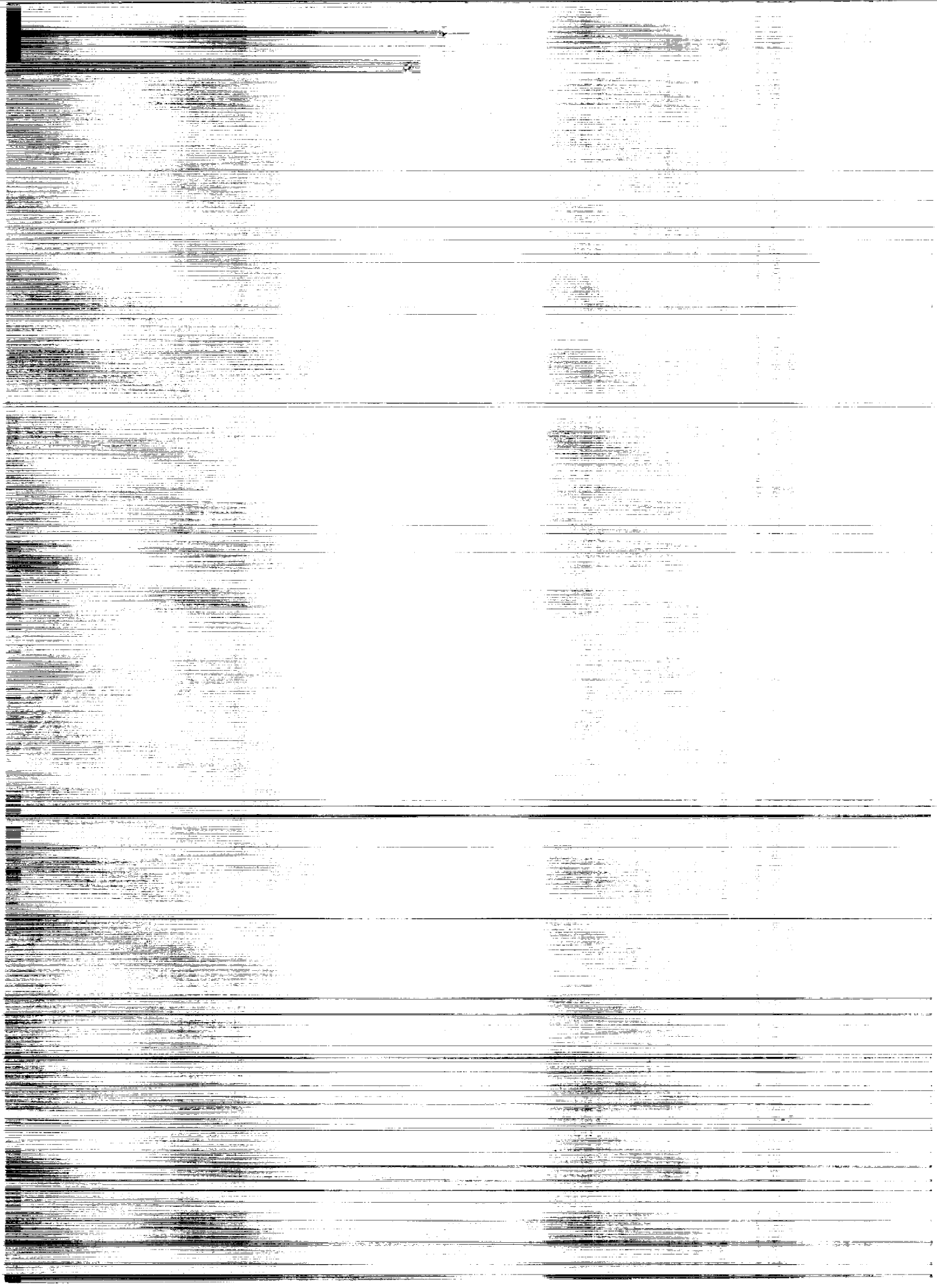
NASA Contractor Report 4263

Data Compression Techniques Applied to High Resolution High Frame Rate Video Technology

William G. Hartz, Robert E. Alexovich,
and Marc S. Neustadter

CONTRACT NAS3-24564
DECEMBER 1989



NASA Contractor Report 4263

Data Compression Techniques Applied to High Resolution High Frame Rate Video Technology

William G. Hartz, Robert E. Alexovich,
and Marc S. Neustadter
Analex Corporation
Cleveland, Ohio

Prepared for
Lewis Research Center
under Contract NAS3-24564

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1989

Table of Contents

SUMMARY

This report details an investigation of video data compression applied to microgravity space experiments using High Resolution High Frame Rate Video Technology (HHVT). An extensive survey of methods of video data compression, described in the open literature, was conducted. The survey examines compression methods employing digital computing. The results of the survey are presented. They include a description of each method and an assessment of image degradation and video data parameters. An assessment is made of present and near term future technology for implementation of video data compression in a high speed imaging system. Results of the assessment are discussed and summarized in a tabular listing of implementation status.

The results of a study of a baseline HHVT video system, and approaches for implementation of video data compression, are presented. Case studies of three microgravity experiments are presented and specific compression techniques and implementations are recommended.

The results of the investigation conclude that video data compression approaches for microgravity space experiments are experiment peculiar in requirements and no one single approach is universally optimum. It is shown, for the experiments studied, that data compression required is separable into two approaches: the first to limit data rates for storage, and the second to reduce data rates for transmission. For high resolution and/or high frame rate experiment requirements and real time compression, hardware implementations are currently limited, by technology, to methods that can be implemented using parallel processing and simple compression algorithms. Although theoretically attractive, no approach could be identified for focal plane processing alone, that could be implemented with state of the art hardware.

ACRONYMS

A

AC	Alternating Current, the remaining coefficients in a image transform
A/D	Analog to Digital conversion
ASIC	Application Specific Integrated Circuit
A-VQ	Address-Vector Quantization

B

BTC	Block Truncation Coding

C

CAQ	Constant Area Quantization
CCIR	Consultative Committee, International Radio
CCD	Charge Coupled Device
CID	Charge Injection Device
CMOS	Complementary Metal Oxide Semiconductor
CR	Conditional Replenishment or Compression Ratio

D

D/A	Digital to Analog conversion
DC	Direct Current, refers to the average pixel value in a transform
DCT	Discrete Cosine Transform
DCT/DPCM	Discrete Cosine Transform/Differential Pulse Code Modulation
DCT/MC	Discrete Cosine Transform/Motion Compensation
DFT	Discrete Fourier Transform
DM	Delta Modulation
DPCM	Differential Pulse Code Modulation
DSP	Digital Signal Processing

G

GaAs	Gallium Arsenide, a high speed semiconductor device
GSP	Graphics System Processor

H

HDTV	High Definition Television
HHVT	High Resolution, High Frame Rate Video Technology
HVS	Human Visual System

I

IDS	Intensity Dependent Spread

K

KLT	Karhunen-Loeve Transform

L

LPC	Linear Predictive Coding
LPF	Low Pass Filter
LZW	Lempel-Ziv-Welch algorithm

M

MAPS	Micro-adaptive Picture Sequencing
MC	Motion Compensation
MC/2D-DCT	Motion Compensation/Two-Dimensional Transform
MNVC	Minimum Noise Visibility Coding
MSE	Mean Square Error

N

NMSE	Normalized Mean Square Error
NTSC	National Television System Committee

O

O_2/N_2	Oxygen/Nitrogen gas atmosphere

P

PAL	Phase Alternation Line, color television system designed by Telefunken
PCAQ	Predictive Constant Area Quantization
PCM	Pulse Code Modulation
PE	Processing Element
Pixel	Picture Element
PROM	Pockel's Readout Optical Modulator
PRN	Pseudo-Random Noise
PSC	Perceptual Space Coding

R

RAM	Random Access Memory
RGB	Red Green Blue, a common digital color coordinate system

RLC	Run-length Coding
RS 170	Electronic Industries Association performance standards for monochrome display systems; RS 170A applies to color displays.

S

SNR	Signal-to-Noise Ratio
SQ	Scalar Quantization
SS	Space Station
SSF	Space Station Freedom

T

TDRSS	Tracking and Data Relay Satellite System
2-D	Two-dimensions
3-D	Three-dimensions

V

VDC	Video Data Compression
VLSI	Very Large Scale Integrated circuits
VQ	Vector Quantization

W

WHT	Walsh-Hadamard Transform

X

XFORM	Transform

Y

YIQ	A color coordinate system employed in broadcast television: The Y-component is luminance, The I- and Q-components are chrominance.

UNITS

This list contains abbreviations for units of measure used throughout this report.

B	byte
b	bit
bpp	bits per pixel
Byte/p	Bytes per pixel
cm	centimeter
dB	decibel
fr/s	frames per second
fsc	frequency of subcarrier
GB	Gigabytes (10^9 bytes)
Gp	Gigapixels (10^9 pixels)
hr	hour
lines/mm	lines per millimeter, a measure of optical resolution
Mb	Megabits (10^6 bits)
Mbps	Megabits per second (10^6 bits/sec)
MHz	Megahertz (10^6 cycle per second)
min	minute
Mp/s	Megapixels per second (10^6 pixels/sec)
ns	nanosecond (10^9 seconds)
p/fr	pixels per frame
sec	second

COMPRESSION TECHNIQUES

Introduction

The performance of each of the video data compression techniques presented here is very much dependent upon the statistics of the image data. Some of the relevant statistics include the entropy of the data within a frame, the correlation of picture elements (pixels) within and/or between frames, and the amount of detail and/or motion contained in a frame.

Performance Results

In this report we will include some of the experimental results from computer simulations of various techniques being applied to actual images. These results, including bit rates, compression ratios, error measurements, and image quality judgments, are simply the data collected from a small number of experiments performed with specific images. Many of the images are from standard, RS 170 television signals. For the images in microgravity experiments, these results may differ greatly. Also, the results of these techniques are expected to change with increased image resolution. The Baseline HHVT system employs a camera with resolution greater than RS 170 images. Even if the results are found to be similar, the images to be produced from the various microgravity experiments are extremely dissimilar in content to both broadcast television and aerial reconnaissance images which are commonly used for simulations. Therefore, the performance data of the image compression techniques we will present may not be indicative of their performance in the context of an HHVT system.

Evaluation Criteria

The reproduced images from the simulations of codecs are usually evaluated either on an objective (i.e. minimum error) basis or for subjective image appearance. Some of the objective error measures are

$$\mathrm{MSE} = \frac{\sum_{x,y}[i_r(x,y) - i_o(x,y)]^2}{N^2}$$

$$\mathrm{NMSE} = \frac{\mathrm{MSE}}{i_{\max}^2}$$

$$\mathrm{SNR} = 10\log_{10}\left(\frac{1}{\mathrm{NMSE}}\right),$$

where i_r is the reconstructed intensity value, i_o is the original intensity value, and $i_{\max}$ is the maximum possible intensity value, and N is the number of picture elements (pixels) in the image.

For microgravity experiment applications, the interest is usually in obtaining measurements of specific physical quantities from the reconstructed data. Therefore, the evaluation criteria may be very different.

Spectral Information

Nassau [1] gives three uses for the word "color". For the purposes of this report, we mean "a class of sensations" produced by the human visual system (HVS). An image can appear to the HVS to have colors that are identical to the real scene being imaged even though the actual spectral composition of the reproduced image may be very different from that of the actual light incident on the sensor(s). Apparently, this is possible because the human eye has only three types of color sensors (cones) each of which responds to red, green, or blue light, with some overlap in the spectral sensitivities. Therefore, the color information available to higher levels of the HVS consists solely of three scalar values. This is not enough information to uniquely determine the spectral distribution of the incident light. A specific combination of sensor outputs produces a specific color sensation regardless of the spectral distribution of the light that caused the sensor responses.

This phenomenon of the HVS is the basis of the standard RGB system of producing color television pictures. Most of the colors that can be perceived by the HVS can be produced by a combination of the proper proportions of red, green, and blue monochromatic light. The reproduced colors will match the originals well if the spectral responses of the camera sensors closely approximate the spectral responses of the cones in the human eye. The match does not have to be exact since the HVS has a perception threshold for the discrimination of color. Two RGB signals that differ by amounts under the threshold will produce the same color. The threshold is not identical for the three components of the signal. It is also dependent on the color.

An experimenter may be more concerned with the spectral distribution of the light source or the surface spectral reflection of the objects being imaged than in the color perceived by a human observer. In general, it is not possible to uniquely determine this information using a finite number of sensors and/or filters with different spectral responses. If the interest is only in a finite number of discrete frequencies, the intensities can be uniquely determined from an equal number of sensors.

Wandell [2] describes a method for extracting spectral information from a multi-spectral video signal consisting of the outputs of a finite number of sensors. This method approximates the spectral curve (intensity vs. frequency or intensity vs. wavelength) by using basis functions. The number of basis functions that can be used is less than, or possibly equal to, the number of sensors. This method will provide a reasonable approximation to the curve only if the variation of intensity with wavelength is slow (low-pass) or if some information about the shape of the curve is known so that appropriate basis functions can be used.

Because the relationships among the color signals received from the sensors, the spectral information they represent, and the perceptions of the HVS are so complex; an analysis of the effects of data compression techniques on spectral information is very difficult. Any analysis would have to consider the responses of each of the sensors and the type of spectral information to be determined in addition to the type and magnitude of the errors introduced by a specific data compression technique.

When the information to be derived from the color signal consists solely of HVS perception, significant compression can be achieved by separating the signal into luminance and chrominance components and sampling the chrominance signals at a lower spatial frequency. This has little effect on the perceived colors. Most likely, this is because the HVS performs some similar operation before interpreting the color information. If other types of spectral information are needed, this type of compression will not be acceptable.

The results reported in the literature on compression of color images usually involve images for television. In typical color television pictures, the illumination is broad-band across the visual spectrum. Therefore, there is some statistical redundancy between the three color components. For some of the planned microgravity experiments, particularly the self-illuminating ones, this may be far from true.

---

1. Nassau, K., *The Physics and Chemistry of Color,* Wiley-Interscience Publication, pp. 3 and 13, 1983.

2. Wandell, B., "The Synthesis and Analysis of Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, pp. 2-13, Jan. 1987.

Overview of Compression Technique Performance

For all of the reasons mentioned above, the results of various compression techniques reported in the literature may have little applicability to HHVT. If a reliable and accurate analysis of the effects of errors on the information contained in the image is to be performed, it will have to be done for specific types of images from experiments once the specifications of the sensors and the content of the images are known. A general evaluation of the effects of a data compression technique on HHVT images would have limited value. The results reported in the literature are presented here only to provide a basis for further analysis.

PERFORMANCE OF COMPRESSION TECHNIQUES

COMPRESSION TECHNIQUES	COMPRESSION bits per pixel	ERRORS %MSE	COMMENTS
REVERSIBLE METHODS			
1. Run-Length Coding	1.5 - 2.0 3.5 up to 16	Not applicable	This compression is possible with low detail images, encoded at 8 bpp. This compression is possible with typical television images. This compression is possible with high detail images.
2. Contour Coding	depends on number of contours	Not applicable	This technique is most effective when used with two-tone, line drawings.
3. Huffman Coding	7.5	Not applicable	This technique is most often used in addition to lossy, entropy reducing methods.
4. Arithmetic Coding	7.5	Not applicable	Technique's performance is similar to Huffman Coding
5. Conditional Replenishment	depends on motion or image content ≥1.0	Not applicable †	Reversible compression ratios depend on amount of motion or background change in image. This compression yields lossy compression with good quality reconstructed images.
6. Bit-plane Encoding	†	†	This method offers additional improvements over previous methods, especially when using gray codes.
PREDICTIVE METHODS			
1. Linear Predictive Coding	depends on image entropy; requires transmission of difference signal	highly dependent on quantization	Prediction is function of image's statistics.
2. Differential Pulse Code Modulation (DPCM)	3.0 - 4.0 2.0 - 3.0	highly susceptible to transmission errors highly susceptible to transmission errors	This compression is possible using non-adaptive quantization. This compression is possible using adaptive quantization.
3. Delta Modulation	≥1.0	highly dependent on quantization; susceptible to transmission errors	Analog input signal simplifies implementation, but must be sampled at rate higher than Nyquist rate. Marginally acceptable quality.
4. Motion Compensation (MC)	1.5		Average compression for good quality pictures.

† Data not available in literature reviewed

This table continues on the following page.

PERFORMANCE OF COMPRESSION TECHNIQUES			
COMPRESSION TECHNIQUES	**COMPRESSION** bits per pixel	**ERRORS** %MSE	**COMMENTS**
BLOCK METHODS			
1. Vector Quantization (VQ)	0.5 - 0.8	0.1	This compression is possible using monochrome images.
	1.5 - 2.0	0.1	This compression is possible using color images.
	0.1 - 0.2	†	This compression is possible with a motion compensation technique. This bpp holds true if motion is ≤ 20% in the image.
2. Vector DPCM	0.5	‡	This compression is achieved using monochrome images.
	0.75 - 1.08	‡‡	This compression is achieved using color images.
			In general, Vector DPCM produces better results than VQ at same bit rate.
3. Block Truncation Coding	1.625	†	This compression is achieved using monochrome images.
	2.13	†	This compression is achieved using color images.
	0.9	†	This compression is achieved using interframe coding.
4. Variable Resolution Coding MAPS	0.593	0.82	
Tree coding	0.2 - 9.0	†	Depending on the amount of detail, this method can be reversible.
HVS COMPENSATION			
1. Synthetic Highs	[a]	†	Amount of compression depends on threshold values and desired image quality.
2. Pyramid Coding	0.7 - 1.6	< 1.0	Quantization errors occur at high frequencies.
3. Region Growing	[b]	†	This technique does not yield good results for small objects or details in original image.
4. Directional Decomposition	[c]	†	This compression was achieved using 8 bit original.
	[d]	†	At this compression, this method produces blurred but still recognizable images.
5. Anisotropic Nonstationary Predictive Coding	[e]	†	This technique does not handle fine texture at high compression rates.

† Data not available in literature reviewed
‡ Square error < 18
‡‡ Square error < 200
Compression ratios depend on number of bits used to encode original and reconstructed images:
[a] 4:1 to 23:1
[b] 30:1
[c] 60:1
[d] 90:1 to 200:1
[e] 20:1 to 30:1

This table continues on the following page.

PERFORMANCE OF COMPRESSION TECHNIQUES			
COMPRESSION TECHNIQUES	**COMPRESSION** bits per pixel	**ERRORS** %MSE	**COMMENTS**
HVS COMPENSATION (CONTINUED)			
6. Minimum Noise Visibility Coding	4.5 5.6	* *	This compression was achieved using monochrome images. This compression was achieved using color images.
7. Constant Area Quantization (CAQ)	1.08 1.2 1.0 - 1.3	3.0 < 2.0 1.5 - 1.0	This compression was achieved by introducing overshoot into the method. This compression was achieved using predictive CAQ.
8. Perceptual Space Coding	0.1 0.25	0.72 0.36 - 3.30	A low detail, monochrome image was encoded resulting in usable quality reproductions. A number of color images of varying detail were encoded resulting in excellent quality reproductions.
TRANSFORM CODING			
1. Karhunen-Loeve (KLT)	0.5 - 1.0	1.5 - 0.5	For widest range of images, this transform produces best results; but it lacks fast implementation.
2. Discrete Cosine (DCT)	0.5 - 1.0	0.75 - 0.2	Using adaptive techniques, DCT performs closely to KLT
3. Slant	1.0 - 1.5	< 1.0 0.5	Using adaptive techniques, this transform performs almost as well as DCT and much better than Hadamard and Haar.
4. Hadamard	1.0 - 1.5	1.5 - 1.0	Classic, straightforward hardware implementation exists for this transform; however with VLSI, it is being replaced by better performing DCT.
5. Haar	0.7 - 1.7	0.8 - 0.2	With adaptive quantization, this transform has better performance than Hadamard; but it is also being replaced by DCT.
HYBRID METHODS			
1. DCT/VQ	0.7 - 0.8 1.1	** **	This level of compression did not result in high quality reconstructed images. At this compression, reconstructed images contained no visible distortion.
2. DCT/MC	0.1 - 0.4	†	This performance uses adaptive DCT.

† Data not available in literature reviewed
* Quantization errors at high frequencies
** 1-2 db higher signal-to-noise ratio than DCT alone

Reversible Image Compression

Image compression schemes are said to be reversible, or information- lossless, if the original digital representation of the image can be fully reconstructed at the receiver from the compressed data. Compression can be achieved without any loss of information only if the digital representation contains redundancies. This is usually the case for digital video images. The measure of the amount of information contained in a set of data indicates the entropy of the information source producing the data.

In order to define the entropy, we must first define the source. A source has an alphabet of symbols that it can produce, as well as a set of probabilities for the production of each symbol in the alphabet. If the probability of occurrence of each symbol is independent of all other symbols, the entropy (bits/symbol) is defined as

$$H(S) = -\sum_{i=1}^{n} [P_i \log_2(P_i)]$$

where P_i is the probability of symbol i occurring. This is known as the zeroth-order entropy.

It is also possible to have a source where the probabilities of the production of a given symbol depend on m previously produced symbols. This is known as an mth-order Markov source. The entropy for this source (sometimes known as a conditional entropy) is

$$H(S) = -\sum [P(s_{i_m}, \ldots, s_{i_1}, s_i) \log_2 P(s_i \mid s_{i_m}, \ldots, s_{i_1})]$$

where the sum is taken over all n members of the symbol alphabet for each of the m symbols obtained from the source, i. e., nm terms. This entropy will be not greater than the zeroth-order entropy, with equality only if the symbols are independent.

For a digital video image, the symbols are the intensity values at the pixels. The alphabet depends on the quantization. For eight- bit PCM quantization there are 256 symbols. The entropy is generally stated in bits per pixel (bpp). In general, the values at nearby pixels are highly correlated. Therefore, a lower entropy is obtained, since the image can be represented as the output of a low-order Markov source.

Some reversible image compression schemes are

1) Run-length Coding
2) Contour Coding
3) Huffman Coding
4) Arithmetic Coding
5) Conditional Replenishment

All of these reversible techniques will, in general, produce output at a variable rate. In order to transmit at a constant rate, a (large) buffer will be required in the transmitter. With today's memory technology, this does not represent a problem.

Run-length Coding

Very often, large regions of an image are relatively uniform in intensity and/or color. This can be used to our advantage by replacing long strings of identical pixels with short strings containing the intensity (once) and the number of repetitions. One problem with the practical implementation of this scheme is how to allow the receiver to distinguish between intensity codes and repetition codes. There are two different ways of solving this problem. One method is to divide all of the pixels into runs (of length 1 or more) and to include a length code between every intensity code. A second method involves reserving one value for an escape code to signal the start of a run. The advantage of the latter method is that the worst case scenario will result in no compression, whereas for the first method the number of bits per pixel could be increased. The cost of the second method is the number of intensity values possible for a given number of bits is reduced by one. If the second method is applied to an image which was quantized using the full scale of values, the compression will not be reversible [3][4].

Compression

Usually, the number of bits used to encode the length of a single run is constant. Therefore, there is a maximum run length, m, that can be encoded, with longer runs being divided into multiple runs. The length of each run depends on the correlation between each pixel and the m previous pixels. The achievable compression is therefore limited by the entropy of the image when it is modelled as the output of an mth-order Markov source. However, there is no guarantee that run-length coding (RLC) will approach this limit. Low detail images may be encoded at about 1.5 - 2.0 bpp for an 8-bit original. Typical television images will require about 3.5 bpp. High detail images could require up to 16 bpp if the first method discussed above is used.

Contour Coding

If the image to be compressed is a two-level (binary) image, the entire image can be reconstructed from knowledge of the contours that define the boundaries between the regions. Therefore, binary images can be encoded at low bit rates by transmitting only the pixels that are part of the contours [5]. More savings can be achieved by dividing the contours into line segments and assigning each segment a code [6]. These techniques can also be applied to multilevel images by including either an intensity value for each area or gradient values along the contours.

Compression

The compression depends on the number of contours in the image. This technique is most effective for line drawings which are mostly background. It is also used when coding coefficients in transform coding.

Huffman Coding

In an entropy encoding scheme the amount of compression achievable is limited by the entropy of the data. Huffman coding [7] is such an entropy encoding scheme. It takes advantage of the non-uniform distribution of the occurrences of pixel intensities, regardless of position in the image. (If the distribution is uniform across all possible values, the bit rate is equal to the entropy already, and no compression will be achieved.) The technique involves assigning a code to each intensity value with the shorter codes going to the more probable events. The compression limit given by the entropy of the image is almost achieved. If the probability distribution is known in advance, the receiver can store the codes. However, in many cases the codes must be generated based on the actual statistics of the image. In this case, the codes must be transmitted as overhead information. For large images, the overhead will not be significant.

Compression

The bit rate is limited by the zeroth-order entropy of the image. Huffman coding will achieve bit rates very close to this limit for large numbers of pixels. The zeroth-order entropy for 8-bit video images is often about 7.5 bpp, so Huffman coding is most often used in combination with a lossy, entropy-reducing compression technique.

Arithmetic Coding

Another entropy encoding scheme is Arithmetic coding [8][9][10]. In this technique the interval from zero to one (including zero, excluding one) is divided according to the probabilities of the occurrences of the intensities. Each subinterval is assigned a code representing a fractional value that it contains. The division process can be repeated many times by insuring that the boundaries are rational values and storing the numerator and denominator as integers. Therefore, long codes can be produced for long strings of pixels. As with Huffman Coding, shorter codes are assigned to more probable strings. The possibility of stringing together pixels when encoding allows the coding rate to approach the entropy of the image even more closely than Huffman codes which must assign an integral number of bits to the code of each pixel. The codebook is not transmitted, but the decoding is done through knowledge of the probabilities of the intensity levels.

Compression

Similar to Huffman coding. (See above)

Conditional Replenishment

When there is very little motion in a sequence of video frames, there will be high correlation between individual pixels of adjacent frames. Much compression can be achieved by only transmitting the locations and intensity values of those pixels that changed since the previous frame. The receiver saves the previous frame and replenishes the changed pixels with their new values. The amount of compression that can be achieved depends on the correlation between frames. If the replenished pixels appear in runs, the overhead information could be reduced by transmitting a starting address and a run length instead of a location for each pixel. Much greater compression can be achieved by using conditional replenishment as a non-reversible technique. This is accomplished by not transmitting any pixel whose value is "close" to that of the previous frame [11][12][13].

Compression

Lossy conditional replenishment (CR) can obtain bit rates as low as 1.0 bpp with good quality. Reversible compression ratios depend on the amount of motion or background change in the specific scene.

Bit-plane Encoding

Some of the lossless compression techniques we have discussed, specifically run-length coding, and contour coding can be used more effectively if they are applied to each bit-plane separately. This can be done by using the binary representation of the pixel intensity values of the image. The most significant bit at each pixel is treated as a binary image. The compression scheme is applied to this image. Each of the remaining bits is treated likewise. At the receiver each bit-plane is decoded, and the bit-planes are combined to reconstruct the image. This scheme will often provide the possibility of more compression since the more significant bit-planes will have large uniform areas which can be greatly compressed [14].

If bit-plane coding is to be used, the amount of achievable compression can usually be increased by replacing the standard binary representation of the intensity values with a gray code. The gray code reorders the binary symbols such that consecutive symbols differ by exactly one bit. Since real images often have consecutive intensity values at adjacent pixels, using the gray code can increase the size of uniform regions.

---

3. Pracht, B.R. and Bowyer, K.W., "Adaptations of Run-Length Encoding for Image Data," *IEEE,* pp. 6-10, 1987.

4. Lansing, D.L., "Experiments in Encoding Multilevel Images as Quadtrees," *NASA Technical Paper,* no. 2722, 1987.

5. Schreiber, W.F., Huang, T.S., et al., "Contour Coding of Images," pp. 443-8, 1972.

6. Chaudhuri, B.B. and Kundu, M.K., "Digital Line Segment Coding: A New Efficient Contour Coding Scheme," *IEEE Proceedings,* vol. 131 E, pp. 143-147, July 1984.

7. Huffman, D.A., "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE,* pp. 1098-1101, September 1952.

8. Jones, C.B., "An Efficient Coding System for Long Source Sequences," *IEEE Transactions on Instrumentation Technology,* vol. IT-27, pp. 280-291, May 1981.

9. Langdon, Jr., G.G., "An Introduction to Arithmetic Coding," *IBM Journal of Research and Development,* vol. 28, pp. 135-149, March 1984.

10. Elnahas, S.E. and Dunham, J.G., "Entropy Coding for Low-Bit-Rate Visual Telecommunications," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1175-1183, August 1987.

11. Mounts, F.W., "Frame-to-Frame Digital Processing of TV Pictures to Remove Redundancy," pp. 653-672, 1972.

12. Haskell, B.G. and Gordon, P.L., "Source Coding of Television Signals Using Interframe Techniques," *Proceedings of the SPIE,* vol. 66, pp. 9-22, 1975.

13. Hsing, T.R. and Tzou, K.-H., "Video Compression Techniques: A Review," *Proceedings of Globecom,* pp. 1521-1526, November 1984.

14. Rabbani, M., "Digital Image Compression - Tutorial Short Course Notes," *SPIE 32nd Annual International Technical Symposium on Optical and Optoelectronic Applied Science and Engineering,* vol. T53, August 1988.

Predictive Methods

Differential Pulse Code Modulation

Predictive methods involve predicting the intensity value at a given pixel based on the values of previously processed pixels. Usually, the difference between the predicted value and the actual value is transmitted. This technique is generally known as Differential Pulse Code Modulation (DPCM). The receiver makes the same prediction as the transmitter and then adds the difference signal to it in order to reconstruct the original value. A predictive method usually involves three steps: prediction, quantization, and coding. These functions are shown in Figure 1.

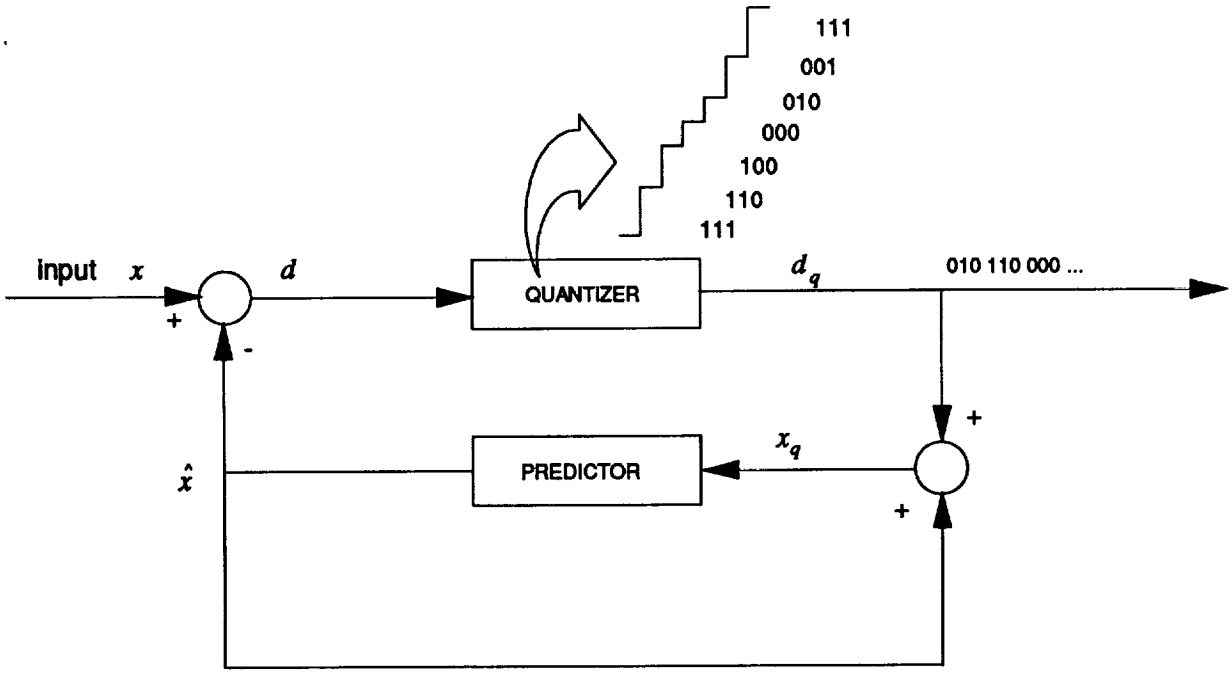


Figure 1: Differential Pulse Code Modulation encoder

The difference, d, is obtained by subtracting $\hat{x}$, the prediction, from the input x. The difference signal, d, is quantized as shown and the signal d_q is transmitted. The quantized input to the predictor, x_q, is obtained by combining $\hat{x}$ with d_q, as shown.

Various reversible coding techniques, discussed in **Reversible Image Compression**, can be applied to the quantized difference signal as presented. Therefore, only prediction and quantization will be covered in this section.

Predictive methods can be applied in one or two dimensions within a frame and/or on a frame-to-frame basis. The methods can either be fixed or adaptive. Adaptivity can be included in the prediction and/or in the quantization.

Prediction:

In the prediction step we try to predict the intensity value at the current pixel as closely as possible. This is done to take advantage of the correlation existing between pixels in a region. If the correlation is high, the predictions will usually be fairly close to the actual values. Therefore, most of the differences will be of low absolute value, and the zeroth-order entropy of the difference signal will be lower than that of the original signal.

The simplest prediction methods use the value at the previous pixel as the predicted value. Somewhat better results can be obtained by using a linear combination of n previous pixels. Better prediction can also be achieved by using two-dimensional prediction. This is usually done by using a linear combination of the previous pixel and adjacent pixels from the previous row. The optimal coefficients of the linear combinations can be derived from the statistics of the image. They can be computed separately for each small region of the image. In this case the coefficients must be sent to the receiver along with the difference signal [15][16].

Another possibility, used in 2-D, is to choose a small number of combinations of coefficients [17][18]. For each pixel the prediction is made using each of these combinations. The best prediction is chosen, and a code is sent to the receiver to indicate which combination was used. Making the predictions in this way is especially effective when the image contains sharp edges with arbitrary orientations.

Linear Predictive Coding:

The technique known as Linear Predictive Coding (LPC) [19][20][21] is commonly used for speech compression. It is essentially an adaptive nth-order predictive method as described above. When LPC is used for speech compression, the difference signal does not need to be transmitted since it can be approximated by either white noise or a periodic pulse train. These can be generated at the receiver and applied to the appropriate prediction algorithm. This tremendous advantage does not exist, in general, for video images. Therefore, just as with other predictive techniques, the difference signal must be transmitted. Even so, LPC is an effective technique since the prediction is adaptively optimized for the local statistics of the image.

Quantization:

To transmit the difference signal, it must first be quantized. If the difference signal is quantized to the same precision as the original signal, the process is reversible as mentioned above. If the prediction is highly accurate, the entropy (information content) of the full-precision difference signal will result mostly from the less significant bits. This means using fewer bits to quantize the signal can significantly reduce the entropy without too much reduction in image quality. If some reduction in image quality is acceptable, much better compression can be obtained by reducing the number of quantization levels, thereby reducing the number of bits required to transmit the difference signal. The optimal distribution of the reduced number of quantization levels depends on the statistics of the image and on the relative significance to the user of various types of errors in the intensity values.

To keep the maximum error below a certain threshold, the levels will be distributed uniformly over the possible values of the difference. To minimize the MSE, the smaller magnitudes will be quantized with more values. The larger magnitudes will be quantized with few values since they occur less frequently. Non-uniform

quantization is also appropriate for minimal subjective degradation of the image since the sensitivity of the human eye to small variations is much greater within smooth regions than near sharp edges. The signal can also be quantized to preserve certain statistical properties of the image [22].

