Compressed/Reconstructed Test Images for CRAF/Cassini

S. Dolinar, K.-M. Cheung, I. Onyszchuk, F. Pollara, and S. Arnold
Communications Systems Research Section

This article briefly describes a set of compressed, then reconstructed, test images submitted to the CRAF/Cassini project as part of its evaluation of near-lossless high-compression algorithms for representing image data. A total of seven test image files were provided by the project.

The seven test images have been compressed, then reconstructed with high quality (root-mean-square error of approximately one or two gray levels on an 8-bit gray scale), using discrete cosine transforms or Hadamard transforms and efficient entropy coders. The resulting compression ratios varied from about 2:1 to about 10:1, depending on the activity or randomness in the source image. This was accomplished without any special effort to optimize the quantizer or to introduce special postprocessing to filter the reconstruction errors.

A more complete set of measurements, showing the relative performance of the compression algorithms over a wider range of compression ratios and reconstruction errors, shows that additional compression is possible at a small sacrifice in fidelity.

I. Introduction

This article briefly describes a set of compressed, then reconstructed, test images submitted to the CRAF/Cassini project as part of its evaluation of near-lossless high-compression algorithms for representing image data. A total of seven test image files were provided by the project. Five test images (d1, f2, h2, j1, and l2) are star fields from the Hubble Space Telescope, and two images (saturn1 and saturn2) are views of Saturn from Voyager. Three of these original images are shown in Figs. 1(a), 1(b), and 1(c). The dimensions of the Hubble images and the Saturn images are 256 × 256 and 800 × 800, respectively. All images are represented by 8-bit pixel values in the range 0 to 255.

A total of 12 compressed/reconstructed images were returned to the project, as listed in Table 1. Three of the reconstructed images (marked by arrows ⇒ in Table 1) are shown in Figs. 2(a), 2(b), and 2(c) for comparison with the originals. Each of the seven test images was compressed using an algorithm that produces high quality reconstructed images (left-hand portion of Table 1)

with a root-mean-square error (RMSE) of about one gray level. Alternate compressed/reconstructed image versions (right-hand portion of Table 1) were also provided for five of the seven test images. Three of these alternate images show how much additional compression is possible at a small sacrifice in image fidelity, and the other two alternate images illustrate the effectiveness of a different compression algorithm, which is simpler to implement on the spacecraft.

In Table 1, the RMSE is computed in absolute units on an 8-bit gray level scale. The quoted bit rate is the number of bits per pixel required to encode the compressed image before reconstruction. The compression ratio is calculated as 8 bits divided by the bit rate.

II. Description of the Compression System

The specific algorithms used to compress, then reconstruct the twelve images listed in Table 1 can be described with reference to the block diagram in Fig. 3. The various blocks in this diagram are described in the following sections.

A. Data Transform/Inverse Data Transform

A discrete cosine transform (DCT) was applied to obtain ten of the twelve compressed/reconstructed images. The DCT is near-optimal for a wide variety of images, and is fast becoming an industry standard for high compression. The DCT was calculated using floating point arithmetic and applied to 8 × 8 sub-blocks of the image.

A Hadamard transform (HT), also applied to 8 × 8 blocks, was used for the remaining two compressed/reconstructed images. The HT is generally not as effective as the DCT, but it performed reasonably well for the seven test images. The HT is simpler to implement than the DCT, because it can be computed with integer arithmetic and without multiplications.

Mathematically, both transforms are defined as unitary transformations on each 8 × 8 block of data. The image array X is decomposed into 8 × 8 blocks X^{IJ}, i.e., $X = [X^{IJ}]$, and the array of transform coefficients T is built from 8 × 8 blocks T^{IJ}, i.e., $T = [T^{IJ}]$. The blocks of transform coefficients are given by

$$T^{IJ} = \begin{cases} \frac{1}{8}\, C X^{IJ} C^T & \text{for DCT} \\ \frac{1}{8}\, H X^{IJ} H & \text{for HT} \end{cases}$$

where the 8 × 8 matrices $C = [c_{ij}]$ and $H = [h_{ij}]$ are defined by

$$c_{ij} = \begin{cases} \sqrt{2}\cos\left(\frac{\pi\, i(2j+1)}{16}\right) & i = 1,2,3,4,5,6,7 \\ & j = 0,1,2,3,4,5,6,7 \\ 1 & i = 0,\ j = 0,1,2,3,4,5,6,7 \end{cases}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} = H^T$$

