Wavelet TCQ: submission to JPEG-2000

Philip J. Sementilli[b], Ali Bilgin[a], James H. Kasner[c], Michael W. Marcellin[a]

[a]Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721
[b]Science Applications International Corp. (SAIC), 101 N. Wilmot Rd., Tucson, AZ 85711
[c]Optivision, Inc., 3450 Hillview Ave., Palo Alto, CA 94304

ABSTRACT

The Joint Photographic Experts Group (JPEG) within the ISO international standards organization is defining a new standard for still image compression – JPEG-2000. This paper describes the Wavelet Trellis Coded Quantization (WTCQ) algorithm submitted by SAIC and The University of Arizona to the JPEG-2000 standardization activity. WTCQ is the basis of the current Verification Model (VM) being used by JPEG participants to conduct algorithm experiments. The outcomes from these experiments will lead to the ultimate specification of the JPEG-2000 algorithm. Prior to describing WTCQ and its subsequent evolution into the initial JPEG-2000 VM, a brief overview of the objectives of JPEG-2000 and the process by which it is being developed is presented.

Keywords: JPEG-2000, image compression, wavelet, trellis coded quantization.

1. JPEG-2000 OVERVIEW

The Joint Photographic Experts Group (JPEG) within the ISO international standards organization is defining a new standard for still image compression – JPEG-2000. JPEG-2000 is intended to complement the existing suite of JPEG and JBIG (Joint Bi-level Image Experts Group) still image compression standards. JPEG-2000 is being designed to support a variety of image types, including bi-level, grayscale and color, across a broad spectrum of applications, such as amateur photography, remote sensing, medical imaging and color facsimile. Compression features that support emerging image transmission applications, such as world-wide web, wireless communications, and client-server computing are of particular interest to the JPEG-2000 developers.

1.1 JPEG-2000 requirements and features. The following requirements for the JPEG-2000 algorithm are summarized from Reference 1:

> This standard is intended to advance standardized image coding systems to serve applications into the next millennium. It will provide a set of features vital to many high-end and emerging image applications by taking advantage of new modern technologies. Specifically, this new standard will address areas where current standards fail to produce the best quality or performance including the following:
>
> - Low bit-rate compression performance
> - Lossless and lossy compression
> - Large images
> - Single decompression architecture
> - Transmission in noisy environments
> - Computer generated imagery
> - Compound documents
>
> It will also provide capabilities to markets that currently do not use compression.

Also from Reference 1, it is desired that JPEG-2000 provide as many of the following features as possible:

- Superior low bit-rate performance: e.g., below 0.25 bpp for highly detailed grayscale imagery. This should be achieved without sacrificing performance on the rest of the rate-distortion spectrum. This is the highest priority feature.

Further author information -
M.W.M (correspondence): mwm@ece.arizona.edu, http://www-spacl.ece.arizona.edu
P.J.S: sementip@aries.tucson.saic.com, A.B: bilgin@ece.arizona.edu, J.H.K. kasner@optivision.com

Part of the SPIE Conference on Applications of Digital Image Processing XXI • San Diego, California • July 1998
SPIE Vol. 3460 • 0277-786X/98/$10.00

- Continuous-tone and bi-level compression
- Lossless and lossy compression
- Progressive transmission by pixel accuracy and resolution
- Fixed-rate, fixed-size, limited workspace memory: Fixed-rate (fixed local rate) means that the number of bits for a given number of consecutive pixels is less than or equal to a certain value. Fixed-size (fixed global rate) means that the total size of the codestream for a complete image equals a certain value.
- Random codestream access and processing
- Robustness to bit-errors
- Open architecture
- Sequential build-up capability: the capability to compress and decompress images with a single sequential pass. Also, the capability to process an image using component interleave order or non-interleaved order.
- Backwards compatibility with JPEG
- Content-based description
- Protective image security: including one or more of watermarking, labeling, stamping, encryption.
- Interface with MPEG-4
- Side channel spatial information (transparency)

In addition, features to support desired markets and applications include: region of interest coding and fast transcoding.

1.2 JPEG-2000 development process. The JPEG-2000 activity within ISO, SC 29 / WG 1, issued a Call for Contributions in March, 1997. (Dr. Daniel T. Lee of Hewlett-Packard is the Convener of SC 29 / WG 1.) The Call for Contributions stipulated that compression technologies be submitted to an evaluation during the November, 1997 ISO meeting in Sydney, Australia. Further, SC 29 / WG 1 released a CD-ROM containing 40 test images to be processed and submitted for the Sydney evaluation, all submittals being due September 30, 1997. For the Sydney evaluations, it was stipulated that compressed and decompressed imagery be submitted for six different bit rates (ranging from 0.0625 to 2.0 bits per pixel (bpp)) and for lossless encoding. Eastman Kodak conducted a subjective evaluation in Sydney using evaluators from among the ISO meeting attendees. The imagery from 24 algorithms was evaluated by ranking the perceived image quality of hard-copy prints. The test imagery included results encoded at 0.0625, 0.125, and 0.25 bpp. Eighteen of the 40 test images were included in the evaluation, with samples from several different image types (not including color). In addition to this subjective evaluation, quantitative metrics were computed for all 24 algorithms over the entire set of test imagery.