Max [23] provided a method for determining the optimal distribution of quantization levels for a given distortion criterion when the statistics of the signal are known. He also provided solutions for 1 to 36 quantization levels which will minimize the MSE for a normal distribution. The Max quantizer has been widely used in both predictive and transform compression schemes.

Quantization is implemented adaptively for a variety of purposes. These include better handling of sharp intensity gradations without sacrificing good quality in smooth regions [24]. Also, adaptive quantization adjusts to local statistics to reduce the local error, as well as insuring a more uniform output bit rate when entropy encoding is used on the difference signal. The adaptivity can be implemented either by changing the placement of quantization levels or by switching between quantizers using different numbers of levels.

One difficulty encountered with DPCM is that errors in transmission will propagate through the image because predictions at the receiver will not be identical to those at the transmitter. A common solution to this problem is to reduce the magnitude of the predicted value by a few percent before calculating the difference signal. This is known as introducing "leak" into the prediction. It will cause the effect of the error to diminish from pixel to pixel, thereby confining the effects to a small region.

Aside from transmission errors, the information which is lost in using a predictive method results from the quantization error. The quality of the prediction affects the error only indirectly in the sense that a better prediction allows a finer quantization of the differences at the same bit rate, resulting in a better quality reproduction of the image. The number of quantization levels, as well as their placement can be adjusted to match various error criteria.

Compression

Good quality images can be obtained at bit rates of 3 - 4 bpp for nonadaptive DPCM and 2 - 3 bpp for adaptive DPCM [25]. Test pictures that were encoded with 3-bit adaptive DPCM had signal-to-noise ratios of over 40 dB [26]. There have been claims of very good image quality at around 1 bpp [27]. The discrepancy may lie in the subjective determination of "good" images. Color images were reproduced with no perceptible degradation at a bit rate of 5 bits/sample where a composite PAL signal was sampled at 3 times the subcarrier frequency (20% over the Nyquist rate) [28].

Spatial Domain

Most of the compression is achieved by taking advantage of the statistical distribution of the differences (most of them will be small in magnitude) through the use of coarse quantization. Using coarse quantization will result in quantization noise that can cause distortion in the reconstructed image. This distortion can generally be classified into three types: granular noise, slope overload, and edge busyness.

Granular noise refers to small variations in intensity in regions where the intensity should be constant. It results from not having enough quantization levels for very small differences. Slope overload refers to errors resulting from not having a large enough quantization level for the difference signal that occurs at a sharp edge. This type of error will tend to blur the edges since it will take a few pixels for the reconstructed image to "catch up" to the original image. It will also tend to shift the edges "downstream" in the image since the predictions are causal which makes the difference signal lag the original signal. Edge busyness can result from not having enough middle-to-large quantization levels so that a continuous edge may appear discontinuous between consecutive pixel rows.

The nature of the degradation of the image for a given bit rate can be controlled by the distribution of the quantization levels. Placing more levels at smaller difference values (reduce granularity) will improve the picture in regions of low detail at the cost of large errors at sharp edges. Many adaptive quantization schemes are designed to reduce the specific type of distortion that would be most critical in the local region. Adaptive 2-D prediction schemes have also been developed for the purpose of predicting along contours to reduce situations that would lead to slope overload and edge busyness.

If the difference signal is subsequently entropy encoded, the amount of detail in an image can affect the number of quantization levels available at a given bit rate. If there is a great amount of detail in the image, a significant number of the differences will be large. This will raise the entropy of the difference signal, and, therefore, less compression will be possible for a given number of quantization levels.

Temporal Domain

Temporal effects will only be of interest for interframe DPCM. (See Motion Compensation)

Aesthetic Appearance

The appearance of the reconstructed image depends on the amount and distribution of the degradations mentioned above. The visibility of each type of degradation depends on its location in the image. Granular noise, for instance, is more visible in constant intensity regions of large area than it is near edges.

One method that is used to reduce the visibility of quantization noise is to add pseudo-random noise (PRN) before quantizing the difference signal. The same noise is subtracted at the receiver. The effect of doing this is to break up any noticeable patterns in the quantization errors.

Spectral Information

The distortions produced by encoding color images were generally similar to those produced by using monochrome images.

Delta Modulation

One technique, known as Delta Modulation (DM), involves quantizing the difference signal with one bit, i. e., two levels, one positive and one negative. The locations of the levels are usually determined adaptively in order to reasonably represent both smooth regions and edges [29][30][31]. Figure 2 shows a block diagram of a Delta Modulation encoder.

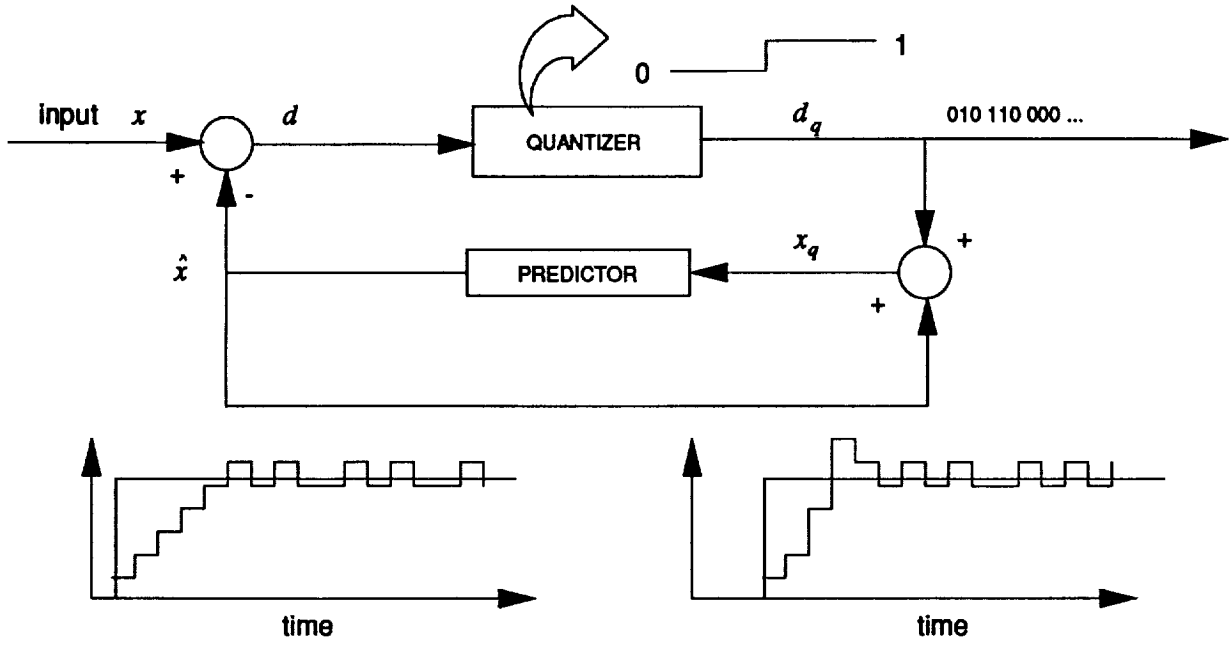


Figure 2: Delta Modulation encoder

The DM encoder is similar to the DPCM encoder, where the quantizer uses two levels. The effects of adaptively changing the quantizing levels are illustrated in the two time plots in Figure 2, showing the response to a step change. In both plots the effects of slope overload can be seen. However, in the right-hand plot, using adaptive quantizing, the response arrives at the steady state level more rapidly.

An advantage of DM over other forms of DPCM is that it does not require a digitizer. This makes it much simpler to implement if the input signal that the encoder receives is in analog form.

Compression

Nonadaptive DM produces 1 bit per sample. However, in order to maintain reasonable quality the analog signal must be sampled at a rate higher than the Nyquist rate. When the input signal is digital, adaptive DM can be used to improve image quality at a bit rate of 2 bpp. The image quality at a bit rate of 2 bpp is marginally acceptable.

Spatial Domain

The coarse quantization used in DM results in quantization noise that can cause distortion in the reconstructed image. The distortion produces three types of noise similar to DPCM: granular noise, slope overload, and edge busyness. Adaptive DM will reduce the magnitude of these effects.

Temporal Domain

Temporal effects will only be of interest for interframe DPCM. (See Motion Compensation)

Aesthetic Appearance

DM is susceptible to transmission errors. The errors result in streaks, where the duration of the streaks is an exponential function of the leak introduced in the transmitter. Edge delay, edge wiggle and edge busyness are distortions characteristic of DM. The magnitude of these distortions increase with increasing compression.

Spectral Information

The distortions produced by encoding color images were generally similar to those produced by using monochrome images.

Motion Compensation

Motion compensation (MC)is a popular technique for improving interframe prediction. It can be used in combination with one of many intraframe compression methods.

There are two general categories of MC techniques: pixel recursive algorithms and block matching algorithms [32]. In both of these techniques, the predictor uses the intensity value of a pixel in the previous frame which may be displaced spatially from the current pixel in order to compensate for the motion of the physical objects being depicted in the image. The goal of the algorithm is to find the spatial displacement that gives the best prediction.

Pixel recursive algorithms update, at each pixel, the spatial displacement that is used in the prediction. The update is based on the difference value at the previous pixel of the current frame [33][34].

Block matching algorithms compare small blocks of pixels in the new frame to displaced blocks in the previous frame, as shown in Figure 3.

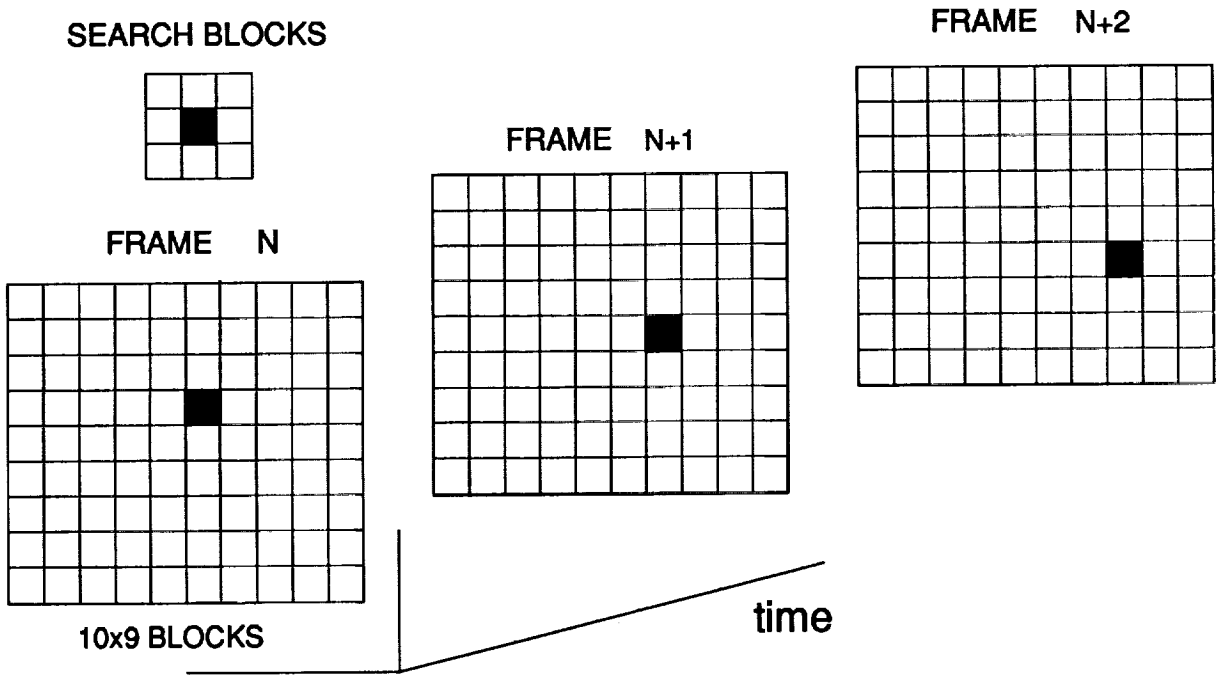


Figure 3: Motion Compensation block search

The spatial displacement which gives the best match is used in the pixel-by-pixel prediction. Since most frame-to-frame displacements due to motion are of small magnitude, regions of the image containing motion can usually be predicted accurately by a MC technique with a reasonable number of calculations. If none of the comparisons result in a sufficiently good match, intraframe coding is used on the current block.

Compression

Motion compensated prediction can produce good quality pictures at an average rate of 1.5 bpp.

Spatial Domain

Granular noise in the region near edges of moving patterns is, in general, attributable to the use of MC.

Temporal Domain

Holding the output bit rate of an interframe predictive method, such as MC, at a given level could produce very low quality images when the amount of motion is high. In general, however, these techniques are designed to produce an output bit rate that is variable and depends on the amount of motion in the image. The quality of the image is maintained at or above a given level. The entropy of the difference signal for interframe methods will clearly be lower for sequences with less activity since the frame-to-frame predictions will be more accurate.

Aesthetic Appearance

Motion compensated predictive methods attempt to predict image intensities accurately when there is some motion in a region. They will only be able to do so for slowly moving objects. In effect, they consider low-motion regions to be stationary, thereby increasing the fraction of the image that is stationary and reducing the bit rate. Just as interframe DPCM, moving edges result in large prediction differences, in turn resulting in slope overload. The aesthetic appearance however, is distributed from frame to frame. A scene change will be treated as high-speed motion (unpredictable) over most of the image.

Spectral Information

The distortions produced by encoding color images were generally similar to those produced by using monochrome images. Chrominance noise appears in the region of moving edges.

---

15. Woods, J.W. and Paul, I., "Adaptive Prediction DPCM Coding of Images," pp. 31.8.1-5, 1980.

16. Desoky, A. and McDonald, J.W., "Compression of Image Data Using Adaptive-Predictive Coding," *Proceedings of the IEEE,* pp. 11-15, 1987.

17. Zschunke, W., "DPCM Picture Coding with Adaptive Prediction," *IEEE Transactions on Communications,* vol. COM-25, pp. 1295-1302, November 1977.

18. Yasuda, H. and Kawanishi, H., "Predictor Adaptive Differential Pulse Code Modulation," *Proceedings of the SPIE,* vol. 149, pp. 189-195, 1978.

19. Poehler, P.L. and Choi, J., "Linear Predictive Coding of Imagery for Data Compression Applications," *Proceedings of ICASSP 83,* pp. 1240-1243, 1983.

20. Nam, M.K. and O'Neill, W.D., "Adaptive LPC of Time-Varying Images Using Multidimensional Recursive..," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1115-1126, August 1987.

21. Kwan, H.K., "Image Data Compression Using 2-D Lattice Predictor," *Electronics Letters,* vol. 20, pp. 994-995, 22 November 1984.

22. Delp, E.J. and Mitchell, O.R., "Use of Moment Preserving Quantizers in DPCM Image Coding," *Proceedings of the SPIE,* vol. 249, pp. 123-130, 1980.

23. Max, J., "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory,* vol. IT-6, pp. 7-12, March 1960.

24. Brown, E.F., "Sliding-Scale Operation of Differential Type PCM Coders for Television," pp. 303-322, 1972.

25. Rabbani, M., "Digital Image Compression - Tutorial Short Course Notes," *SPIE 32nd Annual International Symposium on Optical and Optoelectronic Applied Science and Engineering,* vol. T53, 1988.

26. Zschunke, W., "DPCM Picture Coding with Adaptive Prediction," *IEEE Transactions on Communications,* vol. COM-25, pp. 1295-1302, November 1977.

27. Paul, I. and Woods, J.W., "Some Experimental Results in Adaptive Prediction DPCM Coding of Images," *Proceedings of ICASSP,* pp. 1220-1223, 1983.

28. Thompson, J.E., "Data Compression of Composite Color TV Signals Using Planer Prediction," *International Conference on Digital Satellite Communication,* pp. 315-321, 1975.

29. Candy, J.C., "Refinement of a Delta Modulator," pp. 323-339, 1972.

30. Barba, J., Scheinberg, N., Schilling, D.L., "Entropy Encoding Applied to a Video DM Bit Stream," *International Conference on Communications,* pp. 22.7.1-5, 1981.

31. Scheinberg, N., et al., "A Composite NTSC Color Video Bandwidth Compressor," *IEEE Transactions on Communications,* vol. COM-32, pp. 1331-1335, December 1984.

32. Kappagantula, S. and Rao, K.R., "Motion Compensated Predictive Coding," pp. 64-69, 1983.

33. Robbins, J.D. and Netravali, A.N., "Interframe Television Coding Using Movement Compensation," pp. 23.4.1-5, 1979.

34. Walker, D.R. and Rao, K.R., "Motion-Compensated Coder," *IEEE Transactions on Communications*, vol. COM-35, pp. 1171-1177, November 1987.

Block Methods

Vector Quantization

Vector Quantization (VQ) [35][36][37], which is also called block quantization or block source coding, uses a block of intensity values, treated as a vector. The block size is usually 4 x 4 since it is difficult to design a good representative codebook for larger blocks [38]. The vector is compared to a codebook of vectors and the code for the "closest" (minimizing some measure of distortion) match is transmitted. This process is illustrated in the figure below using a block of 3 x 3 pixels.

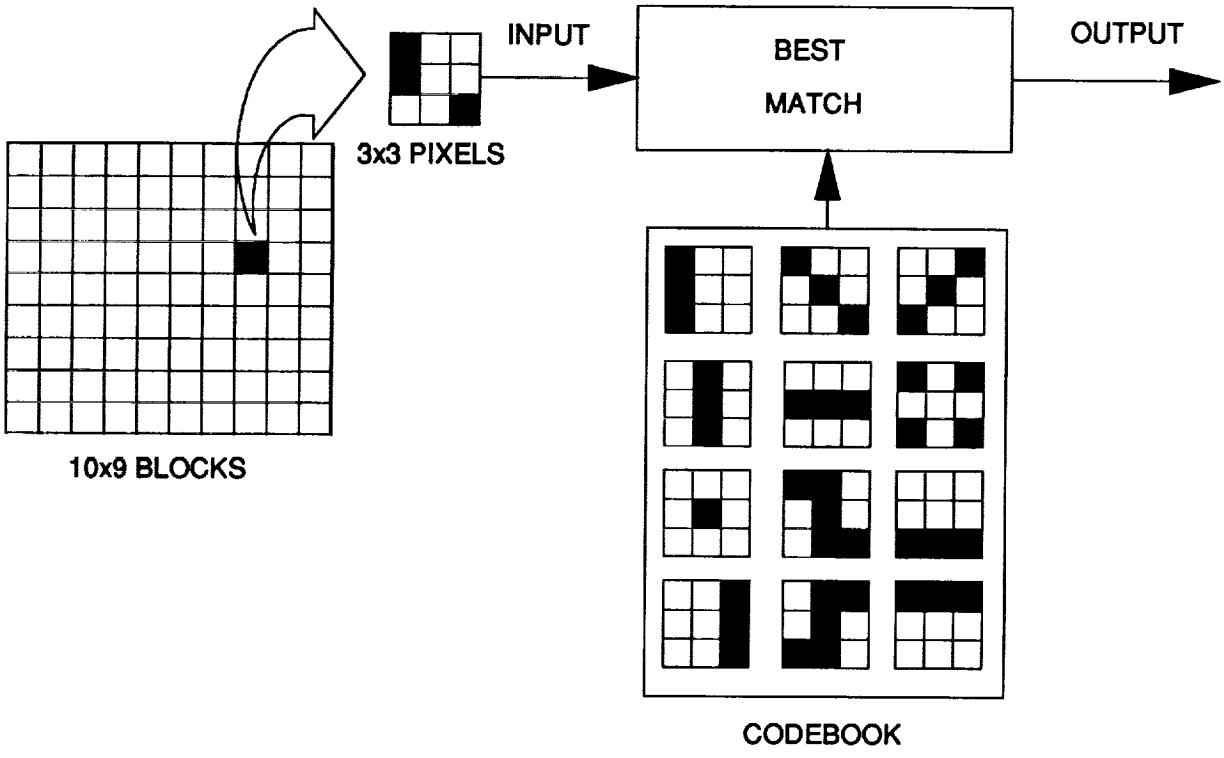


Figure 4: A Vector Quantization encoder using a block of 3 x 3 pixels

It is the choice of the size and contents of the codebook that requires some ingenuity. The distortion measure, used for both the design of the codebook and for the comparisons during encoding of the image, should quantify the most critical type of distortion for the end user. The vector quantizer will then minimize this distortion measure [39].

The codebook consists of a small subset of all the possible vectors of intensity values. It is produced from the probability distribution of the image. Still more likely if the statistics of the image are not known *a priori*, the codebook is made from a string of training vectors which are assumed to be representative of the data to be transmitted. In the latter case, training vectors will produce less distortion if they are taken from the image itself. However, this requires an adaptive implementation involving the production and transmission of a new codebook for each image. A goal of the design

of vector quantizers is to design codebooks that are universal without being too large. A large codebook, while reducing distortion, also reduces the effectiveness of VQ in two ways. First, the bit rate is higher since the number of bits required for each codeword is $\log_2(N)$ where N is the number of codevectors. Second, the time required to search the codebook increases rapidly with its size.

In order to reduce the size of the codebook for a given distortion level, the mean and standard deviation are removed from each block and transmitted separately [40]. Another method that reduces the bit rate without increasing the distortion is called an Address-Vector Quantizer (A-VQ) [41]. For this method the codebook is divided into two parts. Most of the codevectors contain the codes (addresses) of four standard codevectors. To encode an image, each 4 x 4 block is vector quantized using the standard codevectors. Then the blocks are combined into groups of four. If the four codes match those contained in the address-codebook, this one code is substituted for the four standard codes. Otherwise, the four codes are transmitted as usual. In this way A-VQ can reduce the bit rate nearly 50% with no reduction in image quality.

Other methods have been developed to improve the subjective image quality for a given codebook size by segmenting the codebook and assigning more codevectors to "edge" vectors since the accurate reproduction of edges is necessary for good quality images.

VQ can be used for multi-spectral images by treating the multiple intensity values at each pixel as a vector. The spatial and spectral aspects of a color image can also be combined into a single vector [42].

Compression

Straightforward implementations of VQ (segmented codebook or mean subtraction) can produce good quality images at rates of 0.5 - 0.8 bpp for monochrome images and 1.5 - 2.0 bpp for color images. A motion compensated technique has been developed [43] that produces color images at 0.1 - 0.2 bpp if the motion covers less than 20% of the image.

Spatial Domain

The objective errors resulting from VQ depend on the distortion measure that is used. If the acceptable and unacceptable types of distortion can be quantified, a codebook can, in theory, be produced to meet these criteria. In practice, algorithms for producing vector codebooks have only been developed for a few distortion measures. The MSE is about 0.1% for standard VQ.

Temporal Domain

Not Applicable.

Aesthetic Appearance

MSE does not accurately represent the perceived image quality. This can be seen at sharp edges where a "staircase effect" is produced by standard VQ [44]. The degradation of these edges does not significantly increase the MSE, but it does decrease the perceived quality. The number of possible edge orientations are too large to be include all the respective "edge" vectors in a code set. Therefore edges are degraded.

Another less significant effect which can occur is some blocking in smooth areas [45]. The techniques mentioned above which transmit the mean separately should eliminate most of this problem.

Spectral Information

The reports of using VQ for color images did not include an analysis of the effects on spectral information.

Vector DPCM

Vector DPCM [46] performs prediction on a block by block basis. The intensities at all the pixels in a block are predicted based on the intensities at the pixels adjacent to the left and upper borders of the block (the nearest pixels that have already been transmitted). The differences at all the pixels in the block are coded by vector quantization.

Compression

Very good reproduced images were obtained at 0.5 bpp for monochrome images and 0.75 - 1.08 bpp for color images with sub-sampling of the chrominance signals. The square error per pixel for these images were less than 18 for monochrome and less than 200 for color.

The distortions produced by this method are similar to those produced by VQ. In general, Vector DPCM produces better results than VQ at the same bit rate.

Block Truncation Coding

Block Truncation Coding (BTC) [47][48] divides the image to be coded into blocks. The block size is usually 3 x 3 or 4 x 4 pixels. Within each block a threshold is chosen, and the value at each pixel is coded as a 0 or 1 depending on whether it is above or below the threshold. To decode the image, a high value is assigned to each pixel that has a 1, and a low value is assigned to the others. The most common techniques attempt to preserve the mean and variance of each block. They choose the mean as the threshold value. The low, a, and high, b, values at the decoder are computed as follows:

$$a = \eta - \sigma\sqrt{\frac{q}{m-q}}$$

$$b = \eta + \sigma\sqrt{\frac{m-q}{q}}$$

where η is the mean and σ is the standard deviation. m is the total number of pixels in the block and q is the number of pixels whose value is greater than the threshold.

Interframe BTC can be implemented on 4 x 4 x 3 blocks of pixels if motion compensation is included. Whenever the motion is too great, the algorithm switches to 2-D BTC [49].

BTC lends itself very nicely to parallel processing since the coding of each of the blocks is totally independent. Some parallel algorithms have been developed [50].

Compression

Using 4 x 4 blocks will produce a bit rate of 1.625 bpp for a monochrome image, assuming that the mean and standard deviation together are transmitted with 10 bits. NTSC color images can be transmitted at 2.13 bpp by sub-sampling the I-component at 2:1 and the Q-component at 4:1 and using BTC for each component independently [51]. Interframe coding of 4 x 4 x 3 blocks can bring the bit rate down to 0.9 bpp.

Spatial Domain

Some of the reported distortion effects are edge raggedness and misrepresentation of mid-range values [52]. BTC also tends to produce artifacts of sharp, well-defined edges [53].

Temporal Domain

The motion appeared to be continuous and clear for the 3-D BTC.

Aesthetic Appearance

In addition to the distortions reported above, BTC tends to enhance edges at the expense of secondary changes which are less perceptible. This is actually a subjective improvement in the image [54]. In general the images appeared sharp at the transmission rates reported above.

Spectral Information

When coding a color image, there are small areas that contain errors in the color. This is due to different amounts of distortion in the three components. However, the overall color quality is good.

Variable-Resolution Coding

Variable-resolution coding includes methods which use local contrast to adaptively vary the resolution of subareas. They do this by representing blocks of pixels in low detail regions with one intensity value, usually the mean of the intensity values in that block. We will discuss two variable-resolution coding techniques, Micro-adaptive Picture Sequencing (MAPS) and Tree Coding. The two techniques can be implemented to retain the same intensity information. They differ only in the details of the compression algorithm and the overhead information. They both can be implemented for reversible compression by requiring regions of identical intensity values, rather than just low contrast.

Micro-adaptive Picture Sequencing

The MAPS [55][56] algorithm starts with the lowest level block, which is usually individual pixels, and attempts to replace four blocks at a given level with a block at the next level. If the contrasts among the four lower level blocks in the appropriate locations are below the contrast thresholds, the lower level blocks are combined into one larger block. MAPS includes an ordering of blocks where every lower level block of a level *n* block is completely covered before the next block of the same level is begun. Practically, this means the position of the blocks of different sizes can be determined implicitly from the sizes of the blocks. The overhead information, therefore, includes only one number for each block to indicate its level. Normally, this number will be 2 or 3 bits since the case of blocks larger than 128 x 128 pixels is unlikely to occur. Using only two bits will result in lower overhead, but will limit the block size to 8 x 8.

Tree Coding

Tree coding [57][58] starts with the entire image. The image is divided into square subregions. Any subregion having low contrast is encoded with a single intensity value. A subregion having high contrast is divided further. This process is repeated down to the level of individual pixels. By processing the image in this fashion, the intensity information is contained in a tree data structure. Uniform subregions are leaf nodes of the tree, while sub-divided regions are interior nodes. In addition to the intensity value of each region, information is transmitted to tell the receiver the shape of the tree. For intraframe compression the tree is usually a quad-tree. This means that each region is divided into four quadrants, and each interior node has four nodes at the level beneath it. In this case, the overhead for tree structure information is found to be about 14% on the average. For interframe compression oct-trees, eight quad-trees, are sometimes used.

A disadvantage of these two methods of variable-resolution coding is that the possible positions of large blocks are fixed so that large uniform regions can be "missed" if they are not located and/or oriented properly with respect to the subdivisions of the image.

Compression

Tree coding generally achieves slightly more compression than MAPS since the overhead is lower (14% compared to 25-38%). The IEEE Facsimile Test Chart was digitized at 2048 x 2048 pixels and compressed to 0.593 bpp using MAPS. The MSE was 0.82%. At compressions of 0.5 - 2.0 bpp, MAPS produced MSE 25% to 50% lower than fixed block coding in which blocks of pixels are replaced by the mean without regard to contrast. If fine detail can be eliminated, some images can be encoded using tree coding at less than 0.2 bpp. An image with low detail can be reversibly encoded using tree coding at about 2 bpp, whereas a high detail image might need more than 9 bpp (higher than PCM).

Spatial Domain

Information is lost in the fine detail of low contrast regions. The contrast thresholds are adjusted globally and/or locally to ensure that any important detail is retained.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The appropriateness of variable-resolution coding is based on the idea that "when an image is viewed as a whole, fine detail is noticed only when it exhibits sharp contrast" [59]. Therefore, the loss of detail should not be noticeable if the thresholds are chosen properly. The only visible artifact is blockiness when large block sizes are used.

Spectral Information

Not Applicable.

---

35. Linde, Y., Buzo, A., and Gray, R.M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications,* vol. COM-28, pp. 84-94, January 1980.

36. Gersho, A. and Ramamurthi, B., "Image Coding Using Vector Quantization," *Proceedings of ICASSP,* pp. 428-431, 1982.

37. Goldberg, M., Boucher, P.R. and Shlien, S., "Image Compression Using Adaptive Vector Quantization," *IEEE Transactions on Communications,* vol. COM-34, pp. 180-187, February 1986.

38. Feng, Y. and Nasrabadi, N.M., "Address-VQ: An Adaptive VQ Scheme Using Interblock Correlation," *SPIE Proceedings on Visual Communications and Image Processing '88,* vol. 1001, part 1, pp. 214-222, 1988.

39. Ramamurthi, B., Gersho, A., and Sekey, A., "Low-Rate Image Coding Using Vector Quantization," *Conference Record of Globecom '83,* pp. 184-187, 1983.

40. Murakami, T., Asai, K., and Itoh, A., "Vector Quantization of Color Images," *Proceedings of ICASSP 86,* pp. 133-136, 1986.

41. Feng, Y. and Nasrabadi, N.M., "Address-VQ: An Adaptive VQ Scheme Using Interblock Correlation," *SPIE Proceedings on Visual Communications and Image Processing '88,* vol. 1001, part 1, pp. 214-222, 1988.

42. Goldberg, M., Boucher, P.R. and Shlien, S., "Image Compression Using Adaptive Vector Quantization," *IEEE Transactions on Communications,* vol. COM-34, pp. 180-187, February 1986.

43. Murakami, T., Asai, K., and Itoh, A., "Vector Quantization of Color Images," *Proceedings of ICASSP 86,* pp. 133-136, 1986.

44. Gersho, A. and Ramamurthi, B., "Image Coding Using Vector Quantization," *Proceedings of ICASSP,* pp. 428-431, 1982.

45. Lee, H.J. and Lee, D.T.L., "A Gain-Shape Vector Quantizer for Image Coding," *Proceedings of ICASSP 86,* pp. 141-144, 1986.

46. Rutledge, C.W., "Vector DPCM: Vector Predictive Coding of Color Images," pp. 1158-1164, 1986.

47. Delp, E. J. and Mitchell, O. R., "Image Compression By Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-27, pp. 1335-1342, September 1979.

48. Halverson, D. R., Griswold, and Wise, "A Generalized Block Truncation Coding Algorithm for Image Compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-32, pp. 664-668, June 1984.

49. Healy, D.J. and Mitchell, O.R., "Digital Video Bandwidth Compression Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-29, pp. 1809-1817, December 1981.

50. Siegel, L.J., et al., "Block Truncation Coding on PASM," *Proceedings of the Ann Allerton Conference on CCC,* pp. 891-900, 1982.

51. Lema, M.D. and Mitchell, O.R., 'Absolute Moment BTC and Its Application to Color Images," *IEEE Transactions on Communications,* vol. COM-32, pp. 1148-1157, October 1984.

52. Lema, M.D. and Mitchell, O.R., "Absolute Moment BTC and Its Application to Color Images," *IEEE Transactions on Communications,* vol. COM-32, pp. 1148-1157, October 1984.

53. Healy, D.J. and Mitchell, O.R., "Digital Video Bandwidth Compression Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-29, pp. 1809-1817, December 1981.

54. Healy, D.J. and Mitchell, O.R., "Digital Video Bandwidth Compression Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-29, pp. 1809-1817, December 1981.

55. LaBonte, A.E., "Two-Dimensional Image Coding by Micro-Adaptive Picture Sequencing," *Proceedings of the SPIE,* vol. 119, pp. 99-106, 1977.

56. LaBonte, A.E., "Micro-Adaptive Picture Sequencing in a Display Environment," *Proceedings of the SPIE,* vol. 249, pp. 61-70, 1980.

57. Lansing, D.L., "Experiments in Encoding Multilevel Images as Quadtrees," *NASA Technical Paper,* no. 2722, 1987.

58. Farrelle, P.M. and Jain, A.K., "Quad-tree Based Two Source Image Coding," 1986.

59. LaBonte, A.E., "Two-Dimensional Image Coding by Micro-Adaptive Picture Sequencing," *Proceedings of the SPIE,* vol. 119, pp. 99-106, 1977.

Human Visual System Compensation

Human Visual System (HVS) compensation techniques attempt to compress video images by eliminating any data not perceptible to the human visual system, even if they are important from an information theory point of view. Some techniques apply a model of the HVS directly to the image data. Many different models have been developed to represent as many features of the HVS as possible. They generally include linear filters, as well as a non-linear element. Once a model has been chosen, a compression technique can be developed that, when applied to the output of the model, "loses" information that would be lost by the HVS anyway, e. g., high-frequency signals that would be filtered out. At the receiver, the image is restored by applying the inverse of the HVS model.

Other techniques involve taking advantage of specific characteristics of the HVS without attempting to explicitly model it. When measured subjectively, the accurate reproduction of edges is very important for the reproduction of images of high quality. Therefore, one technique concentrates the transmitted data on the reproduction of edges at the expense of other features. Such techniques should attain higher compression while yielding better quality reconstructed images.

Many of the HVS compensation techniques involve extracting edge information from the video signal. Many methods have been developed for this purpose. Some involve separating the contour (high spatial frequency) portion of the video signal from the texture (low spatial frequency) portion by linear filtering in the frequency domain. Other techniques attempt to categorize each pixel as "edge" or "not edge". This categorization are done either by applying local derivative operators and thresholding the results or by matching small regions of the image to templates of standard edge configurations. These techniques are generally difficult to investigate analytically, and they are compared by the results they reproduce on test images. The application of some of these methods to data compression are addressed in the discussion of the compression techniques in this section.

A variety of HVS techniques are described here. It would seem many of these techniques consist independent steps which can be mixed and matched carefully. Specifically, the filtering and/or edge detection algorithms are somewhat interchangeable between different techniques. This leads to the possibility of a compression technique tailored to a specific application.

