Group Testing for Video Compression*

Gidon Shavit,† Michael F. Ringenburg,† Jeff West,‡
Richard E. Ladner,† Eve A. Riskin§
University of Washington

Abstract

Hong and Ladner [6] used context-based group testing to implement bit-plane coding for image compression. We extend this technique to video coding, by replacing the quantization and entropy-coding stages, of an H.263 standard video coder, with bit-plane coding. We experiment with ways to improve baseline coder, including different classification schemes and cross-frame adaptive coding. Our results indicate that our new coder, GTV (Group Testing for Video), significantly outperforms H.263 at medium to high bit-rates (300+ kbps) on most sequences, while allowing very precise rate scalability.

1 Introduction

Most standard video coders, including H.26x and MPEG-x, are very similar in structure: block-based motion compensation is used; the prediction error (residual image) is then transformed (typically with DCT); and the transform coefficients are quantized and entropy-coded. Everything that happens after the motion-compensation stage is, in principle, very similar to traditional still image coding.

The image compression world, however, has been steadily transitioning from quantization based methods to bit-plane coding ones (e.g. JPEG-2000). Introduced in a seminal paper by Shapiro in 1993 [16], bit-plane coding (BPC) has shown significant gains over scalar quantization methods [16, 15]. BPC coders are embedded, allowing precise bit-rate control and excellent rate scalability properties. One of the arguments *against* using bit-plane coders for video has been its relative slowness compared with classic scalar quantization methods. The constant improvement in computing speed, however, reduces the weight of this argument.

*This work was supported in part by NSF under grant number CCR-0104800, by the ARO under grant number DAAD1919-02-1-0309 and by an ARCS Fellowship sponsored by the Washington Research Foundation (WRF).

†Address: Department of Computer Science and Engineering, Box 352350, University of Washington, Seattle, WA 98195–2350, Tel: +1-206-543-1695; Fax: +1-206-543-2969. E-mail: *gidon,miker,ladner@cs.washington.edu.*

‡E-mail: *Jeff.West@guidant.com*

§Address: Department of Electrical Engineering Box 352500, University of Washington, Seattle, WA 98195–2500, Tel: +1-206-543-2150; Fax: +1-206-543-3842. E-mail: *riskin@ee.washington.edu.*

Proceedings of the Data Compression Conference (DCC'04)
1068-0314/04 $ 20.00 © 2004 IEEE

One BPC algorithm, for encoding wavelet-transformed images, was presented by Hong and Ladner in [5], and later adapted to handle block-transforms, including DCT [6]. This algorithm, GT-DCT, is similar to JPEG, but replaces the scalar quantization and entropy coding stages, with bit-plane coding using context-based group testing. GT-DCT consistently and significantly outperforms JPEG, and is competitive with other BPC methods, such as SPIHT [15]. Since video compression standards use a very JPEG-like algorithm for coding residual images, they might be improved by replacing this with an appropriate GT-DCT-like method.

In this work, we begin by essentially replacing the quantization and entropy-coding stages of H.263 [9] with group testing bit-plane coding, copied exactly from GT-DCT. Using this basic coder as a baseline, we experiment with several methods to enhance the algorithm: we study the statistical structure of residual frame DCT coefficients, to design contexts for the coder; and we propose a mechanism for coder adaptivity across frames, based on measurements of the stability of coefficient statistics in time.

As far as we know, our new coder GTV (Group Testing for Video) is the first context-based bit-plane coder for DCT-transformed video. Our tests show that, while H.263 is better than GTV at low bit-rates (100-200kbps), GTV catches up at about 300kbps, and significantly outperforms H.263 at higher rates. This, together with the excellent rate scalability, makes GTV appropriate for applications such as DVD encoding and broadband streaming video (with appropriate error resilience additions). A discussion of how the rate scalability of GTV can be used to produce near-constant quality video can be found in [14].

The rest of this paper is organized as follows: section 2 surveys related work on bit-plane coding and video compression; section 3 describes the baseline GTV coder in detail; section 4 describes some enhancements to the baseline algorithm; section 5 presents experimental results, and section 6 discusses future work.