Both the DCT and the HT are exactly invertible in principle. The inverse transform formulas are simply:

$$X^{IJ} = \begin{cases} \frac{1}{8}\, C^T T^{IJ} C & \text{for DCT} \\ \frac{1}{8}\, H T^{IJ} H & \text{for HT} \end{cases}$$

Both the DCT and HT can be implemented with fast algorithms requiring fewer arithmetic operations than direct implementation of the matrix multiplications in the above expressions. The DCT requires real multiplications and additions, whereas the HT requires only integer additions and no multiplications. For this study, the DCT's real arithmetic was approximated by 32-bit floating point multiplications and additions, whereas integer arithmetic must generally be substituted in practice. Integer approximations in the computation of the transform coefficients can produce additional errors in the reconstructed images.

Transforming a block of data does not change its information content. Useful transforms concentrate most of the data's energy into a small number of transform coefficients. Low-energy transform coefficients can be encoded with a small number of bits.

B. Quantizer/Dequantizer

The DCT produces real transform coefficients, which of necessity must be quantized to a finite number of bits. The HT produces quantized coefficients, but the quantization is impractically fine: the transform coefficients have eight times the dynamic range and one-eighth the granularity of the input, requiring six extra bits to represent exactly. So additional quantization is also performed for the HT.

Once the transform coefficients have been quantized, the dequantizer in Fig. 3 can only reconstruct an approximate version of the true coefficients, and the inverse data transform can no longer regenerate the exact original image. Except for the possible errors (noted above) in computing the transform coefficients, quantization of the computed coefficients is the only step in Fig. 3 that introduces errors in the reconstructed image data. The choice of quantization coarseness and uniformity thus sets the fidelity of the reconstructed image. This choice also limits the extent to which the entropy encoder can compress the image.

A uniform quantizer was used for all twelve of the compressed/reconstructed images. Mathematically, the output of the quantizer is an array of 8×8 blocks, $Q = [Q^{IJ}]$, where each block, $Q^{IJ} = [Q_{ij}^{IJ}]$, is obtained from the 8×8 block of transform coefficients, $T^{IJ} = [T_{ij}^{IJ}]$, as

$$Q_{ij}^{IJ} = \left\lfloor \frac{T_{ij}^{IJ}}{q} + \frac{1}{2} \right\rfloor$$

where q is the quantizer step size and $\lfloor a \rfloor$ is the largest integer less than or equal to a. This uniform quantizer is actually "triply uniform." Not only are the quantization levels for each transform coefficient equally spaced, but the quantization step size q is the same for all 64 coefficients within each 8×8 block, and the step size does not change from block to block. A step size of $q = 4$ was used for nine of the twelve images, and a coarser step size of $q = 8$ was used for three of the alternate images.

Nonuniform quantization rules are available [1,2] to match the quantizer to the human visual response by selectively quantizing low-frequency DCT or HT coefficients more finely than high-frequency coefficients. Similarly, other algorithms can adapt the quantizer to the local statistics of the data on a block by block basis. However, such quantizers have so many adjustable parameters that a nonuniform quantizer optimized for a small set of test images would not fairly reflect the performance of the quantizer for untested images. Research is ongoing to find universal nonuniform, adaptive quantizers that consistently outperform the uniform quantizer.

C. Entropy Encoder/Entropy Decoder

The entropy encoder losslessly encodes the array of quantized transform coefficients Q into a bit stream $\mathbf{b}$ with bit rate approaching the entropy per coefficient of Q. Several types of highly efficient coders are available for this purpose. Among these are the Gallager–van-Voorhis–Huffman (GVH) coder [3], a variant of IBM's arithmetic Q-coder [4] being developed for the Joint Photographic Experts Group (JPEG) standard [2], and a baseline Huffman coder for this same standard. Variations of these coders were used to compress, then reconstruct the images listed in Table 1.