Synthetic Highs

The method of synthetic highs [60][61] involves the separation of the video signal into two components, a low spatial-frequency component and an edge component. The low-frequency component is extracted by a low-pass filter. This component can be sampled at a lower rate than the original signal without loss of information. The edge information is extracted by applying a differential operator to the original signal. The result is compared to a threshold, and only the location and value of the "important" edge pixels are transmitted. This threshold comparison is the only information-lossy process in this technique. At the receiver the low-frequency component is interpolated to the full resolution. A reconstruction filter is used to synthesize a high-frequency signal from the edge information. The two components are then combined to reconstruct the video image.

Compression

The value of the threshold used for the edge detection provides a trade-off between compression ratio and image quality. High quality images were obtained with compression ratios ranging from 4 to 23 depending on the image.

Spatial Domain

The only information lost is the portion of the high-frequency signal below the threshold. This can result in a loss of texture in the reconstructed image.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The images appear sharp. However, some texture may be missing, as stated above.

Spectral Information

Not Applicable.

Pyramid Coding

Pyramid coding [62] involves separating the video signal into multiple frequency bands. Compression is achieved because the high-frequency signals have low entropy and are encoded using fewer bits per pixel, and the low-frequency components are transmitted using fewer pixels.

The HVS is less sensitive to contrast errors at high spatial frequencies. Therefore, the highest-frequency components which contain the largest number of pixels can be quantized with the least levels, thereby significantly reducing the entropy and the bit rate. The lower frequency components require more bits per pixel, but the number of pixels is so much smaller that the effect on the overall bit rate is insignificant.

The set of frequency band components is known as a pyramid because each band contains fewer samples than the previous higher-frequency band. Each level of the pyramid is extracted by low-pass filtering the image and subtracting the low-pass component. The remainder is the current level of the pyramid, while the low-pass component is used as the starting point for the next iteration in which the filter has a lower cutoff frequency.

To produce the pyramid at high speed, the filtering results from convolution of a weighting function with the image. (This is actually a non-causal form of prediction.) The sample rate is reduced from level to level, typically by a factor of two in each dimension.

Each level is encoded independently. Significant compression usually is achieved at the cost of loss of information by quantizing the components more coarsely than the original signal. The number of quantization levels for each component are chosen so as to have little effect on the subjective appearance of the image.

At the receiver the low-pass components are interpolated back to full resolution and all of the components are added together to reconstruct the image. This method lends itself nicely to progressive transmission schemes since most of the transmitted bits are for the high-frequency components, and a blurred version of the image can be transmitted at a much lower bit rate.

Compression

Data rates from 0.7 to 1.6 bpp were reported with normalized MSE of less than 1%.

Spatial Domain

The only errors result from the quantization noise added to each frequency component. The amount of noise in each frequency band is controlled by the number of quantization levels used.

Temporal Domain

Not Applicable.

Aesthetic Appearance

Most of the quantization noise occurs in the high spatial frequency ranges where the human visual system is less sensitive. Therefore, the subjective quality of the reconstructed images is very high.

Spectral Information

Not Applicable.

Region Growing

Region growing techniques [63][64] also separate the contour information from the texture information. The pixels of the image are first divided into regions. The boundaries of the image are defined as contours. The contours and the texture information are transmitted separately and combined at the receiver.

A region is defined by a property common to all the pixels within it. The ideal property insures the boundaries of the regions correspond to physical boundaries of the objects being imaged. In practice, however, this is not achievable. Instead, the property is usually a range within which all the intensity values must lie.

The regions are grown by starting with one pixel and adding all surrounding pixels having the correct property. When no more pixels can be added to a region, a new region is started. This procedure continues until every pixel belongs to a region. (Some of the regions may contain only one pixel.) The pixels on the boundary of a region are treated as contour pixels. The image is processed to reduce the width of the contours from two pixels to one by putting some of the contour pixels back into the interior of the regions. To reduce the complexity, small regions are merged into larger ones and regions which are similar along their common boundary can be combined. These operations will most likely not eliminate contours having physical significance.

The contours are coded using a contour coding scheme (see Contour Coding under **Reversible Image Compression**). The remaining texture information is coded by using an approximating function for each region. Since sharp discontinuities will not appear within a region, low-order polynomials are usually reasonable approximations.

Compression

Good reproductions of an image can be obtained at a compression ratio of about 30:1. Higher compression can be achieved if the image is composed of a small number of large regions of near-uniform intensity.

Spatial Domain

It is assumed that very small regions do not represent physical objects. If they do, these small objects and/or fine details may be lost.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The reconstructed picture has sharp edges and the proper texture. Degradation will be obvious if the original image contained much fine detail which is subjectively significant, e. g., facial features on people at a distance.

Spectral Information

Not Applicable.

Directional Decomposition

Directional Decomposition [65] is based on evidence the HVS contains direction sensitive cells which extract features at specific orientations. Imitation of this directional decomposition should not introduce large subjective errors. The compression technique involves decomposing the image into a low-pass component and N (typically 16) components which are high-pass in a specific direction and low-pass in all the other directions. The decomposition is done in the 2-D Fourier domain by multiplying the transform of the image with directional filter transfer functions. Each component is then inverse transformed to the spatial domain.

As in the other techniques in this section, the low-pass component is under-sampled and transmitted. The directional high-pass components are used for edge detection. The edge detection is performed most effectively by producing an isotropic high-pass image consisting of a combination of the directional components. The pixels are compared to a threshold and selected to retain only the "important" edges. Every edge pixel found in the isotropic image is then classified into one of the directions based on a comparison of the form of the signal in the vicinity of that pixel in each directional component.

The edge information of each directional component is encoded independently. The location of the edge points can be encoded by sub-sampling along the edge since each component is low-pass along the direction of the edges. This additional compression is the advantage of directional decomposition. Some information about the profile of the edge must also be coded and transmitted. This can be done by modelling the profile with a simple analytic function and transmitting some parameters of the function.

Since the directional filters used for decomposition are not perfectly sharp, some edges will appear in more than one directional component. This redundancy can be reduced by using a form of prediction between components.

At the receiver the low-pass component is interpolated to full resolution. Each of the high-pass directional components is synthesized from the edge location and profile information. All of the components are combined to form the reconstructed image. The relative weight of the high-frequency portion of the image can be adjusted to control the sharpness of the edges.

Compression

Reasonable quality images can be obtained at compression ratios up to 60:1 for 8-bit originals. Somewhat blurred, but still easily recognizable, images can be obtained at compression ratios in the range of 90:1 to 120:1.

Spatial Domain

Since both the high and low spatial frequency components are compressed, degradations can result from either or both of these components. Loss of detail is the major degradation. The amount of lost detail increases with the compression ratio since the thresholds are increased in order to obtain more compression.

Temporal Domain

Not Applicable.

Aesthetic Appearance

By assigning a large weight to the high-frequency component, the image can be made sharp even at high compression. However, the loss of detail will still be noticeable.

Spectral Information

Not Applicable.

Anisotropic Nonstationary Predictive Coding

Anisotropic Nonstationary Predictive Coding [66] is basically a predictive technique. The prediction uses a combination of a low-pass filter, a high-pass filter, and an anisotropic filter to enhance edges. These filters are weighted with non-stationary weighting functions and linearly combined to form the actual prediction filter. In one scheme, the difference signal is coded using a Discrete Cosine Transform on each row. In addition to the difference signal, the weighting functions must also be transmitted. These functions are low-pass so they are under-sampled at 1:6 and quantized fairly coarsely. The high degree of adaptivity of the predictor to local anisotropies allows the difference signal to also be encoded with few bits.

Compression

Good quality images can be obtained with compression ratios as high as 30:1 from 8-bit originals, although 20:1 is more typical.

Spatial Domain

This technique does not handle well fine texture at low bit rates. However, if the texture is not important, it can be filtered out before encoding. Otherwise, the major degradation is wide band quantization noise.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The reproduced edges are sharp and coarse texture is reproduced well. The visible errors are in highly detailed regions. Without close inspection, these errors are not highly visible.

Spectral Information

Not Applicable.

Minimum Noise Visibility Coding

One effective way to take advantage of HVS characteristics is to "hide" the quantization noise by moving it to areas where it is less visible. As a result, lower bit rates can be used to achieve comparable quality images. Minimum Noise Visibility Coding (MNVC) uses two effects to hide the noise. One is that noise is more visible in darker areas than in lighter areas. However, there exists a scale, called the "lightness" scale, on which equal increments are equally visible. Therefore, the luminance values are transformed to the "lightness" scale prior to quantization in order to distribute the noise evenly over the intensity range in terms of visibility. The other effect is less visibility of noise in areas of high detail, e. g., near high-contrast edges. Figure 5 illustrates the MNVC concept.

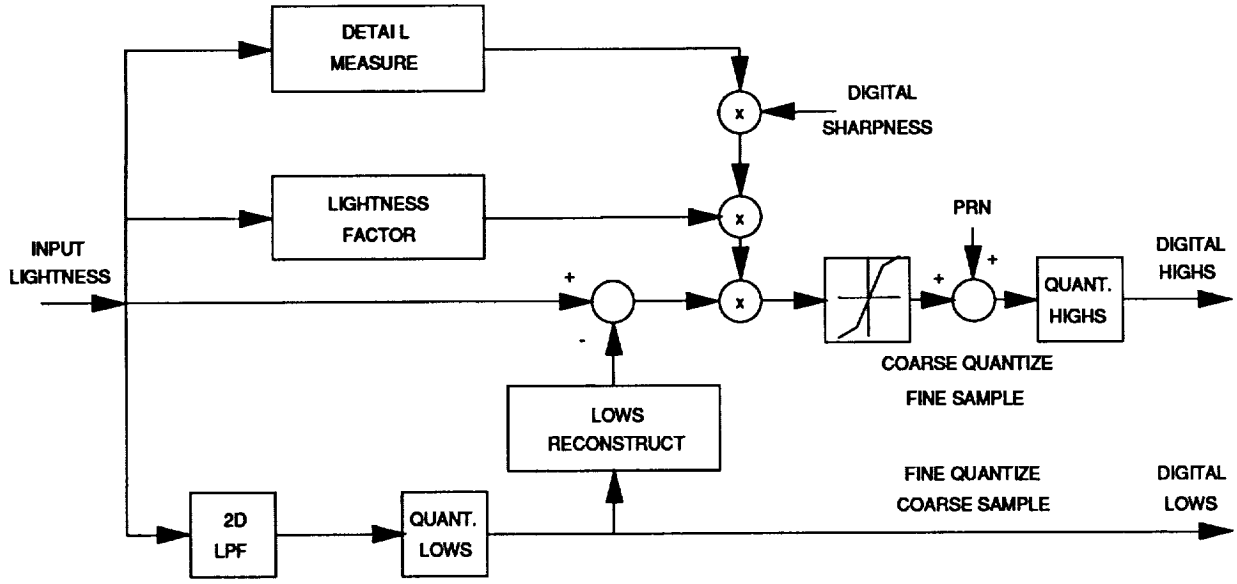


Figure 5: A Minimum Noise Visibility Coding encoder

As with many other HVS compensation techniques, MNVC [67] involves separating the high and low spatial frequency components. The low frequency component is sub-sampled at 1:5 and transmitted with the full 8-bit representation. The high frequency component is quantized at a much lower bit rate per pixel. First, however, pseudo-random noise (PRN) is added in order to eliminate the correlation that would exist in normal quantization noise. Then a tapered quantizer is used to place more error in high detail areas in order to take advantage of the property of the HVS mentioned above.

Color images are encoded in the YIQ format. The Y (luminance) signal is encoded identically to a monochrome signal. The I and Q (chrominance) signals are sub-sampled in the same manner as the low frequency luminance component.

At the receiver the PRN is subtracted from the high frequency component. The low frequency component is interpolated to full resolution, and the two components are combined to produce the reconstructed image.

Compression

Using standard television resolution, encoding the high frequency luminance component at 4 bpp produced images subjectively as good as the original 8 bpp signal. The low frequency and chrominance components each use 0.53 bpp, so the overall result was 5.6 bpp for a color signal and 4.5 bpp for monochrome. MNVC is a low-compression technique where the emphasis is on invisibility of the quantization noise when the bit rate is reduced. It was found to produce subjectively less degradation than DPCM for a given bit rate.

Spatial Domain

The degradations involve quantization noise in the high frequency component and under-sampling in the low frequency component. The latter is not significant due to the sampling theorem.

Temporal Domain

The PRN that is added to the high frequency signal can produce a "dirty window" effect if it is synchronized to the frame rate. Using totally unsynchronized PRN makes the image appear more noisy. This trade-off has to be optimized.

Aesthetic Appearance

The number of bits per pixel mentioned above yields a reconstructed image whose noise is essentially invisible to a human observer. As the number of bits per pixel decreases, results similar to those from DPCM become likely.

Spectral Information

Under-sampling the I- and Q-components at 5:1 had little effect on the overall quality of the color image.

Constant Area Quantization

Constant Area Quantization (CAQ) [68][69][70] is a predictive technique similar to Delta Modulation. Unlike DM, however, the difference is compared to a threshold; and if the difference is small enough, a 0 is transmitted. If the magnitude of the signal is above threshold, either a P or N is produced to indicate positive or negative change. The technique is called Constant Area Quantization because the threshold, as well as the positive and negative quantization levels, are adjusted to keep the area, A, (distance x luminance) under the triangle from the previous P or N to the current one constant. This is equivalent to setting the threshold equal to A/n where n is the number of pixels since the previous P or N was produced. This scheme provides high resolution for high contrast regions and high compression for low contrast regions. The motivation for CAQ is the property of human vision where the eye sees more detail in high contrast regions.

A number of modifications have been made to CAQ in order to reduce the error and/or the bit rate. One modification makes the area threshold adaptive to the detail of the image. Another possibility is to have a 2-D predictor instead of the using the value at the last P or N as the prediction. This is called Predictive CAQ (PCAQ). The entropy of the output reduces 30% by providing a better prediction, thereby reducing the number of Ps and Ns. A third possibility for reducing the error incorporates overshoot into the scheme, i. e., to make the quantization levels larger than the thresholds. This allows the reconstructed signal to follow the original signal more closely. Combining a Hadamard transform in the perpendicular (vertical) direction with the basic CAQ will also reduce the bit rate by taking advantage of the correlation in that direction also.

Some of the great advantages of the basic CAQ are its minimal complexity, power consumption, and cost. The method was originally designed for a remotely piloted vehicle where these factors are critical. Adding overshoot does not increase the complexity, whereas improving the predictor does. However, with the improvements in hardware in the last decade, all of the variations considered above should be practical even under severe power and cost constraints.

Compression

The basic CAQ will produce an output signal having a maximum zeroth-order entropy of 1.58 ($=\log_2 3$) bpp. Typically, the entropy is about 1.1 bpp since 0 is more common than P or N. Huffman coding is used to approach the actual entropy values. When the entropy is near or below 1 bpp, Huffman coding should be implemented on blocks of 2 or more pixels in order to include code symbols that require less than 1 bpp. Using a high detail test image, the MSE was about 3% at 1.08 bpp. Introducing overshoot produced an MSE of less than 2% at about 1.2 bpp. PCAQ resulted in an MSE of about 1% at 1.3 bpp and about 1.5% at 1.0 bpp, using the same test image.

Spatial Domain

Basic CAQ has the same type of problems as DM, i. e., not being able to handle both large slopes and low contrast regions. Also, the reconstructed signal always lags the original and contains blurring of edges. Large slopes and low contrast regions are reduced somewhat by using PCAQ to provide a better prediction or by making the method adaptive. The lag and blur are relieved by using overshoot.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The images produced by the basic CAQ appear blurry. However, it is still possible to use the images for object recognition. PCAQ produced a much clearer image.

Spectral Information

Not Applicable.

Perceptual Space Coding

A method which minimizes the perception of distortion takes advantage of the HVS model. Perceptual Space Coding [71][72][73] attempts to minimize the distortion in a "perceptual" space, i. e., in a domain obtained by passing the image through the model of the HVS.

A commonly used model of the HVS involves a linear transformation of the input color components (RGB, YIQ, etc.) into a luminance component and the two color components that correspond to the types of cones in the retina. Each component passes through a logarithmic compression in an attempt to model the neural response of the receptors of the retina. The next step in the HVS involves what are known as color-opponent cells. The action of these cells is modelled by two weighted linear difference circuits which subtract weighted multiples of the chrominance signals from the luminance signal to produce the perceptual chromatic information. Each perceptual component is then passed through a linear band-pass filter which replicates the lateral inhibition mechanism of the ganglion cells.

Using this model, the distortion criterion is the MSE in the perceptual space at the output of the system. To compress the image, it is first passed through the system, i. e., converted to the perceptual space. The linear filter is implemented by multiplication in the spatial frequency domain. The Fourier coefficients, at the filter's output, are then quantized with fewer bits than the original image using a method which minimizes the MSE. The bit allocation for the Fourier coefficients is determined by the power spectra of all three components so that more bits are used for the coefficients that contain more power.

At the receiver the inverse of the HVS model is applied. The reconstructed image may have a relatively high MSE in the image domain. However, the errors are "hidden" so that they are not highly noticeable. Therefore, the subjective image quality is much higher than other images with the same MSE.

If the image is monochrome, only the logarithmic compression and bandpass filter steps of the HVS model are used. Otherwise, the method remains the same.

Compression

A low detail 512 x 512 monochrome image was encoded at 0.1 bpp with an MSE in the image domain of 0.72%. This image was of usable quality. A number of color images of varying detail were encoded at 0.25 bpp with MSEs ranging from 0.36% to 3.3%. These images were reported to be of excellent quality.

Spatial Domain

The errors are due to quantization errors in the spatial frequency domain. Since the transform is done on the entire image, no blockiness should occur.

Temporal Domain

Not Applicable.

Aesthetic Appearance

The degradation of quality with reduced bit rates was reported to be "graceful".

Spectral Information

No specific information was reported.

---

60. Schreiber, W.F., "Picture Coding," *Proceedings of the IEEE,* vol. 55, pp. 320-330, March 1967.

61. Kunt, M., Ikonomopoulos, A., and Kucher, M, "Second-Generation Image-Coding Techniques," *Proceedings of the IEEE,* vol. 73, pp. 549-574, April 1985.

62. Burt, P.J. and Adelson, E.H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications,* vol. COM-31, pp. 532-540, April 1983.

63. Kunt, M., Ikonomopoulos, A., and Kucher, M, "Second-Generation Image-Coding Techniques," *Proceedings of the IEEE,* vol. 73, pp. 549-574, April 1985.

64. Kocher, M. and Kunt, M., "A Contour-Texture Approach to Picture Coding," *Proceedings of the ICASSP-82,* pp. 436-440, May 1982.

65. Kunt, M., Ikonomopoulos, A., and Kucher, M, "Second-Generation Image-Coding Techniques," *Proceedings of the IEEE,* vol. 73, pp. 549-574, April 1985.

66. Wilson, R., et al., "Anisotropic Nonstationary Image Estimation & Its Applications: Part II," *IEEE Transactions on Communications,* vol. COM-31, pp. 398-406, March 1983.

67. Troxel, D.E., et al., "A Two-Channel Picture Coding System: I - Real-Time Implementation," *IEEE Transactions on Communications,* vol. COM-29, pp. 1841-1848, December 1981.

68. Pearson, J.J., "A CAQ Bandwidth Reduction System for RPV Video Transmission," *Proceedings of the SPIE,* vol. 66, pp. 101-107, 1975.

69. Pearson, J.J. and Simonds, R.M., "Adaptive, Hybrid, and Multi-Threshold CAQ Algorithms," *Proceedings of the SPIE,* vol. 87, pp. 19-23, 1976.

70. Arnold, J.F. and Cavenor, M.C., "Improvements to the CAQ Bandwidth Compression Scheme," *IEEE Transactions on Communications,* vol. COM-29, pp. 1818-1823, December 1981.

71. Frei, W. and Baxter, B., "Rate-Distortion Coding Simulation for Color Images," *IEEE Transactions on Communications,* vol. COM-25, pp. 1385-1392, November 1977.

72. Hall, C.F., "Perceptual Coding in the Fourier Transform Domain," *NTC '80 Conf. Rec.,* pp. 36.1.1, 1980.

73. Hall, C.F. and Andrews, H.C., "Digital Color Image Compression in a Perceptual Space," *Proceedings of the SPIE,* vol. 149, pp. 182-188, August 1978.

Transform Coding

Transform coding is an information lossy technique which uses a mathematical operator on the data representing a digital image. The input data are typically highly correlated, so the goal of transform coding is to derive an array of uncorrelated or nearly uncorrelated data from the input data. A typical transform coding scheme is shown below.

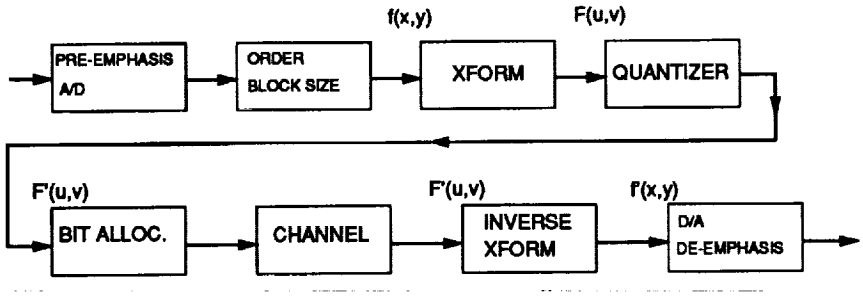


Figure 6: The process of Transform Coding

The mathematical operators used in transform coding form a complete orthogonal set of basis vectors. A complete set of basis vectors allows the original data to be described as a linear combination of all basis vectors in the set. Orthogonality implies, in a given set of basis vectors, any one basis vector cannot be represented as a linear combination of the other basis vectors. In other words, each of the basis vectors is unique. The Fourier transform is a very familiar transform whose basis vectors are complex exponentials.

A block of digital image data, $f(x,y)$, in an M x N array can be expressed as a new M x N array, $F(u,v)$, via a two-dimensional transform according to the relationship

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\phi_{uv}(x,y)$$

where $\phi_{uv}(x,y)$ are a set of orthogonal basis vectors.

The image data are fully recoverable using the inverse transform

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\phi_{xy}^{-1}(u,v)$$

again, where $\phi_{xy}^{-1}(u,v)$ represents the transform kernel or basis vectors for the image space.

It is common to think of transform coefficients as representing a "frequency domain", although this term is truly accurate only in the case of the Fourier transform. Some authors have coined the word "sequency" to describe the pseudo-frequency domain behavior of the other transforms.

A transform acts to "pack" a large number of highly correlated image data samples into a smaller number of uncorrelated coefficients. If the image's information is analogous to the energy stored in a mechanical system, then transforms pack diffuse energy into more compact energy "packets". The majority of the energy, the average, is packed into the first "packet". In discussing images, this first value is known as the DC coefficient. This term represents the average intensity, or value, of the pixels in the image block. The remaining terms are known as AC coefficients. The amount of the image's information in each packet, or the amount of correlation between pixels, decreases as the order of the coefficients increases. For most transform kernels, this order is the "sequency" order. Many bit allocation and quantization schemes (digitization) act to reduce or filter this high "sequency" data. The effect is analogous to frequency filtering in electrical and electronic systems.

In broadcasting and other types of electronic communication, a technique known as pre-emphasis/de-emphasis is employed to overcome deleterious effects of bandwidth compression. This electronic processing is proven beneficial in image processing. Electronic pre-emphasis is added to the analog image signal prior to digitization. At the receiver, electronic de-emphasis is added after the digital, reconstructed image is converted to the analog domain. In addition, an optical filtering process is employed with spatial bandpass filtering prior to the video camera's focal plane. The section on optical implementations of VDC in HHVT discuss this second method.

Both techniques help to reduce the blockiness and edge busyness associated with transform methods. They contribute very little to implementation complexity, while increasing the effectiveness of image compression. A technique such as these should be an integral part of any image compression or processing scheme.

The transform is a reversible process until quantization of the coefficients and no data compression occur. Both of these processes will introduce distortion. Also, usually M = N.

Quantization of coefficients can be performed using a uniform quantizer, an optimal quantizer, a compander/expander or by adaptive methods. The quantization process is the key step in a transform coding algorithm since it is here data compression is performed and output image fidelity is affected. $F(u,v)$, known as the coefficient array, and $f(x,y)$, the original image block, are both N x N arrays with n bits per array element or $N^2 \times n$ bits total. Data compression of the $F(u,v)$ array is achieved by reducing the number of bits used to quantize some of the coefficients. A common feature of the various transform coding algorithms is many of the coefficients will be of very small magnitude. These coefficients are thus coarsely quantized or even omitted, with negligible effect on image quality.

This process will result in a compressed coefficient array $F'(u,v)$ which may be further compressed by entropy coding the coefficients. The result will be transmitted over the communication link with some type of error correcting code. At the receiver, decoding followed by the inverse transformation (and de-emphasis) will be performed on the $F'(u,v)$ array, resulting in $f'(x,y)$, the output image.

The following sections describe the various parameters of transform coding systems. It is assumed we are dealing with square images, e. g., 512 x 512 or 1024 x 1024 pixels.

Block Size and Dimensionality

Transforms are performed on line segments of length N of an image (one dimensional), on N x N blocks of the image (two dimensional), or on N x N x M blocks of an image sequence (three dimensional). Often, the third dimension is time. One and two dimensional transforms are intraframe coding techniques, whereas three dimensional transforms are interframe coding techniques.

One dimensional transforms are usually performed on the horizontal lines of an image since data are often raster scanned horizontally. A whole line or a segment of a line may be used. One dimensional transform coding has advantages over multidimensional transform coding of speed and implementation simplicity. However, one dimensional transforms are inefficient since they only take advantage of spatial correlation of an image in the horizontal direction. Thus, under the same image fidelity criteria, an one dimensional transform coder will not be able to achieve as high a compression ratio as the multidimensional transforms.

In practice, one dimensional transform coding is seldom implemented by itself. A common technique uses transform coding in the horizontal direction and predictive coding in the vertical and temporal directions.

Two dimensional transform coding usually begins by subdividing the whole image into N x N sub-blocks. Transform coding is then performed on the sub-blocks. Two dimensional transforms take full advantage of intraframe correlation in both directions, unlike one dimensional transforms. They are not as fast since N lines of the image must first be stored before the transform process can begin, and they are somewhat more complex to implement. Usually, the increased coding efficiency more than offsets the other factors.

The choice of block size (the choice of N) is an important parameter in two dimensional transform coding schemes. For simplicity, N is always chosen as an integer power of 2. Large and small values of N each have unique advantages and disadvantages. Transform coding schemes with large values of N perform better under a given fidelity criterion at high compression ratios [74]. These schemes take advantage of correlation over a larger area. Those with smaller values of N are faster, easier to implement and more receptive to adaptivity. One author [75] suggests the intraframe correlation for most images is negligible beyond a spatial distance of 20 pixels and recommends smaller blocks. 4 x 4, 8 x 8, 16 x 16, 32 x 32 and 64 x 64 transform coding schemes have been implemented in cases appearing in the literature.

Three dimensional transform coding schemes subdivide a sequence of image frames into N x N x M blocks. They are the most efficient of transform coding schemes since they fully exploit correlation in the spatial and temporal directions. However, they are high in implementation complexity. Also, they are much slower since M-1 frames plus N lines of the image sequence must be stored before the transform coding can start. This decreases their attractiveness for use in real-time systems.

To minimize complexity, small values of N and M are chosen. Research indicates higher values of M (i.e., coding more frames at a time) result in a lower MSE [76], but choosing M = N allows for vector processing to speed the address evaluations of three dimensional arrays [77]. Block sizes of 4 x 4 x 4 and 8 x 8 x 4 and 4 x 4 x 8 are most common in the literature [78].

Quantization and Bit Allocation

Once the array of transform coefficients is determined, the next task is to quantize them for storage or transmission. Data compression is achieved by deleting low magnitude coefficients and coarsely quantizing. Coarse quantization assigns fewer bits to these coefficients. There are several methods for accomplishing this.

One method is known as threshold sampling or magnitude sampling. By this method, all coefficients above a certain magnitude are retained while those below the threshold are deleted. This results in an address set, $F'(u,v)$ given by

$$F'(u,v) = \{F(u,v): \quad |F(u,v)| > T\},$$

where T is the predetermined threshold.

Extra bits are required to address the transmitted coefficients [79], so magnitude sampling requires overhead. However, magnitude sampling adapts to the local statistics of the image, and thus performs well. The transmitter's coding operation is independent of the receiver's decoding operation. Also, this overhead is often much less than the overhead required to transmit a codebook in other block based methods.

Another quantization method, zonal sampling, uses on the variance of the transform coefficients. The variance of the coefficients decreases for the higher sequency coefficients, and the coefficients with the largest variances contribute most to the reconstructed image [80]. Zonal sampling assumes a given coefficient's variance will be constant for a given class of images, even though its amplitude will fluctuate. This allocation scheme relies on average statistics across a class of images. However, it is suboptimal since it does not adapt to the local statistics of each individual image as was the case with threshold sampling. The address set, $F'(u,v)$, for zonal sampling is obtained by

$$F'(u,v) = \{F(u,v): \quad \sigma^2_{u,v} > V_T\},$$

where V_T is the minimum variance with which a coefficient is retained.

In addition, since zonal sampling is based on average statistics of a class of images, a bit allocation map may be developed *a priori* which results in coarser quantization based on smaller variance (dynamic range). The DC coefficient (representing the average value) has the largest variance and will be quantized with a full eight bits, but other coefficients will receive fewer and fewer bits based on smaller and smaller variances. These zones, within which coefficients share the same number of bits, may be determined empirically or using a numerical fidelity criterion such as

$$n_b = \text{trunc}\left[\frac{1}{2}\log_2\frac{\sigma^2_{u,v}}{D} + \frac{1}{2}\right]$$

where $n_b(u,v)$ is the number of bits assigned to coefficient $F'(u,v)$, trunc represents the truncation operator, and D is the maximum distortion penalty of quantizing with n_b bits [81]. The result of a zonal sampling algorithm is a bit allocation map such as the one shown below in Figure 7.

```
8 7 6 5 4 3 3 2 2 1 1 1 0 0 0 0
7 6 6 5 4 3 3 2 2 1 1 1 0 0 0 0
6 5 5 4 4 3 3 2 2 1 1 1 0 0 0 0
5 5 4 4 4 3 3 3 2 2 1 1 1 0 0 0
4 4 4 3 3 3 3 2 2 2 1 1 1 0 0 0
3 3 3 3 3 2 2 2 2 2 1 1 1 1 0 0
3 3 3 3 3 2 2 2 2 1 1 1 1 1 0 0
2 2 2 3 2 2 2 1 1 1 1 1 1 0 0 0
2 2 2 2 3 2 2 1 1 1 1 1 1 1 0 0
2 2 2 2 2 2 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
```

Figure 7: Bit Allocation Map from Zonal Sampling

Once the coefficients and their respective bit allocations are known, quantization is performed. This can be accomplished with either a uniform or nonuniform quantizer. Statistically, it has been shown the DC coefficient is best approximated by a Gaussian distribution and the remaining coefficients are best approximated by a Laplacian distribution [82]. Therefore, optimally, the DC coefficient would be quantized with a Gaussian quantizer and the remaining coefficients with a Laplacian quantizer.

The quantization and bit allocation process results in data compression, but this process also introduces distortion in the reconstructed image, $f'(x,y)$. The mean square error resulting from quantization is given by

$$\varepsilon^2 = E\{\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}[f(x,y)-f'(x,y)]^2\}$$

where E represents the expectation operator.

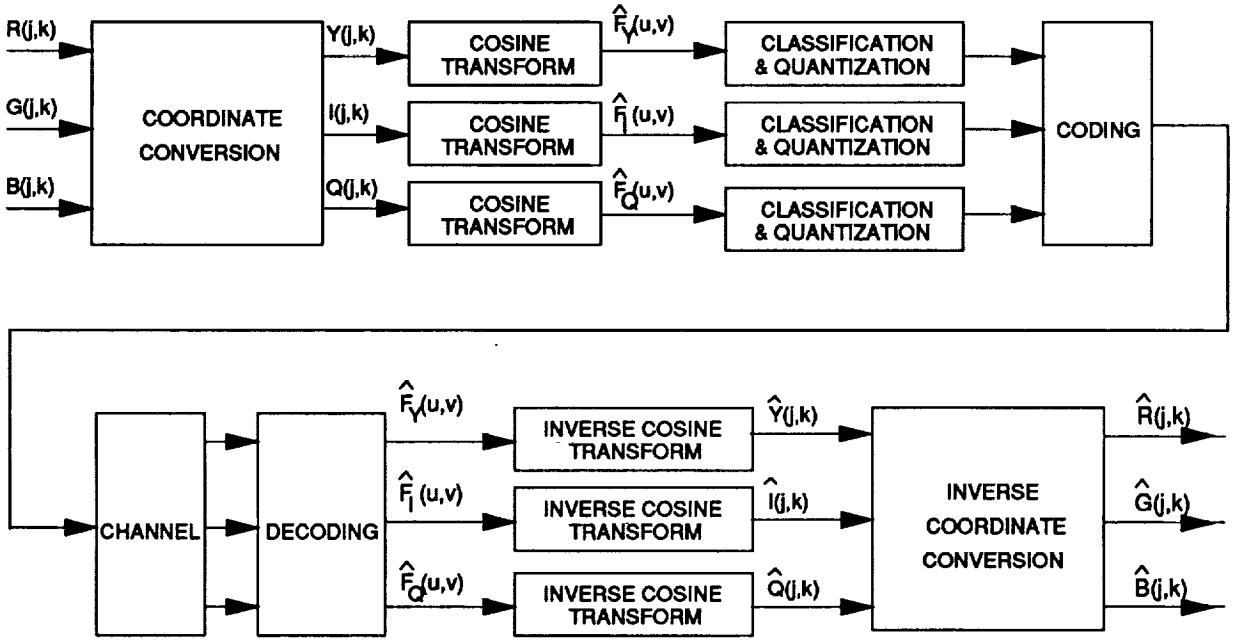
For the same mean square error, zonal sampling results in reconstructed images more objectionable to subjective human observers than threshold sampling [83]. The quantization noise from zonal sampling is more noticeable than the low pass filtering from threshold sampling. This implies a complexity vs. performance trade-off, which may be optimized with a hybrid sampling technique [84].

Color Signal Transform Coding

Color images may be transform coded by coding the individual color components separately or coding the composite signal. For digital video systems, component coding is more desirable [85]. Even though the component system conversions may cause additional degradation, the better coding efficiency usually compensates for this [86].

YIQ co-ordinate conversion provides almost as high an energy compactness for color images as does the Karhunen-Loeve Transform (KLT) color co-ordinate conversion [87]. By its definition, the KLT (or principle components) produces the most uncorrelated, and therefore optimal, transform coefficients.

For component coding, the YIQ signals are available after a component conversion process. One wishes to transform code with this component system, since these three signals are nearly uncorrelated (unlike the RGB signals which are quite correlated) and since monochrome systems will only require the encoded Y component. This will result in the following three coefficient arrays: $F_Y(u,v)$, $F_I(u,v)$, and $F_Q(u,v)$ derived from the three component signal arrays $f_Y(x,y)$, $f_I(x,y)$, and $f_Q(x,y)$ which represent a digitized sub-block of the image. This information permits quantizing the I and Q signals more coarsely (with fewer bits) than the Y signal. This achieves additional data compression.