Although the top third of the submitted algorithms performed well in the Sydney evaluation, WTCQ ranked first overall in both the subjective and objective evaluations. In the subjective evaluation, WTCQ ranked first - averaged over the entire set of evaluated imagery - at 0.25 and 0.125 bpp. WTCQ ranked second at 0.0625 bpp. In terms of RMS error averaged over all images, WTCQ ranked first at each of the six bit rates between 0.0625 and 2 bpp, inclusive. WTCQ ranked seventh in lossless performance. Based on the superior image quality of WTCQ, it was selected as the baseline JPEG-2000 algorithm at the conclusion of the Sydney meeting. It was further decided that a series of "core experiments" would be conducted to evaluate WTCQ and other techniques in terms of the JPEG-2000 desired features and in terms of algorithm complexity.

The first round of core experiments was conducted using a C code implementation of WTCQ, with results presented at the March, 1998 ISO meeting in Geneva. Based on the core experiments, several key decisions were made. It was decided that a JPEG-2000 "Verification Model (VM)" would be created that would lead to a reference implementation of JPEG-2000. The VM is the software in which future rounds of core experiments will be conducted. It will be updated after each ISO JPEG-2000 meeting based on the results of core experiments. (The next two ISO JPEG-2000 meetings are July, 1998 in Copenhagen and Nov., 1998 in Los Angeles.) Michael Marcellin was appointed Chair of the VM Ad Hoc Group with Val Vaughn of The Aerospace Corp, Co-Chair and Charis Christopoulos of Ericsson, Editor. SAIC was appointed to develop and maintain the VM software. The VM Ad Hoc Group selected results from round 1 core experiments to be used to modify WTCQ into the first release of the VM (VM 0). These modifications are described in Section 3.

2. WAVELET TCQ ALGORITHM – SYDNEY SUBMISSION

SAIC and The University of Arizona (SAIC/UA) submitted the wavelet TCQ (WTCQ) still image compression algorithm in response to the JPEG-2000 call for contributions. Many elements in the JPEG-2000 submission of WTCQ were first published in 1994.[2] At that time, the performance of WTCQ was among the best in the literature. During 1997, several enhancements and additions were made to WTCQ to satisfy all requirements of JPEG-2000, e.g., support for bi-level imagery, lossless compression, etc. In this section we describe the main features of the Sydney submission of WTCQ. For the remainder of this section, "WTCQ" refers to the version of the algorithm submitted to the Sydney evaluation.

2.1 WTCQ encoder: The WTCQ encoder for grayscale imagery is illustrated in Figure 1. The WTCQ encoder accepts raster ordered input imagery of arbitrary dimensions. WTCQ accepts a variety of grayscale pixel data types, including 8, 10, 11, 12, and 16 bit per pixel data as well as floating point pixels.

2.1.1 Discrete Wavelet Transform (DWT): The DWT function includes two floating point wavelet filters, selectable by the user - (9,7) biorthogonoal filters[3] and (10,18) filters[4]. The DWT implementation accommodates even and odd image dimensions by placing the "extra" filter output in the low-pass subband. Symmetric extension is applied at image boundaries, with the reflection point based on whether the wavelet filter length is even or odd. The number of levels in the wavelet decomposition is such that the lowest resolution subband is nominally 16x16. The exact size varies based on the size of the input image. The wavelet decomposition "tree" is user selectable from among three types - a dyadic decomposition, a dyadic decomposition with an additional "split" of the three highest frequency subbands, and a three-level uniform decomposition followed by dyadic decomposition of the low resolution subband. Although somewhat non-descriptive, the command line arguments for selecting among these three decompositions are; mallat, spacl (an acronym for the UA Signal Processing and Coding Lab) and packet, respectively.

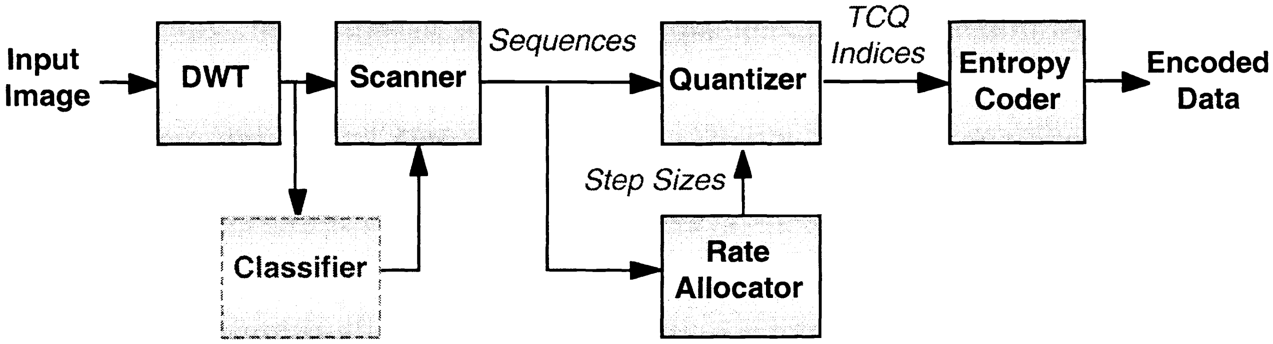


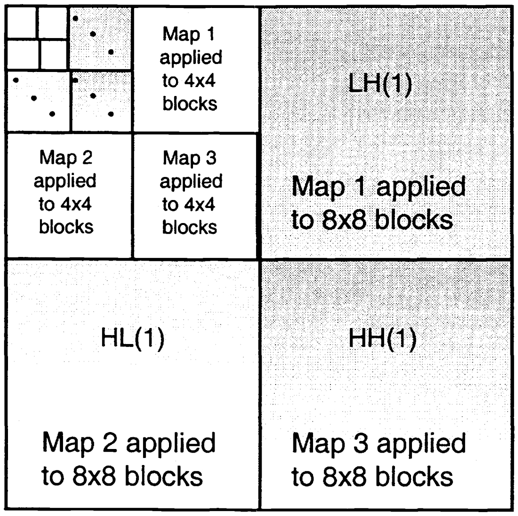
Figure 1 -- Wavelet TCQ Grayscale Imagery Encoder Block Diagram