The first step in all three coding schemes is to arrange the quantized transform coefficients $Q^{IJ} = [Q_{ij}^{IJ}]$ into an ordered sequence, starting with the DC coefficient ($i = j = 0$) in the upper left-hand corner of the transformed block. The remaining 63 coefficients (AC coefficients) are ordered in some fashion, generally via a zigzag readout starting at the upper left-hand corner and working toward the lower right-hand corner. This zigzag sequence arranges the AC coefficients in increasing order of spatial frequency.

The GVH coding scheme is derived based on two observations on the quantized AC and DC coefficients. First, the AC coefficients and the differences between adjacent DC coefficients have two-sided geometric distributions. Second, runs of zeros occur frequently in the zigzag sequences of AC coefficients, especially at high compression ratios. By extending a result originally shown by Gallager and van Voorhis, a near-optimal adaptive coding scheme for prefix coding the two-sided geometric source is derived, avoiding both binning calculations and the Huffman tree generation algorithm. Instead, this scheme estimates the local activity of each 8×8 block by counting the number of zeros in the block or in some preceding blocks, and adaptively encodes the transform coefficients using simple pipelined table lookup operations. An optional runlength code can also be used to encode runs of zeros in the zigzag sequence of AC coefficients.

The Q-coder is a lossless, binary entropy coder, developed by researchers at IBM, that efficiently implements an Elias code [5] on an input bit sequence. A coarsely quantized approximation to the real interval [0,1], or a scaled version thereof, is recursively subdivided into two sections, whose sizes are proportional to probability estimates that the bit currently being coded is a 0 or 1. By dynamically updating these estimates using a finite-state machine, the Q-coder adapts to input data statistics (unlike a Huffman encoder, which requires statistics before coding), which makes it both robust and efficient. A coding model forms a binary sequence from a raster-scan ordering of the 64 quantized DCT or HT coefficients in each block. For each integer, an equal-to-zero flag bit, the sign bit, the position of the most significant bit, and then the least significant bits are sent through the Q-coder. Runlength coding was not performed because the small quantizer step sizes used make it unprofitable. A simple model, using only 12 prob-

ability estimates, each of which is a 5-bit number (state), was used instead of the complex model in Section 8 of [2] that requires 252 estimates. Negligible bit rate reductions are expected with the latter model for the images tested. The output sequence length is very close to a value calculated from the input stream entropy. Since only additions, subtractions, and comparisons are utilized by a Q-coder, it is simple and fast in practice. The particular variant implemented is described in Section 12 of [2].

The Joint Photographic Experts Group of the International Standards Organization/International Telegraph and Telephone Consultative Committee (ISO/CCITT) [2] is currently developing an international standard for still-image compression. In its baseline version, the proposed algorithm consists of an 8×8 DCT, coefficient quantization, and Huffman or arithmetic coding. This scheme provides a near-lossless, high-compression image coding capability, which preserves image fidelity at compression rates competitive or superior to most known techniques. The DCT's 64 coefficients are independently uniformly quantized with a different step size for each coefficient. The DC component is differentially encoded, and the AC components are runlength encoded. Finally, some of the most significant bits of each resulting code are further encoded with a variable length code; the remaining bits are transmitted as they are. In the case of Huffman encoding, the JPEG default tables were used. The tables for the Huffman codes can be easily customized to adapt to the particular image source of interest.

The bit rates listed in Table 1 are the bit rates achieved by a variant of the Q-coder. The GVH and JPEG coders achieve comparable bit rates averaging 0.2 to 0.3 bits per pixel higher than the Q-coder for the images in Table 1.

D. Noiseless Channel

Because the entropy code is lossless, the entropy decoder is able to reconstruct an exact replica of the quantized transform coefficients, given the compressed files of coded bits. However, if the channel in Fig. 3 were not noiseless, the decoding process would be severely disrupted and errors might propagate wildly.

E. Preprocessing and Postprocessing (Not Implemented)

The original image data supplied by the project were actually obtained by preprocessing 12-bit data available from the cameras. The 12-bit data were subjected to a square-root operation, then quantized to 8 bits. No additional preprocessing was performed on the 8-bit data before the transform operation depicted in Fig. 3. The 8-bit data provided by the project were considered to be the original image data for the purposes of the tests reported here.