2 Related Work

Standard video coders: Since the 1990's, the two standards that have dominated applications and served as a basis for most research, are ITU H.263 [9] and MPEG-2 [8]. Both are based on motion compensation, followed by a DCT block transform, almost-uniform quantization of transform coefficients, and entropy-coding of the quantized coefficients. H.263 tends to outperform MPEG-2 due to the latter's frequent use of synchronization intra-frames (which are not useful in error-free applications).

A new standard, H.264/AVC, developed in cooperation by the MPEG and ITU, was recently finalized [10]. It is still similar in principle to H.263 and MPEG-2, but adds a host of new features and improvements to almost every aspect of the coding process, e.g. higher-resolution motion prediction from multiple frames, small-block integer transforms, and syntax-based arithmetic coding. However, the quantization stage of H.264 is still similar to that of H.263.

Bit-plane coding for image compression: Bit-plane coding for image compression was introduced by Shapiro [16]. The original algorithm, EZW, applied a

discrete wavelet transform (DWT) to the image, and then used bit-plane coding to compress the coefficients. The inherent hierarchical structure of wavelet coefficients facilitates efficient coding of coefficient trees, using a method called zerotree coding. EZW consistently outperformed JPEG.

Said and Pearlman developed SPIHT [15], which is similar to EZW in concept, but takes advantage of further statistical structure in the wavelet transform. Since then, a number of algorithms in the same tack have been published. Another approach, using context-based arithmetic coding, was introduced by Taubman in EBCOT [18], and later served as the basis for the JPEG-2000 image compression standard.

Some work has been applied to block-transform compression as well, e.g. [20] for DCT, and [12, 13] for lapped transforms. Both are based on zerotrees, although the hierarchical structure is defined somewhat arbitrarily.

BPC for video: Several BPC-based video coders were published since the late 1990's. Most of those are based on zerotree methods, applied to either motion-compensated residuals (e.g. [17]) or to 3D (time/space) wavelet transform coefficients (e.g. [11]). Context-based coding is applied to 2D and 3D wavelet coefficients by several authors, e.g. [21, 7, 19], following the ideas introduced in EBCOT.

3 The Basic GTV Algorithm

GTV is a hybrid video coder, and borrows much of its structure from H.263 [9]. Motion compensation is done on 16×16-pixel macroblocks, and motion vectors have a 15-pixel range and half-pixel resolution. For implementation reasons, GTV does not use the H.263 variable-length code method for compressing motion vectors. Instead, the difference-coded vectors are packed and compressed with BZIP2, based on the Burrows-Wheeler transform [1]. The residual undergoes 8×8 DCT. Only the first frame is intra-coded, and is transformed and coded exactly like residuals.

GTV diverges from H.263 mostly in that it replaces the quantization and entropy coding stages of H.263 with group testing bit-plane coding. The baseline implementation of GTV uses the coding part of GT-DCT [6], which will be explained next. GTV assigns the same bit-rate to all inter-frames, and allows the user to specify how many bits to assign to the first, intra-coded frame. More elaborate bit-allocation algorithms are developed and tested in [14]. However, those affect the variance of the PSNR rather than its average, so the results in section 5 remain mostly unchanged.

3.1 Bit-plane coding

Scalar quantization methods code coefficients one at a time. In contrast, BPC codes coefficients "in parallel": first, the most significant bits of *all* coefficients are coded, then all second-most significant bits, etc. Lossless entropy coding is used to compress the bit-planes, and coding stops when the output stream reaches a specified length. BPC is an *embedded* coding method, since the most important bits are always coded first.

In [16] Shapiro applied the following paradigm for BPC, which has since become predominant. First, assume all coefficients are normalized to the range (-1,1). A coefficient is said to be *significant* in bit-plane b, if $|C| \geq 2^{-b}$. Each bit-plane is coded in two passes: the *significance pass* codes which coefficients have become significant at the current bit-plane (as well as their signs); the *refinement pass* codes the current bit of previously-significant coefficients. Compression is achieved mostly at the significance pass, where statistics on the magnitude of coefficients can be used to achieve good compression ratios.