2.1.2 Classifier and Scanner: The classifier and scan algorithms are used to group wavelet coefficients into "sequences" of similar statistics prior to quantization.[5] The classifier is optional - its use must be manually selected by the user. The classifier algorithm partitions each wavelet subband at the four lowest levels of the tree into four sequences. The class membership for the wavelet coefficients is specified using "class maps". These class maps are included in the codestream as overhead information. To avoid excessive overhead, wavelet coefficients are not classified individually. They are classified on a block by block basis. The block size depends on the level within the wavelet tree. Only three maps (one for each "orientation") are transmitted. The maps are then re-used by orientation (see Figure 2) at each level within the tree. This approach attempts to exploit the energy dependencies that are present between wavelet bands (i.e., low and/or high energy areas tend to occur in the same spatial location for parent/child wavelet bands).

Classification is performed on the three high-frequency bands following the first 2D filtering step of the DWT. Each band is partitioned into a set of non-overlapping 8x8 blocks. The variance (local energy) is computed for each block, and the variances from all blocks are clustered using the K-means algorithm. The result of clustering is that each block is a member of one of four classes.

The Scanner function is the mechanism that forms sequences from the wavelet subband coefficients. If the classifier is not used, then a sequence is created by a standard raster scan of the subband. If a class map is available for a subband, the scanner creates four sequences from that subband, one sequence per class.

The benefits obtained by this classification procedure depend on the encoding bit rate and on the model used in the final entropy coding. The technique is of most benefit when coding for higher quality (higher bit-rates) and when using a memoryless (context free) model in the entropy coding. In this case, coding gains in excess of 1.0 dB are achievable.



- **Calculate variance of 8x8 blocks in LH(1), HL(1), HH(1)**
- **K-means cluster the variances**
- **Label each 8x8 block as belonging to one of K classes**
- **Propagate labels up through tree**
- **Include entropy encoded class maps in codestream header**

Figure 2 -- Illustration of Class Map Propagation Through Dyadic Wavelet Tree

2.1.3 Rate Allocator: The rate allocator is used to compute step sizes for the quantizer, one step size for each sequence. This process is analogous to Q-table selection in the current JPEG standard[6]. The step sizes are computed with the objective of minimizing mean-squared encoding error while achieving a compressed bit rate within some user specified range. The rate allocation method uses statistical models to predict the rate-distortion performance for quantizing each sequence. The rate-distortion models are analytic expressions derived empirically from memoryless sources with generalized Gaussian probability density functions (GGD's). Thus, no training is required. The rate-distortion models are used in a Lagrangian formulation to minimize (predicted) mean-squared-error, subject to a constraint on the (predicted) encoding rate. Specifically, a set of step sizes are chosen so that the objective function

$$J(R_i) = \sum_i \omega_i d_i(R_i) + \lambda \sum_i \gamma_i R_i \qquad (1)$$

is minimized. In (1), R_i is the bit rate associated with sequence i, $d_i(R_i)$ is the distortion model associated with sequence i as a function of rate R_i, γ_i is a "size weight" for sequence i, and ω_i is an "energy weight". The size weight indicates the percentage of the total number of wavelet coefficients contained in sequence i. The energy weights account for the quantization noise amplification that occurs during wavelet synthesis, in the manner described in Reference 7. The energy weights are strictly a function of the DWT filter weights and the wavelet decomposition tree. They are not image dependent. One of five rate-distortion models is used for each sequence, by selecting one of five possible estimates of the value for the parameter of a GGD. The GGD parameter is estimated for each sequence based on quantization of the sample kurtosis (ratio of the fourth central moment to the square of the variance) computed for each sequence. To optimize (1), a rapidly converging, low complexity iteration is done to compute the Lagrange multiplier λ. Once the Lagrange multiplier is selected, step sizes for quantizing unit variance GGD sequences are extracted from the model. These step sizes are scaled by the sample standard deviation to compute the TCQ step size for each sequence.

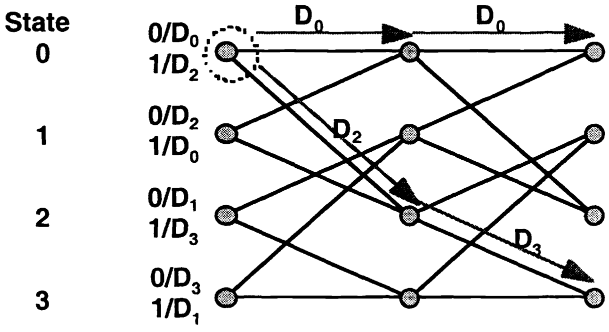
Since the rate-distortion models *predict* the actual encoding rate as a function of quantizer step size, the actual bit rate is not always within the user selected range. If the user inputs mandate that the actual bit rate must fall within a given range, the rate allocation / quantization / entropy coding functions are repeated until such a rate is achieved. This "rate iteration" adjusts the step sizes based on a simple feedback loop, and usually converges within two or three iterations.