Classification Map for Y

4	3	4	4	4	3	4	1	1	1	2	4	2	1	3	3
4	3	4	4	4	2	1	1	2	1	1	1	1	2	2	2
4	3	4	4	4	1	1	1	1	1	2	1	1	2	2	3
4	3	4	4	1	2	1	2	3	3	3	1	2	3	2	3
4	3	4	4	1	2	2	2	2	3	1	1	2	2	3	4
4	4	4	4	3	1	2	3	2	2	2	1	2	3	3	3
4	4	4	4	4	1	1	1	2	3	3	1	2	2	3	3
4	4	4	4	4	1	1	1	1	3	1	1	3	2	3	3
4	4	4	4	4	2	1	1	2	2	1	2	3	2	3	3
4	4	4	4	4	2	1	1	2	2	2	2	2	3	2	3
3	4	4	4	4	3	1	1	3	3	1	2	2	2	3	3
2	2	3	3	2	2	3	1	2	3	4	3	1	1	3	3
2	2	2	1	3	4	3	4	3	4	4	3	2	1	1	3
2	3	1	2	3	4	4	3	3	1	1	3	4	3	1	2
4	4	2	2	4	3	3	3	2	1	1	1	1	3	3	1
4	4	2	2	4	3	3	1	1	1	1	1	1	1	4	1

Classification Map for I

4	4	4	4	4	4	3	3	2	3	3	3	2	2	3	3
4	4	4	4	4	3	1	2	2	3	2	2	2	2	2	2
4	4	4	4	3	1	2	2	2	2	3	2	1	2	3	3
4	4	4	4	2	3	2	2	3	3	2	2	2	2	3	3
4	4	4	4	2	2	2	2	2	3	2	2	2	3	3	3
4	4	4	4	3	2	2	2	2	3	3	2	2	3	3	3
4	4	4	4	4	1	1	1	1	2	3	2	3	3	3	4
4	4	4	4	4	1	2	2	2	2	2	2	3	3	3	4
4	4	4	4	4	1	2	2	2	3	1	3	3	3	3	4
4	4	4	4	4	1	1	2	2	3	2	3	3	3	3	3
2	4	4	4	3	2	1	1	3	3	1	1	3	3	3	3
1	1	4	4	1	1	1	1	1	1	1	1	1	1	3	3
1	1	2	1	2	3	1	2	1	1	2	1	1	3	3	3
1	2	2	1	1	2	2	1	1	1	1	2	4	4	3	3
4	4	3	1	2	1	1	1	1	1	1	1	1	4	4	3
3	4	3	1	1	1	1	1	1	1	1	1	1	1	4	4

Classification Map for Q

4	4	3	4	4	4	4	3	2	2	4	4	4	2	2	2
4	4	3	4	3	4	2	1	2	2	2	3	3	2	2	2
3	3	3	4	4	2	1	2	2	3	3	2	3	2	2	2
4	3	3	3	3	3	3	4	4	4	4	2	2	2	2	2
4	3	3	4	3	3	3	3	3	4	3	3	2	2	3	2
4	3	3	4	3	3	3	3	3	3	4	3	2	3	3	3
4	4	3	4	3	2	1	1	2	2	4	2	2	3	3	3
4	4	4	4	4	1	1	2	2	2	2	2	2	3	3	3
4	4	4	4	4	1	1	2	2	3	1	2	2	2	3	3
4	4	4	4	4	1	1	3	3	1	2	2	3	2	3	2
4	4	4	4	3	2	1	1	4	4	1	1	2	2	2	2
3	4	4	4	1	1	1	1	1	1	1	1	1	2	2	
4	3	2	1	2	3	1	2	1	1	2	1	1	1	1	2
4	3	1	1	1	3	2	1	1	1	1	1	4	4	2	2
4	4	2	1	2	1	1	1	1	1	1	1	1	3	4	1
4	4	3	1	1	1	1	1	1	1	1	1	1	1	4	2

Figure 8: Typical YIQ component coding scheme, with bit allocation tables based on zonal sampling.

Adaptivity

Adaptive transform coding schemes compensate for statistically non-stationary images by changing quantization levels or bit allocations. Optimal adaptive methods use changes in local statistics. Adaptive techniques tend to improve coding efficiency at the expense of raising complexity. A number of simple adaptive and quasi-adaptive techniques have been developed. This report investigates some of them.

The most common adaptive technique uses coarse coefficient bit allocations in regions of the image where detail (i. e, high sequency coefficient contribution) is low and finer quantization in regions of high detail. A drawback which is immediately apparent is the problem of informing the receiver of the quantization scheme.

An effective way to overcome this problem is the technique known as class adaptive transform coding. Depending on the sum of the magnitudes or variances of the coefficients (known as the activity index), a particular coefficient array is classified into one of K distinct classes. K = 4 is by far the most common [88]. This requires only a $\log_2 M$ bit overhead per block to identify of which class that particular block is a member. The added complexity is moderate and the coding efficiency at a given fidelity criterion can improve significantly, depending on how stationary the image's statistics are. Figure 9 shows the bit allocations for the four classes for the luminance component of one such coding scheme. Note that since different numbers of bits are used to code different blocks, there must be an output buffer to allow a constant output bit rate [89].

Class adaptive transform coding based on coefficient variances is more efficient than coding based on coefficient magnitudes [90]. We will describe two techniques here.

With the first variance-based technique, a procedure known as recursive block quantization is used to determine coefficient variances. The coefficients are placed into a one-dimensional array using the ordering procedure shown in Figure 10 -- $F(u,v)$ becomes F_i -- and the recursive relationship for coefficient variance is given by $\hat{\sigma}^2_{i+1} = w\hat{\sigma}^2_i + (1-w)\hat{F}_i$, where $\hat{\sigma}^2_i$ is the quantized value of the variance of the coefficient F_i, $\hat{F}_i$ is the quantized value of F_i, and w is a weighting factor found to be 0.75 for best results.

8	7	6	5	4	3	3	2	2	1	1	1	0	0	0	0
7	6	6	5	4	3	3	2	2	1	1	1	0	0	0	0
6	5	5	4	4	3	3	2	2	1	1	1	0	0	0	0
5	5	4	4	4	3	3	3	2	2	1	1	1	0	0	0
4	4	4	3	3	3	3	2	2	2	1	1	1	0	0	0
3	3	3	3	3	2	2	2	2	2	1	1	1	1	0	0
3	3	3	3	3	2	2	2	2	1	1	1	1	1	0	0
2	2	2	3	2	2	2	1	1	1	1	1	1	0	0	0
2	2	2	2	3	2	2	1	1	1	1	1	1	1	0	0
2	2	2	2	2	2	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

Class 1

8	6	5	4	4	3	2	2	2	1	1	1	0	0	0	0
6	5	5	4	4	3	2	2	2	1	1	1	0	0	0	0
5	5	4	4	4	3	3	2	2	1	1	1	0	0	0	0
4	4	4	3	3	3	3	2	2	2	1	1	0	0	0	0
3	3	3	3	3	3	3	2	2	2	1	1	1	0	0	0
3	3	3	3	2	2	2	2	2	2	1	1	1	1	0	0
2	2	2	2	2	2	2	2	2	1	1	1	1	0	0	0
2	2	2	3	3	2	1	1	1	1	1	1	1	1	0	0
2	2	2	3	4	3	2	1	1	1	1	1	0	0	0	0
1	2	1	2	3	2	1	1	1	1	1	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	2	2	1	0	0	0	0	0

Class 2

8	5	4	3	3	2	2	1	1	1	1	0	0	0	0	0
5	5	4	4	3	3	2	2	1	1	0	0	0	0	0	0
4	4	3	3	3	3	2	2	1	1	1	0	0	0	0	0
3	3	3	3	3	2	2	2	1	1	1	0	0	0	0	0
3	3	3	2	2	2	2	1	1	1	1	0	0	0	0	0
2	2	2	2	2	2	1	1	1	1	1	0	0	0	0	0
2	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	2	1	1	1	1	0	0	0	0	0	0	0
1	1	1	2	2	2	1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Class 3

8	3	3	2	1	1	0	0	0	0	0	0	0	0	0	0
3	3	2	2	1	1	1	1	0	0	0	0	0	0	0	0
2	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Class 4

Figure 9: Bit allocations for four classes

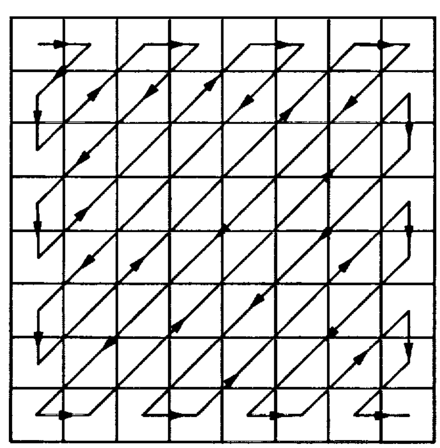


Figure 10: Ordering of Coefficients in recursive block quantization

The DC coefficient is quantized with the full 8 bits, and the initial condition for the recursive relationship is found by averaging the square of the first four AC coefficients to obtain $\hat{\sigma}_2$.

The second technique [91][92][93] makes two assumptions based on the original image, $f(x,y)$. First, this technique assumes the mean value of all AC coefficients is 0. It also assumes the mean of the DC coefficient is the average brightness of the original image $f(x,y)$ which is denoted as m. From the transform definition, this implies $E\{F(u,v)\}$ is $2m$. The variances of the coefficients, $F(u,v)$, are then given by

$$\sigma_{0,0}^2 = E\{[F(0,0)]^2\} - 4m^2; \quad u = v = 0,$$

and

$$\sigma_{u,v}^2 = E\{[F(u,v)]^2\}; \quad (u,v) \neq (0,0).$$

Approximate Gaussian density functions for each coefficient are formulated from these parameters. Using the results of [94] we can set up Laplacian densities for the AC coefficients. The Laplacian density is given by

$$p_{F(u,v)}(\beta) = \frac{\alpha_{u,v}}{2} e^{-\alpha_{u,v}|\beta|},$$

where $\alpha_{u,v}$ is a parameter which can be calculated from $\sigma_{u,v}^2$ by

$$\alpha_{u,v} = \sqrt{\frac{\sigma_{u,v}^2}{2}} \qquad (u,v) \neq (0,0).$$

and β represents an AC coefficient.

The AC energy of a transform block is computed as

$$E_{AC} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} [F(u,v)]^2 - [F(0,0)]^2.$$

The magnitude of the AC energy is used to classify the transform block into the high energy class or the low energy class. Several frequency regions, consisting of 16 x 16 blocks, are of particular interest. These regions include low-frequency, mid-frequency, high-frequency, and horizontal and vertical edges. Figure 11 shows the locations of typical regions.

Then, the ratio of low frequency AC energy to high frequency AC energy will subdivide each of these classes into a high frequency class and a low frequency class. High and low frequency coefficients are grouped. This system works better than classifying based on AC energy alone since human vision is more sensitive to high frequencies at low illuminations [95].

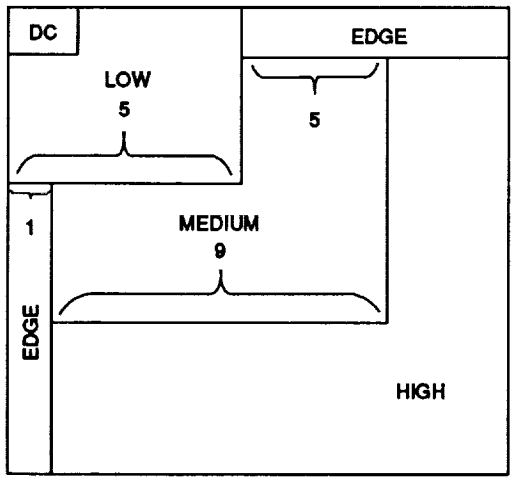


Figure 11: AC energy classification by frequency

The result of this procedure will be each of the four classes will contain different numbers of image blocks, which hinders the maintaining of a constant bit rate. The solution assigns bits to the classes with the constraint that the average bit rate is constant.

The next step calculates the ensemble average of the variances of each F(u,v) coefficient in the whole frame for each class k by

$$\sigma_k^2(0,0) = \frac{N^2K}{L^2} \sum_{m=0}^{\frac{L}{N\sqrt{K}}-1} \sum_{n=0}^{\frac{L}{N\sqrt{K}}-1} [F_{m,n}(0,0)]^2 - 4m^2;$$

and

$$\sigma_k^2(u,v) = \frac{N^2K}{L^2} \sum_{m=0}^{\frac{L}{N\sqrt{K}}-1} \sum_{n=0}^{\frac{L}{N\sqrt{K}}-1} [F_{m,n}(u,v)]^2; \quad (u,v) \neq (0,0);$$

where $k = 1,2,\ldots,K$ and the variables m,n index the various sub-blocks within the total image. The total image size is L x L and the sub-block size is N x N.

Now we can consider bit allocations for each class. The bit allocation matrix for class k is given by the following equations

$$N_{b_k}(u,v) = \frac{1}{2}\log_2[\sigma_k^2(u,v)] - \log_2 D; \quad (u,v) \neq (0,0),$$

and

$$\log_2 D = \frac{N}{2L(N-1)}\left[c_k \sum_{u\neq 0}^{N-1}\sum_{v\neq 0}^{N-1} \log_2[\sigma_k^2(u,v)]\right] + \frac{b_{DC} - Nb_{avg}}{N-1}$$

where c_k is the number of N x N blocks assigned to class k, b_{DC} is the number of bits assigned to the DC coefficient (usually 8 bits) and b_{avg} is the desired average bits/pixel (bpp) obtained from the desired compression ratio, not including overhead. This may need to be done iteratively until the desired number of bits is used exactly.

Now, the classes and bit allocations have been determined. The next task is to normalize the transform samples prior to quantization. For a given class and coefficient, the normalization coefficient is given by

$$\sigma_k(u,v) = c\left(2^{N_{b_k}(u,v)-1}\right); \quad (u,v) \neq (0,0).$$

where c is a normalization factor which is the maximum standard deviation value of all transform sample values which are assigned one bit.

The overhead in bits/frame for this rather involved but effective technique is given by

$$B = 2\left(\frac{L}{2N}\right)^2 + b_c + 6N^2,$$

where b_c is the number of bits to encode c. For N = 8, L = 1024 and $b_c = 6$, the overhead is about 8600 bits/frame or about 0.008 bit per pixel [96][97]. Clearly, overhead is minimal.

Interestingly, the concept of threshold sampling of the coefficients described in a previous section is by nature an adaptive technique since more coefficients will be transmitted when the image block has finer detail [98]. This has the drawback of needing to code the positions of the coefficients, however.

Another adaptive technique is to vary quantization parameters so the MSE is maintained at a constant value. This technique is complex to implement and has been shown to perform no better than class adaptive transform coding [99].

Coding The Coefficients

Once the $F'(u,v)$ array representing the quantized coefficients is known, further data compression can be achieved by coding the coefficients for transmission using the entropy coding techniques described earlier.

Run length coding is a logical technique since the coefficient array will contain many zero-valued coefficients.

It has been suggested that chain (or contour) coding of transform coefficients could reduce data by 10-30% [100][101]. By chain coding the boundaries between the zero coefficient and non-zero coefficient regions, the non-zero coefficients are clustered together and can be more efficiently identified.

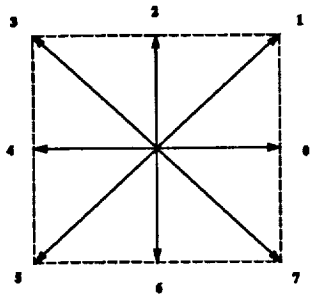


Figure 12: 8-direction Freeman chain code

Chain codes are a set of directed line segments known as links. A candidate link is selected for the chain's next link if it spans only zero-coefficients and if at least one neighboring coefficient, to the right of the link, is non-zero. An eight direction Freeman code is shown in the adjacent figure. This eight-direction code is only the first member of a large family of $(8 \times m)$-direction codes, where m is an integer. This family is suitable for representing planar curves.

In the case of adaptive transform coding, the chain coding algorithm outlines the boundaries of non-zero coefficients in the transform domain. The non-zero coefficients and the chain links are then coded for transmission. The coded chain links provide a more efficient coding of zero coefficients than does run-length coding. The additional complexity of implementing this algorithm is modest.

The Basis Vectors

Finally, we need to consider different types of basis vector sets. These sets of functions need to be orthogonal and complete. Orthogonality implies a given basis vector cannot be represented as a linear combination of the other basis vectors. Completeness means any image is specified in terms of a transform coefficient matrix and is fully recovered (no distortion) via the inverse transform as long as quantization and compression have not yet been performed. The basis vector set is perhaps the most widely studied parameter of transform coding and will be treated in-depth here.

Karhunen-Loeve Transform

The Karhunen-Loeve transform (KLT) is also known as the Hotelling transform or the method of principal components. The transform algorithm selects the optimal set of orthogonal basis vectors so the elements of the coefficient array are uncorrelated [102], that is

$$E\{F(u_1, v_1)F(u_2, v_2)\} = 0; \quad \text{if} \quad (u, v) \neq (0, 0)$$

with the transform coefficients assumed to be random variables. The basis vectors determined by the KLT are actually the eigenvectors of the covariance matrix.

Since it produces completely uncorrelated coefficients, the KLT represents the optimum transform based on lowest distortion at a given bit rate and the least bits required to encode the coefficient array at a given distortion criterion.

However, the KLT is not easily implemented since it requires by far the greatest number of calculations of any transform coding technique (it has no known "fast" algorithm). Also since the basis vectors are not known at the receiver, they must be encoded along with the coefficient array. This large overhead requirement defeats the optimum coding of the coefficient array.

Yet, the KLT is useful because it sets the limit on the performance achievable with any transform coding scheme. In practice, transform coding techniques use a suboptimal set of basis vectors which are already known at the receiver. As we shall see, the performance of the KLT can be closely approximated by some of these less optimal transforms at great computational savings.

The KLT requires the formation of a set of vectors (this depends on the output of the sensing device). It also requires the estimation of the covariance matrix and the calculation of its eigenvectors and eigenvalues. The time needed to calculate eigenvectors and eigenvalues is lengthy. It precludes real time, video rate operations. Not only is memory required when calculating the eigenvectors and eigenvalues; but additional memory is required for storing the elements of the covariance matrices.

NUMBER OF COVARIANCE MATRIX ELEMENTS VS. IMAGE SIZE		
IMAGE SIZE		NUMBER OF COVARIANCE MATRIX ELEMENTS
M	N	M^2N^2
4	4	256
8	4	1024
8	8	4096
16	8	16384
16	16	65536
32	16	262144
32	32	1048576
64	32	4194304
64	64	16777216
128	64	67108864
128	128	268435456

This table reflects the enormous memory required for the covariance matrix as the block size increases. "[A] major drawback is that the required number of computation steps is also proportional to M^2N^2 for an M x N image, which [this table] implies, is very large for many images" [103].

This lengthy, iterative nature of the KLT is known as principal components analysis. By using backward error propagation in a neural network implementation of this process, researchers hope to learn the elements of the optimal algorithm. Backward error propagation is a supervised learning scheme which changes the weights between non-linear units in a neural network. The non-linear units compute a sigmoidal function of their inputs. Learning occurs in each unit in the network by reducing the MSE of the output image. The learning algorithm produces a nearly linear transformation of the input, which are image pixels. The researchers termed the results "respectable . . . when compared to current techniques." [104].

Compression:

Good quality reconstructed pictures result when using the KLT, while achieving compression as low as 0.5 to 1.0 bpp. Experiments demonstrate the MSE ranges from 1.5% to 0.5% at these rates.

Spatial Domain

The KLT is capable of maintaining crisp edges and texture in the reconstructed image. These two features, being of higher spatial frequency, are the most sensitive to the quantization scheme which is employed. Some blockiness is observed in non-adaptive methods. For the best results, an adaptive scheme must be used. This suggests a continued interest in neural network implementations of this algorithm. With a compression rate of 1.0 to 1.5 bpp, the reconstructed images exhibit a good definition of detail.

Temporal Domain

No reports of studies involving motion or interframe applications were found. The studies emphasize still images, due to the computational complexity of the algorithm.

Aesthetic Appearance

The KLT proves robust in minimizing transmission errors. In general, this holds true for transform techniques. Still, as the bits per pixel decreases, transform encoding will show block errors. These errors are caused, in part, by transmission errors. However, because of the block oriented processing, transform encoding does not result in the compounded errors of other techniques. These compounded errors can cause streaking, lost frames, and jerky motion.

Quantizing errors are distributed throughout the reconstructed image. This benefit results in visually less objectionable images. Errors are often undetectable to an untrained observer. To show the errors, researchers often employ an image which maps %MSE x 4. In such images, one sees that errors are not concentrated to one specific feature, such as edges or low spatial frequency areas (contours). To the casual, or inexperienced, observer, the greatest errors occur in areas of very high detail or texture.

The KLT and DCT perform the best at preserving edges. Again, edge fidelity improves with adaptive quantization.

Spectral Information

The distortions produced by encoding color images were generally similar to those produced by using monochrome images.

Discrete Cosine Transform

The two dimensional Discrete Cosine Transform (DCT) of an N x N array is given by

$$F_{DCT}(u,v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)\pi u}{2N}\right] \cos\left[\frac{(2y+1)\pi v}{2N}\right]$$

where $C(0) = \frac{1}{\sqrt{2}}$ and $C(u) = 1$ if $u > 1$.

The inverse two dimensional DCT is given by

$$f(x,y)=\frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}C(u)C(v)F_{DCT}(u,v)\cos\left[\frac{(2x+1)\pi u}{2N}\right]\cos\left[\frac{(2y+1)\pi v}{2N}\right].$$

For N x N x M sub-blocks, the three-dimensional DCT is given by

$$F_{DCT}(u,v,w)=\frac{8}{N^3}C(u)C(v)C(t)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{t=0}^{M-1}f(x,y,t)\cos\left[\frac{(2x+1)\pi u}{2N}\right]\cos\left[\frac{(2y+1)\pi v}{2N}\right]\cos\left[\frac{(2t+1)\pi w}{2M}\right].$$

The DCT is an attractive choice for a basis vector set since the performance of the DCT closely approximates that of the KLT [105]. In fact, for an image which can be modeled by a Markov source with correlation coefficient close to unity, the KLT reduces to the DCT [106]. For most classes of images, this is a valid assumption.

Yet, the DCT has enormous advantages over the KLT in terms of simplicity. Since the basis vectors of the DCT are known in advance, they need not be calculated for every transform block as they do with the KLT. Also, the basis vectors need not be transmitted along with the coefficients, and there are several "fast" algorithms for computing the DCT which make real time applications feasible [107]. These savings in overhead actually make the DCT more efficient at a fixed bit rate than the KLT. This justifies removing the KLT from further consideration for HHVT. In fact, the DCT has been shown to have the highest coding efficiency (i.e., highest compression ratio with least distortion) of any transform coding scheme. However, in spite of the vast reduction in complexity over the KLT, the DCT is still moderate to high in complexity of implementation compared to other algorithms.

Computational simplicity suggests that the Hadamard transform is the best choice for an encoding implementation. Yet, as will be demonstrated, it suffers a significant performance degradation, especially when compared to the DCT. Specific VLSI hardware provides efficient, fast implementations of the DCT. Hence, it appears best for applications requiring the highest degree of compression while minimizing coding distortion.

Compression

Use of the Discrete Cosine Transform results in good quality reconstructed pictures, while achieving compression as low as 0.5 to 1.0 bpp. Experiments demonstrate the MSE ranges from 1.50% to 0.50% at these rates. Adaptive quantization yields even lower MSE. Experimental results range from approximately 0.75% to 0.20%. The higher bit rate, i. e., less compression, yields the lower MSE.

Spatial Domain

The DCT maintains crisp edges and texture in the reconstructed image. These two features, being of higher spatial frequency, are the most sensitive to the quantization scheme which is employed. To minimize blockiness, an adaptive scheme is preferred. With a compression rate of 1.0 to 1.5 bpp, the reconstructed images exhibit a good definition of detail.

Temporal Domain

Several reports of studies involving motion or interframe applications indicate gains in both compression and %MSE. This is a result of greater correlation between pixels in successive frames. Unfortunately, the increase in computation time, and circuit complexity and size, has limited the practice of interframe techniques. However, the use of a charge coupled device (CCD) or charge injection device (CID) sensors would greatly facilitate these algorithms.

A three dimensional DCT yields roughly a 30% improvement over the performance of a two dimensional DCT. When a motion compensated (MC) algorithm performs interframe compression, one achieves a 100% improvement over the 2D-DCT. These improvements are most noticeable in areas of no motion. As the motion increases, the advantages of the interframe encoding decrease; but this suggests a preservation of motion.

Aesthetic Appearance

The DCT proves robust in minimizing transmission errors. Schemes using very low levels of encoding, from 0.5 to 1.0 bpp, result in a great degree of compression. However, at these low levels, the reconstructed image becomes more sensitive to transmission errors. One begins to notice block errors.

The reconstructed image has very evenly distributed quantizing errors. This benefit results in visually less objectionable images. As with the Karhunen-Loeve transform, errors are often undetectable to an untrained observer. The most noticeable errors occur in areas of very high detail or texture.

The DCT performs the best at preserving edges. Again, edge fidelity improves with adaptive quantization.

A hybrid MC/2D-DCT does not portray blurred motion seen in some non-transform based techniques. While yielding a spatially, good quality reconstructed image, it does have some slight flicker in intensity around the boundaries of the motion. This flicker does not inhibit an observer's viewing or evaluation of the content of the image. A discussion of this method in the section on hybrid techniques will show some means to eliminate this.

Spectral Information

RGB distributed errors produce a slight reduction in color purity. For this reason, RGB values are converted to less correlated YIQ values. Discrete cosine transformations of the YIQ yield very accurate color in reconstructed images. Between adjacent pixels, the DCT maintains slight variations in hue and saturation in the reconstructed image. Very little contouring or banding appears. However, if the encoding is horizontally oriented, vertically errors appear as blocks in background areas. Often, the quantization procedure will place most of the errors in the Q- (and to a lesser degree, the I-) components. This minimizes the distortions in the reconstructed color images.

Slant Transform

The Slant transform was developed to take advantage of linear changes in brightness which occur in some classes of images. The transform is an orthogonal set of sawtooth waveforms.

These waveforms are generated via a recursive matrix procedure. The Slant matrix of order two is given by

$$S_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

and the matrix of order four is given by

$$S_4 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & b_4 \\ 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{bmatrix} \cdot \begin{bmatrix} S_2 & \underline{0} \\ \underline{0} & S_2 \end{bmatrix} \quad \text{where} \quad \underline{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where a_4 and b_4 are determined by requiring a linear (constant negative slope) function formed in row two, which yields that $a_4 = 2b_4$. This and the orthonormality condition $SS^T=I$ lead to the following for S_4:

$$S_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix}$$

A second iteration of the process given in [108] results in the eighth order Slant matrix, S_8, as shown in Enomoto and Shibata [109].

$$S_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{\sqrt{7}}{\sqrt{3}} & \frac{5}{\sqrt{3}\sqrt{7}} & \frac{\sqrt{3}}{\sqrt{7}} & \frac{1}{\sqrt{3}\sqrt{7}} & -\frac{1}{\sqrt{3}\sqrt{7}} & -\frac{\sqrt{3}}{\sqrt{7}} & -\frac{5}{\sqrt{3}\sqrt{7}} & -\frac{\sqrt{7}}{\sqrt{3}} \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{3}{\sqrt{5}} \\ \frac{7}{\sqrt{5}\sqrt{21}} & -\frac{1}{\sqrt{5}\sqrt{21}} & -\frac{9}{\sqrt{5}\sqrt{21}} & -\frac{17}{\sqrt{5}\sqrt{21}} & \frac{17}{\sqrt{5}\sqrt{21}} & \frac{9}{\sqrt{5}\sqrt{21}} & \frac{1}{\sqrt{5}\sqrt{21}} & -\frac{7}{\sqrt{5}\sqrt{21}} \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

The Slant transform does not perform as well as the DCT, but is significantly easier to implement. The irrational constants the matrices contain may be stored in a look up table in a hardware implementation [110]. Pratt, et al, develop a reordering of these matrices to yield a signal flow graph from input image values to Slant transform values. This signal flow graph leads directly to a systolic array based, fast, hardware implementation.

Compression

With a compression of 1.0 to 1.5 bpp, the Slant transform produces fine reconstructed images. Use of zonal, adaptive coding of the coefficients, help maintain a %MSE of 1.0% or below. Adaptive techniques provide additional bandwidth reduction of 50% for the same degree of image quality.

The optimum block size for this transform appears to be 8 x 8 or 16 x 16 pixels. As block sizes within the image increase, the performance of the transformation approaches that of the Haar and Hadamard transforms. The Slant transform results in a lower MSE for moderate size image blocks when compared to the Haar and Hadamard transforms [111].

Spatial Domain

The Slant transform reproduces linear variations of brightness quite well. However, its performance at edges is not as optimal as the KLT or DCT. Because of the "slant" nature of the lower order coefficients, its effect is to smear edges. At 1.5 bpp, the Slant transformed image is much more desirable than the Hadamard or Haar transforms.

Temporal Domain

No information was reviewed.

Aesthetic Appearance

High quality are produced images at a compression ratio of approximately one half. But, as the rate of compression increases, the quality of the images drops off. Compared to the Hadamard and Haar transforms, the Slant transform showed almost no block effects.

The Slant technique is relatively tolerant to transmission errors. Such errors tend to extend only to the end of one block. This depends, partly, on the coding method used.

Spectral Information

Colors, when expressed in YIQ terms, seem quite accurate in reconstructed images. At 2.0 bpp, block effects appear most noticeable in areas of low spatial frequency, i. e., backgrounds. Edges in the reconstructed image are slightly less crisp than in the original image. Still, they are quite clearly defined and discernable. Slight variations in hue and saturation within adjacent areas are still preserved. Texture in background elements, e. g., carpet and upholstery, is preserved very well in the reconstructed image. At the same compression, reconstructed color images appear to perform better than monochrome reconstructed images in maintaining texture and fine detail. This suggests using color images when one wants to preserve texture.

Hadamard Transform

The basis vectors of the Hadamard transform comprise a set of orthogonal rectangular waveforms which only assume values of 1 or -1 and are defined over a given spatial interval. These waveforms are known as Walsh functions and the transform has also been called the Walsh transform or the Walsh-Hadamard transform [112].

There are several ways of deriving these rectangular waveforms. The Hadamard basis vectors in natural order can be found using Hadamard matrices. A Hadamard matrix is square and has elements of 1 and -1 only and the rows and columns of the matrix are orthogonal. The lowest order Hadamard matrix is of order two,

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

For N x N matrices where N is a power of 2, H_N can be recursively derived as

$$H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

Since a Hadamard matrix equals its transpose, then the Hadamard transform in two dimensions [113] is given by

$$F_H(u,v) = [H_N]\,[f(x,y)]\,[H_N]$$

and since the inverse of a Hadamard matrix is itself multiplied by a scalar -1/N, then the inverse transform is given by

$$f(x,y) = \frac{1}{N^2}[H_N]\,[F_H(u,v)]\,[H_N]$$

The natural ordered set of basis vectors obtained via Hadamard matrices is not ordered by increasing sequency. Thus, the coefficients must be reordered before techniques such as zonal sampling are applied.

The sequency ordered Hadamard transform may be obtained by Boolean synthesis. Here the transform is given by

$$F_H(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1}[g_i(u)x_i + g_i(v)y_i]}$$

where $g_0(u) = u_{n-1}$; $g_1(u) = u_{n-1} + u_{n-2}$; $g_2(u) = u_{n-2} + u_{n-3}$; ... $g_{n-1}(u) = u_1 + u_0$;

and $u_{decimal} = (u_{n-1}u_{n-2}u_{n-3}\ldots u_1u_0)_{binary}$

and similarly for v.

The additions within the summation in the exponent are performed modulo-2, that is, binary additions with no carry.

One of the most advantageous properties of the Hadamard transform is how the basis vectors assume values of 1 or -1 only. Thus, computing the transform involves only combinations of addition and subtraction of the samples in the image array. The "fast" algorithm for the Hadamard transform involves the least calculations of any transform algorithm (only $N\log_2 N$ additions/subtractions and no long multiplications since all multiplications involves powers of two and can be performed by shift operations) [114].

Another useful property of the Hadamard transform is that the coefficients in the $F(u,v)$ array will either be all even or all odd [115]. This permits further coding efficiency by assigning a bit for even or odd to the whole array and then truncating the least significant bit from each coefficient with no loss of information.

The Hadamard transform is the transform technique most easily implemented in VLSI for real time applications [116][117][118][119]. Rather than zonal or threshold sampling, ranking of coefficients yields the best results [120]. This ranking is determined by spatial and temporal precedence guidelines [121]. A given coefficient should not be transmitted unless those coefficients with higher precedence are also transmitted. The result of not following this procedure is undesired edges or blurring. Logarithmic quantization of coefficients is also desirable [122].

Compression

Despite its computational simplicity, the coding efficiency of the Hadamard transform is significantly below that of the DCT or Slant transforms. Thus, its compression is slightly less than the compression achieved with the previous transform kernels (or basis vectors). As the block size increases above 8 x 8, experiments show a marked increase in the MSE values between this transform and the KLT, DCT, and Slant transforms.

Compression down to the range of 1.0 to 1.5 bpp produce good quality reconstructed images. The %MSE between the original and reconstructed images falls between 1.5% and 1.0%.

Spatial Domain

At equal compression ratios, more distortion appears in a Hadamard transformed image than a cosine transformed image. This applies both to mean square error criteria and subjective evaluation [123]. Mean square error images (%MSE) show how quantization error tends to gather in areas of high spatial frequency such as edges and texture. This indicates poor coding performance. Uncorrelated pixels in MSE images indicate good performance. Quantization errors, which are more localized than previously discussed transforms, do not distort picture elements. Yet, compared to the others, the reconstructed image appears out of focus.

Temporal Domain

By taking advantage of interframe pixel correlation, the Hadamard transform yields a 30% to 100% improvement in compression (expressed as bits per pixel) and %MSE.

Aesthetic Appearance

Some block errors appear as the pixel rate approaches 1.0 bpp. Undesired edges or blurring may result from improper quantization of the coefficients.

The results of Hadamard transform encoding produce visually less objectionable pictures than those resulting from lossy, predictive schemes. The pictures contain fair edge reproduction. However, subjective comparisons of images employing other transforms indicate the Hadamard provides the least desirable images. Experiments, where quantization and bit allocation were the same for all transforms, supports this conclusion. The only part of the implementation which varied was the particular transform algorithm.

Spectral Information

The distortions produced by encoding color images were generally similar to those produced by using monochrome images.

Haar Transform

The basis vectors for the Haar transform are also rectangular waveforms, but they may assume values of A, 0 and -A, where A is a constant which depends on the "order" of the basis vector. The first Haar basis vector, $HAR_0(x)$ in one dimension is a DC value defined over the interval (from 0 to N-1) which the transform is taken. The rest of the set of Haar functions over the same interval can be found by

$$HAR_{2^p+n}(x)=\begin{cases}\sqrt{2^p} & \text{for } \frac{n}{2^p}\le x<\frac{\left[n+\frac{1}{2}\right]}{2^p}\\ -\sqrt{2^p} & \text{for } \frac{\left[n+\frac{1}{2}\right]}{2^p}\le x\le[n+1]2^p\\ 0 & \text{for all other } x\end{cases};$$

and the two dimensional Haar transform is found by

$$F_H(u,v)=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)HAR_u(x)HAR_v(y).$$

The Haar transform represents a measure of locally concentrated differential energy within a sub-picture. The Haar transform is not a "sequency" type transform; thus, zonal sampling is generally not used.