Some sort of postprocessing is often desirable, following the inverse data transform, to make the reconstruction errors less noticeable. Postprocessing can be applied to remove visually disturbing blockiness in images that have been highly compressed by block-transform techniques. No such postprocessing was performed for any of the twelve compressed/reconstructed images, because it would unfairly mask the true efficacy of the compression algorithms.

A crude form of nonoptimum postprocessing actually did take place after the inverse transform, because the output values had to be quantized to 8 bits to fit the original image format. This quantization step should be skipped or deferred if the output data are subjected to further postprocessing (such as removal of the square-root operation mentioned above).

III. Additional Performance Results

The twelve reconstructed images listed in Table 1, including the three images shown in Figs. 2(a), 2(b), and 2(c), were chosen to reflect the desires of the project to obtain compression ratios in the range of about 2:1 to about 10:1 with essentially zero reconstruction error. A more complete set of measurements showing the relative performance of the compression algorithms over a wider range of compression ratios and reconstruction errors was also obtained. These results are plotted in Figs. 4(a), 4(b), 4(c), 5(a), 5(b), and 5(c) for the three images shown in Figs. 1(a), 1(b), and 1(c), using DCT-based algorithms. Figures 4(a), 4(b), and 4(c) show the bit rate (bits per pixel) of the compressed images as a function of the RMSE distortion, and Figs. 5(a), 5(b), and 5(c) show the corresponding compression ratios. In these figures the bit rates and compression ratios achieved by the three entropy coders are compared with each other and with an approximate bound based on the estimated entropy of the quantized DCT coefficients. This entropy "bound" is derived assuming stationary statistics throughout the image; it can sometimes be beaten by algorithms capable of adapting to locally varying statistics.

IV. Summary

The seven test images have been compressed, then reconstructed with high quality (RMSE of approximately one or two gray levels on an 8-bit gray scale) using DCT- or HT-based schemes and efficient entropy coders. The re-

sulting compression ratios varied from about 2:1 to about 10:1, depending on the activity or randomness in the source image. This was accomplished without making any special effort to optimize the quantizer or to introduce special postprocessing to filter the reconstruction errors.

A more complete set of measurements, showing the relative performance of the compression algorithms over a wider range of compression ratios and reconstruction errors, shows that additional compression is possible at a small sacrifice in fidelity.

References

[1] J. D. Eggerton and M. D. Srinath, "A Visually Weighted Quantization Scheme for Image Bandwidth Compression at Low Data Rates," *IEEE Trans. Commun.*, vol. COM-34, no. 8, pp. 840–847, August 1986.

[2] *Joint Photographic Experts Group (JPEG) Draft Technical Specification (Rev. 5)*, ISO/CCITT, Washington, D.C., January 15, 1990.

[3] K.-M. Cheung, P. Smyth, and H. Wang, "A High-Speed Distortionless Predictive Image-Compression Scheme," *TDA Progress Report 42-102*, vol. April–June 1990, Jet Propulsion Laboratory, Pasadena, California, pp. 73–90, August 15, 1990.

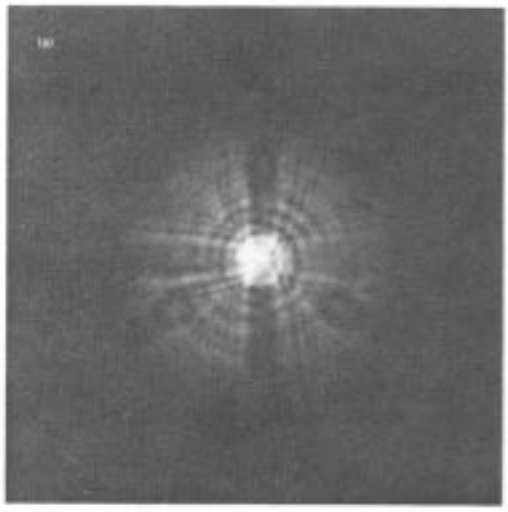
[4] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, Jr., and R. B. Arps, "An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder," *IBM J. Res. Develop.*, vol. 32, no. 6, pp. 717–726, November 1988.

