. 1997.

[20] P. Schöder and W. Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," Computer Graphics Proc., (SIGGRAPH) pp. 161–172, 1995.

[21] J.M. Shapiro, "An embedded wavelet hierarchical image coder," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, San Francisco, CA, pp. IV 657–660, March 1992.

[22] J.M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," IEEE Trans. Signal Processing, vol. 41, pp. 3445–3462, Dec. 1993.

[23] J.M. Shapiro, "A fast technique for identifying zerotrees in the EZW algorithm," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 3, pp. 1455–1458, May 1996.

[24] P.G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," IEEE Signal Processing Letters, vol. 4(7), pp. 189–191, July 1997.

[25] E.P. Simoncelli, "Statistical models for images: compression, restoration and synthesis," Asilomar Conf. Signals, Syst., Computers, Pacific Grove, CA, Nov. 1997.

[26] G.J. Sullivan and R.L. Baker, "Efficient quadtree coding of images and video," IEEE Trans. Image Processing, vol. 3, pp. 327–331, May 1994.

[27] W. Sweldens, The Lifting Scheme: A Custom-Design Constructions of Biorthogonal Wavelets, Tech. Paper 1994:7, IMI, Dept. of Mathematics, Univ. of South Carolina, 1994.

[28] T. Tran and T. Nguyen, "A progressive transmission image coder using linear phase paraunitary filter banks," Asilomar Conf. Signals, Syst., Computers, Pacific Grove, CA, Nov. 1997.

[29] S. Thillainthan, D. Bull, and N. Canagarajah, "Robust embedded zerotree wavelet coding algorithm," Proc. SPIE, Visual Commun. Image Processing, San Jose, CA, Jan. 1998.

[30] J.W. Woods and S.D. O'Neil, "Subband coding of images," IEEE Trans. Acoust., Speech, Signal Processing, vol. 34, pp. 1278–1288, Oct. 1986.

[31] J.W. Woods, ed., Subband Image Coding, Kluwer Academic Publishers, Boston, MA, 1991.

[32] Z. Xiong, O.G. Guleryuz and M.T. Orchard, "A DCT-based embedded image coder," IEEE Signal Processing Letters, vol. 3(11), pp. 1289–290, Nov. 1996.

[33] Z. Xiong, K. Ramchandran and M.T. Orchard, "Wavelet packet image coding using space-frequency quantization," submitted to IEEE Trans. Image Processing, 1996.