The Haar functions' rectangular shapes make them fairly easy to implement in VLSI circuits, but the scaling factors involving fractional powers of two make the Haar transform more difficult to implement than the Hadamard transform. At the same compression ratio, using the Haar transform over the Hadamard transform saves a small degree of mean-square error.

Compression

Interframe adaptive encoding results in a compression of 0.70 to 1.70 bpp with a %MSE ranging from 0.8% to 0.2%.

Spatial Domain

At maximum compressions, the Haar transform produces accurate reconstructions of monochrome and color images. When the YIQ component coding is utilized, color images retain much texture and detail. The greatest drawback of this transform scheme is its effect on edges. Edges in the reconstructed image show a decided blockiness.

Temporal Domain

Interframe compression yields approximately a 20% improvement in compression when compared to two dimensional DCT, KLT, or Slant transforms.

Aesthetic Appearance

Elements in the image are still discernable; however, the loss of resolution in textured areas proves annoying to many observers. The effect produces an out of focus appearance. Block errors along edges are definitely noticeable at maximum compression rates.

Spectral Information

YIQ component images perform as well as, or even better than, monochrome images. The use of color may help to preserve texture and fine detail.

---

74. Saghri, J.A., Tescher, A.G., Habibi, A., "Block Size Considerations for Adaptive Image Coding," pp. E1.2.1-4, 1982.

75. Essman, J.E., Hua, Q.D., and Griswold, N.C., "Video Link Data Compression for Remote Sensors," *Proceedings of the SPIE*, vol. 87, pp. 55-77, 1976.

76. Ocylok, G., "A Comparison of Interframe Coding Techniques," *Proceedings of ICASSP 83*, pp. 1224-1227, 1983.

77. Natarajan, T.R. and Ahmed, N., "On Interframe Transform Coding," *IEEE Transactions on Communications*, vol. COM-25, pp. 1323-1329, November 1977.

78. Ahmed, N., Natarajan, T., and Rao, K.R., "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90-93, January 1974.

79. Jain, P.C. and Delogne, P., "Real-Time Hadamard Transform Coding for TV Signals," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-23, pp. 103-107, May 1981.

80. Wintz, P.A., "Transform Picture Coding," *Proceedings of the IEEE*, vol. 60, pp. 809-820, July 1972.

81. Habibi, A. and Samulon, A.S., "Bandwidth Compression of Multispectral Data," *Proceedings of the SPIE*, vol. 66, pp. 23-35, 1975.

82. Reininger, R.C. and Gibson, J.D., "Distributions of the 2-D DCT Coefficients for Images," *IEEE Transactions on Communications*, vol. COM-31, pp. 835-839, June 1983.

83. Parsons, J.R. and Tescher, A.G., "An Investigation of MSE Contributions in Transform Image Coding Scheme," *Proceedings of the SPIE*, vol. 66, pp. 196-206, 1975.

84. Narasimhan, M.A., Rao, K.R., and Raghava, "Image Data Processing by Hybrid Sampling," *Proceedings of the SPIE*, vol. 119, pp. 130-136, 1977.

85. Limb, J.O., Rubinstein, C.B., and Thompson, "Digital Coding of Color Video Signals - A Review," *IEEE Transactions on Communications*, vol. COM-25, pp. 1349-1385, November 1977.

86. Limb, J.O., Rubinstein, C.B., and Thompson, "Digital Coding of Color Video Signals - A Review," *IEEE Transactions on Communication*, vol. COM-25, pp. 1349-1385, November 1977.

87. Pratt, W.K., "Spatial Transform Coding of Color Images," *IEEE Transactions on Communication Technology*, vol. COM-19, pp. 980-992, December 1971.

88. Habibi, A. and Samulon, A.S., "Bandwidth Compression of Multispectral Data," *Proceedings of the SPIE*, vol. 66, pp. 23-35, 1975.

89. Habibi, A., "Survey of Adaptive Image Coding Techniques," *IEEE Transactions on Communications*, vol. COM-25, pp. 1275-1284, November 1977.

90. Habibi, A. and Samulon, A.S., "Bandwidth Compression of Multispectral Data," *Proceedings of the SPIE*, vol. 66, pp. 23-35, 1975.

91. Chen, W., Smith, C. H., And Fralick, S., "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communication*, vol. COM-25, pp. 1004-1009, 1977.

92. Mitchell, O.R. and Tabatabai, A., "Adaptive Transform Image Coding for Human Analysis," *Proceedings of ICC '79*, vol. 2, pp. 23.2.1-5, 1979.

93. Burge, R.E. and Wu, J.K., "An Adaptive Transform Image Data Compression Scheme Incorporating Pattern Recognition Procedures," *Proceedings of the Conference on Pattern Recognition & Image Processing*, pp. 230-236, 1981.

94. Reininger, R.C. and Gibson, J.D., "Distributions of the 2-D DCT Coefficients for Images," *IEEE Transactions on Communications*, vol. COM-31, pp. 835-839, June 1983.

95. Mitchell, O.R. and Tabatabai, A., "Adaptive Transform Image Coding for Human Analysis," *Proceedings of ICC '79*, vol. 2, pp. 23.2.1-5, 1979.

96. Chen, W., Smith, C. H., and Fralick, S., "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications*, vol. COM-25, pp. 1004-1009, 1977.

97. Mitchell, O.R. and Tabatabai, A., "Adaptive Transform Image Coding for Human Analysis," *Proceedings of ICC '79*, vol. 2, pp. 23.2.1-5, 1979.

98. Habibi, A., "Survey of Adaptive Image Coding Techniques," *IEEE Transactions on Communications*, vol. COM-25, pp. 1275-1284, November 1977.

99. Habibi, A., "Survey of Adaptive Image Coding Techniques," *IEEE Transactions on Communications*, vol. COM-25, pp. 1275-1284, November 1977.

100. Saghri, J.A. and Tescher, A.G., "Adaptive Transform Coding Based on Chain Coding Concepts," *IEEE Transactions on Communications*, vol. COM-34, pp. 112-117, February 1986.

101. Saghri, J.A. and Tescher, A.G., "Adaptive Transform Coding Based on Chain Coding Concepts," *SPIE Proceedings on Applications of Digital Image Processing VI*, pp. 75-79, August 1983.

102. Tasto, M. and Wintz, P.A., "Image Coding by Adaptive Block Quantization," *IEEE Transactions on Communication Technology*, vol. COM-19, pp. 957-971, December 1971.

103. Queen, G.W., Bryan, J.K. and Gowdy, J.N., "Investigations of Several Discrete Transform Methods for Image Processing Applications," *Proceedings of SOUTHEASTCON '81*, pp. 881-886, 1981.

104. Cottrell, Garrison W., and Munro, Paul, "Principal components analysis of images via back propagation', *SPIE Proceedings of the Conference on Visual Communications and Image Processing '88*, vol. 1001, part 2, pp. 1070-1077, November 1988.

105. Ahmed, N., Natarajan, T., and Rao, K.R., "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90-93, January 1974.

106. Flickner, M.D. and Ahmed, N., "A Derivation for the Discrete Cosine Transform," *Proceedings of the IEEE*, vol. 70, pp. 1132-1134, September 1982.

107. Hou, H.S., "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, pp. 1455-1461, October 1987.

108. Pratt, W.K., Chen, W-H, and Welch, L.R., "Slant Transform Image Coding," *IEEE Transactions on Communications*, vol. COM-22, pp. 1075-1093, August 1974.

109. Enomoto, H. and Shibata, K., "Orthogonal Transform Coding System for Television Signals," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-13, pp. 11-17, August 1971.

110. Reader, C., "Intraframe and Interframe Adaptive Transform Coding," *Proceedings of the SPIE On Efficient Transmission of Pictorial Information*, vol. 66, pp. 108-117, 1975.

111. Gonzales, R., and Wintz, P., *Digital Image Processing*, Addison-Wesley Publishing Co., 2nd Ed, pp. 117-121, November 1987.

112. Gonzales, R., and Wintz, P., *Digital Image Processing*, Addison-Wesley Publishing Co., vol. 2nd Ed, pp. 117-121, November 1987.

113. Pratt, W.K., Kane, J., and Andrews, H.C., "Hadamard Transform Image Coding," *Proceedings of the IEEE*, vol. 57, pp. 58-68, January 1969.

114. Pratt, W.K., Kane, J., and Andrews, H.C., "Hadamard Transform Image Coding," *Proceedings of the IEEE*, vol. 57, pp. 58-68, January 1969.

115. Lyle, Jr., W.D. and Forte, F., "A Useful Property of the Coefficients of a Walsh-Hadamard Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp 479-480, August 1980.

116. Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-18, pp. 28-36, February 1976.

117. Tescher, A.G. and Cox, R.V., "An Adaptive Transform Coding Algorithm," pp. 47.20-3.

118. Jones, R.A., "Adaptive Hybrid Picture Coding," *Proceedings of the SPIE,* vol. 87, pp. 247-255, 1976.

119. Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *Proceedings of the SPIE,* vol. 66, pp. 58-69, 1975.

120. Dillard, G.M., "Application of Ranking Techniques to Data Compression for Image Transmission," *Proceedings of the NTC '75,* vol. 1, pp. 22.18-22, 1975.

121. Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *Proceedings of the SPIE,* vol. 66, pp 58-69, 1975.

122. Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *Proceedings of the SPIE,* vol. 66, pp. 58-69, 1975.

123. Murphy, M.S., "Comparison of Transform Image Coding Techniques for Compression ...," *Proceedings of the SPIE,* vol. 309, pp. 212-219, 1981.

Hybrid Techniques

Discrete Cosine Transform/Vector Quantization

Discrete Cosine Transform/Vector Quantization (DCT/VQ) [124][125][126] involves using VQ on the DCT coefficients. It provides an opportunity to improve upon the compression/quality achievable by either of the techniques individually. According to rate-distortion theory, for a given amount of distortion, a lower rate can be obtained by using vector, rather than scalar coding. Therefore, using VQ on the DCT coefficients will provide better performance than any quantization scheme which treats them as a set of scalars. Also, the statistics of the DCT coefficients can be modeled more easily and with greater accuracy than the statistics of the image itself. This allows for more efficiency in the design of the vector codebooks and less computation time in the implementation of VQ. In some cases the higher sequency coefficients are dropped. This reduces the dimension of the codevectors which significantly reduces the number of computations needed to use VQ.

Compression

Images of 480 x 768 pixels digitized at 8 bpp have been encoded at 0.7 - 0.8 bpp with absolute average error of 5 - 10 levels. This is not high quality. Another method used 1.1 bpp to reproduce an image with no visible distortion. The SNR of DCT/VQ was found to be 1 - 2 dB higher than DCT with scalar quantization for the same bit rate.

Spatial Domain

The types of errors that occur are not easy to predict. Degradations due to VQ would cause errors similar to those produced by using DCT at a low bit rate. The degradations due to throwing away the high sequency coefficients are discussed in the transform section.

Temporal Domain

Not Applicable.

Aesthetic Appearance

Low bit rates (below 0.5 bpp) produce some blockiness.

Spectral Information

Not reported.

Discrete Cosine Transform/Motion Compensation

Television information contains temporal redundancy as well as spatial redundancy. That is, a pixel of one frame is the same as the pixel in the same position in the following frame and therefore need not be transmitted. The pixel description of the earlier frame could be used. The compression techniques addressed thus far, in this report, are directed toward reducing spatial redundancy and lossy compression only in the spatial domain of a single frame. Conditional Replenishment is used, in part, to remove temporal redundancy where corresponding pixels of one frame are the same as a prior frame. In Conditional Replenishment the pixels are separated into two groups, those pixels, called background pixels, which are the same as their corresponding pixel in the previous frame; and those pixels classed as moving area pixels, which are not the same as their corresponding pixel. Only the moving area, changed pixels, and their locations are transmitted. This method can be improved by sending an estimate of the displacements of groups of pixels, such as those that might be representative of a moving object, to provide MC.

In hybrid transforms, interframe DPCM has been combined with MC, such as DCT/DPCM. The approach used in DCT with interframe DPCM is shown in the following figure.

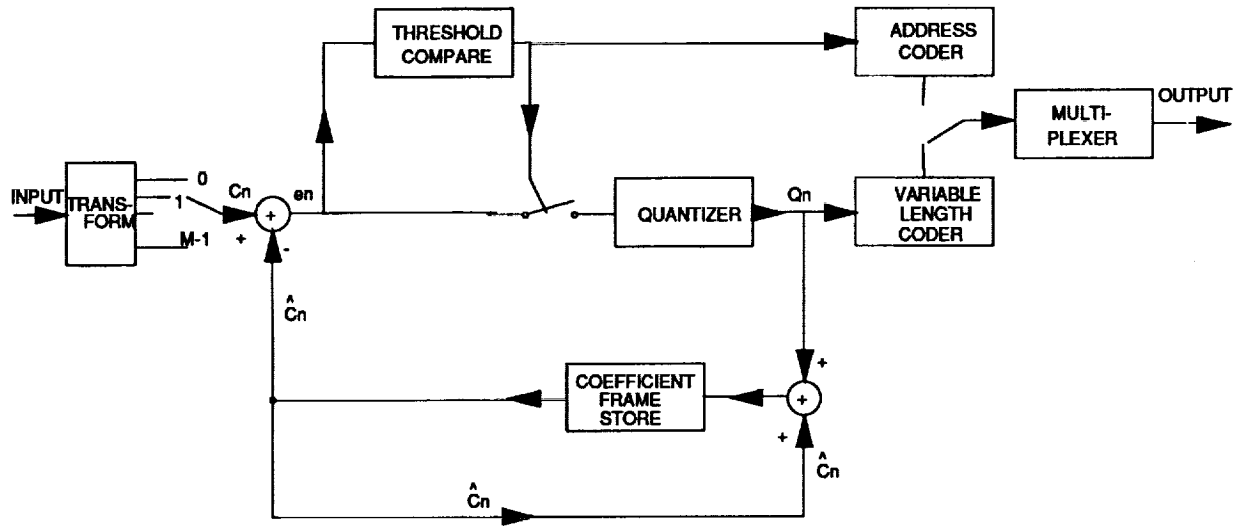


Figure 13: Hybrid interframe Transform/DPCM encoder.

The steps used in the Transform/DPCM interframe coding are

1. Partition the frame into blocks.
2. Take the two dimensional, spatial Discrete Cosine Transform of each block to yield, for each block, a block of transform coefficients.
3. Predict the coefficients of the k^{th} block of the present frame from the corresponding coefficients of the k^{th} block of the prior frame.
4. If the difference is less than a selected threshold, send a zero block, if it is greater send the prediction error.

Compression is achieved by redundancy removal and truncation of the coefficient set using the DCT in the spatial domain. And by redundancy removal and lossy compression using DPCM in the temporal domain. The use of MC can be added to the above process in the following manner:

5. If the difference is less than a selected threshold, send a zero block.
6. If it is greater, search adjacent pixels or coefficients to define spatial displacement between frames. A recursive algorithm using pixel intensities or transform coefficients can be used, or a block matching algorithm can be used to define a displacement vector.
7. Send the prediction error along with the displacement vector.

Compression

The applications of DCT/DPCM with MC, described in the recent literature, are for video conferencing. The resolution of the video test image sequences are not representative of NTSC video images. The motions are limited and more constrained, representing video conferencing material content, showing individual or groups of people engaged in normal conference or meeting activities. In comparison of Transform/Conditional Replenishment without MC, the use of MC provides a 30% to 40% improvement in bit rate [127]. Achieving compressions, for "good" quality pictures judged subjectively, of 0.1 to 0.4 bpp using adaptive DCT [128], and separate coding of pulse-like components which cause slope overload in DPCM [129].

Spatial Domain

As compression become large, 0.1 bpp, block noise and granular noise become more visible. Common distortions of transform coding.

Temporal Domain

The use of MC with DCT/DPCM produces fluctuations in luminance in the region of moving edges. This effect has the appearance of mosquitoes, termed the mosquito effect, and although not large in amplitude, can be very annoying. Moving areas, or objects, cover stationary background in the direction of motion, and uncover background away from the direction of motion. This produces a step change between successive frames, resulting in pulse-like peaks using DPCM. The separate coding of these peaks using Scalar Quantization (SQ) has been used to significantly reduce this distortion [130].

Aesthetic Appearance

Block matching at the edge of the transform blocks, granular noise due to coarse quantization, and the mosquito effect are concentrated in localized regions and therefore are more visible than uniformly distributed errors. MSE are therefore not representative of the subjective effects of the errors.

Spectral Information

Edge blockiness and the mosquito effect result in discontinuities and noise fluctuations in chrominance as well as luminance, resulting in spectral errors in the region of block edges and spectral noise in the region of moving edges.

124. Abdelwahab, A.A. and Kwatra, S.C., "Image Data Compression with VQ in the Transform Domain," *Proceedings of the ICC '86,* vol. 2, pp. 1285-1289, 1986.

125. Saito, T., et al., "Adaptive DCT Image Coding Using Gain/Shape Vector Quantizers," *Proceedings of the ICASSP '86,* pp. 129-132, 1986.

126. Marescq, J.P. and Labit, C., "Vector Quantization in Transformed Image Coding," *Proceedings of the ICASSP '86,* pp. 145-148, 1986.

127. Netravali, A.N. and Stuller, J.A., "Motion-Compensated Transform Coding," *Bell System Technical Journal,* vol. 58, pp. 1703-1718, September 1979.

128. Kato, Y., Mukawa, N., and Okubo, S., "A Motion Picture Coding Algorithm Using Adaptive DCT Encoding Based...," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1090-1099, August 1987.

129. Kaneko, M., Hatori, Y., and Koike, A., "Improvements of Transform Coding Algorithm for Motion-Compensated ...," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1068-1077, August 1987.

130. Kaneko, M., Hatori, Y., and Koike, A., "Improvements of Transform Coding Algorithm for Motion-Compensated ...," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1068-1077, August 1987.

HIGH SPEED IMPLEMENTATIONS OF VDC ALGORITHMS

The literature on video data compression techniques consists mostly of papers detailing the development or improvement of specific techniques. Usually, the paper includes experimental results comparing the compression ratio and image degradation of the new technique with previously existing ones. The experiments are generally performed by implementing the compression techniques in software and applying them to standard test images. The speed at which the compression can be performed is not addressed as a major concern, and it is seldom discussed at all. Many papers, addressing the practical implementations of VDC techniques for specific applications, jump immediately to hardware implementations. Some of the implementations involve designs of chips for specific tasks. Others use standard digital signal processing (DSP) chips. Using today's technology, VDC software written for a general purpose microprocessor are not fast enough for real-time applications. Therefore, software applications will not be considered any further.

Some representative examples of hardware implementations of high speed video data compression techniques are presented in the Table of Hardware Implementations at the end of this section. Most of these are implementations of predictive or transform techniques. The applications for which the techniques have been implemented generally involve television signals at video rates (30 frames/sec). In terms of pixel rate, "video rate" covers a large range. The actual pixel rate depends on the method of digitization as well as the frame rate. The highest standard throughput for digitized video signals is 14.32 Mp/s for an NTSC signal sampled at 4 fsc. Other standards produce rates in the range of 7.8 - 13.5 Mp/s. Some of the implementations can be used at higher rates than were reported, but the developers did not try to do so because the practical applications did not exist.

One-dimensional DPCM implementations have produced output of 3 - 5 bpp with input rates of 10 - 14.3 Mp/s. Two-dimensional DPCM can compress an image down to 3 - 4 bpp at input rates as high as 10.7 Mp/s. Interframe DPCM has achieved compression to 1.6 bpp at 10.6 Mp/s.

Separable N x N transforms can be implemented using a technique known as row/column decomposition. This technique involves performing N 1 x N transforms, a matrix transposition (accomplished by careful addressing of RAM), and another N 1 x N transforms. Most of the popular transforms are separable and also have "fast" implementations which reduce the number of arithmetic operations. These two characteristics make simple, high speed hardware implementations feasible.

Hadamard transform techniques have been implemented to produce 2 bpp at 8.1 Mp/s and 0.5 bpp at 1.8 Mp/s. Also, a system using a pipeline fast Hadamard transform configuration has been implemented at 128 Mp/s [131].

DCT techniques can achieve 1.6 bpp at 10.4 Mp/s and 0.82 bpp at 9.7 Mp/s. There have also been DCT chips produced which perform the two-dimensional transform at 14.3 Mp/s and above. A chip architecture to implement DCT at data rates up to 27 Mp/s (the combined video rate of component coded television, CCIR Rec. 601) has been proposed. In neither of these cases was a quantizer for the transform coefficients included, so neither is a complete compression implementation.

A proposed technique performs discrete unitary transforms (DCT, DFT, etc.) optically with incoherent light by using acousto-optic spatial modulators and a CCD camera for detection and storage [132]. This technique can perform two-dimensional transforms of entire images at "video rate" since the speed is limited by the CCD camera. Performing many smaller transforms would require many modulators and light sources in parallel. Therefore, this implementation would be most practical for performing transforms on large blocks or entire images. With the appropriate bit-allocation schemes, using larger blocks can produce better compression according to information theory. This technique is not a focal plane implementation since row/column decomposition is used. The second one-dimensional transform uses the output of the first one-dimensional transform which is fed back from the camera to a spatial modulator.

Optical transforms can be performed using the electro-optical spatial modulators and CCD camera if the transform is separable. Current parallel scan architecture of the HHVT system does limit the usefulness of this approach if large blocks are to be considered.

An edge detection technique that produces a 1 bpp edge map of the image was implemented at 10 Mp/s. Other types of filters have been developed that run at clock rates of 30 MHz and above and process images at rates of 5 - 10 Mp/s.

A parallel processing implementation of BTC has been proposed. It is based on early 1980's technology and would compress about 0.5 Mp/s per processing element (PE). The output would be 1.625 bpp. With today's technology it is likely that such a technique could be implemented at a higher data rate, thereby allowing a high-speed implementation without an unreasonable number of PEs.

A hardware implementation of a vector quantization technique is planned. It would operate at 11.8 Mp/s. No indication was given of the planned compression ratio.

Recently, experimental work has been done on the compression of HDTV signals which use about five times the bandwidth of conventional television signals. Various compression techniques have been developed. None of these techniques attempt digital compression at the full Nyquist rate. They use sub-sampling or analog filtering to reduce the pixel rate before applying digital compression algorithms. A previous pixel DPCM coder has been developed that operates at 16 MHz and produces 5 bpp. Another DPCM technique has been demonstrated at 16.2 Mp/s [133][134].

TABLE OF HARDWARE IMPLEMENTATIONS OF VDC ALGORITHMS						
METHOD	YEAR	INPUT RATE	CLOCK RATE	OUTPUT RATE	GENERAL DESCRIPTION OF METHOD	REFERENCE
p = proposed d = designed		(Mp/s)	(MHz)	(bpp)		
2-D HT	1976	8.06		2.0 / 1.0	transf is 2 bpp - reduc to 1 b by 3-mode adapt incl f-f diff & fr repeat	Jones
1-D DPCM	1977	10.00		3	fixed pred., tapered quantizer	McClung
2-D Hadamard chip	1979				CID image plane trnsform, 128x128, 256x256	Camana
2-D WHT	1979	1.84	2.5	0.5 - 2.0	8x8 WHT w/ variable quant., digital CCD, bit-serial input	Spencer
2-D DCT	1981	9.70		0.82	CCD, 8 samples in parallel input, 8x8 DCT	Ludington
BTC p	1982	0.52/PE	10	1.625	4x4 BTC using parallel proc. - 80 Mp/s would require 153 procs.	Mudge
2-D transform p	1984	video rate			optical incoherent unitary X-form, rate limited by CCD camera	Lebreton
edge detection	1984	10.00		1	2x2 Roberts operator w/ thresholding, produces 1 bpp edge/no edge picture	McIlroy
2-D DPCM	1985	14.00		1,2,3, or 4	2-D fixed pred., quant., can be made adapt or lossless by adding modules	Jain
DPCM - simple	1985	10.00		4		Kretz
DPCM - complex	1985	10.00		3		Kretz
2-D DPCM	1986	7.25		3	non-linear adapt pred, adapt quant	Driessen
2-D HT	1986	7.25		2.63	4x4, adapt quant	Driessen
DPCM / MC	1986	8.40		~0.04	adapt prev pixel/prev fr pred w/MC(BMA), adapt quant/subsmpl, entropy cod	Koga
vector quant. p	1986	11.80			systolic array architecture	Ramamoorthy
2-D DCT	1986	0.04		1	8x8 block, fixed bit alloc. & Max quant, simulated for TMS32010 DSP chip	Srinivasan
2-D DCT p	1986	7.86			blocks up to 128x64, using 64 TMS32010s + ext HW, faster w/ more chips	Srinivasan
2-D filter	1986	7.86	up to 8.85	N/A	independent of filter order	Ty
2-D filter	1987		33	N/A	LSI, up to 9th order filter, for HDTV processing	Abe
2-D DPCM	1987	10.70		4	2-D fixed predictor, fixed quantizer	Brainard
2-D DCT chip	1987	10.00	10		16x16, row/column decomp., analog CCD, 7-bit accuracy, 1-D in 100 ns	Chiang
transform chip	1987	6.15			performs any separable linear transform by storing constants in RAM	Heiman
2-D DCT chip p	1987	27.00	0 - 27		8x8 block, pixel-serial I/O, quantization not included	Leger
3-D DPCM	1987	10.56		1.6	adaptive prediction, adaptive quant., motion comp, CR, entropy coding	Murakami
2-D filter p	1987	.015/PE	10		3x3 adapt l.p. filter, parallel proc., 1 PE/pixel, VLSI	Papananos
2-D DCT chip p	1987	14.32	up to 30's		8x8, 16x16, or 32x32	Sun
2-D DCT chip d	1987	14.32	14.32 +		16x16 DCT, row/col decomp, can run faster?, design checked w/ simulators	Sun
3-D DPCM	1988	10.13		1.8	adaptive pred. w/ motion comp.	Gaggioni
DCT	1988	10.43		1.6	8x8 block, adaptive scaling of coefficients, entropy coding	Gaggioni
3-D DPCM	1988	10.43		4.8 - 5.8	fixed pred., adaptive quant. to maintain required bit rate	Gaggioni
1-D DPCM	1988	13.00	16	5	previous pixel pred.	Gaggioni
1-D DPCM	1988	14.32	14.32	4	previous pixel pred., 15-level fixed Max quantizer	Jalali

131. Noble, S.C., "A Comparison of Hardware Implementations of the Hadamard Transform...," *Proceedings of the SPIE,* vol. 66, pp. 207-11, 1975.

132. Lebreton, G., "Image Data Compression Using Unitary Transforms: Video-Rate Optical...," *Proceedings of the SPIE,* vol. 492, pp. 278-283, 1984.

133. Gaggioni, H. and Le Gall, D., "Digital Video Transmission and Coding for the Broadband ISDN," *IEEE Transactions on Consumer Electronics,* vol. CE-34, pp. 16-34, February 1988.

134. Hopkins, R., "Advanced Television Systems," *IEEE Transactions on Consumer Electronics,* vol. CE-34, pp. 1-15, February 1988.

USER REQUIREMENTS CASE STUDIES

Background

Baseline Near-term HHVT Video System

Some of the significant specifications of this system are

1. Monochrome only.
2. 1024 x 1024 pixels maximum resolution of the sensor.
3. Sub-framing capability with pixels addressable in 8 x 8 blocks.
4. 8-pixel parallel scan format.
5. 80 Mp/s (8×10^7 pixels/second) maximum sensor pixel addressing rate, i.e. (pixels/frame)(frames/sec) $\leq 8 \times 10^7$.
   Therefore, the maximum frame rate at full (1024 x 1024) resolution is $(8 \times 10^7 / 220) = 76.29$ fr/s. The resolution of each frame can be traded off for a higher frame rate.
6. Each pixel can be coded in 1, 2, 4, or 8 bits/pixel.
   The maximum data output rate from the sensor is 640 Mbps. (1 Mbps = 10^6 bits per second)
7. 512 Mb (229 bytes) dynamic RAM with data transfer rate of up to 1160 Mbps.
8. 99 GB magnetic tape recorder with data transfer rate of up to 240 Mbps (MIL-STD-2179).

Communications Link Capabilities

The following downlink data rates are derived from "An Investigation of Available Communications Link Capabilities for Space Experiments Employing HHVT" which was submitted by Analex Corporation on February 1, 1988. This report contains more detailed explanations of the scenarios considered.

1. Short Transmissions

If the data is transmitted over a period of time comprising a small fraction (less than 1/10) of an orbit, it should be scheduled for a section of the orbit for which there is TDRSS coverage. In this situation the best and worst case downlink rates for the various vehicles are

A. Space Shuttle	
1. Best case - no time sharing	48 Mbps
2. Worst case - 5-way time sharing	9.6 Mbps
B. Spacelab / Space Shuttle	
1. Best case - no time sharing or multiplexing	48 Mbps
2. Worst case - 2-way time sharing and 16:1 multiplexing	1.5 Mbps
C. Space Station Freedom	
1. Best case - time sharing where HHVT has the necessary block of time reserved	50 Mbps
2. Worst case - 6:1 multiplexing	7.0 Mbps
D. USLab / Space Station Freedom	
1. Best case - time sharing where HHVT has the necessary block of time reserved	75 Mbps
2. Worst case - 16:1 multiplexing	4.0 Mbps

2. Long Transmissions

If the data is to be transmitted over a period of time comprising a significant fraction of an orbit or more, the coverage of the carrier by TDRSS, as well as antenna blockage, must be included in the link availability calculations. In this situation the best and worst cases are:

A. Space Shuttle	
1. Best case - 90% coverage	43.2 Mbps
2. Worst case - 5-way time sharing, 52% coverage, rendezvous activity	2.4 Mbps
B. Spacelab / Space Shuttle	
1. Best case - 90% coverage	43.2 Mbps
2. Worst case - 2-way time sharing and 16:1 multiplexing, 52% coverage, rendezvous activity	0.375 Mbps
C. Space Station Freedom	
1. Best case - 6-way time sharing, 85% coverage	7.1 Mbps
2. Worst case - 6:1 multiplexing, 85% coverage	6.0 Mbps
D. USLab / Space Station Freedom	
1. Best case - 10-way time sharing, 85% coverage	6.4 Mbps
2. Worst case - 16:1 multiplexing, 85% coverage	3.4 Mbps

Experiment #102 - Solid Surface Combustion

General Description of Experiment and Image Content

A piece of ashless filter paper is ignited via a hot wire in an O_2/N_2 environment and burns. The propagating flame is recorded as it moves across the paper.

Information to be Derived from the Video Record

Edge of flame, color.

Video System Requirements

1. 2 views, color, 256 intensity levels
2. Resolution: measure dimensions of 0.02 cm to within 10% in a field of view of 10 cm x 5 cm $\rightarrow$ 5000 x 2500 pixels
3. Frame rate: 64 fr/s for 3 min
4. Runs per flight: 3

Required Data Acquisition and Storage

(64 frames/second)(180 seconds) = 11,520 frames

(11,520 frames)(5000 x 2500 pixels/frames) = 144 Gigapixel

(144 Gigapixel)(3 Byte/pixel) = 4.32×10^{11} B required storage/view/run

(4.32×10^{11} B)(3 runs)(2 views) = 2.59×10^{12} B required storage/flight

Baseline Data Acquisition and Storage

The baseline sensor will be monochrome only, with a maximum resolution of 1024 x 1024 pixels.

(64 frames/second)(180 seconds) = 11,520 frames
(11,520 frames)(1024 x 1024 pixels/frame) = 12,080 Megapixels
(12,080 Megapixels)(1 Byte/pixel) = 1.208×10^{10} B storage/view/run

This is much more data than can be stored in the dynamic RAM, so magnetic tape storage is needed. The use of the magnetic tape recorder imposes a maximum data storage rate of 240 Mbps. The possible approaches to compressing the data for storage will be discussed later. In the meantime the data storage can be calculated as follows.

(240 Mbps)(180 s) = 43.2 Gb = 5.40×10^{9} B storage/view/run
(5.40×10^{9} B)(3 runs) = 1.62×10^{10} B storage /view /flight

This total is less than the capacity of one reel of magnetic tape.

Required Downlink Rate

The data from each run must be transmitted within 12 hours.

(4.32×10^{11} B/view)(2 views) = 8.64×10^{11} B = 6912 Gb
(6912 Gb) / [(12 hr)(3600 sec/hr)] = 160 Mbps

Baseline Downlink Rate

The data from each run must be transmitted within 12 hours.

(43.2 Gb/view)(2 views) = 86,400 Mb
(86,400 Mb) / [(12 hr)(3600 sec/hr)] = 2.0 Mbps

Data Compression Requirements

The baseline system without data compression falls short of meeting the acquisition and storage requirements of this experiment in a number of areas. Therefore, video data compression must be considered. Since the baseline system does not provide color information, the remaining important features to be derived from the image are the intensity of the flame and its edges. Therefore, any compression technique to be considered should preserve, as much as possible, edge information and edge location.

The first area in which the baseline system is inadequate is resolution. The limit of 1024 x 1024 pixels is below the required resolution by a factor of 5 in one dimension and 2.5 in the other. An optical system containing Fourier transform optics implemented in front of the sensor might be able to use the 1024 x 1024 available pixels to record high spatial frequency information at the expense of low frequency information in order to allow the detection of objects beyond the original resolution of the sensor. However, this type of system does not seem to be very practical for this experiment since low frequency information is also necessary. An alternate approach is to reduce the field-of-view to 2 cm x 1 cm combined with electro-mechanical tracking of the image. This would permit direct use of the RAM.

The second issue involves the storage of the video data. As mentioned above, the amount of data which needs to be stored precludes the use of dynamic RAM for storage. Even using the RAM as a buffer for magnetic tape storage will not suffice. Therefore, the data cannot be stored at a rate exceeding the maximum data transfer rate of the magnetic tape unit, 240 Mbps. The sensor, however, is producing data at the rate of

(1,048,576 pixels/fr)(64 fr/s) = 67.1 Mp/s.

In order to take full advantage of the data acquisition rate of the sensor, all of the data produced by the sensor must be compressed to 240 Mbps in real time. In other words, the image must be compressed from 8 bpp to (240 / 67.1) = 3.58 bpp at an input rate of 67.1 Mp/s. The entropy of the experiment images is expected to exceed 3.58 bpp during the time of experiment execution. It may be possible to use a lossless technique. However, without specific knowledge of the image statistics, it is not possible to determine if the RAM (with a storage capacity of 1/2% of the total data storage) can be used as a buffer for a lossless technique. Therefore, an analysis of the choice of a lossy technique follows.

The specifications of the compression technique are high speed, relatively low compression, implemented between the sensor and the tape recorder. These requirements point to using a straightforward technique that can be implemented in the parallel pipeline. As discussed above, two-dimensional predictive and transform techniques are appropriate.

Predictive techniques tend not to accurately preserve the location of edges because of slope overload. Also, the output bit rate of transform techniques can be adjusted more easily than that of predictive techniques. Therefore, a transform technique is the most appropriate for this application. It can provide the necessary compression in real time while preserving the location of the edges fairly well.

An alternate approach would be to use a sub-frame of 800 x 800 pixels with a reduced field-of-view to maintain resolution. This would require electronic or electro-mechanical tracking of the burning region, either of which would be technically feasible.

The third area needing compression is downlink transmission. The downlink time will cover many orbits, so the "long transmission" bit rates should be used in this analysis. Although the baseline system requires a downlink rate of only 2.0 Mbps, running this experiment on a Spacelab mission (which is where it is presently proposed to run) might require significantly lower data rates, depending on the requirements of the other experiments. In the worst case the data would have to be compressed by an additional factor of 5.33 to 0.375 Mbps (0.67 bpp). This compression can be done at a comparatively slow downlink rate (1 Mp/s) which will allow more complex techniques to be considered.