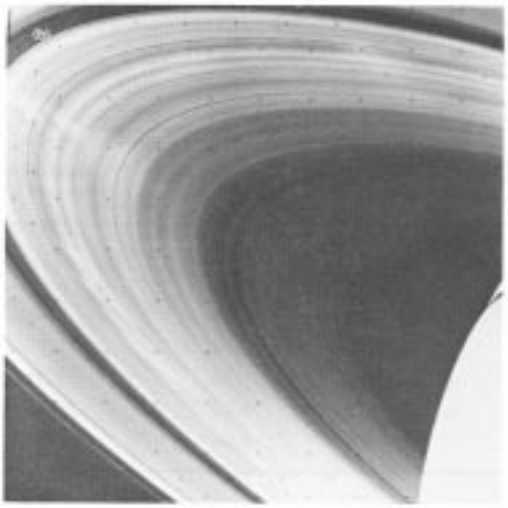
[5] R. E. Blahut, *Digital Transmission of Information*, Reading, Massachusetts: Addison-Wesley, p. 308, 1990.

Table 1. List of original and compressed/reconstructed test images. Arrows ⇒ denote the three images shown in Figs. 2(a), 2(b), and 2(c).

Original images	Compressed/Reconstructed images[a]			Alternate compressed/ reconstructed images[b]		
	RMSE, absolute	Bit rate, bits/pixel	Compression ratio	RMSE, absolute	Bit rate, bits/pixel	Compression ratio
d1	1.19	1.50	5.33	—	—	—
f2	1.15	1.15	6.96	—	—	—
h2	1.16	1.21	6.61	(H) 1.18	1.25	6.40
j1	1.19	3.77	2.12	(D) 2.33	2.56	3.12
l2	1.19	2.04	3.92	⇒ (D) 2.25	1.10	7.27
saturn1	0.97	1.31	6.11	⇒ (D) 1.43	0.75	10.67
saturn2	⇒ 0.87	0.82	9.76	(H) 0.95	1.00	8.00

[a] Using Q-coder and DCT with quantization step size $q = 4$.
[b] Using Q-coder and either DCT with quantization step size $q = 8$ (D) or HT with step size $q = 4$ (H).





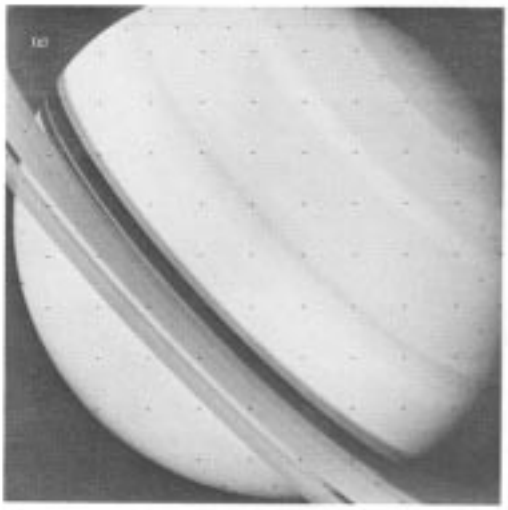
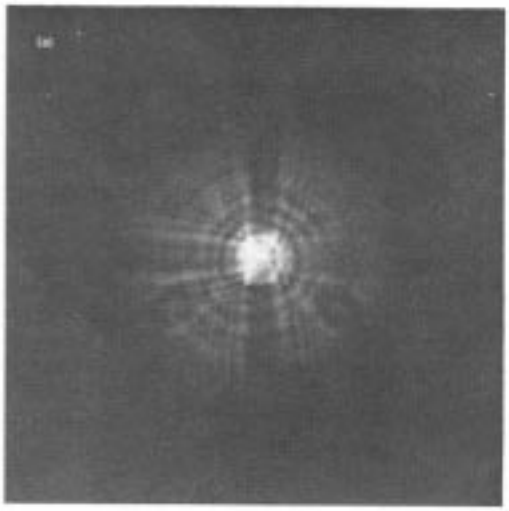
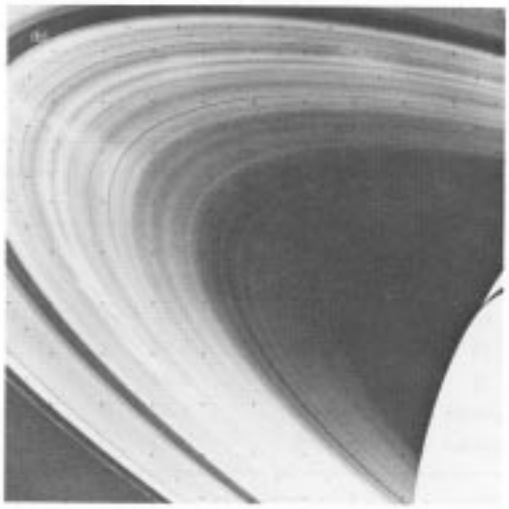