2.1.4 Trellis Coded Quantizer (TCQ): TCQ can be thought of as a computationally efficient form of vector quantization (VQ). TCQ can process very long vectors (limited only by memory considerations) with a computational complexity (operations per sample) that is independent of vector length. This is in stark contrast to the exponential growth in complexity for full-search VQ.

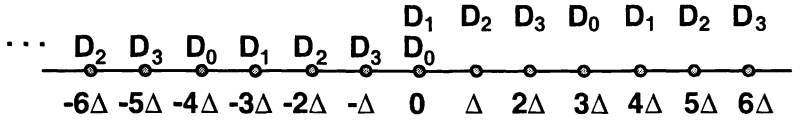
TCQ provides a theoretical advantage of up to 1.5 dB in mean-squared-error (MSE), or equivalently, 0.25 bits per sample. This amount assumes variable rate coding (e.g., Huffman or arithmetic) and high quality (high encoding rate). The perceptual difference in image quality when using TCQ is often higher than the MSE improvement might imply. TCQ is based on the same technology as trellis coded modulation, which is employed in 28.8 kbps voice band phone modems, and is highly amenable to hardware implementation.

In addition to the VQ interpretation discussed above, TCQ can also be thought of as a time-varying scalar quantizer. Each sample is quantized using one of four separate scalar quantizers. The allowable way of choosing among the scalar quantizers for each sample is specified in the form of a trellis. WTCQ uses an eight-state trellis, but for simplicity, Figure 3a depicts a four-state trellis. Legal sequences of "quantizers" are specified by "walks" through the trellis from left to right. For example, some of the legal quantizer sequences of length two (the ones starting from the upper left corner of the trellis) are: D_0, D_0; D_0, D_2; D_2, D_1; D_2, D_3.

Figure 3b depicts the codewords (or output points) of the four scalar quantizers, named D_0, D_1, D_2, and D_3 (i.e., each point labeled by D_i belongs to quantizer D_i.) The quantizers are chosen in the interlaced fashion shown and satisfy a set of heuristic design rules. A complete description of TCQ as implemented in the WTCQ coder can be found in Reference 8.



a) Four-State Trellis Diagram, illustrating quantizer sequences (D_0, D_0) and (D_2, D_3)



b) n-State Trellis Specifies Four Scalar Quantizers

Figure 3 -- Basic Mechanisms of Trellis Coded Quantization

2.1.5 Entropy Coder: Variable length coding is used for lossless compression of the quantizer output indices. WTCQ has demonstrated superior performance using m-ary arithmetic coding and Huffman coding. To meet the scalability (progressive transmission) requirements of JPEG-2000, the WTCQ implementation submitted to the Sydney evaluation employs binary arithmetic "bit-plane" coding with causal contexts. As illustrated in Figure 4, bit-plane coding treats the *magnitude* of the TCQ indices as a three-dimensional array of bits. The number of bit-planes, N, is determined based on the maximum magnitude over all TCQ indices. According to the *embedding principle*,[9,10,11,12] the encoded bit stream should be ordered in a way that maximally reduces coding MSE per bit transmitted. This is accomplished at a coarse granularity by sending bit-planes in decreasing order from most- to least-significant.

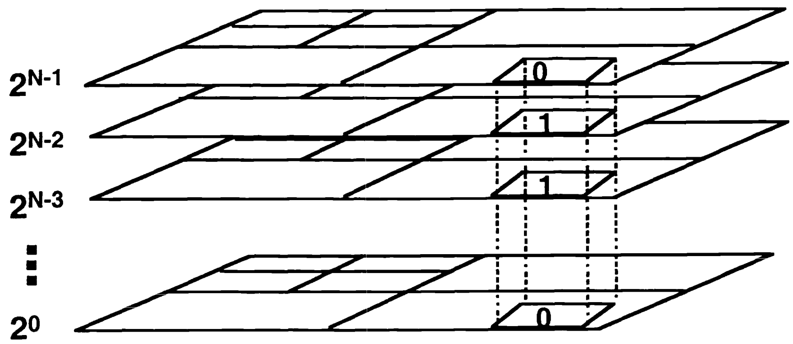


Figure 4 -- Illustration of Bit-plane Coding of TCQ Index Magnitude

Within a bit-plane, bits are input to an arithmetic coder in raster order, one sequence at a time. Sequences within a bit-plane are in order of non-decreasing resolution (i.e., from the upper left to lower right of the wavelet tree). The sign bit for a TCQ index is encoded immediately following the first non-zero magnitude bit for that index.[10,13] This approach is particularly well-suited to progressive transmission, since sign bits are not entered into the codestream until the point at which they will be needed at the decoder.