A significant amount of compression could likely be obtained from interframe prediction, with or without motion compensation.

Based on a comparison of the above techniques, it is recommended that the two-dimensional fast Hadamard transform be considered to achieve the compression from 8 bpp to 3.58 bpp. To achieve the additional compression of 5.33 it is recommended that interframe prediction in the transform domain without motion compensation be used, followed by run-length entropy encoding.

Experiment #228 - Bubble-in-Liquid Mass Transport Phenomena

General Description of Experiment and Image Content

A bubble of gas is injected into a liquid under controlled pressure conditions. The pressure is adjusted to maintain an unstable equilibrium between the bubble and the surrounding liquid. At some point the pressure is increased, and the bubble begins to dissolve.

Information to be Derived from the Video Record

The precise diameter of the bubble is measured in each image. The diameter measurement is resolved to two microns.

Video System Requirements

1. 2 views, monochrome, 256 gray levels.
2. Resolution:
   A. Desired
      Resolve bubble diameter to 0.002 mm in a field of view of 4 mm x 4 mm (obtained by zooming from full field of 1.5 cm x 1.5 cm) - To accomplish this, we need 2000 x 2000 pixels.
   B. Acceptable
      1024 x 1024 pixels (baseline maximum resolution)
3. Frame rate:

A. Injection	1000 fps	for 1 s	(Desired)
	100 fps	for 1 s	(Acceptable)
B. Equilibrium	1 fps	for 300 s	[480 sec]*
C. Initiation	1000 fps	for 1 s	(Desired)
	100 fps	for 1 s	(Acceptable)
D. Dissolution	1 fps	for 300 s	[1800 sec]*

* from Experiment Timeline

4. Runs per flight: 4

Required Data Acquisition and Storage

A. Desired

(1000 fr/s)(2 s) + (1 fr/s)(2280 s) = 4280 fr

(4280 fr)(2000 x 2000 p/fr) = 17,120 Mp

(17,120 Mp)(1 Byte/p) = 1.71×10^{10} B req. storage/view/run

(1.71×10^{10} B)(2 views)(4 runs) = 1.37×10^{11} B req. storage/flight

B. Acceptable

(100 fr/s)(2 s) + (1 fr/s)(2280 s) = 2480 fr

(2480 fr)(1024 x 1024 p/fr) = 2.60 Gp

(2.60 Gp)(1 Byte/p) = 2.60×10^{9} B req. storage/view/run

(2.60×10^{9} B)(2 views)(4 runs) = 2.08×10^{10} B req. storage/flight

Baseline Data Acquisition and Storage

Since the sensor is being used at full resolution, the frame rate is limited to 76.29 fps (see Background above).

(76.29 fr/s)(2 s) + (1 fr/s)(2280 s) = 2433 fr
(2433 fr)(1024 x 1024 p/fr) = 2.55 Gp
(2.55 Gp)(1 Byte/p) = 2.55 x 10^9 B storage /view /run

Since the capacity of the RAM will be only 512 Mb, storage on magnetic tape will be required. Although the data rate of the tape drive will be limited to 240 Mbps, using the RAM as a buffer will allow all of the above data to be captured at the given frame rates, since the 76.29 Mb/s capture is for just one second at a time. Therefore, the storage in the buffer only reaches about 50 Mb before the input rate drops to 1 Mb/s, and the buffer can be emptied over the next few seconds, as shown in Figure 14.

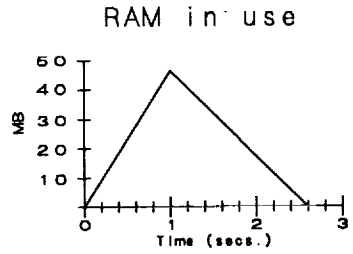


Figure 14: Dynamic RAM buffer capacity for Experiment #288

The storage capacity required for this experiment is
(2.55 x 10^9 B)(4 runs) = 1.02 x 10^{10} B storage /view /flight.

This total is within the capacity of one reel of magnetic tape.

Required Downlink Rate

Approximately 1 fps (almost real time) for up to 2 minutes for ground control (position, zoom) of sensor.

A. Desired
(1 fr/s)(2000 x 2000 p/fr) = 4.0 Mp/s
(4.0 Mp/s)(8 bpp) = 32.0 Mbps

B. Acceptable
(1 fr/s)(1024 x 1024 p/fr) = 1.049 Mp/s
(1.049 Mp/s)(8 bpp) = 8.39 Mbps

Baseline Downlink Rate

(1 fr/s)(1024 x 1024 p/fr) = 1.049 Mp/s
(1.049 Mp/s)(8 bpp) = 8.39 Mbps

Since the requirements are for almost real time downlink for a short period of time, we will assume the downlink is taking place during a period of time in which the carrier vehicle (Space Shuttle or SS Freedom) has TDRSS coverage.

Data Compression Requirements

The baseline HHVT system almost meets the video image acquisition and storage "acceptable" requirements of this experiment without the need for data compression. The only exception is the reduction in frame rate from 100 fps to 76.29 fps for the injection and initiation phases of the experiment. Since the frame rate is limited by the pixel addressing rate of the sensor, not much can be done about it.

The downlink bpp requirement varies greatly depending on the vehicle and the number of other experiments sharing the Ku-band link. Most likely, no compression will be necessary for the downlink. In the worst case the image might have to be compressed to 1.5 Mbps, or 1.43 bpp. Since the only feature of interest in the transmitted image is the boundary of the bubble, a reasonable approach to compression is to transmit only the edges. This can be done by using an edge detection technique

and transmitting only the locations of pixels classified as edges, with or without the intensity values. A technique of this type will easily be able to achieve the necessary compression ratio which is less than 6:1.

There are several techniques that could be very useful for this experiment. Part of the synthetic highs technique [135] can be used by extracting and transmitting only the edge information. The method of Robinson [136] uses two-dimensional directional mask operators to extract edges. The edge pixels are then chain coded for transmission. Although either method would probably be sufficient for this experiment, the latter one is more advanced and will likely produce a better edge picture at a lower bit rate. Both of these techniques involve filtering by convolution which uses many multiplications per pixel, but high-speed compression is not required for downlink transmission because the input rate is only 1.05 Mp/s. Another technique that will provide high-speed edge detection involves the intensity-dependent spatial summation (IDS) operator [137]. This technique is being implemented in hardware for Langley Research Center [138].

Based on a comparison of the above techniques and projected development status, it is concluded that a two-dimensional directional mask operator or IDS operator should be used to extract edge maps for compressed data transmission.

---

135. Schreiber, W.F., Knapp, C.F., Kay, N.D., "Synthetic Highs - An Experimental TV Bandwidth Reduction System," *Journal of SMPTE*, vol. 68, pp. 525-537, August 1959.

136. Robinson, G.S., "Detection and Coding of Edges Using Directional Masks," *Proceedings of the SPIE* (also in *Optical Engineering*), vol. 87, pp. 117-125, 1976.

137. Cornsweet, T.N. and Yellott, Jr., J.I., "Intensity-Dependent Spatial Summation," *Journal of the Optical Society of America A*, vol. 2, pp. 1769-1786, October 1985.

138. Huck, F.O., "Local Intensity Adaptive Image Coding," *NASA Data Compression Workshop*, 1988.

Experiment #230 - Nucleate Pool Boiling

General Description of Experiment and Image Content

Freon is heated locally by means of a large current passed through a thin gold coating on quartz. At some point the freon begins to boil. Vapor bubbles form, grow, and depart from the surface.

Information to be Derived from the Video Record

Nucleation site density
Nucleation frequency at a given site
Bubble shape
Bubble growth, collapse, departure, motion after departure
Existence of fluid micro-layer underneath bubble

Video System Requirements

1. monochrome, 10 - 20 gray levels (we will assume 16 levels since this would take full advantage of 4 bpp)
2. Resolution: resolve an object of dimension 0.005 inch in a field of view of 2.5 inch x 5 inch - To accomplish this, we need 500 x 1000 pixels. (We will use 512 x 1024.)
3. Frame rate: First 6 sec (avg.): 1000 fps desired, 100 fps acceptable. Next 120 sec: 10 fps
4. Runs per flight: 9

Required Data Acquisition and Storage

A. Desired

(1000 fr/s)(6 s) + (10 fr/s)(120 s) = 7200 fr
(7200 fr)(1024 x 512 p/fr) = 3.77 Gp

A 16-gray-level image requires 4 bpp (0.5 bytes/pixel).

(3.77 Gp)(0.5 bytes/p) = 1.89×10^9 B req. storage/run
(1.89×10^9 B)(9 runs) = 1.70×10^{10} B required storage/flight

B. Acceptable

(100 fr/s)(6 s) + (10 fr/s)(120 s) = 1800 fr
(1800 fr)(1024 x 512 p/fr) = 944 Mp
(944 Mp)(0.5 bytes/p) = 4.72×10^8 B req. storage/run
(4.72×10^8 B)(9 runs) = 4.25×10^9 B required storage/flight

Baseline Data Acquisition and Storage

Sub-imaging will allow the sensor to produce 152.6 fps since each frame is only 0.524 Mp.

(152.6 fr/s)(6 s) + (10 fr/s)(120 s) = 2115 fr
(2115 fr)(1024 x 512 p/fr) = 1109 Mp
(1109 Mp)(0.5 bytes/p) = 5.55×10^8 B storage per run

Since the capacity of the RAM will be only 512 Mb, storage on magnetic tape will be required. Although the data rate of the tape drive will be limited to 240 Mbps, using the RAM as a buffer will allow all of the above data to be captured at the given frame rates, since the 76.29 Mb/s capture is only required for 6 seconds. Therefore, the storage in the buffer only reaches about 300 Mb before the input rate drops to 5 Mb/s, and the buffer can be emptied over the next several seconds, as shown in Figure 15.

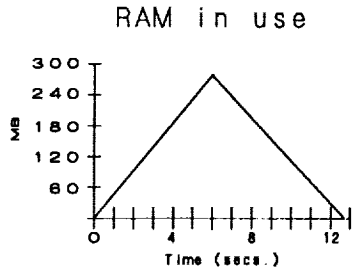


Figure 15: Dynamic RAM buffer capacity for Experiment #230

$(5.55 \times 10^8 \text{ B})(9 \text{ runs}) = 4.99 \times 10^9 \text{ B}$ storage/flight

This total is less than the capacity of one reel of magnetic tape.

Required Downlink Rate

No downlink requirements.

Baseline Downlink Rate

No downlink requirements.

Data Compression Requirements

The baseline HHVT system meets the video imaging "acceptable" requirements of this experiment without the need for data compression. The only area in which there is room for improvement is the frame rate. Although the "acceptable" frame rate is exceeded, the "desired" frame rate of 1000 fps cannot be approached. Since the frame rate is limited by the pixel addressing rate of the sensor, only an optical compression technique that is implemented in front of the sensor could help. This technique would have to reduce the number of pixels required to achieve the necessary resolution, thereby allowing the sensor to trade fewer pixels per frame for a higher frame rate.

INTEGRATION OF VDC INTO HHVT SYSTEM

Since the goal of the HHVT system is to provide as much scientific information as possible, video data compression (VDC) should preserve the salient features of the image. In addition, the HHVT system must meet storage and downlink limitations. Therefore, we need to consider two stages of compression, one before storage on tape and one for downlink. Whenever the data acquisition rate is above the magnetic tape storage rate (30 Mb/s), the data should be initially compressed only enough so that it can be stored on tape. If further compression is required for downlink transmission, a second compression algorithm can be applied. Finally, much of the compression, especially before storage, must occur in real- or near-real-time. Hardware implementations of VDC algorithms deserve serious consideration.

Several features of the baseline HHVT system's design suggest prominent points for the integration of VDC. HHVT will employ eight parallel lines out of the sensor. This architecture suggests the use of a parallel processing architecture, especially in the first stage of compression before storage. If the sensor's output is a nominal 80 Mp/s, each line could process at 10 MHz. This is within the capability of current hardware video processing technology.

Some current hardware-based VDC implementations involve predictors like DPCM and DM. An predictive algorithm which has much promise for VDC is the Lempel-Ziv-Welch algorithm [139][140]. While the algorithm requires buffering for a variable output bit rate and a code table [141], it can provide either lossy or lossless compression. Good quality images can be reconstructed at 1.24 bpp. Using DCT coefficients, the LZW algorithm yields compression ratios of 16:1 [142]. Linear filtering (convolutions using DFTs) occurs at high processing speeds through the use of DSPs. Also, DSPs have evolved into graphics system processors such as Texas Instruments' TMS34010 and 34020 devices, National Semiconductor's Advanced Graphics Chip Set, the AMD 95C60, the Intel 82786, and Hitachi's 63484, among others. These processors, operating at rates up to 60 MHz, are optimized to perform graphic and image processing functions. These processors perform block operations, such as matrix rotations, in a single operation. They also work with floating point number accuracy. Another area of great potential for HHVT is the success with application specific integrated circuits or ASICs. ASICs are semiconductor devices, often in GaAs or CMOS, which use basic logic elements, special memory cells, adders, and multipliers integrated onto a single die or chip [143]. The processing speed of these devices is often the fastest for any particular hardware implementation of a VDC algorithm. DPCM, DCT, MC, and DFT operations are a few of the algorithms which have been successfully implemented.

The following two sections examine some of the unique VDC requirements of HHVT. They also address in more detail the implementation of VDC into the system. The first section examines electronic, or hardware, implementations. The second section examines the use of optical filtering to improve high spatial frequency response and edge preservation in VDC.

---

139. Ziv, J. and Lempel, A., "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. IT-23, No. 3, pp. 337-343, May 1977.

140. Welch, Terry, "A Technique for High-Performance Data Compression," *Computer,* pp 8-19, June 1984

141. Nelson, Mark R., "LZW Data Compression," *Dr. Dobb's Journal of Software Tools and Programming Techniques,* vol. 14, no. 10, pp. 29-36, October 1989.

142. Lewis, Jr. H. Garton and Forsyth, William B., "Hybrid LZW Compression," Proceedings of the NASA Langley sponsored *International Workshop on Visual Information Processing for Television and Telerobotics,* Williamsburg, Virginia, May 1989.

143. Pirsch, Peter, "VLSI Implementation for Visual Communications - Tutorial Short Course Notes," *SPIE Symposium on Optical and Optoelectronic Engineering,* vol. T5, November 1988.

Electronic Implementations

The baseline system's sensor will have an 80 Mp/s output rate. Hardware implementations are likely to require some form of parallel processing. DPCM using a one-dimensional predictor can be implemented in the parallel pipeline of the baseline system. Eight processors, each operating at 10 Mp/s, can compress the eight parallel bit streams emerging from the sensor. However, as indicated in the discussion of predictive methods, vertical edges will be poorly represented by a horizontal predictor.

It may be possible to implement a two-dimensional predictor for seven out of the eight bit streams by delaying each bit stream by one pixel more than the stream above it. Doing so will allow the current predictor to use the pixel that was originally directly above the current pixel. This also could be implemented at 10 Mp/s per processor. Three-dimensional predictive schemes clearly cannot be implemented in the parallel pipeline since they require an entire frame to be stored in RAM for use by the predictor.

The Hadamard transform, although it produces suboptimal compression when compared to the DCT, has a great advantage in its speed of implementation. This is because the computation of the transform involves only additions without multiplications. A pipeline implementation of a two-dimensional Hadamard transform can operate at very high rates. The speed of such an implementation is limited by whichever is slower: one addition operation or the access time of the RAM. The RAM stores the matrix of the intermediate results produced by the first stage of one-dimensional transforms. Two one-dimensional transforms determine the two-dimensional transform. With today's high-speed electronics, it should be possible to implement a Hadamard transform processor in the parallel pipeline of the HHVT system. It would perform an 8 x 8 transform, taking as input 8 groups of 8 parallel pixels, and producing 8 groups of 8 transform coefficients. Thus, it could process the image at a data rate of 80 Mp/s using the system clock rate of 10 MHz. The delay in the line for computation of the transform would be about 10 clock periods (1 μs).

Two-dimensional DCTs can be performed on parallel data quite easily. However, total throughput is usually no higher than "video rate". Therefore, the transforms cannot be implemented in the parallel pipeline when the combined data rate is much above the conventional "video rates". One possibility for high throughput involves the use of an analog CCD device that performs a one-dimensional DCT in 100 ns (10 MHz rate) using parallel I/O [144]. Use of this device in the parallel pipeline would require D/A and A/D conversion.

An advantage of transform and some other block techniques is that each block is processed and compressed independently. Once the video data is stored in RAM, many processors can be used in parallel to perform the compression. Therefore, as long as the implementation is fast enough that one block can be compressed in the time that one frame is acquired, real time processing can be achieved by using the appropriate number of processors in parallel.

Block Truncation Coding is performed on each block independently. Therefore, it can be implemented in parallel by using multiple processors. However, the amount of compression is not easily adjustable.

The HVS compensation schemes do not appear to have hardware implementations. However, there are DSP implementations which perform linear filtering operations at very high speeds. Compression schemes based on simple edge detection and coding would utilize these devices. With the image broken down into a few sub-images, multiple processors would perform the edge detection. This has been done for a very simple 2x2 edge detector. The high speed GSP chips planned for the near future (TMS 34020) should implement more effective filters at reasonable speeds. Implementations of the complex, high-compression HVS schemes at high speeds seem beyond the capabilities of current hardware technology.

In most cases the experimenter would probably want a copy of as much information as possible for detailed analysis in addition to the compressed, transmitted image. Therefore, whenever the data acquisition rate is above the magnetic tape storage rate (30 Mb/s), the data should be initially compressed only enough to be stored on tape. If further compression is required for downlink transmission, the data can be read into RAM, the image can be reconstructed if necessary, and a second compression algorithm can be applied. Therefore, we need to consider two stages of compression, one before storage on tape and one for downlink.

The first stage must be able to handle input data rates as high as 80 Mp/s in the parallel pipeline configuration. The output rate should be 30 Mb/s which means the algorithm must be able to produce a fixed, but adjustable, bit rate in the range of 3 - 8 bpp. Within these specifications, we should attempt to minimize degradations of the image quality.

DPCM techniques can run at a high enough rate. However, the simple algorithms do not handle edges well at lower bit rates (even a two-dimensional predictor would have to revert to one-dimension for one line out of eight in the pipeline). More complex algorithms such as adaptive predictors may provide sufficient image quality.

Transform coding techniques can produce good quality reconstructed images when the bit rate is 3 bpp or above. Even edges are fairly well reproduced. The Hadamard transform and DCT are easily implemented in the parallel pipeline at 10 MHz.

Of the techniques which produce high-quality images, the fast Hadamard transform is the easiest to implement. However, if a non-uniform bit allocation scheme is used, the bit rate will vary from line to line in the parallel pipeline. This will require a multiplexer that can adjust to the bit allocation scheme. A uniform bit allocation scheme is also possible, but it will result in lower image quality.

The compression for downlink can generally be done at a much lower pixel rate than the first stage. Also, the output bit rate will be lower. At lower bit rates, less information is preserved. Therefore, it becomes more important to tailor the compression technique to the specific type of information which must be preserved. As a result, a programmable system may be necessary. This would allow for flexibility in the choice of compression method to match the experimenter's needs. The reduced speed requirement should make such a system possible. The needs of most of the experimenters can probably be met with a short list of techniques. A good edge detection technique will have to be included.

144. Chiang, A.M., "A Video-Rate CCD Two-Dimensional Cosine Transform Processor," *Proceedings of the SPIE*, vol. 845, pp. 2-5, 1987.

Optical Implementations

Speed is the main advantage of coherent optical transform processing. The entire image is transformed at the speed of light. Filtering operations in the transform domain are done continuously and are used in a system with any subsequent frame rate without synchronization.

A coherent light source is required for an optical method which uses Fourier transforms of monochromatic images. Since the Fourier transform is independent of spatial position, optical Fourier transforms are the most practical for image processing tasks such as identification and classification. However, image compression algorithms involving only spatial-frequency filtering are also easily implemented using Fourier optics.

Spatial filtering is used to vary the spatial bandwidth prior to the digitizing process. A He-Ne (helium-neon) laser source is used with a dibutyl-phthalate liquid gate, mixing with light from the image to produce a coherent light source. A Fourier transform lens forms the spatial Fourier transform of the image at the transparency. Various obstacles at the focal plane (of this first lens) result in spatial filtering. The image is retransformed by a second Fourier transform lens. This spatially filtered image, at the focal plane of a video camera, is then digitized and digitally compressed [145].

This filtering process has been used with Delta Modulation (DM) by Eichmann, et al [146][147]. A computer simulation also verifies significant compression when this technique is used in conjunction with a DCT encoder. The optical technique reduces high frequency distortions which manifest themselves as edge blockiness [148]. Consequently, this focal plane process increases the resolvable detail in the compressed image. It serves a function similar to pre-emphasis employed in broadcast television.

Even if the images are not produced with coherent light, coherent optical processing can be performed through the use of optically accessible spatial light modulators. Incoherent light initializes a modulator, which converts the light to a coherent beam of light. This coherent light is proportional in intensity to the original image produced with incoherent light. The result is an equivalent image consisting of coherent light, which can then be processed using Fourier optics.

Pockel's Readout Optical Modulator (PROM) is one device that can modulate spatial light at and above video rates. It has a maximum resolution of 500 lines/mm and is typically 25-30 mm in diameter. PROM can obtain a 10,000:1 contrast ratio. However, PROM requires blue light for writing and red light for reading. In situations where image information is contained in lower frequency light, such as a flame experiment where image information occurs in the red and infrared bands, input could become a problem.

Optical techniques may be well suited to certain applications. However, they are not easy to implement and may be inappropriate for all situations where a coherent light source is not available. Although a modulator can be used to convert an incoherent light source to a coherent light beam, the spectra of light from certain experiments studied in this report exceed the useful range of the modulator. It may be feasible to integrate this technique into some experiments using incoherent light sources within the operating range of the modulator. Because of this, caution is required when considering a general application of optical, spatial filtering.

145. Eichmann, G., Mammone, R., Stirbl, J., Keybl, J., Silber, R., "Hybrid Image Source Encoding for Real-Time Image Transmission," *Proceedings of the SPIE on Real-Time Signal Processing,* vol. 154, pp. 204-209, 1978.

146. Eichmann, G., Mammone, R., Stirbl, J., Keybl, J., Silber, R., "Hybrid Image Source Encoding for Real-Time Image Transmission," *Proceedings of the SPIE on Real-Time Signal Processing,* vol. 154, pp. 204-209, 1978.

147. Eichmann, G., Stirbl, J., Mommone, R., "Hybrid video encoding for real-time image transmission," *Proceedings of the SPIE on Optical Signal Processing for C^3I,* vol. 209, pp. 24-29, 1979.

148. Eichmann, G., Stirbl, J., Mommone, R., "Hybrid video encoding for real-time image transmission," *Proceedings of the SPIE on Optical Signal Processing for C^3I,* vol. 209, pp. 24-29, 1979.

CONCLUSIONS

The following conclusions were developed while studying the literature:

1. Three experiments requirements were studied in detail. The three experiments studied were Solid Surface Combustion, Bubble-in-Liquid Mass Transport Phenomena, and Nucleate Pool Boiling. The results of the investigation conclude that video data compression approaches for microgravity space experiments are experiment peculiar in requirements and no one single approach is universally optimum.

2. It is shown, for the experiments studied, required data compression is separable into two approaches, the first to limit data rates for storage, and the second to reduce data rates for transmission.

3. Hardware implementations for high resolution and/or high frame rate experiment requirements, and real time compression are currently limited, by technology, to methods that can be implemented using parallel processing, digital filtering, decomposition, and tree searches.

4. In general, based on this survey and the state of the art in image coding, transform algorithms are preferred over predictive methods. Of the transform methods, the Discrete Cosine Transform is the optimal method. It provides the most efficient compression.

5. For HHVT applications, the DCT is the one method which can best meet the stringent compression requirements, and maintain image fidelity. Several fast algorithms been developed [149][150]. These algorithms perform at least six times faster than the Fast Fourier Transform, and these algorithms directly lend themselves to hardware implementations.

6. Coupled with a motion compensation or block matching algorithm in the temporal direction, the DCT is currently the best method for high compression, interframe coding.

7. Although theoretically attractive, no approach could be identified for focal plane processing alone, that could be implemented with state of art hardware. Still, optical techniques are advantageous when used with digital compression to help maintain edges and high frequency detail.

---

149. Chen, W., Smith, C. H., and Fralick, S., "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications,* vol. COM-25, pp. 1004-1009, 1977.

150. Lee, B. G., "A New Algorithm to Compute the Discrete Cosine Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-32, pp. 1243-1245, 1984.

RECOMMENDATIONS FOR FURTHER STUDY

1. User controlled, dynamic image processing which provides image enhancement, reconstruction, and manipulation. These techniques could be used to offset the perceived degradations in compressed images. They will also aid investigators in the analysis of video data. Such techniques also provide needed functions in a telescience environment. This development will help to make the HHVT system a more useful, and easier to use, system. Investigators can use these methods to control the operation of the HHVT system.

2. A detailed study of experimenters' requirements.

3. A study to define specific spectral requirements.

4. Continued study of promising compression techniques. For example, at the time of this report, great advances are being made of fractals, and other shape matching, algorithms. These techniques provide several orders of magnitude improvement in compression. Also, current Human Visual System approaches will attempt to identify crucial elements to maintain the perception of a high fidelity image. These techniques, however, need further refinement of both algorithms and implementations to be useful in the HHVT system.

5. Further study of optical processing techniques for high data rates. We addressed the use of optical techniques primarily as a pre-processing step to digital encoding of the image. Implementation of purely optical techniques has been difficult and costly. In the past, such systems' performances have been disappointing. Still, since images ultimately start with light, an optical process seems to be the ideal method for image compression. Recent literature indicates a renewed effort in this area. Of special interest is the renewed effort to model and develop focal plane processors based upon human perception, neural responses, and pattern matching.

6. Development of standard methods to evaluate and compare digital image compression algorithms. Such standards should be "blind" to specific operating system and computer hardware advantages. Ideally, all algorithms would be tested in hardware "in situ". However, associated development time precludes this, and one should consider software-based benchmark comparisons. Also, one should respect the proprietary concerns of algorithm developers.

BIBLIOGRAPHY

'A Comparison of the Visual Effects of Two Transform Domain Encoding...," *Proceedings of the SPIE,* vol. 119, pp. 137-146, 1977.

A

Abdelwahab, A.A. and Kwatra, S.C., "Image Data Compression with VQ in the Transform Domain," *Proceedings of the ICC '86,* vol. 2, pp. 1285-1289, 1986.

Abe, M., et al., "A High-Speed Digital Filter LSI for Video Signal Processing," *IEEE Transactions on Solid-State Circuits,* vol. SC-22, pp. 396-402, June 1987.

Agbinya, J.I., "Fast Interpolation Algorithm Using Fast Hartley Transform," *Proceedings of the IEEE,* vol. 75, pp. 523-524, April 1987.

Ahmed, N. and Natarajan, T., "Some Aspects of Adaptive Transform Coding of Multispectral Data," *Proceedings of the Ann Asilomar Conf Circuits,* vol. 10, pp. 583-597, 1977.

Ahmed, N. and Natarajan, T.R., "Interframe Transform Coding of Picture Data," *Proceedings of the Ann Asilomar Conf Circuit,* vol. 9, pp. 553-557, 1976.

Ahmed, N., Natarajan, T., and Rao, K.R., "Discrete Cosine Transform," *IEEE Transactions on Computers,* vol. C-23, pp. 90-93, January 1974.

Allott, D. and Clarke, R.J., "Shape Adaptive Activity Controlled Multistage GSVQ of Images," *Electronics Letters,* vol. 21, pp. 393-395, 25 April 1985.

Anastassiou, D., et al., "Gray-Scale Image Coding for Freeze-Frame Videoconferencing," *IEEE Transactions on Communications,* vol. COM-34, pp. 382-394, April 1986.

Andrews, H.C. and Patterson, C.L., "Singular Value Decompositions and Digital Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-24, pp. 26-53, February 1976.

Arce, G.R. and Gallagher, Jr., N.C., "Image Source Coding Using Median Filter Roots," *Ann Allerton Conf. on CCC,* pp. 869-878, 1982.

Arnold, J.F. and Cavenor, M.C., "Improvements to the CAQ Bandwidth Compression Scheme," *IEEE Transactions on Communications,* vol. COM-29, pp. 1818-1823, December 1981.

Artieri, A., Jutand, F., et al., "An Efficient Architecture for Real-Time Motion Estimation Based on the Block Matching Technique," presented at the SPIE Conference on Visual Communications and Image Processing '88, Cambridge, Massachusetts, November 9-11, 1988.

B

Barba, J., et al., "A Modified Adaptive Delta Modulator," *IEEE Transactions on Communications,* vol. COM-29, pp. 1767-1785, December 1981.

Barba, J., Scheinberg, N., Schilling, D.L., "Entropy Encoding Applied to a Video DM Bit Stream," *Int Conf Communications,* pp. 22.7.1-5, 1981.

Bessette, O.E. and Schaming, W.B., "A 2-D Discrete Cosine Transform Video Bandwidth Compression System," *Proceedings of NAECON 1980,* vol. 3, pp. 1146-1152, 1980.

Bourbakis, N.G. and Alexandridis, N.A., "An Efficient, Real-Time Method for Transmitting Walsh-Hadamard Transformed Pictures," pp. 452-455, 1982.

Bowyer, D.E., "Walsh Functions, Hadamard Matrices, and Data Compression," pp. 33-37, 1971.

Brainard, R.C. and Othmer, J.H., "VLSI Implementation of a DPCM Compression Algorithm for Digital TV," *IEEE Transactions on Communications,* vol. COM-35, pp. 854-856, August 1987.

Brown, E.F., "Sliding-Scale Operation of Differential Type PCM Coders for Television," pp. 303-322, 1972.

Burge, R.E. and Wu, J.K., "An Adaptive Transform Image Data Compression Scheme Incorporating Pattern Recognition Procedures," *Proceedings of the Conference on Pattern Recognition and Image Processing,* pp. 230-236, 1981.

Burt, P.J. and Adelson, E.H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications,* vol. COM-31, pp. 532-540, April 1983.

Buss, D.D., et al., "The Technology of Charge-Coupled Devices for Video Bandwidth Reduction," *Proceedings of the SPIE,* vol. 66, pp. 48-56, 1975.

C

Camana, P., "Video-Bandwidth Compression: A Study in Tradeoffs," *IEEE Spectrum,* vol. 16, pp. 24-29, June 1979.

Camana, P.C., "Image Processing Techniques for Compression," *NAECON '79 Record,* pp. 1298-1302, 1979.

Candy, J.C., "Refinement of a Delta Modulator," pp. 323-339, 1972.

Cappellini, V., Chini, A., and Lotti, F., "Some Data Compression Methods for Processing the Images Received from ERS," *Proc Int'l Sci-Tech Conf Space,* pp. 33-43, 1976.

Cham, W.K. and Clarke, R.J., "DC Coefficient Restoration in Transform Image Coding," *IEE Proceedings,* vol. 131 F, pp. 709-713, December 1984.

Chan, L.C. and Whiteman, P., "Hardware-Constrained Hybrid Coding of Video Imagery," *IEEE Transactions on Aerospace and Electronic Systems,* vol AES-19, pp. 71-83, January 1983.

Charalambous, "Appendix B - Dimensionality Reduction of the KLT," NASA Contractor Report, no. 164286, pp. 44-69, January 1981.

Chaudhuri, B.B. and Kundu, M.K., "Digital Line Segment Coding: A New Efficient Contour Coding Scheme," *IEE Proceedings,* vol. 131 E, pp. 143-147, July 1984.

Chaudhuri, B.B., Mali, P.C., Majumder, D., "An Optimal Property of Walsh-Hadamard Transforms and Some Related Algorithms in Image Processing," pp. 791-795, 1983.

Chen W-H. and Smith, C.H., "Adaptive Coding of Color Images Using Cosine Transform'.

Chen, W-H., Smith, C. H., and Fralick, S., "A Fast Computational Algorithm for the Discrete Cosine Transform, *IEEE Transactions on Communications,* vol. COM-25, pp. 1004-1009, 1977.

Chen, W-H., "Scene Adaptive Coder," *Proceedings of the ICC '81,* vol. 2, pp. 22.5.1-6, 1981.

Chiang, A.M., "A Video-Rate CCD Two-Dimensional Cosine Transform Processor," *Proceedings of the SPIE,* vol. 845, pp. 2-5, 1987.

Choras, R.S., "Application of Hadamard, Haar, and H-H Transformations to Image Coding and Bandwidth Compression," *Proceedings of the SPIE,* vol. 359, pp. 336-342, 1982.

Claire, E.J., Farber, S.M., and Green, R.R., "Practical Techniques for Transform Data Compression/Image Coding," pp. 2-6.

Clarke, R.J., "Hybrid Intraframe Transform Coding of Image Data," *IEE Proceedings,* vol. 131 F, pp. 2-6, February 1984.

Cohen, Y., Landy, M.S. and Pavel, M., "Hierarchical Coding of Binary Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-7, pp. 284-298, May 1985.

Colef, M., Barba, J., et al., "NTSC Composite Coding for Video Conference Systems Using the Hadamard Transform," pp. 183-187, 1987.

Connor, D.J., et al., "Intraframe Coding for Picture Transmission," *Proceedings of the IEEE,* vol. 60, pp. 779-791, July 1972.

Cornsweet, T.N., "A Simple Retinal Mechanism That Has Complex and Profound Effects on Perception," Prentice Award Lecture, December 1984.

Cornsweet, T.N. and Yellott, Jr., J.I., "Intensity-Dependent Spatial Summation," *Journal of the Optical Society of America A,* vol. 2, pp. 1769-1786, October 1985.

Cottrell, G.W. and Munro, P., "Principal Components Analysis of Images via Back Propagation," *Proceedings of the SPIE, Visual Communication and Image Processing '88,* vol. 1001, part 2, pp. 1070-1077, November 1988.

D

Dail, J.E., "Polynomial Transform Coding of Image Signals," pp. 47.14-19.

Dapeng, Z. and Zhongrong, L., "The Design of a Smart Sensor System for Real-Time Remote Sensing Image Processing," *Proceedings of the SPIE,* vol. 657, pp. 164-70, 1986.

Delp, E. J. and Mitchell, O. R., "Image Compression By Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-27, pp. 1335-1342, September 1979.

Delp, E.J. and Chu, C.H., "Detecting Edge Segments," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. SMC-15, pp. 144-152, February 1985.

Delp, E.J. and Mitchell, O.R., "Use of Moment Preserving Quantizers in DPCM Image Coding," *Proceedings of the SPIE,* vol. 249, pp. 123-130, 1980.

Desoky, A. and McDonald, J.W., "Compression of Image Data Using Adaptive-Predictive Coding," *IEEE,* pp. 11-15, 1987.

Dillard, G.M., "A Walsh-Like Transform Requiring Only Additions, With Applications to Data Communications," *NAECON '76 Record,* pp. 101-105, 1976.

Dillard, G.M., "Application of Ranking Techniques to Data Compression for Image Transmission," *Proceedings of NTC '75,* vol. 1, pp. 22.18-22, 1975.

Dosik, P.H. and Schwartz, M., "An Optimized Buffer Controlled Data Compression System," *IEEE Transactions on Communications,* vol. COM-22, pp. 1506-1515, October 1974.