3.2 Group testing and GT-DCT

Group testing is a binary entropy coder, based on the concept of group-tests: a subset (*group*) of bits is tested to see if any of them is 1. In the encoder, the bits are known; the test is performed, and the result is sent to the decoder. The decoder is synchronized to the same sequence of tests, and can reconstruct the original sequence. Hwang [3] has described a group testing coding algorithm for sequences of i.i.d. bits. This code is equivalent to elementary Golomb code [4], and almost achieves the entropy bound for i.i.d. sources. The key parameter for the algorithm is the group size (number of bits) tested at each iteration.

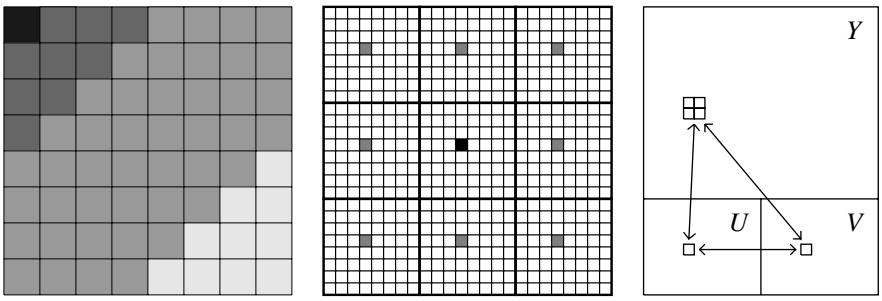
Hong and Ladner [5] applied context-based group testing to the significance pass of the bit-plane coding paradigm. Insignificant coefficients are divided into *classes*, or contexts, and each class is coded separately using group testing, as if it were an i.i.d. source. Since the actual distribution of coefficients in the class is not known in advance, adaptive coding is used, whereby the group size is continuously adjusted to match the empirical distribution observed so far.

Significance-pass coding proceeds iteratively: in every iteration, a class is picked according to a predefined priority order, and a group of coefficients of appropriate size (as determined by the observed distribution for that class) is coded using the Hwang algorithm. Any neighbors of newly-found significant coefficients are updated, and possibly re-classified. An end-case occurs when a class has fewer coefficients than its proper group size. We call this an "inadequate" class. GT-DCT attempts to pick only adequate classes, and when none is available, it combines coefficients from inadequate classes to form a group. The result is that sometimes, especially when classes are small, coefficients may be coded in an inefficient manner.

3.3 Basic GTV classification scheme

GT-DCT and GTV classify coefficients by the following criteria, which will be later referred to as the *baseline classification* scheme:

Sub-band level: Every DCT coefficient belongs to one of 64 sub-bands, determined by its frequency content, i.e. its position in the transformed block. Low-frequency coefficients tend to have higher magnitudes than high-frequency ones, making this a good classification criterion. GT-DCT defines 4 levels of sub-bands (see figure 1a).
Neighbor significance: Neighboring blocks in transformed still images tend to



(a) (b) (c)

Figure 1: Classification criteria: in (a), coefficients in an 8x8 DCT block are shown in shades corresponding to the sub-band level they belong to. In (b), the coefficient in black (middle) has 8 spatial neighbors, shown here in grey. In (c), the 4 coefficients in the Y band are spatially co-located with the U and V coefficients.

exhibit high levels of correlation. GT-DCT uses this by counting how many of a coefficient's *spatial neighbors* (see figure 1b) are already known to be significant at the current or previous bit-plane. The higher this number, the more likely the coefficient itself is to become significant. It turns out that little information is gained beyond 3 significant neighbors, so GT-DCT classifies coefficients as having 0, 1, 2, or 3+ significant neighbors.