Fig. 1. Original test images for CRAF/Cassini: (a) Hubble image "I2," (b) Saturn image "saturn1," and (c) Saturn image "saturn2."





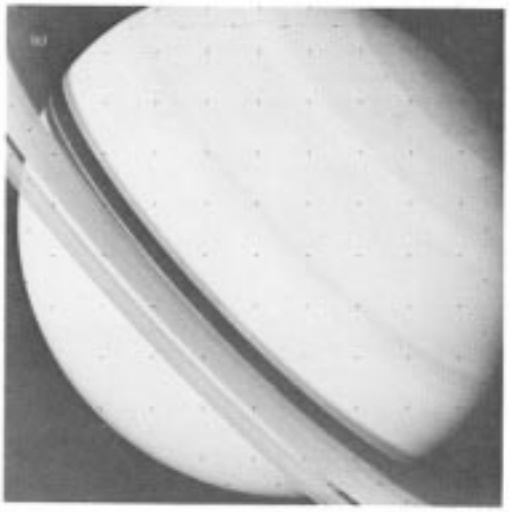


Fig. 2. Reconstructed test images for CRAF/Cassini: (a) Hubble image "I2," (b) Saturn image "saturn1," and (c) Saturn image "saturn2."

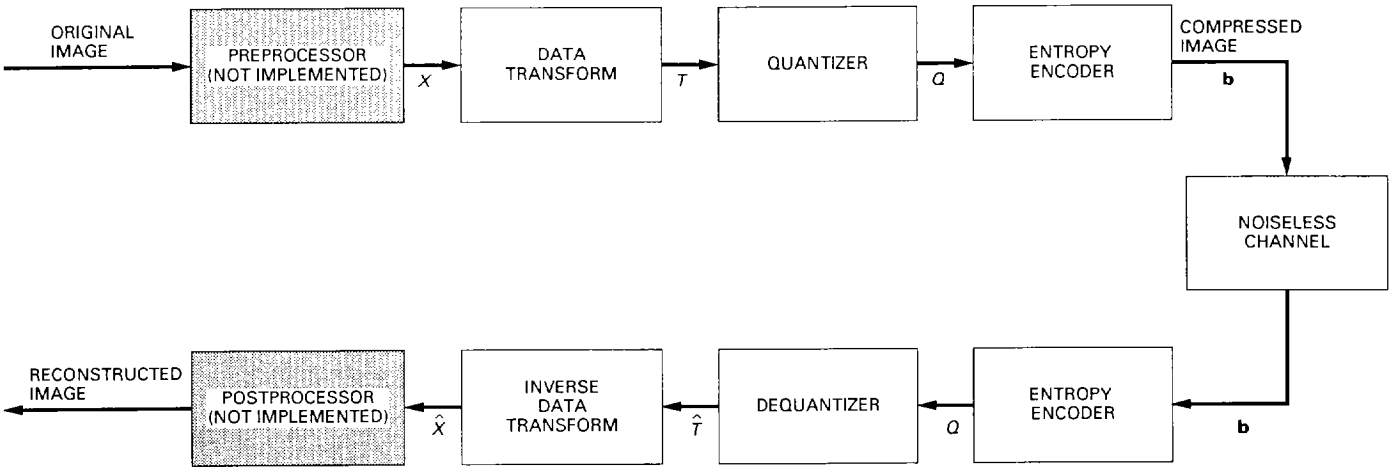


Fig. 3. Block diagram of compression system.