Driessen, L.M.H.E., et al., "An Experimental Digital Video Recording System," *IEEE Transactions on Consumer Electronics,* vol. CE-32, pp. 362-371, August 1986.

Dukhovich, I.J., "A DPCM System Based on a Composite Image Model," *National Telecommunications Conference,* pp. 36.2.1-6, 1980.w

E

Eichmann, G., et al., "Hybrid Image Source Encoding for Real-Time Image Transmission," *Proceedings of the SPIE,* vol. 154, pp. 204-210, 1978.

Eichmann, G., Stribl, R., and Mammone, R, "Hybrid Video Encoding for Real-Time Image Transmission," *Proceedings of the SPIE,* vol. 209, pp. 24-29, 1979.

Ekambaram, C. and Kwatra, S.C., "A New Architecture for Adaptive Transform Compression of NTSC Composite Video," *Proceedings of the NTC '81,* vol. 2, pp. C9.6.1-5, 1981.

Elnahas, S.E. and Dunham, J.G., "Entropy Coding for Low-Bit-Rate Visual Telecommunications," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1175-1183, August 1987.

Enomoto, H. and Shibata, K., "Orthogonal Transform Coding System for Television Signals," pp. 11-17.

Essman, J.E., Hua, Q.D., and Griswold, N.C., "Video Link Data Compression for Remote Sensors," *Proceedings of the SPIE,* vol. 87, pp. 55-77, 1976.

F

Farrelle, P.M. and Jain, A.K., "Recursive Block Coding - A New Approach to Transform Coding," *IEEE Transactions on Communications,* vol. COM-34, pp. 161-179, February 1986.

Farrelle, P.M. and Jain, A.K., "Quad-tree Based Two Source Image Coding," 1986.

Feng, Y. and Nasrabadi, N.M., "Address-VQ: An Adaptive Vector Quantization Scheme Using Interblock Correlation," *Proceedings of the SPIE, Visual Communication and Image Processing '88,* vol. 1001, part 1, pp. 214-222, 1988.

Flickner, M.D. and Ahmed, N., "A Derivation for the Discrete Cosine Transform," *Proceedings of the IEEE,* vol. 70, pp. 1132-1134, September 1982.

Fong, C.B. and Kanefsky, M., "An Efficient Source Coding Technique for Data Compression of Multil...," *Proceedings of the Ann Allerton Conf Comm Cont Co,* vol., pp. 860-869, 1981.

Frei, W. and Baxter, B., "Rate-Distortion Coding Simulation for Color Images," *IEEE Transactions on Communications,* vol. COM-25, pp. 1385-1392, November 1977.

Fukinuki, T. and Miyata, M., "Intraframe Image Coding by Cascaded Hadamard Transforms," *IEEE Transactions on Communications,* vol. COM-21, pp. 175-80, March 1973.

G

Gaggioni, H. and Le Gall, D., "Digital Video Transmission and Coding for the Broadband ISDN," *IEEE Transactions on Consumer Electronics,* vol. CE-34, pp. 16-34, February 1988.

Gerken, P. and Schiller, H., "A Low Bit-Rate Image Sequence Coder Combining a Progressive DPCM on Interleaved Rasters with a Hybrid DCT Technique," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1079-1089, August 1987.

Gersho, A. and Ramamurthi, B., "Image Coding Using Vector Quantization," *Proceedings of the ICASSP,* pp. 428-431, 1982.

Gharavi, H. and Jalali, A., "Entropy Coding of Cosine Transformed Images," pp. 1165-1169, 1986.

Girod, B., "The Efficiency of Motion-Compensated Prediction for Hybrid Coding ...," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1140-1154, August 1987.

Goldberg, M., Boucher, P.R. and Shlien, S., "Image Compression Using Adaptive Vector Quantization," *IEEE Transactions on Communications,* vol. COM-34, pp. 180-187, February 1986.

Gonsalves, R.A. and Johnson, G.R., "Image Compression with Approximate Transform Implementation," *Proceedings of the SPIE,* vol. 249, pp. 138-145, 1980.

Gonsalves, R.A. and Shea, A., "Bounded-Error Coding of Cosine Transformed Images," *Proceedings of the SPIE,* vol. 207, pp. 291-298, 1979.

Gonsalves, R.A., et al., "Fixed-Error Encoding for Bandwidth Compression," *Proceedings of the SPIE,* vol. 149, pp. 27-42, 1978.

Gonsalves, R.A., Evans, N.E., Shea, A., "Bounded Error Data Compression at Constant Rate," *Proceedings of the International Telemetering Conference,* pp. 77-85, 1978.

Grallert, H.-J., "Application of Orthonormalized M-Sequences for Data Reduced and Error Prot...."

Gray, R.M., "Vector Quantization for Signal Coding and Recognition," NASA Data Compression Workshop, 1988.

Griswold, N.C. and Haralick, R.M., "A Critical Comparison of Fast Transforms for Image Data Compression," *Proceedings of the SPIE,* vol. 87, pp. 180-188, 1976.

Griswold, N.C. and Sayood, K., "Unsupervised Learning Approach to Adaptive DPCM," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-4, pp. 380-391, July 1982.

Griswold, N.C., Halverson, D.R., and Wise, "A Note on Adaptive Block Truncation Coding for Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-35, pp. 1201-1203, August 1987.

Guglielmo, M., Marion, R., Sciarappa, A., "Subjective Quality Evaluation of Different Intraframe Adaptive Coding ...," pp. 467-70, 1982.

H

Habibi, A., "Survey of Adaptive Image Coding Techniques," *IEEE Transactions on Communications,* vol. COM-25, pp. 1275-1284, November 1977.

Habibi, A., "Hybrid Coding of Pictorial Data," *IEEE Transactions on Communications,* vol. COM-22, pp. 614-624, May 1974.

Habibi, A., "Comparison of n^{th}-Order DPCM Encoder With Linear Transformations and..," *IEEE Transactions on Communications,* vol. COM-19, pp. 948-956, December 1971.

Habibi, A., "Adaptive Hybrid Coding of Images," *Proceedings of the SPIE,* vol. 207, pp. 233-239, 1979.

Habibi, A. and Samulon, A.S., "Bandwidth Compression of Multispectral Data," *Proceedings of the SPIE,* vol. 66, pp. 23-35, 1975.

Habibi, A. and Wintz, P., "Image Coding by Linear Transformation and Block Quantization," *IEEE Transactions on Communication Technology,* vol. COM-19, pp. 50-62, February 1971.

Habibi, A. and Wintz, P., "Optimum Linear Transformations for Encoding Two-Dimensional Data," pp. 577-619, 1972.

Habibi, A., Samulon, A.S., and Fultz, G.L., "Adaptive Coding of MSS Imagery," *Proceedings of the National Television Conference '77,* pp. 10:2.1-7, 1977.

Hall, C.F., "Perceptual Coding in the Fourier Transform Domain," *NTC '80 Conf. Rec.,* pp. 36.1.1, 1980.

Hall, C.F. and Andrews, H.C., "Digital Color Image Compression in a Perceptual Space," *Proceedings of the SPIE,* vol. 149, pp. 182-188, August 1978.

Hall, C.F. and Carl, J.W., "A Hybrid Image Compression Technique," *Proceedings of NAECON 1985,* vol. 1, pp. 188-194, 1985.

Hall, G., "Large-Kernel Convolutions in Image Processing," *Digital Design,* vol. 16, pp. 46-48, August 1986.

Halverson, D. R., Griswold, and Wise, "A Generalized Block Truncation Coding Algorithm for Image Compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-32, pp. 664-668, June 1984.

Haralick, R.M., "Digital Step Edges from Zero Crossing of Second Directional Derivative," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-6, pp. 58-68, January 1984.

Haralick, R.M. and Griswold, N., "A Fast Two-Dimensional Karhunen-Loeve Transform," *Proceedings of the SPIE,* vol. 66, pp. 144-159, 1975.

Haralick, R.M. and Shanmugam, K., "Comparative Study of Discrete Linear Basis for Image Data Compression," *IEEE Transactions on Systems, Man, Cybernetics,* vol. SMC-4, pp. 16-27, January 1974.

Haralick, R.M., Griswold, N.C., and Paul, "An Annihilation Transform Compression Method for Permuted Images," *Proceedings of the SPIE,* vol. 87, pp. 189-196, 1976.

Hart, M.M., "An Image Compression Survey and Algorithm Switching Based on Scene Activity," NASA Technical Paper 2458, 1985.

Haskell, B.G. and Gordon, P.L., "Source Coding of Television Signals Using Interframe Techniques," *Proceedings of the SPIE,* vol. 66, pp. 9-22, 1975.

Haskell, B.G. and Steele, R., "Audio and Video Bit-Rate Reduction," *Proceedings of the IEEE,* vol. 69, pp. 252-262, February 1981.

Hatori, Y., Murakami, H., Yamamoto, H., "30 Mbits/s Codec for NTSC-CTV by Interfield and Intrafield Adaptive Prediction," pp. 23.6.1-5, 1979.

Healy, D.J. and Mitchell, O.R., "Digital Video Bandwidth Compression Using Block Truncation Coding," *IEEE Transactions on Communications,* vol. COM-29, pp. 1809-1817, December 1981.

Heiman, Arie, "A Look-Up-Based Universal Real-Time Transformer for Image Coding," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1207-1213, August 1987.

Hopkins, R., "Advanced Television Systems," *IEEE Transactions on Consumer Electronics,* vol. CE-34, pp. 1-15, February 1988.

Hou, H.S., "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-35, pp. 1455-1461, October 1987.

Hsing, T.R. and Tzou, K.-H., "Video Compression Techniques: A Review," *Proceedings of Globecom,* pp. 1521-1526, November 1984.

Hsu, Y.S., et al., "Pattern Recognition Experiments in the Mandala/Cosine Domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-5, pp. 512-520, September 1983.

Huck, F.O., "Local Intensity Adaptive Image Coding," NASA Data Compression Workshop, 1988.

Huffman, D.A., "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE,* pp. 1098-1101, September 1952.

I

Ikonomopoulos, A. and Kunt, M., "High Compression Image Coding via Directional Filtering," *Signal Processing,* vol. 8, pp. 179-203, 1985.

J

Jain, A.K., "A Fast Karhunen-Loeve Transform for a Class of Random Processes," *IEEE Transactions on Communications,* vol. COM-24, pp. 1023-1029, September 1976.

Jain, A.K., "Advances in Mathematical Models for Image Processing," *Proceedings of the IEEE,* vol. 69, pp. 502-528, May 1981.

Jain, A.K., "Image Data Compression: A Review," *Proceedings of the IEEE,* vol. 69, pp. 349-387, March 1981.

Jain, A.K. and Harrington, D.G., "A 10 MHz Data Compression System for Real-Time Image Storage and Transmission," *Proceedings of the SPIE,* vol. 575, pp. 62-65, 1985.

Jain, A.K., Wang, S.H., and Liao, Y.Z., "Fast Karhunen-Loeve Transform Data Compression Studies," pp. 6.5.1-5.

Jain, P.C. and Delogne, P., "Real-Time Hadamard Transform Coding for TV Signals," *IEEE Transactions on Electromagnetic Compatibility,* vol. EMC-23, pp. 103-107, May 1981.

Jalali, A., et al., "A Component Codec and Line Multiplexer," *IEEE Transactions on Consumer Electronics,* vol. CE-34, pp. 156-165, February 1988.

Jayant, N. S. and Noll, Peter, *Digital Coding of Waveforms,* Prentice Hall, 1984.

Jones, C.B., "An Efficient Coding System for Long Source Sequences," *IEEE Transactions on Information Theory,* vol. IT-27, pp. 280-291, May 1981.

Jones, H.W., Hein, D.N., and Knauer, S.C., "The K-L, Discrete Cosine, and Related Transforms Obtained Via the HT," *Proceedings of the ITC '78,* pp. 87-98, 1978.

Jones, Jr., H.W., "A Comparison of Theoretical and Experimental Video Compression Designs," *IEEE Transactions on Electromagnetic Compatibility,* vol. EMC-21, pp. 50-6, February 1979.

Jones, Jr., H.W., "Performance of a Fixed Rate Conditional Replenishment Transform Compressor," *Proceedings of the NTC '77,* pp. 10:1.1-5, 1977.

Jones, Jr., H.W., "A Conditional Replenishment Hadamard Video Compressor," *Proceedings of the SPIE,* vol. 119, pp. 91-97, 1977.

Jones, Jr., H.W., "A Real-Time Adaptive Hadamard Transform Video Compressor," *Proceedings of the SPIE,* vol. 87, pp. 2-9, 1976.

Jones, Jr., H.W. and Hofman, L.B., "Comparison of Video Fields and Frames for Transform Compression," *Proceedings of the SPIE,* vol. 149, pp. 214-221, 1978.

Jones, R.A., "Adaptive Hybrid Picture Coding," *Proceedings of the SPIE,* vol. 87, pp. 247-255, 1976.

Jones, R.A. and Mix, D.F., "A Rate Distortion Analysis of Adaptive Hybrid Picture Coding," *Proceedings of the SPIE,* vol. 149, pp. 10-17, 1978.

Jorgenson, F.B., Michel, G., Wagner, C., "Predite: A Real Time Processor for Bandwidth Compression in TV Transmission," pp. 1195-1198, 1982.

K

Kamangar, F.A. and Rao, K.R., "Interfield Hybrid Coding of Component Color Television Signals," *IEEE Transactions on Communications,* vol. COM-29, pp. 1740-1752, December 1981.

Kamangar, F.A. and Rao, K.R., "Adaptive Coding of NTSC Component Video Signals," *Proceedings of the NTC '80,* pp. 36.4.1-6, 1980.

Kamangar, F.A. and Rao, K.R., "Interframe Hybrid Coding of NTSC Component Video Signal," pp. 53.2.1-5, 1979.

Kaneko, M., Hatori, Y., and Koike, A., "Improvements of Transform Coding Algorithm for Motion-Compensated ...," *IEEE Journal on Selected Areas in Communication,* vol. SAC-5, pp. 1068-1077, August 1987.

Kappagantula, S. and Rao, K.R., "Motion Compensated Predictive Coding," pp. 64-69, 1983.

Kato, Y., Mukawa, N., and Okubo, S., "A Motion Picture Coding Algorithm Using Adaptive DCT Encoding Based...," *IEEE Journal on Selected Areas in Communication,* vol. SAC-5, pp. 1090-1099, August 1987.

Kauth, R.J. et al., "BLOB: An Unsupervised Clustering Approach to Spatial Preprocessing of MSS..," *International Symposium on Remote Sensing Environments,* pp. 1309-1317, 1977.

Keshavan, H.R., et al., "Image Enhancement Using Orthogonal Transforms," pp. 593-597, 1976.

Kim, E.S. and Rao, K.R., "Walsh-Hadamard Transform/DPCM Processing of Intraframe Color Video," *Proceedings of the SPIE,* vol. 207, pp. 240-246, 1979.

Knab, J.J., "Effects of Round-Off Noise on Hadamard Transformed Imagery," *IEEE Transactions on Communications,* vol. COM-25, pp. 1292-1294, November 1977.

Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *IEEE Transactions on Electromagnetic Compatibility,* vol. EMC-18, pp. 28-36, February 1976.

Knauer, S.C., "Real-Time Video Compression Algorithm for Hadamard Transform Processing," *Proceedings of the SPIE,* vol. 66, pp. 58-69, 1975.

Knight, G.R., "Interface Devices and Memory Materials," *Optical Information Processing,* ed. Lee, S.H., 1981.

Kocher, M. and Kunt, M., "A Contour-Texture Approach to Picture Coding," *Proceedings of the ICASSP-82,* pp. 436-440, May 1982.

Koga, T. and Ohta, M., "Entropy Coding for a Hybrid Scheme with Motion Compensation in Subprimary ...," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1166-1174, August 1987.

Koga, T., et al., "A 384 kbit/sec Motion Video Codec with Scene Change Detection," *Proceedings of the ICC '86,* vol. 1, pp. 366-370, 1986.

Kretz, F. and Nasse, D., "Digital Television: Transmission and Coding," *Proceedings of the IEEE,* vol. 73, pp. 575-591, April 1985.

Kunt, M., "Edge Detection: A Tutorial Review," pp. 1172-1175, 1982.

Kunt, M. and Johnsen, O., "Block Coding of Graphics: A Tutorial Review," *Proceedings of the IEEE,* vol. 68, pp. 770-786, July 1980.

Kunt, M., Ikonomopoulos, A., and Kucher, M, "Second-Generation Image-Coding Techniques," *Proceedings of the IEEE,* vol. 73, pp. 549-574, April 1985.

Kwak, H.S., Srinivasan, R., and Rao, K.R, "C-Matrix Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-31, pp. 1304-1307, October 1983.

Kwan, H.K., "Image Data Compression Using 2-D Lattice Predictor," *Electronics Letters,* vol. 20, pp. 994-995, 22 November 1984.

Kwatra, S.C. and Ekambaram, C., "An Adaptive Technique for Bandwidth Compression of Color Video Signal," pp. 350-354, 1983.

Kwatra, S.C. and Fatmi, H., "NTSC Composite Video at 1.6 Bits / Pixel," *Proceedings of the ICC '83,* vol. 1, pp. 458-462, 1983.

L

LaBonte, A.E., "Micro-Adaptive Picture Sequencing in a Display Environment," *Proceedings of the SPIE,* vol. 249, pp. 61-70, 1980.

LaBonte, A.E., "Two-Dimensional Image Coding by Micro-Adaptive Picture Sequencing," *Proceedings of the SPIE,* vol. 119, pp. 99-106, 1977.

Langdon, Jr., G.G., "An Introduction to Arithmetic Coding," *IBM Journal of Research and Development,* vol. 28, pp. 135-149, March 1984.

Lansing, D.L., "Experiments in Encoding Multilevel Images as Quadtrees," NASA Technical Paper, no. 2722, 1987.

Lebreton, G., "Image Data Compression Using Unitary Transforms: Video-Rate Optical...," *Proceedings of the SPIE,* vol. 492, pp. 278-283, 1984.

Lee, B. G., "A New Algorithm to Compute the Discrete Cosine Transform," *IEEE Transactions on Acoustic, Speech, and Signal Processing,* vol. ASSP-32, no. 6, pp. 1243-1245, 1984.

Lee, C. and Nadler, M., "Predictive Image Coding with Pseudo-Laplacian Edge Detector," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1190-1206, August 1987.

Lee, H.J. and Lee, D.T.L., "A Gain-Shape Vector Quantizer for Image Coding," *Proceedings of the ICASSP 86,* pp. 141-144, 1986.

Lee, S.H., "Basic Principles of Optical Information Processing," *Optical Information Processing,* ed. Lee, S.H., 1981.

Leger, A., et al., "Distributed Arithmetic Implementation of the DCT for Real Time ...," *Proceedings of the SPIE,* vol. 804, pp. 364-370, 1987.

Lei, T-L, Scheinberg, R., Schilling, N., "Adaptive Delta Modulation Systems for Video Encoding," *IEEE Transactions on Communications,* vol. COM-25, pp. 1302-1314, November 1977.

Lema, M.D. and Mitchell, O.R., "Absolute Moment BTC and Its Application to Color Images," *IEEE Transactions on Communications,* vol. COM-32, pp. 1148-1157, October 1984.

Lemne, G. and Johansson, B., "Design Considerations for High-Speed Transform Image Compression," *Proceedings of the SPIE,* vol. 694, pp. 55-61, 1986.

Lewis, Jr. H. Garton and Forsyth, William B., "Hybrid LZW Compression," Proceedings of the NASA Langley sponsored *International Workshop on Visual Information Processing for Television and Telerobotics,* Williamsburg, Virginia, May 1989.

Li, C.E. and Rao, K.R., "DCT and Hybrid Coding of the Component Color TV Signal," pp. 161-165, 1983.

Li, C.E. and Rao, K.R., "Composite Predictive Coding of NTSC Color TV Signal," *Proceedings of the SPIE,* vol. 359, pp. 343-346, 1982.

Limb, J.O., "Visual Perception Applied to the Encoding of Pictures," *Proceedings of the SPIE,* vol. 87, pp. 80-87, 1976.

Limb, J.O., Rubinstein, C.B., Thompson, "Digital Coding of Color Video Signals - A Review," *IEEE Transactions on Communications,* vol. COM-25, pp. 1349-1385, November 1977.

Linde, Y., Buzo, A., and Gray, R.M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications,* vol. COM-28, pp. 84-94, January 1980.

Lindsay, R.A., "A Recursive Technique for Adaptive Vector Quantization," NASA Data Compression Workshop, 1988.

Lineback, J.R., "How Long Will It Take Image Processing to Blast Off?," *Electronics,* vol. 60, pp. 65-66, February 19, 1987.

Ludington, D., "Real Time Compression of Video Signals," *Proceedings of NAECON 1979,* vol. 3, pp. 1318-1321, 1979.

Ludington, D.N., "Video Compression Using Sampled Data Analog Devices," *Proceedings of the SPIE,* vol. 298, pp. 142-147, 1981.

Luther, A.C., "You Are There... and In Control," *IEEE Spectrum,* vol. 25, pp. 45-50, September 1988.

Lyle, Jr., W.D. and Forte, F., "A Useful Property of the Coefficients of a Walsh-Hadamard Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-28, pp. 479-480, August 1980.

M

MacCalla, J.R. and Chang, M.U., "Multi-Spectral Data Compression Using Hybrid Cluster Coding," *Communication Satellite Systems Conference,* pp. 404-407, 1980.

Marescq, J.P. and Labit, C., "Vector Quantization in Transformed Image Coding," *Proceedings of the ICASSP '86,* pp. 145-148, 1986.

Marr, D. and Hildreth, E., "Theory of Edge Detection," *Proc Royal Soc London B,* vol. 207, pp. 187-217, 1980.

Max, J., "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory,* vol. IT-6, pp. 7-12, March 1960.

McCaughey, D.G., "An Image Coding Algorithm Using Spline Functions," *Proceedings of the SPIE,* vol. 149, pp. 51-61, 1978.

McClung, T., Morison, R., Gaston, D.W., "RPV Video Digital Multiplexing System," *NAECON '77 Record,* pp. 510-508, 1977.

McIlroy, C.D., Linggard, R., and Monteith, "Edge Detection in Real-Time," *Proceedings of the SPIE,* vol. 504, pp. 445-454, 1984.

Meiri, A.Z., "The Pinned KLT of a Two-Dimensional Gauss-Markov Field," *Proceedings of the SPIE,* vol. 87, pp. 155-163, 1976.

Meiri, A.Z. and Yudilevich, E., "A Pinned Sine Transform Image Coder," *IEEE Transactions on Communications,* vol. COM-29, pp. 1728-1735, December 1981.

Melzer, S.M., "An Image Transform Coding Algorithm Based on a Generalized Correlation Model," *Proceedings of the SPIE,* vol. 149, pp. 205-213, 1978.

Merchant, S.N. and Rao, B.V., "Image Data Compression via Coshar Transform," pp. 1079-1083, 1983.

Merola, P.A., et al., "Charge Injection Device Focal Plane Processor for Video Bandwidth Compression," *SPIE,* vol. 119, pp. 115-120, 1977.

Miller, W.H., "Image Processing Using Gallium Arsenide Technology," NASA Data Compression Workshop, 1988.

Mitchell, O.R. and Tabatabai, A., "Adaptive Transform Image Coding for Human Analysis," *Proceedings of the ICC '79,* vol. 2, pp. 23.2.1-5, 1979.

Modestino, J.W., et al., "Performance of Block Cosine Image Coding with Adaptive Quantization," pp. E1.1.1-6, 1982.

Modestino, J.W., et al., "Combined Source-Channel Coding of Images Using the Block Cosine Transform," *Proceedings of the NTC '80,* pp. 50.4.1-7, 1980.

Moorhead, R.J. II, et al., "Image Sequence Compression Using a Pel-Recursive Motion-Compensated Technique," *IEEE Journal on Selected Areas in Communication,* vol. SAC-5, pp. 1100-1114, August 1987.

Moraitis, C.S., and Maistros, J.G., "A Microcomputer System for Real Time Digitized Image Compression," *ICIASF '85 Record,* pp. 133-139, 1985.

Mounts, F.W., "Frame-to-Frame Digital Processing of TV Pictures to Remove Redundancy," pp. 653-672, 1972.

Mudge, T.N., Delp, E.J., et al., "Image Coding Using the Multimicroprocessor System PASM," *Conf Pattern Recog Image Proc,* pp. 200-205, 1982.

Murakami, H., et al., "15/30 Mbit/s Universal Digital TV Codec Using a Median Adaptive Predictive Coding," *IEEE Transactions on Communications,* vol. COM-35, pp. 637-645, June 1987.

Murakami, T., Asai, K., and Itoh, A., "Vector Quantization of Color Images," *Proceedings of the ICASSP 86,* pp. 133-136, 1986.

Murphy, M.S., "Comparison of Transform Image Coding Techniques for Compression ...," *Proceedings of the SPIE,* vol. 309, pp. 212-219, 1981.

Murray, G.G., "Microprocessor System for TV Imagery Compression," *Proceedings of the SPIE,* vol. 119, pp. 121-129, 1977.

Musmann, H.G., Pirsch, P., and Grallert, "Advances in Picture Coding," *Proceedings of the IEEE,* vol. 73, pp. 523-548, April 1985.

N

Nam, M.K. and O'Neill, W.D., "Adaptive LPC of Time-Varying Images Using Multidimensional Recursive..," *IEEE Journal on Selected Areas in Communication,* vol. SAC-5, pp. 1115-1126, August 1987.

Narasimhan, M.A., Rao, K.R., and Raghava, "Image Data Processing by Hybrid Sampling," *Proceedings of the SPIE,* vol. 119, pp. 130-136, 1977.

Nassau, K., *The Physics and Chemistry of Color,* Wiley-Interscience Publication, 1983.

Natarajan, T.R. and Ahmed, N., "On Interframe Transform Coding," *IEEE Transactions on Communications,* vol. COM-25, pp. 1323-1329, November 1977.

Neagoe, V.E., "Predictive Data Compression of Color Picture Signals Using a Component," pp. 22.6.1-4, 1981.

Nelson, Mark R., "LZW Data Compression," *Dr. Dobb's Journal of Software Tools and Programming Techniques,* vol. 14, no. 10, pp. 29-36, October 1989.

Netravali, A., Prasada, B., Mounts, F.W., "Adaptive Hadamard Transform Coding of Pictures," pp. 6.6.1-4.

Netravali, A.N. and Limb, J.O., "Picture Coding: A Review," *Proceedings of the IEEE,* vol. 68, pp. 366-407, March 1980.

Netravali, A.N. and Stuller, J.A., "Motion-Compensated Transform Coding," *Bell System Technical Journal,* vol. 58, pp. 1703-1718, September 1979.

Ngan, K.N. and Steele, R., "Enhancement by Simple Statiscal Methods of PCM and DPCM Images Corrupted...," *Nat. Telecommunications Conf.,* pp. 50.5.1-5, 1980.

Noble, S.C., "A Comparison of Hardware Implementations of the Hadamard Transform . . .," *Proceedings of the SPIE,* vol. 66, pp. 207-211, 1975.

O

Ocylok, G., "A Comparison of Interframe Coding Techniques," *Proceedings of the ICASSP '83,* pp. 1224-1227, 1983.

Ohira, T., Hayakawa, M., Matsumoto, K., "Orthogonal Transform Coding System for NTSC Color Television Signal," *ICC Conference Record,* vol. 1, pp. 4B.3-86, 1977.

Orfanidis, S.J. and Marshall, T.G., "Two-Dimensional Transforms of the Sampled NTSC Color Video Signal," *Proceedings of the SPIE,* vol. 207, pp. 308-313, 1979.

P

Papananos, Y. and Anastassiou, D., "A VLSI Architecture for a Nonlinear Edge-Preserving Noise-Smoothing Image Filter," *Proceedings of the SPIE,* vol. 845, pp. 321-328, 1987.

Parsons, J.R. and Tescher, A.G., "An Investigation of MSE Contributions in Transform Image Coding Scheme," *Proceedings of the SPIE,* vol. 66, pp. 196-206, 1975.

Paul, I. and Woods, J.W., "Some Experimental Results in Adaptive Prediction DPCM Coding of Images," *Proceedings of the ICASSP,* pp. 1220-1223, 1983.

Pearson, D.E. and Whybray, M.W., "Transform Coding of Images Using Interleaved Blocks," *IEE Proceedings,* vol. 131 F, pp. 466-472, August 1984.

Pearson, J.J., "A CAQ Bandwidth Reduction System for RPV Video Transmission," *SPIE,* vol. 66, pp. 101-107, 1975.

Pearson, J.J. and Simonds, R.M., "Adaptive, Hybrid, and Multi-Threshold CAQ Algorithms," *Proceedings of the SPIE,* vol. 87, pp. 19-23, 1976.

Pirsch, Peter, "VLSI Implementation for Visual Communications - Tutorial Short Course Notes," *SPIE Symposium on Optical and Optoelectronic Engineering,* vol. T5, November 1988.

Poehler, P.L. and Choi, J., "Linear Predictive Coding of Imagery for Data Compression Applications," *Proceedings of the ICASSP '83,* pp. 1240-1243, 1983.

Pracht, B.R. and Bowyer, K.W., "Adaptations of Run-Length Encoding for Image Data," *Proceedings of the IEEE,* pp. 6-10, 1987.

Pratt, W.K., "Spatial Transform Coding of Color Images," *IEEE Transactions on Communication Technology,* vol. COM-19, pp. 980-992, December 1971.

Pratt, W.K., "DPCM Quantization Error Reduction for Image Coding," *Proceedings of the SPIE,* vol. 66, pp. 167-171, 1975.

Pratt, W.K. and Andrews, H.C., "Application of Fourier-Hadamard Transformation to Bandwidth Compression," pp. 515-553, 1972.

Pratt, W.K., Chen, W-H, and Welch, L.R., "Slant Transform Image Coding," *IEEE Transactions on Communications,* vol. COM-22, pp. 1075-1093, August 1974.

Pratt, W.K., Kane, J., and Andrews, H.C., "Hadamard Transform Image Coding," *Proceedings of the IEEE,* vol. 57, pp. 58-68, January 1969.

Q

Queen, G.W., Bryan, J.K. and Gowdy, J.N., "Investigations of Several Discrete Transform Methods for Image Processing Applications," *Proceedings of the SOUTHEASTCON '81,* pp. 881-886, 1981.

R

Rabbani, M., *Digital Image Compression - Tutorial Short Course Notes,* SPIE Symposium on Optical and Optoelectronic Applied Science and Engineering, vol. T53, August 1988.

Rabiner, L.R., Schafer, R.W., Rader, C., "The Chirp z-Transform Algorithm," *IEEE Transactions on Audio and Electroacoustics,* vol. AU-17, pp. 86-92, June 1969.

Raghava, V., Rao, K.R., Narasimhan, M.A., "Simulation of Image Data Processing by Hybrid Sampling," *Proceedings of the Ann Asilomar Conference on Circuits, Systems, and Components,* pp. 495-498, 1980.

Ramamoorthy, P.A., "Techniques for Digital Video Image Coding and Implementation Using Systolic Arrays," report to NASA LeRC NAG 3-582, 1986.

Ramamoorthy, P.A. and Tran, T., "A Hybrid Coding Involving ADM and Vector Quantization for Digital Video Image Compression," *Proceedings of the ICASSP '86,* pp. 153-156, 1986.

Ramamoorthy, P.A. and Tran, T., "A Systolic Architecture for Real-Time Composite Video Image Coding," *Proceedings of MILCOM '86,* pp. 49.6.1-4, 1986.

Ramamurthi, B., Gersho, A., and Sekey, A., "Low-Rate Image Coding Using Vector Quantization," *Conference Record of Globecom '83,* pp. 184-187, 1983.

Rao, K.R., Narasimhan, M.A., Revuluri, K, "Image Data Processing by Hadamard-Haar Transform," *IEEE Transactions on Computers,* vol. C-24, pp. 888-896, September 1975.

Reader, C., "Intraframe and Interframe Adaptive Transform Coding," *Proceedings of the SPIE,* vol. 66, pp. 108-117, 1975.

Reader, C. and Chen W-H, "Local Retransmission of Compressed Imagery Data," *Proceedings of the SPIE,* vol. 149, pp. 196-204, 1978.

Reininger, R.C. and Gibson, J.D., "Distributions of the 2-D DCT Coefficients for Images," *IEEE Transactions on Communications,* vol. COM-31, pp. 835-839, June 1983.

Reis, J.J., Lynch, R.T., and Butman, J., "Adaptive Haar Transform Video Bandwidth Reduction System for RPVs," *Proceedings of the SPIE,* vol. 87, pp. 24-35, 1976.

Rissanen, "An Image Compression System," pp. 49.1.1-3, 1986.

Robbins, J.D. and Netravali, A.N., "Interframe Television Coding Using Movement Compensation," pp. 23.4.1-5, 1979.

Robinson, G.S., "Color Edge Detection," *Proceedings of the SPIE,* vol. 87, pp. 126-133, 1976.

Robinson, G.S., "Detection and Coding of Edges Using Directional Masks," *Proceedings of the SPIE* (also in *Optical Engineering*), vol. 87, pp. 117-125, 1976.

Roese, J.A. and Robinson, G.S., "Combined Spatial and Temporal Coding of Digital Image Sequences," *Proceedings of the SPIE,* vol. 66, pp. 172-180, 1975.

Roese, J.A., Pratt, W.K., and Robinson, G.S., "Interframe Cosine Transform Image Coding," *IEEE Transactions on Communications,* vol. COM-25, pp. 1329-1339, November 1977.

Rutledge, C.W., "Vector DPCM: Vector Predictive Coding of Color Images," pp. 1158-1164, 1986.

S

Saghri, J.A. and Tescher, A.G., "Adaptive Transform Coding Based on Chain Coding Concepts," *IEEE Transactions on Communications,* vol. COM-34, pp. 112-117, February 1986.

Saghri, J.A., Tescher, A.G., Habibi, A., "Block Size Considerations for Adaptive Image Coding," pp. E1.2.1-4, 1982.

Saito, T., et al., "Adaptive DCT Image Coding Using Gain/Shape Vector Quantizers," *Proceedings of the ICASSP '86,* pp. 129-132, 1986.

Santago, P. and Rajala, S.A., "Using Convex Set Techniques for Combined Pixel and Frequency Domain . . .," *IEEE Journal on Selected Areas in Communications,* vol. SAC-5, pp. 1127-1139, August 1987.

Sato, M., Ogawa, H., and Iijima, T., "A Theory of Pseudo-Orthogonal Bases and Its Application to Image Transmission," *Proceedings of the SPIE Applications of Digital Image Processing VI,* pp. 38-44, 1983.

Schaming, W.B. and Bessette, O.E., "Empirical Determination of Processing Parameters for Real-Time 2-D DCT Video Bandwidth Compression," *Proceedings of the SPIE,* vol. 249, pp. 78-84, 1980.

Scheinberg, N., et al., "A Composite NTSC Color Video Bandwidth Compressor," *IEEE Transactions on Communications,* vol. COM-32, pp. 1331-1335, December 1984.

Schmidt, E., et al., "Video Image Bandwidth Reduction/Compression Study Final Report," NTIS AD/A-040 033, October 1976.

Schreiber, W.F., "Picture Coding," *Proceedings of the IEEE,* vol. 55, pp. 320-330, March 1967.

Schreiber, W.F., Huang, T.S., et al., "Contour Coding of Images," pp. 443-438, 1972.