4 Enhancements to the Basic Algorithm

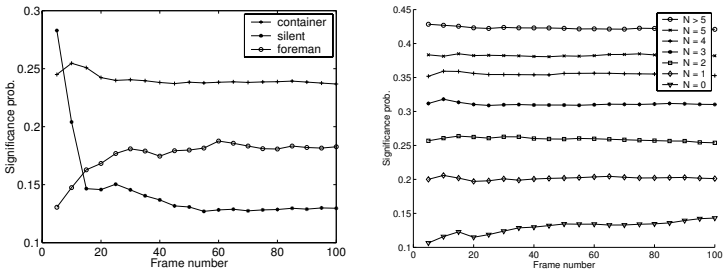
GT-DCT was designed for coding grey-scale still images. Since GTV is a video coder, it is natural to try and adapt the algorithm to the different characteristics of video data. We experiment with two enhancements to the original algorithm: new classification schemes, adapted to the statistics of video data, and an adaptation mechanism that allows the coder to keep adapting to empirical distributions across frames.

4.1 Cross-frame adaptive coding

As described above, GT-DCT applies adaptive group testing to encode classes. In the baseline GTV, the class distribution was reset every bit-plane, starting again from uniform. This causes a problem when trying to increase the number of classes, since small classes are coded most of the time at the wrong distribution.

We analyzed the distributions of classified significance bits, and found that they exhibit remarkable stability across frames (see figure 2). This indicates that it should be possible to use historical distribution data to initialize the coding of the current frame. On the other hand, since the distributions do change with time, the mechanism must allow significant adaptivity at late stages of coding. Simply maintaining the cumulative empirical distribution of all classes is therefore not advisable.

The approach we chose to implement follows these guidelines. For each class c and bit-plane p, we maintain a *weighted observed distribution*, as a pair (N^0_{cp}, N^1_{cp}). Let



(a) (b)

Figure 2: Significance probability averaged on 5-frame chunks. In (a), the overall probability of a significance bit being '1' is shown for several video sequences. In (b), the probability is for coefficients classified by number of significant spatial neighbors ("container").

$n^b_{cp}[t]$ be the number of b's coded in class c and bit-plane p of frame t. Then, after coding frame t_0, we require:

$$N^b_{cp} = \sum_{t=1}^{t_0} \alpha^{t_0 - t} n^b_{cp}[t] \tag{1}$$

where $0 \leq \alpha \leq 1$ is a user-specified constant. This way, the data from k frames ago is weighted down by α^k. An α value of 0 simulates the baseline algorithm, where past distributions are discarded; $\alpha = 1$ uses the entire (unweighted) observed distribution. Maintaining this weighted distribution is easy: the number of observed 0 and 1 bits is constantly added to the current N^b_{cp} values, and after each frame we simply multiply both values by α, which conforms to the requirement (1). The coder constantly adjusts the group size to fit the distribution (N^0_{cp}, N^1_{cp}).

Our results indicate that this mechanism is helpful, especially when the number of classes is increased. Interestingly, the performance is not very sensitive to the specific value of α; all values in The range (0.2,1.0) exhibit similar performance on our data. As expected, the gain from using cross-frame adaptivity grows with the number of classes: when the baseline classification (16 classes) is used, the gain is about 0.1dB across the board; when a scheme with 128 classes is used, the gain is about 0.2-0.3dB.

4.2 Alternative classification schemes

The bulk of the data coded by GTV is in the form of residual (inter) frames, which are very different in nature from the data for which GT-DCT was designed. It is, therefore, important to reexamine the statistical characteristics of the data, and design the classification system of GTV to match it. The goal remains to make classes as homogeneous as possible, ideally consisting of i.i.d. coefficients. For that purpose, we collected statistics on transformed motion-compensated residuals, and the statistical correlations between different coefficients. We concentrated on three types of relationships:

Spatial neighbors: While natural images exhibit much spatial correlation across DCT blocks, it seems that residual images should be far less correlated. However,