Since the inputs to the arithmetic coder are single bits, a simple binary arithmetic coder is sufficient. WTCQ uses an arithmetic coder developed by Don Speck at UC Santa Cruz for use in JPEG related research. To exploit spatial correlations within bit-planes, spatial context models are used. Let x be the bit to be encoded, then $-\log_2(P(x))$ bits are needed to encode x. A modeling context $C = \{x_1, x_2, ..., x_k\}$ can be created, where x_i, $i=1,...,k$ are other bits that are correlated with x. The context can be used to choose a probability table $P(x|C)$, such that $-\log_2(P(x|C))$ bits are needed to encode x. In general, the context can be chosen within a subband and across subbands. The WTCQ bit-plane coder limits the use of inter-subband contexts to enable protection against bit errors (e.g., during transmission of compressed images over noisy channels), to maximize flexibility in scalable decoding, and to facilitate parallel implementation. Note that $P(x|C)$ is estimated on the fly by accumulating histograms for each context. The Sydney submittal of WTCQ used 6 bits of context, requiring that 64 histograms be maintained. Since the arithmetic coding uses a binary alphabet, each histogram has only two bins. Therefore, the memory impact associated with context modeling is minimal. Context bits are derived from two sources; "sign flag bits", which indicate whether or not the sign bit for a neighboring TCQ index has been sent, and magnitude bits within the bit-plane being decoded. The context modeling in WTCQ has been improved since the Sydney submittal, so the specific contexts used in the current WTCQ are described in Section 3.

2.2 WTCQ decoder: The WTCQ decoder for grayscale imagery is illustrated in Figure 5. The WTCQ decoder extracts all necessary header information from the encoded file, including inverse quantization step sizes, class maps for control of the inverse scanner, and indicators of the DWT filter and decomposition type. The decoded image is stored as a raw file with the pixel data type of the original image. Since each step in the decoding process is the inverse of the corresponding encoding process, the details are omitted.

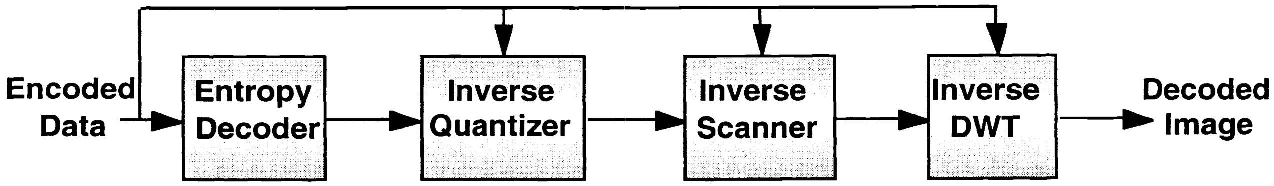


Figure 5 -- WTCQ Grayscale Imagery Decoder Block Diagram

One noteworthy aspect of the WTCQ decoder is the ability to progressively decode trellis coded quantized data.[14] The user specifies a decoding rate, which translates into a number of bytes to decode. The encoded file is truncated so that its size equals the desired number of bytes to decode, then the truncated file is decoded. Although aspects of TCQ prevent optimal progressive decoding in the least significant bit plane, the quality of the progressively decoded imagery is sufficient for most applications.

2.3 WTCQ features beyond grayscale image coding: The WTCQ algorithm for lossy coding of grayscale images is augmented to support lossy coding of color imagery, lossless coding of color and grayscale imagery, and lossless and lossy coding of bi-level and compound (comprised of bi-level and grayscale regions) imagery.

2.3.1 Lossy coding of color imagery: Lossy coding of color imagery applies a color transform - the YCrCb transform[6] - prior to the DWT. The DWT's of luminance and two chrominance bands are taken independently. If the classifier is used, the class maps are generated from the DWT of the luminance band, and this set of class maps is applied to all three bands. After scanning, the sequences associated with Y, Cr, and Cb are concatenated and processed through the rest of the encoding steps in aggregate.

2.3.2 Lossless coding of color and grayscale imagery: To provide an initial capability for lossless coding within WTCQ, a simple residual coding technique was developed. Within the encoder, the quantized TCQ indices are inverse quantized, inverse scanned, and the inverse DWT is computed. A residual (or error) image is computed by subtracting the inverse DWT result from the original image. This residual is encoded using the bit-plane coder, and is appended to the lossy encoded data. The WTCQ decoder can, based on user selection, decode the residual to create a lossless decoded result. Although this method may not guarantee lossless decoding if encoding is done on one type of machine and decoding on another, the performance was such that lossless WTCQ ranked in the top third of the Sydney evaluation.

2.3.3 Coding of bi-level imagery: For lossless coding of bi-level imagery, the image is bit-plane encoded as a single bit-plane, with no transformations or quantization applied. The approach is analogous to JBIG, and JBIG-like bit-plane coder contexts are used when losslessly encoding bi-level images. For lossy encoding of bi-level imagery, the image is scaled and processed through the grayscale lossy encoder. After decoding, the decoded image is thresholded at the 50% point to create the bi-level decoded image.

2.3.4 Tiling and coding of compound imagery: Compound imagery is encoded using the tiling feature of WTCQ. Based on a user-specified tile size, the input image is partitioned into non-overlapping rectangular regions. Each region is encoded separately, and the encoded tiles are concatenated into a single encoded file. Tiling can be used to encode / decode arbitrarily large input images on machines with limited memory. For compound images that can be segmented into rectangular regions, WTCQ used tiling and a bi-level region detector to losslessly encode bi-level regions and encode the grayscale regions with the remaining bits. This emulated a more general segmenter that might be used outside of a standard JPEG-2000 compression algorithm. Since this processing is outside the scope of standardization, special handling of compound imagery has been removed from WTCQ.

3. JPEG-2000 VERIFICATION MODEL 0

At the March, 1998 ISO JPEG-2000 meeting in Geneva, it was determined that the SAIC/UA implementation of WTCQ would be the basis of the initial verification model - VM 0 - with additions and modifications to support the second round of core experiments. The premise of the VM is that various organizations will contribute technologies to the VM, and that SAIC will integrate the various contributions and maintain the VM software. This process has begun with VM 0, and will continue for the duration of the JPEG-2000 standard development.