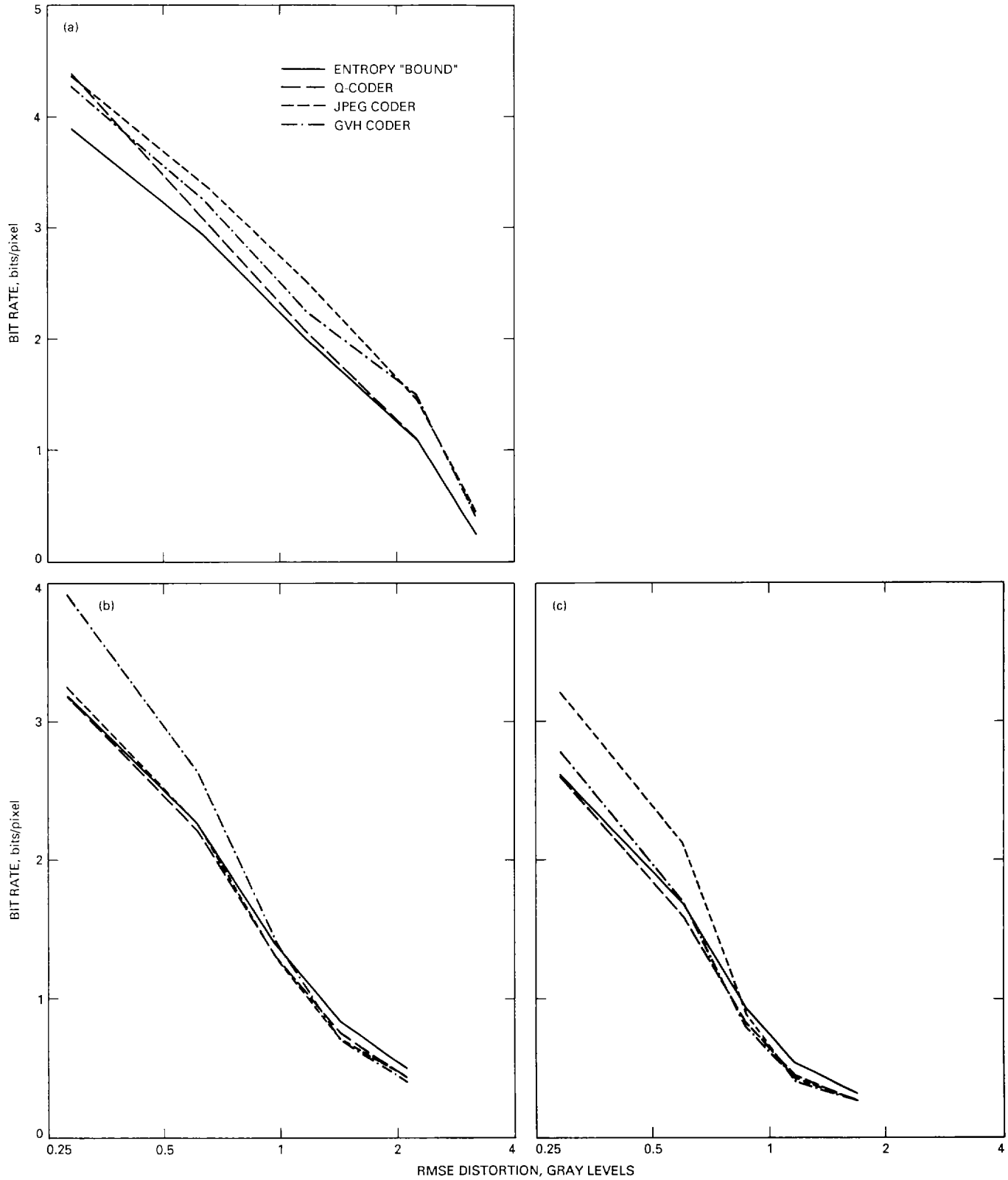


Fig. 4. Bit rate versus RMSE distortion for: (a) Hubble image "I2," (b) Saturn image "saturn1," and (c) Saturn image "saturn2."