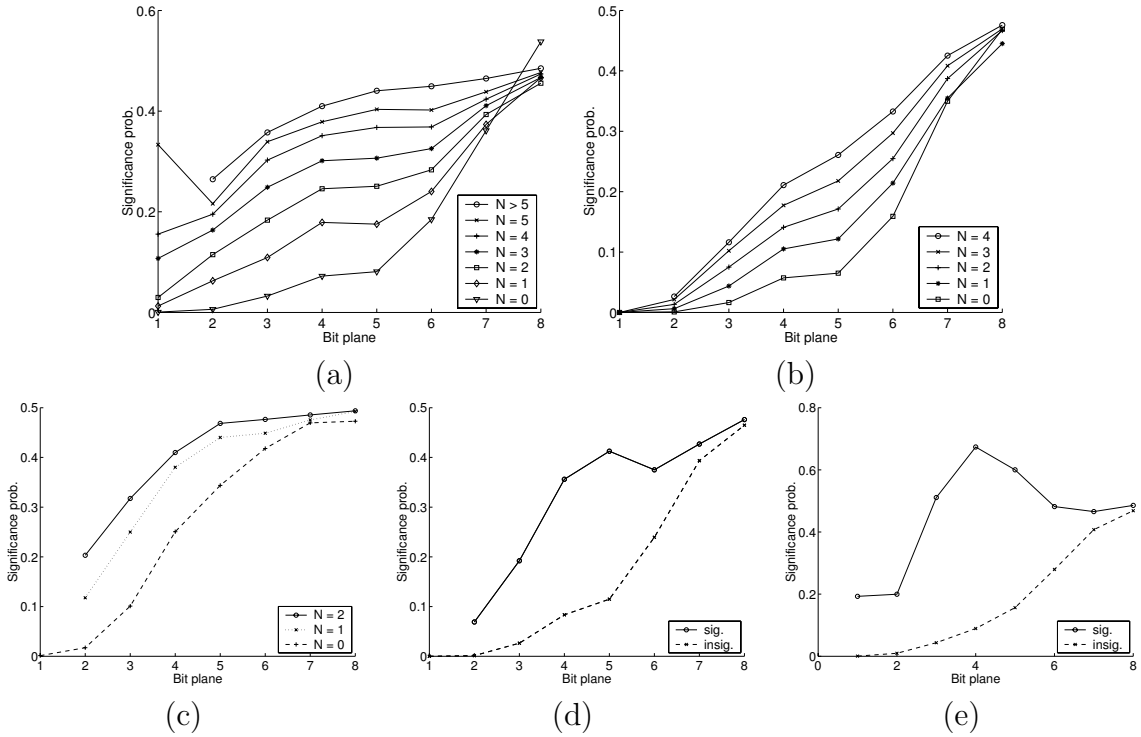


Figure 3: Probability of '1' significance bits by bit-plane, classified by different criteria: (a) number of significant spatial neighbors; (b) for chroma coefs, number of sig. co-located luma coefs; (c) for luma coefs, number of sig. co-located chroma coefs; (d) for chroma coefs, significance of other chroma band coef; (e) significance of same coef in previous frame. All results are for "container".

our tests show that there is still considerable correlation between the significance of spatial neighbors. Figure 3a shows the probability of significance bits being 1, classified by the number of previously-significant spatial neighbors. It is interesting to note, that beyond 3 or 4 significant neighbors, the gain in information becomes less useful: in the first bit-planes, the occurrence of many significant neighbors is unlikely; in later bit-planes, the gain in significance probability is small.

Co-located luma-chroma : GTV input is in QCIF format, where color frames consist of three bands: a luma (or Y) band, and two chroma bands (U and V). Each chroma band is sub-sampled by a ratio of 4:1 (see figure 1c). We expected considerable correlation between spatially co-located coefficients, and were not disappointed. Figures 3b, 3c, and 3d show the probability of significance bits being 1, classified by the significance of corresponding coefficients in the other bands.

Temporal neighbors: Since we are dealing with frame sequences rather than single images, it is natural to examine the relationship between co-located coefficients in consecutive frames. Although the motion prediction is supposed to extract temporal correlation from the video, some of it remains, due to the imperfections of the motion compensation algorithm. Figure 3e indicates that this correlation is far from insignificant.