3.1 VM 0 description: The VM 0 software includes most of the features from the Sydney submission of WTCQ, with the following modifications:

- DWT - In addition to the (9,7) and (10,18) floating point transforms, (2,10) and (13,7) integer transforms are selectable as command line options. The (2,10) transform is from the current implementation of Ricoh California Research Center's CREW algorithm, and the (13,7) transform is provided by TeraLogic, Inc. Other user defined floating point and integer transforms can be specified by supplying a file containing the filter coefficients. The wavelet decomposition trees available as command line options are mallat, spacl, and packet. User defined wavelet decomposition trees can also be specified. The "mallat" tree with (9,7) filter is the default mode. Note, if integer wavelets are used, encoding is done losslessly. There is currently no ability to *explicitly* quantize when using integer wavelets, although implicit quantization can be done during progressive decoding.

- Classifier - The classifier described in Section 2.1.2 has been replaced by a simpler 2-Class classifier developed by Cannon-France. Also, the entropy encoding of the class maps has been made more efficient by using two-dimensional context modeling.

- Rate Allocator - As an alternative to the WTCQ rate allocator using rate-distortion models, a fixed quantization table ("Q-table") can be used. This is analogous to the quality factor in the current JPEG standard. When a Q-table is used, no rate iteration occurs, no sequence statistics are calculated, and the compressed file size varies based on the image. The WTCQ rate iteration is the default rate control technique.

- Quantization - In addition to the default TCQ, a scalar quantizer is available in VM 0. Scalar quantization rate-distortion models are included in the rate allocator, and trained codewords are computed using the technique described in Reference 8.

- Scalability - The VM 0 coder generates a codestream which is progressively decodable by accuracy (SNR) and/or by resolution. The progressive decoding is specified at decode time. Progressive decoding by SNR and/or resolution is supported for grayscale and color imagery, using the techniques described in Section 3.2.

- Lossless coding - The residual image method of lossless coding used in WTCQ is not part of VM 0. Lossless coding is accomplished using integer wavelet transforms, no explicit quantization, and bit-plane coding. A reversible color transform, contributed by Ricoh California Research Center, is available for lossless coding of color images.

- Visual Weighting - Sharp Labs of America has contributed table-driven visual weighting code, which is integrated in VM 0. The visual weighting is intended to improve the quality of printed (hard copy) decoded images at low bit rates. The visual weighting is done during encoding and modifies the rate allocation so that a *weighted* mean square error is minimized. The weighting is on the basis of the human visual system contrast sensitivity function at a certain viewing distance. Visual weighting tables are available for several print resolutions.

- Error Resilience - Texas Instruments and Motorola have contributed code for error resilience (e.g., re-synch markers, error concealment). The bit-plane coder has no inter-sequence dependencies, enabling insertion of re-synch markers between encoded sequences. The VM 0 codestream header contains the length, in bytes, of each encoded sequence. The error resilience function verifies that the correct number of bytes are read between re-synch markers and verifies that the arithmetic decoding of each sequence ends at the correct byte. If either test fails for a sequence, the DWT coefficients for that sequence are set to zero (error concealment).

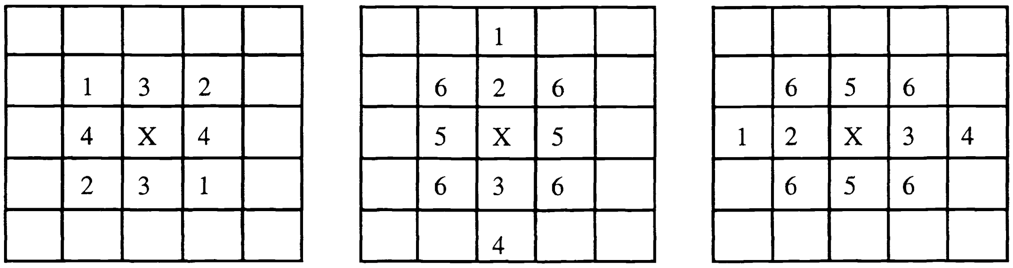
- Independent JPEG Group (IJG) Code - The IJG implementation of JPEG is included in the distribution of VM 0.

- Region of Interest (ROI) Coding - Ericsson has contributed ROI coding functions within VM 0. These functions allow a rectangular ROI to be decoded at a higher quality than the background (image area outside of the ROI).

3.2 VM 0 scalability features: Associated with the improvements to scalability, several refinements were made to the WTCQ bit-plane coder. The major changes were the introduction of de-interleaving within bit-planes and an improvement to the context modeling. Within a given bit-plane of each sequence, the bits are “de-interleaved” into three sequences of the following types: 1) bits predicted to be newly "significant", 2) "refinement" bits, and 3) bits predicted to be remain "insignificant". Significant and refinement are consistent with the terminology of Reference 9. Specifically, a wavelet coefficient becomes significant when its value is "1" in the current bit plane and all previous bit planes for that coefficient have been "0". Refinement bits are bits from wavelet coefficients that have had non-zero bits in at least one previous bit plane. The decision as to whether a given bit is predicted to be significant or insignificant is based on whether neighboring coefficients are already significant. The three de-interleaved bit streams within a bit-plane are arithmetically encoded to the compressed codestream, in order 1), 2), 3).