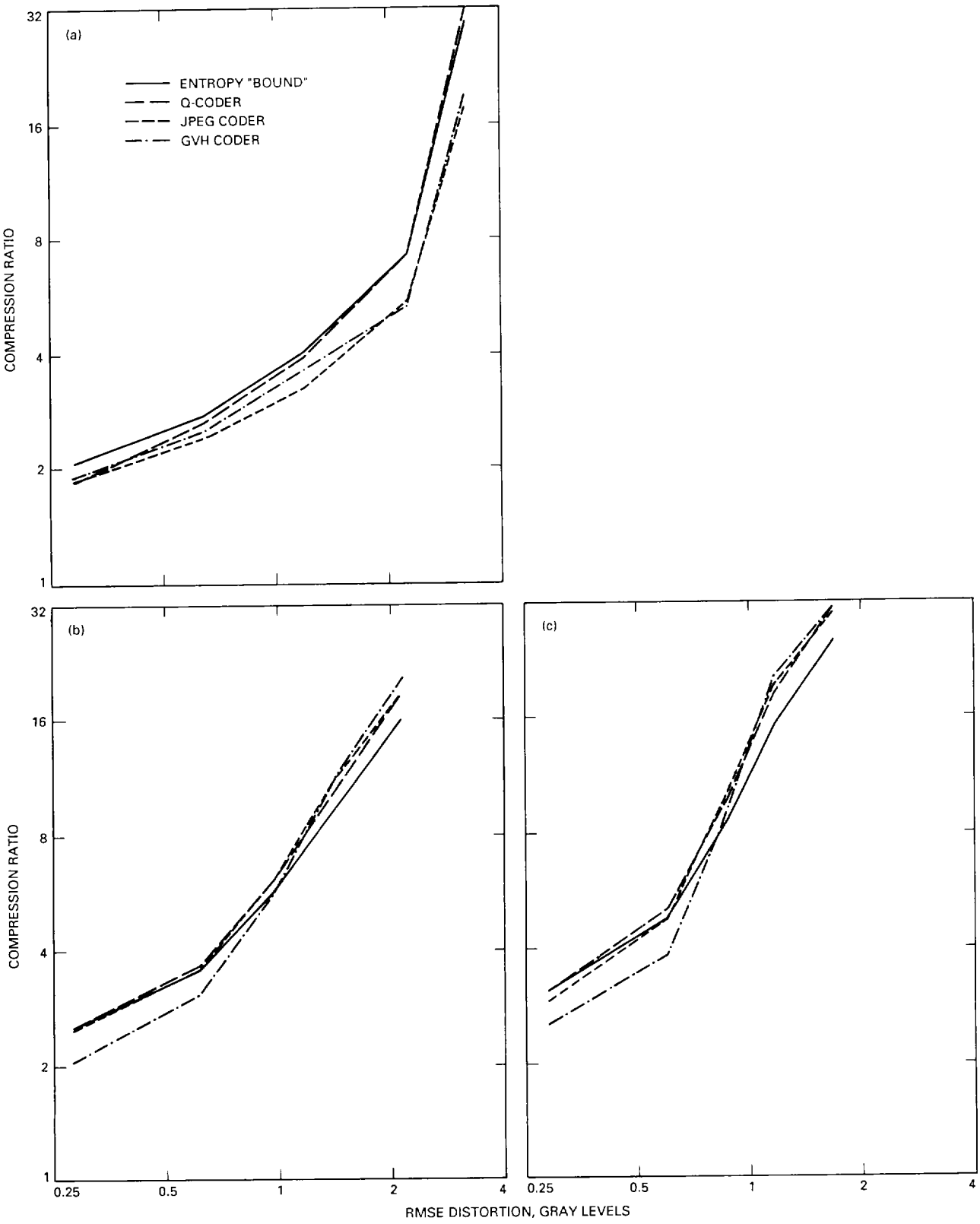


Fig. 5. Compression ratio versus RMSE distortion for: (a) Hubble image "I2," (b) Saturn image "saturn1," and (c) Saturn image "saturn2."