Based on the above results, we experimented with several classification schemes, of growing complexity. We used the GT-DCT scheme as our baseline, and compared it to the others in terms of performance. We added luma-chroma neighbors and/or temporal neighbors to the context. We also tried increasing the number of sub-band levels from the original 4 to 5, 6, or 7.

Perhaps surprisingly, the only significant improvement was achieved by using the luma-chroma classification. This yielded gains of 0.2-0.3dB, whereas all other changes gained at most 0.1dB, and usually less. In fact, increasing the number of classes seems to cause slight degradation sometimes, which may be due in part to the increasing number of bits coded in inadequate classes (up to 13%, compared to about 6% using the original classification).

5 Results

Figure 4 compares the fidelity vs. bit-rate curves of GTV and H.263 for several video sequences, where fidelity is measured as the average PSNR of the first 100 frames. All GTV results were obtained using classification based on spatial and luma-chroma neighbors, as well as sub-band level, with a cross-frame adaptivity parameter $\alpha = 0.5$. H.263 results were generated with the public-domain Telenor H.263 TMN5 implementation [2].

It is evident that GTV does not do as well as H.263 at low bit-rates, i.e. 100 and 200kbps. However, GTV becomes competitive around 300kbps, and at higher bit-rates it outperforms H.263 on most inputs, in some cases by several dB. However, since it's embedded, GTV has much better rate scalability properties than H.263.

One major reason for the relative failure of GTV in low bit-rates is its inefficient motion-vector encoding. For example, on "foreman," GTV spends on average about 1250 bits on coding motion vectors; the number for the H.263 coder is less than 600. The difference is about 20% of the total bits per frame, which is very significant. If GTV were as efficient as H.263 in this task, we would see a gain of about 1dB in the 100kbps results.

6 Conclusions and Future Work

Bit-plane coding, using context-based group testing for the significance pass, appears to be competitive with quantization-based H.263 at medium bit-rates (300kbps), and to improve upon it considerably at higher (500+ kbps) rates. GTV also allows precise control over the bit-rate used for each frame. Cross-frame adaptivity and additional classification criteria appear useful, with gains of about 0.2-0.4dB over the baseline.

Several directions for further research seem to be suggested from this work. First of all, with the emergence of H.264 as the new leading video coding standard, it can replace H.263 as the basis for our coder. This will allow us to take advantage of the numerous new features and improvements in the prediction and transform stages. It should be relatively straight-forward to replace the quantization and entropy coding

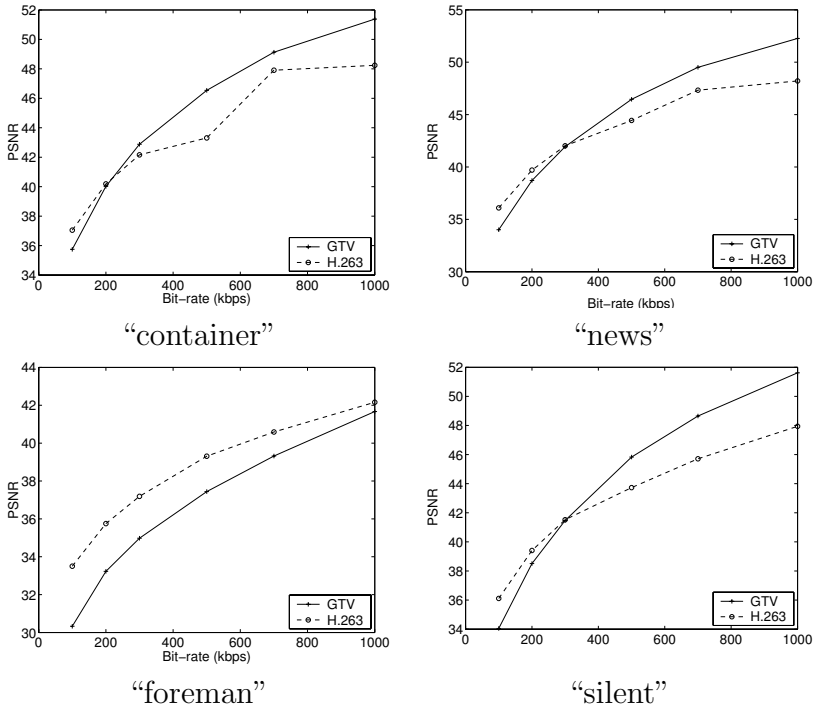


Figure 4: Comparison of reconstruction quality vs. bit-rate curves, for GTV and H.263. PSNR values are averaged over the first 100 frames of each video sequence.

of H.264 with bit-plane coding. Further studies into the statistical nature of H.264 coefficients will be necessary, to construct an appropriate classification scheme.