The main idea behind de-interleaving is, for each bit-plane, to order the bits so that bits of value "1" are placed in the codestream ahead of bits of value "0". This requires predicting which quantizer indices become newly significant at each bit-plane. This prediction must only use information that will be available to the decoder. In Reference 12, significance is predicted in two stages. First, if any of the 8-connected neighbors are significant, then the index in question is predicted to become newly significant (assuming it is insignificant in previous bit-planes). If this first stage does not predict that the index will become significant, then the second test is applied. This test predicts significance if the "parent" index in the wavelet tree is significant, where the "parent" is the index at the corresponding location one level higher in the wavelet tree. The VM 0 significance prediction is similar, but uses 4-connected neighbors and does not use the parent index to predict significance. The choice of 4-connectedness reduces complexity. The parent context is not used in order to eliminate inter-sequence dependencies at the decoder. Inter-sequence dependencies limit error resilience and limit scalable decoding flexibility.

Three types of context models are used in VM 0, one for coding "non-refinement" bits (predicted significant or predicted insignificant), one for refinement bits, and one for coding sign bits. These context models, illustrated in Figure 6, are related to those found in Reference 15. Note that the contexts for non-refinement bits are orientation sensitive to exploit dominant spatial correlation in particular wavelet subbands. For the non-refinement bits, the context is generated using the sign flag bits (as described in 2.1.5) from the bit locations indicated in Figure 6a. Bits from locations having the same numerical label are logically OR-ed to form a context bit. For example, the LH context model is six bits, generated by combining bits from 10 neighboring locations. The refinement context is relatively simple, using neighboring refinement bits to exploit within-plane correlations. The sign bit context model is somewhat different from the other two in that three-valued logic is used at each context model position. Thus, there are 3^4 contexts used for sign bit encoding. At each of the four locations, the context bit value is "0" if the sign bit has not yet been encoded, "+" if a positive sign bit has been encoded, and "-" if a negative sign bit has been encoded.



	1	3	2	
	4	X	4	
	2	3	1	

LL & HH Orientation

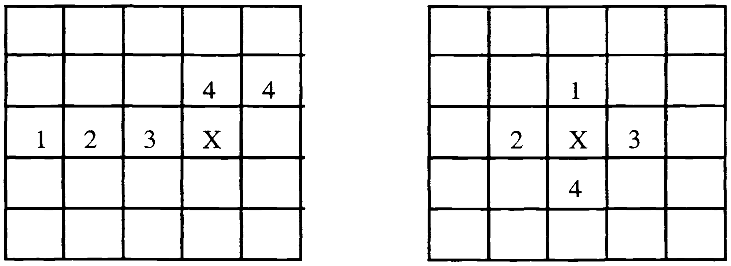
		1		
	6	2	6	
	5	X	5	
	6	3	6	
		4		

LH Orientation

	6	5	6	
1	2	X	3	4
	6	5	6	

HL Orientation

a) Context Bit Locations for Predicted Significant and Predicted Insignificant Bits



			4	4
1	2	3	X	

Refinement Bit Context

		1		
	2	X	3	
		4		

Sign Bit Context

b) Context Bit Locations for Refinement Bits and Sign Bits

Figure 6 -- Context Bit Locations for VM 0 Bit-plane Coder. "X" indicates the location of the bit to be encoded.

The bit-plane coding scheme allows for the extraction of any scaled version (resolution or quality) of the image from a single unified codestream. For example, let b(i,j) represent the bits from the jth bit-plane from sequence i. Here, i=0 indicates the lowest frequency subband, i=1 indicates the LH subband at the lowest resolution, i=2 indicates the HL subband at the lowest resolution, and so on. Similarly, j=N indicates the most significant bit plane, N-1 indicates the next most significant bit-plane, and so on. The codestream then consists of the concatenation of the b(i,j), in the following order: b(0,N), b(0,N-1), ..., b(0,0), b(1,N), b(1,N-1) ,..., b(1,0), b(2,N), b(2,N-1), ..., b(2,0),... With the appropriate pointers and/or a smart parser, an image at any resolution and quality is easily extracted from this codestream. For example, to get the image with resolution scale one size larger than the lowest frequency subband, and quality scale corresponding to three bit-planes, the codestream can be parsed to obtain b(0,N), b(0,N-1),b(0,N-2), b(1,N), b(1,N-1), b(1,N-2), b(2,N), b(2,N-1), b(2,N-2), b(3,N), b(3,N-1), b(3,N-2).

3.3 VM 0 compression performance: It is difficult to fully characterize the performance of the JPEG-2000 VM, due to the many ways in which the coder can be configured. However, for comparison purposes, representative results for lossy and lossless coding of grayscale imagery are presented herein.

Table 1 presents performance results for a low complexity configuration of VM 0 and compares these results to those obtained with the Sydney version of WTCQ. The low complexity coder uses the (9,7) wavelet filter with dyadic decomposition, non-statistical rate allocation ("Q-table"), scalar quantization, and progressive decoding by SNR from a single codestream. The low-complexity coder achieves excellent performance on imagery such as "bike", but does not match the performance of WTCQ in all cases. It is important to note that both the "WTCQ" and the "Low-Complexity" results are achieved using capabilities available within VM 0. Since JPEG-2000 will standardize a *decoder*, it is seen that the VM 0 decoder supports a variety of user interests. Clearly, the JPEG-2000 convergence process is leading to a compression algorithm with the flexibility to support users interested in the highest quality as well as users interested in low complexity - all within a system that provides the desired performance features described in Section 1.