The use of group testing for the coding of significance bits is not essential to our approach. Any adaptive, context-based, entropy coder would work just as well. The obvious candidate seems to be arithmetic coding, which has several important advantages over group testing. The issue of "inadequate" classes will be resolved, performance at significance probability around or above 50% should improve, and implementation will be simpler and more efficient. In addition, arithmetic coding would allow us to code planes of non-binary digits, which might allow better use of statistical structure. Finally, it is a challenge to improve the speed of compression and decompression of GTV, and indeed of any bit-plane video coder.

References

[1] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Research Report 124, Digital Systems Research Center, 1994.

[2] Telenor H.263 codec. *ITU-T/SG-15, video codec test model, TMN5.* Telenor Research, 1995.

[3] D. Du and F. Hwang. *Combinatorial Group Testing.* World Scientific, Singapore, 1972.

[4] S. Golomb. Run-length encodings. *IEEE Transactions on Information Theory*, 12:399–401, Jul 1966.

[5] E. S. Hong and R. E. Ladner. Group testing for image compression. In *Proc. DCC '00*, pages 3–12, March 2000.

[6] E. S. Hong, R. E. Ladner, and E. A. Riskin. Group testing for block transforms. In *Proceedings of the thirty-fifth conference on signals, systems, and computers*, pages 769–772, November 2001.

[7] J. Hua, Z. Xiong, and X. Wu. High-performance 3-D embedded wavelet video (EWV) coding. In *IEEE 4th Workshop on Multimedia Sig. Proc.*, pages 569–574, Oct 2001.

[8] ISO/IEC. The MPEG-2 international standard (ref. number ISO/IEC 13818-2), 1996.

[9] ITU-T. Video coding for low bit rate communications (ITU-T Rec. H.263), 1995.

[10] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050. Draft ITU recommendation and final draft international standard of joint video specification (ITU Rec. H.264/ISO/IEC 14 496-10 AVC), 2003.

[11] B.-J. Kim and W. A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In *Proc. DCC '97*, pages 251–260, March 1997.

[12] H. S. Malvar. Lapped biorthogonal transforms for transform coding with reduced blocking and ringing artifacts. In *Proc. ICASSP-97*, volume 3, pages 2421–2424, Apr 1997.

[13] Henrique S. Malvar. Fast progressive image coding without wavelets. In *Proc. DCC '00*, pages 243–252, March 2000.

[14] M. F. Ringenburg, R. E. Ladner, and E. A. Riskin. Global MINMAX interframe bit allocation for embedded video coding. Submitted to DCC '04, 2004.

[15] Amir Said and William A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE. Trans. CSVT*, 6(3):243–250, June 1996.

[16] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[17] K. Shen and E. J. Delp. Wavelet based rate scalable video compression. *IEEE. Trans. CSVT*, 9(1):109–122, February 1999.

[18] David Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, July 2000.

[19] J. W. Woods and G. Lilienfield. A resolution and frame-rate scalable subband/wavelet video coder. *IEEE Trans. CSVT*, 11(9), Sep 2001.

[20] Z. Xiong, O. G. Guleryuz, and M. T. Orchard. A DCT-based embedded image coder. *IEEE Signal Processing Letters*, 3(11):1289–1290, November 1996.

[21] J. Xu, Z. Xiong, S. Li, and Y. Q. Zhang. 3-D embedded subband coding with optimal truncation (3-D ESCOT). *Applied and Computational Harmonic Analysis: Special Issue on Wavelet Applications in Engineering*, 10:290–315, May 2001.