Table 1 -- Lossy Grayscale Coding Results On Selected JPEG-2000 Test Imagery
Table Entries are Peak Signal-to-Noise Ratio in Decibels

		Rate (bits per pixel)					
Test Image	Encoder	2.0	1.0	0.5	0.25	0.125	0.0625
aerial2	WTCQ	39.66	34.09	31.10	28.90	26.83	24.84
	Low-Complexity	38.54	33.37	30.69	28.53	26.48	24.55
bike	WTCQ	43.83	37.55	32.91	29.11	26.09	23.61
	Low-Complexity	44.06	38.10	33.33	29.28	25.78	23.32
sar2	WTCQ	34.05	28.48	25.88	24.29	23.27	22.45
	Low-Complexity	32.96	27.68	25.48	23.97	23.01	22.28
woman	WTCQ	44.75	38.73	34.15	30.45	27.41	25.63
	Low-Complexity	44.02	38.46	33.60	29.74	27.23	25.34

Table 2 compares performance of the VM 0 lossless compression to that achieved by JPEG-LS. The VM 0 results use the (13,7) integer wavelet and bit-plane coding. Although the VM 0 algorithm is more complex than JPEG-LS, it delivers comparable lossless compression performance along with the many useful scalability features of JPEG-2000. This example shows one way in which JPEG-2000 complements the existing suite of JPEG and JBIG still image compression standards.

Table 2 -- Lossless Grayscale Coding Results On Selected Test Imagery
Table Entries are Bits per Pixel

	target	bike	woman	café	aerial2	barbara	lenna	AVERAGE
JPEG-LS	2.186	4.356	4.451	5.092	5.288	4.863	4.236	4.353
JPEG-2000 VM 0	2.390	4.402	4.364	5.180	5.219	4.582	4.158	4.328

4. ON-GOING WORK

The JPEG-2000 community is actively experimenting with compression technologies in every area described in this paper, with the goal of optimizing the JPEG-2000 algorithm within the quality, functionality, and complexity design space. Areas of experimentation include; fixed-point DWT implementations, entropy coding techniques, codestream syntax, progressive visual weighting, progressive to lossless decoding of regions of interest, and several others. The SAIC/UA team, along with several collaborators, is currently focused on two areas: improving lossy encoding when using reversible wavelet transforms, and improving the functionality and performance of scalability features. One specific activity is development of a parser that converts among different embedded codestream orderings, including arbitrary subband orderings to support various display and printing applications.

5. ACKNOWLEDGMENTS

We gratefully acknowledge the contributions of Tom Flohr, Jim Owens, Don Fisher, Janet Rountree and Bobby Hunt of SAIC to the development of WTCQ and VM 0. We further recognize the participants within the JPEG-2000 community for their contributions to the VM.

6. REFERENCES

1. JPEG-2000 Requirements and Profiles Version 1.1, *ISO/IEC JTC 1/SC 29/WG 1 Doc. No. N716*, Dec. 20, 1997.
2. J. H. Kasner and M. W. Marcellin, "Adaptive wavelet coding of images", *Proc. of IEEE International Conference on Image Proc.*, vol. III, pp. 358-362, 1994.
3. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. on Image Proc.*, vol. 1, pp 205-220, Apr. 1992.
4. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding", *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 519-521, Oct. 1996.
5. R. Joshi, H. Jafarkhani, J. H. Kasner, T. Fischer, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, "Comparison of different methods of classification in subband coding of images", *IEEE Trans. on Image Proc.*, vol. 6, pp. 1473-1486, Nov. 1997.
6. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1992.
7. J. W. Woods and T. Naveen, "A filter based bit allocation scheme for subband compression of HDTV", *IEEE Trans. on Image Proc.*, vol. 1, pp. 436-440, Jul. 1992.
8. J. H. Kasner, M. W. Marcellin, B. R. Hunt, "Universal Trellis Coded Quantization", *submitted to IEEE Trans. on Image Proc.* (see http://www-spacl.ece.arizona.edu for preprint).
9. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Trans. on Sig. Proc.*, vol. 41, pp. 3445-3462, Dec. 1993.
10. 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 Tech.*, vol. 6, pp. 243-250, June 1996.
11. J. Li and S. Lei, "Rate-distortion optimized embedding", *Picture Coding Symposium*, Berlin, Germany, pp. 201-206, Sept. 10-12, 1997. (See also, J. Li, P. Cheng, and C.-C. J. Kuo, "On the improvements of embedded zerotree wavelet (EZW) coding", *Proc. of SPIE Visual Communication and Image Processing*, vol. 2601, Taipei, Taiwan, pp. 1490-1501, May 1995.)
12. E. Ordentlich, M. J. Weinberger, G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients", *Proc. of 1998 IEEE Data Compression Conference*, Snowbird, Utah, March 29 - April 1, 1998.
13. A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, "CREW: Compression with reversible embedded wavelets", *Proc. of IEEE Data Compression Conference*, Snowbird, Utah, pp. 212-221, March 1995.
14. P. J. Sementilli, A. Bilgin, F. Sheng, M. W. Marcellin, and J. Kieffer, "Progressive transmission in trellis coded quantization-based image coders", *Proc. of IEEE International Conference on Image Proc.*, Santa Barbara, CA, Oct. 1997.
15. X. Wu, "Higher-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression", *Proc. of 31st Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1997.