Schreiber, W.F., Knapp, C.F., Kay, N.D., "Synthetic Highs - An Experimental TV Bandwidth Reduction System," *Journal of the SMPTE,* vol. 68, pp. 525-537, August 1959.

Schumpert, J.M. and Jenkins, R.J., "A Two-Component Image Coding Scheme Based on 2-D Interpolation and DCT," *Proceedings of the ICASSP 83,* pp. 1232-1235, 1983.

Shapiro, S.D., "Use of the Hough Transform for Image Data Compression," pp. 576-582, 1979.

Siegel, L.J., et al., "Block Truncation Coding on PASM," *Proc Ann Allerton Conf on CCC,* pp. 891-900, 1982.

Spencer, D.J., "Video Data Processor (VIDAP): A Real-Time Video Data Compressor," *Proceedings of the SPIE,* vol. 207, pp. 284-289, 1979.

Spencer, D.J. and Anderson, J.M., "A Real Time Video Bandwidth Reduction System Based on a CCD Hadamard Transform," *Proceedings of NAECON '79,* vol. 3, pp. 1218-1231, 1979.

Srinivasan, R. and Rao, K.R., "Motion-Compensated Coder for Videoconferencing," *IEEE Transactions on Communications,* vol. COM-35, pp. 297-304, March 1987.

Srinivasan, S., Jain, A.K., and Chin, T.M., "Cosine Transform Block Codec for Images Using TMS 32010," *Proceedings of the IEEE International Symposium on Circuits and Systems,* vol. 1, pp. 299-302, 1986.

Strickland, R.N. and Smith, W.E., "Stationary Transform Processing of Digital Images for Data Compression," *Applied Optics,* vol. 22, pp. 2161-2168, 15 July 1983.

Stuller, J.A. and Netravali, A.N., "Transform Domain Motion Estimation," *Bell System Technical Journal,* vol. 58, pp. 1673-1702, September 1979.

Sun, M.T., Chen, T.C., et al., "A 16 x 16 Discrete Cosine Transform Chip," *Proceedings of the SPIE,* vol. 845, pp. 13-18, 1987.

Sun, M.T., Wu, L., and Liou, M.L., "A Concurrent Architecture for VLSI Implementation of DCT," *IEEE Transactions on Circuits and Systems,* vol CAS-34, pp. 992-994, August 1987.

T

Tanaka, H., Matsuo, H., and Kaneku, S., "Efficient Encoding of Sources with Memory," *IEEE Transactions on Information Theory,* vol. IT-25, pp. 221-225, March 1979.

Tanimoto, M., et al., "A New Bandwidth Compression System of Picture Signals - The TAT," *Proceedings of GLOBECOM '85,* pp. 427-431, 1985.

Tasto, M. and Wintz, P.A., "Image Coding by Adaptive Block Quantization," *IEEE Transactions on Communication Technology,* vol. COM-19, pp. 957-971, December 1971.

Tescher, A.G., "Adaptive Transform Coding of Color Images at Low Rates," *Proceedings of the NTC '80,* pp. 36.3.1-4, 1980.

Tescher, A.G. and Cox, R.V., "An Adaptive Transform Coding Algorithm," pp. 47.20-23.

Tescher, A.G. and Cox, R.V., "Image Coding: Variable Rate DPCM Through Fixed Rate Channel," *Proceedings of the SPIE,* vol. 119, pp. 147-154, 1977.

Tescher, A.G. and Saghri, J.A., "Adaptive Transform Coding and Image Quality," *Optical Engineering,* vol. 25, pp. 979-983, August 1986.

Tewksbury, S.K., et al., "FIR Digital Filters for High Sample Rate Applications," IEEE *Communications* Magazine, vol. 25, pp. 62-72, July 1987.

Thompson, J.E., "Data Compression of Composite Color TV Signals Using Planar Prediction," *International Conference on Digital Satellite Communications,* pp. 315-321, 1975.

Thyagarajan, K.S. and Viswananthan, M., "Matrix Quantization of Homomorphically Processed Images," pp. 49.3.1-5, 1986.

Torre, V. and Poggio, T., "On Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-8, pp. 147-163, March 1986.

Troxel, D.E., "Application of Pseudorandom Noise to DPCM," *IEEE Transactions on Communications,* vol. COM-29, pp. 1763-1767, December 1981.

Troxel, D.E., et al., "Bandwidth Compression of High Quality Images," *ICCC '80 Conference Record,* vol. 2, pp. 31.9.1-5, June 1980.

Troxel, D.E., et al., "A Two-Channel Picture Coding System: I - Real-Time Implementation," *IEEE Transactions on Communications,* vol. COM-29, pp. 1841-1848, December 1981.

Ty, K.M. and Venetsanopoulos, A.N., "A Fast Filter for Real-Time Image Processing," *IEEE Transactions on Circuits and Systems,* vol. CAS-33, pp. 948-957, October 1986.

U

Udpikar, V.R. and Raina, J.P., "BTC Image Coding Using Vector Quantization," *IEEE Transactions on Communications,* vol. COM-35, pp. 352-356, March 1987.

Un, C.K. and Cynn, M.H., "A Performance Comparison of ADPCM and ADM Coders," *National Telecommunications Conference,* pp. 50.1.1-5, 1980.

V

Venkataraman, S. and Rao, K.R., "Applications of Vector Quantizers and BTC in Image Coding," *GLOBECOM '85 Conference Record,* pp. 602-608, 1985.

W

Walker, D.R. and Rao, K.R., "Motion-Compensated Coder," *IEEE Transactions on Communications,* vol. COM-35, pp. 1171-1177, November 1987.

Wandell, B., "The Synthesis and Analysis of Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, pp. 2-13, January 1987.

Wang, L. and Goldberg, M., "Progressive Image Transmission by Residual Error VQ in Transform Domain," pp. 178-182, 1987.

Wang, R.T.P., "RPV Video Communications: A New Challenge to Video Data Compression," *Proceedings of the SPIE,* vol. 87, pp. 36-46, 1976.

Welch, Terry, "A Technique for High-Performance Data Compression," *Computer,* pp 8-19, June 1984

Weidong, K., "A Transform Product Code VQ for Digital Image Coding," pp. 1170-1173, 1986.

Weidong, K. and Zheng, H., "Vector Quantization Incorporating Orthogonal Transforms and Its Applications to Image Coding," *GLOBECOM '85 Conference Record,* pp. 609-612, 1985.

Whitehouse, H., et al., "A Digital Real Time Intraframe Video Bandwidth Compression System," *Proceedings of the SPIE,* vol. 119, pp. 64-78, 1977.

Whitehouse, H.J., Means, R.W., Wrench, "Real Time Television Image Bandwidth Reduction Using Charge Transfer Devices," *Proceedings of the SPIE,* vol. 66, pp. 36-47, 1975.

Whiteman, P., et al., "Hardware Systems Design of an Airborne Video Bandwidth Compressor," *Proceedings of the SPIE,* vol. 309, pp. 93-103, 1981.

Wilson, R., et al., "Anisotropic Nonstationary Image Estimation and Its Applications: Part II," *IEEE Transactions on Communications,* vol. COM-31, pp. 398-406, March 1983.

Wilson, R., Knutson, H., Granlund, G.H., "Image Coding Using a Predictor Controlled by Image Content," pp. 432-435, 1982.

Wintz, P.A., "Transform Picture Coding," *Proceedings of the IEEE,* vol. 60, pp. 809-820, July 1972.

Woods, J.W. and Paul, I., "Adaptive Prediction DPCM Coding of Images," pp. 31.8.1-5, 1980.

Y

Yasuda, H., Kawanishi, H., "Predictor Adaptive Differential Pulse Code Modulation," *Proceedings of the SPIE,* vol. 149, pp. 189-195, 1978.

Yoshida, S. et al., "Pel Pattern Predictive Coding of Dithered Images," pp. B4.7.1-5, 1983.

Z

Zdepski, J. and Daut, D.G., "An Image Vector Quantizer Using a Perceptually-Based Preprocessor," pp. 1150-1157, 1986.

Ziv, J. and Lempel, A., "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory,* vol. IT-23, No. 3, pp. 337-343, May 1977.

Zschunke, W., "DPCM Picture Coding with Adaptive Prediction," *IEEE Transactions on Communications,* vol. COM-25, pp. 1295-1302, November 1977.

Zuniga, O.A., Haralick, R.M., and Klein, R.L., "Adaptive Image Data Compression," *Proceedings of the Conference on Pattern Recognition and Image Processing,* pp. 583-590, 1979.

APPENDIX 1: BIBLIOGRAPHY DATABASE

The bibliography, resulting from the Task 2.0 study, was prepared using R:BASE System V software by Microrim Inc. for the MS-DOS operating system. The database files, the applications files, and the program files are contained on the disk provided with the this report and represents the required deliverable for this study.

The R:BASE files containing the list of articles are ARTICLE?.RBF. The R:BASE application for entering articles and printing the bibliography list is contained in IOIO.APP, IOIO.API, and IOIO.APX.

Load R:BASE. At the R:BASE Main Menu select

```
(1)  R:BASE command mode.
```

The program can be run by typing *run ioio in ioio.apx* from the R> prompt. The program used by *ioio* is REFLIST. The program can be run using menus. The initial menu is shown below.

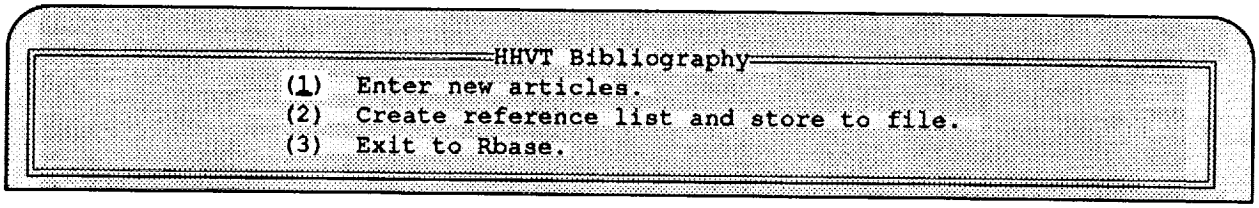


Figure 16: HHVT bibliography menu

Highlight `(1) Enter new articles.` Press [↵ Enter].
The next menu to appear selects the appropriate list.

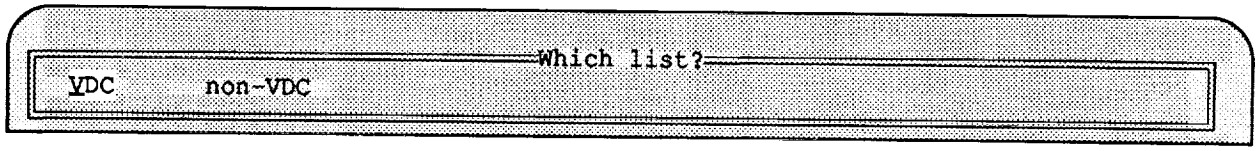


Figure 17: List selection menu

Highlight `VDC`. Press [↵ Enter].

Next, a screen will show where you can add or list data entries in the Bibliography Database. The database contains the following information for each article:

```
Reference Number
Author(s)
Title
Journal
Volume
Pages
Date
Category
Method
```

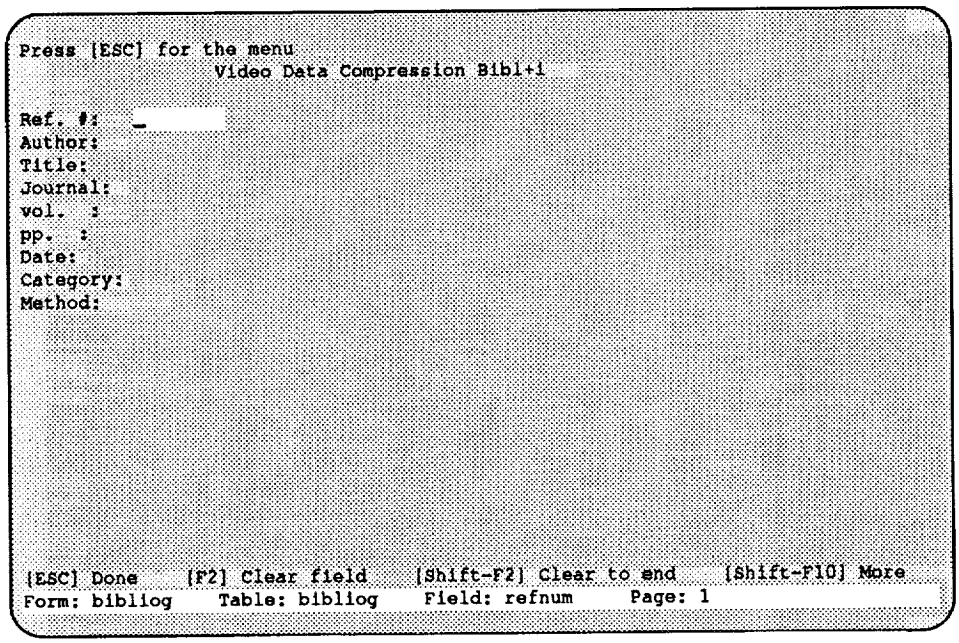


Figure 18: Bibliography data input listing prompts

Each article is identified using descriptors for the general groupings into categories, and a descriptor for the compression methods used in the article. The category descriptors used are

```
Block
HVS
Hybrid
Implementation
Predictive
Reversible
Survey
Transform
```

The menu across the bottom of the screen displays key strokes which modify the database files. When you have completed making changes or additions to the database, press Esc. The following menu appears across the screen top.

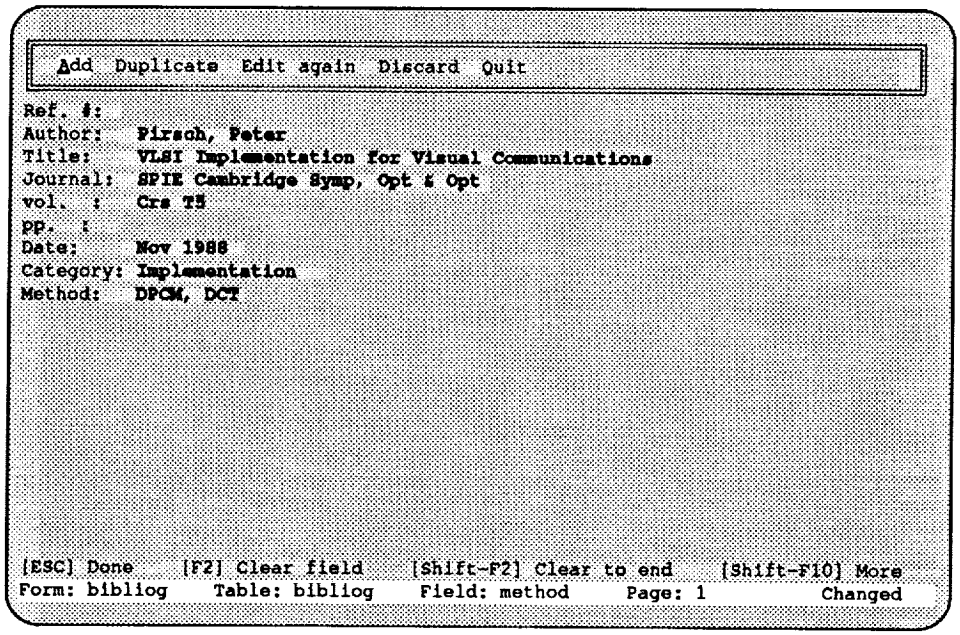


Figure 19: Completed example of bibliography data input

Highlight Add. Press ↵ Enter. This enters the changes to the database. Press Esc three times to return to the HHVT Bibliography menu.

Highlighting item (2) from this menu will generate an ASCII text file from the database.

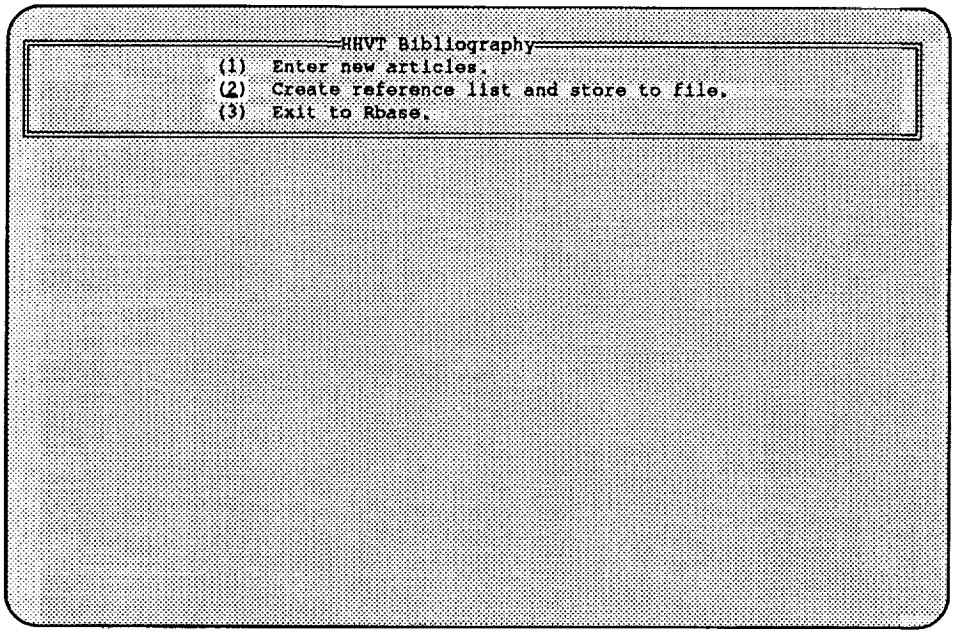


Figure 20: Create a reference list menu

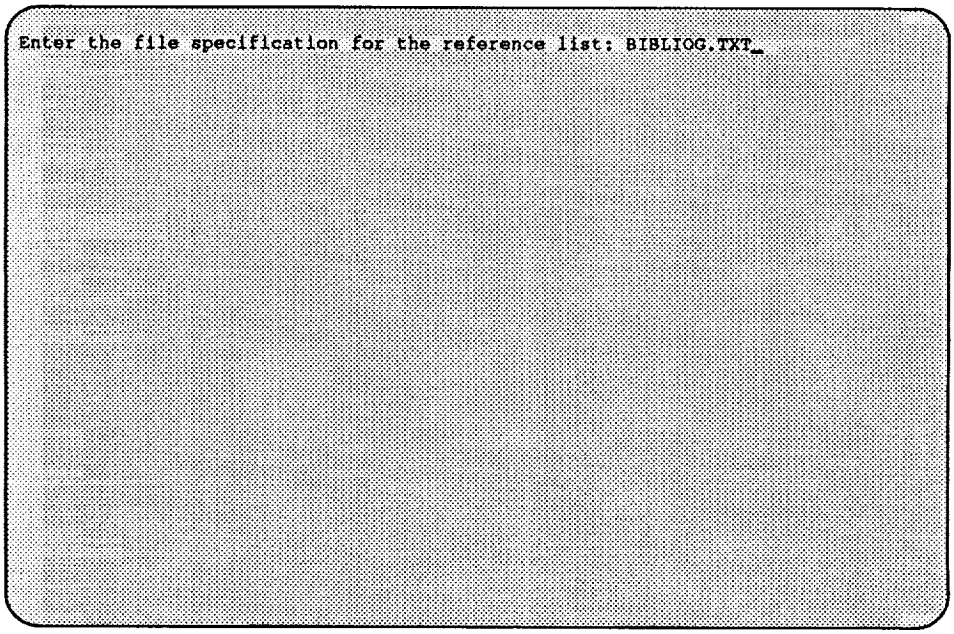


Figure 21: Enter a filename for the ASCII text file

At this screen, in Figure 21, enter the name of the file where R:BASE will store the ASCII text file. In the example shown, the file is named `BIBLIOG.TXT`. This file can be used in a word processor to prepare and print out a bibliography listing.

The following menu demonstrates how to exit from the Database program. Select item `(3)` and press `↵ Enter`. At the `R>` prompt, type `exit`. This is shown in the following figure. Again, the DOS prompt will appear.

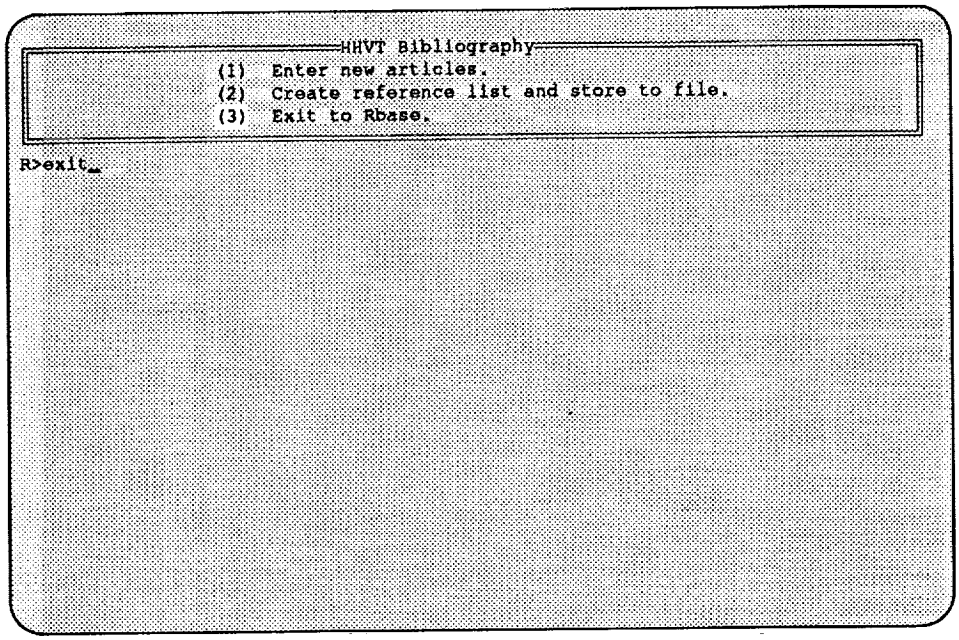


Figure 22: Exiting the Bibliography Database

The database can be queried using R:BASE to manipulate the data, and any sorted and/or selected subset of the bibliography can be presented. For detailed information on how to query the database, refer to the R:BASE manuals.

APPENDIX 2: COMPRESSION PROGRAM

Three computer programs were written to aid in the comparative evaluation of video compression algorithms. The programs were written using QuickBASIC 4.0 by Microsoft. The first of the three, 2D_TRNSF.BAS, is used to evaluate the Cosine, Fast Walsh-Hadamard, and Slant transform algorithms. It is based on an original image using 8 bpp. The 2D_TRNSF.BAS can be used for comparing results of transform compression algorithms by varying the following parameters:

1. Image size, (Random, 8 x 8, 16 x 16, 32 x 32, 64 x 64, and 128 x 128; Bubble, 8 x 8, and 16 x 16; Edge, 8 x 8)
2. Block size, (8 x 8, and 16 x 16)
3. Image, (Random, Bubble, and Edge)
4. Transform algorithm, (DCT, WHT, SLANT, No Transform)
5. Compression, (3, 1, 0.5 bpp)
6. Quantization scheme, (Linear, Max/Gaussian)

The program 2D_TRNSF is listed in the following section. There are two program files for 2D_TRNSF, an executable compiled version of the program which can be run by typing *2D_TRNSF* at the DOS prompt, and a file which can be run using QuickBASIC. The program execution, input and output, and user interface are the same. Use of the program will be illustrated using QuickBASIC (for detailed information refer to the QuickBASIC users' manual).

The QuickBASIC screen is shown in Figure 23.

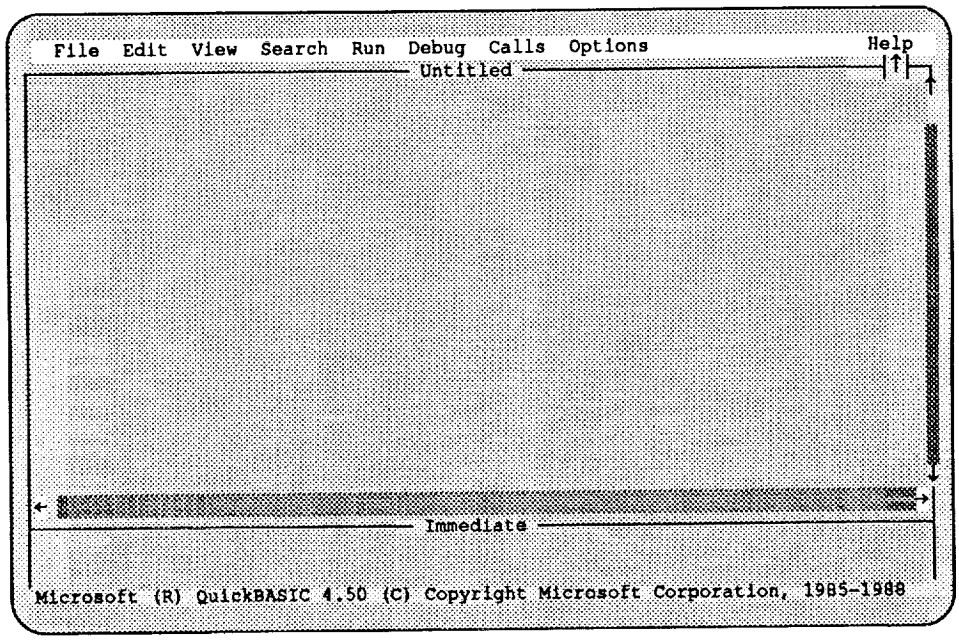


Figure 23: The QuickBASIC screen

The QuickBASIC menu used to open the program is shown in Figure 24. `Open Program` is selected using the cursor movement keys, or using a mouse.

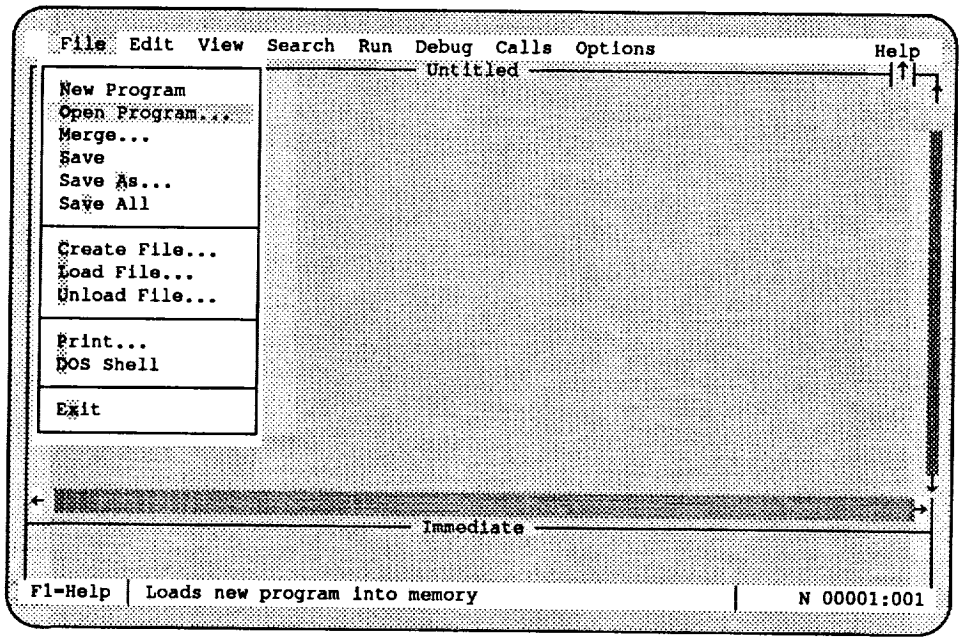


Figure 24: QuickBASIC menu, OPEN PROGRAM

A listing of files will be shown as illustrated in Figure 25. The selected file is typed, or placed in the `File Name:` box and entered.

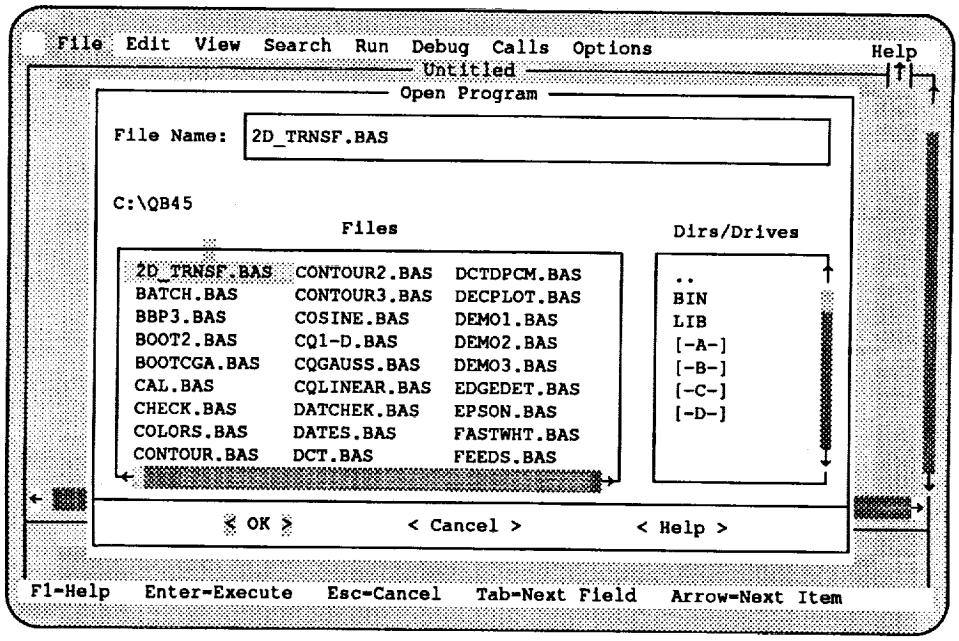


Figure 25: 2D_TRNSF.BAS program selected from list

The selected program will be listed as shown in Figure 26.

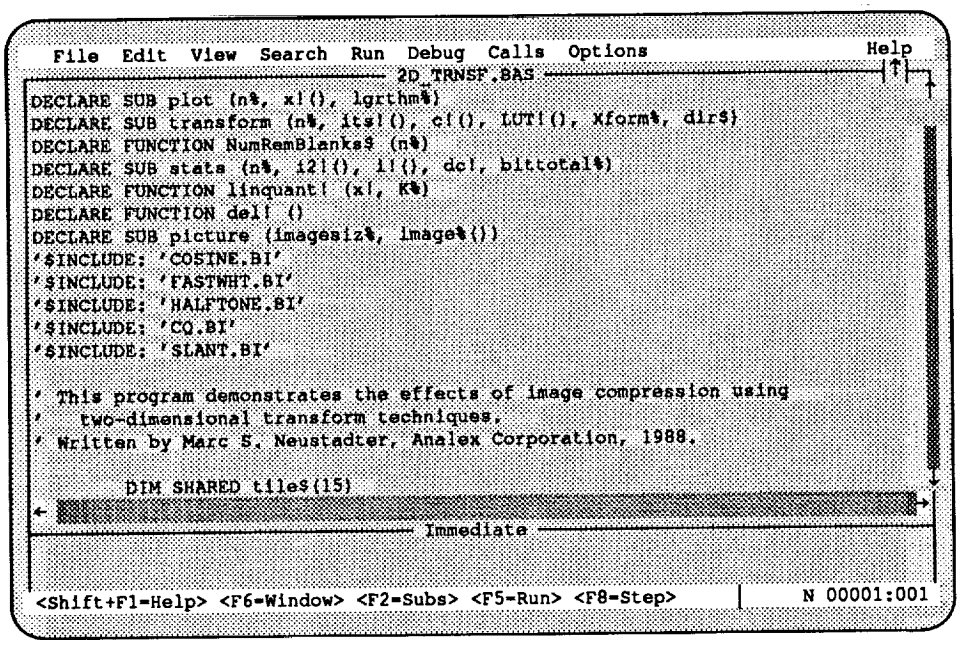


Figure 26: QuickBASIC program, 2D_TRNSF.BAS, loaded

The program is run by selecting `Run`, and from the `Run` menu selecting `Start`, shown in Figure 27.

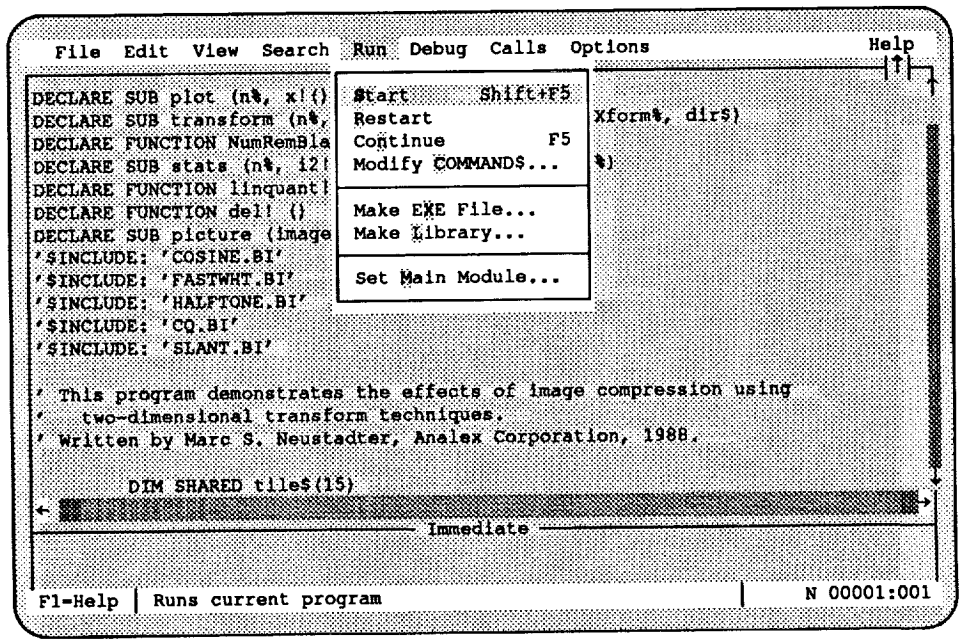


Figure 27: RUN PROGRAM menu

The program will prompt the user for inputs. The program prompts with example responses are shown in the following:

```
Enter the image size: 16
Enter the block size: 16

Is the image in a file? N
```

If the last input had been *Y*, the program would prompt for an input file name. The program at this point produces a 16 x 16 image of pixels having a random distribution of intensities, where the intensities are based on a first order Markov process. The program continues with the following prompts:

```
Which transform would you like to use?
     1. Discrete Cosine Transform (DCT)
     2. Walsh-Hadamard Transform (WHT) [8 or 16 only]
     3. Slant Transform [8 only]
     4. No Transform
     5. Exit

Enter the number of your choice: 1
```

The next set of program prompts will be:

```
Which bit allocation would you like to use?
     1. 3 bpp, uniformly distributed
     2. 3 bpp, sub-optimally distributed
     3. 1 bpp, sub-optimally distributed
     4. 0.5 bpp, sub-optimally distributed
     5. Exit

Enter the number of your choice: 1
```

This selection sets the compression, from 8 bpp to 3 bpp. Finally the program prompts will be:

```
Which quantization scheme would you like to use?
     A. Linear (uniform) quantization
     B. Max quantizer based on Gaussian distribution
Enter the letter of your choice: B
```

This completes the requests for inputs. The program in response to this set of inputs will produce an output display as shown in Figure 28. The output display contains a vertical column in the upper left of the figure. This column is used to show the pixel intensity level key. The four square blocks, proceeding clockwise from upper left, are

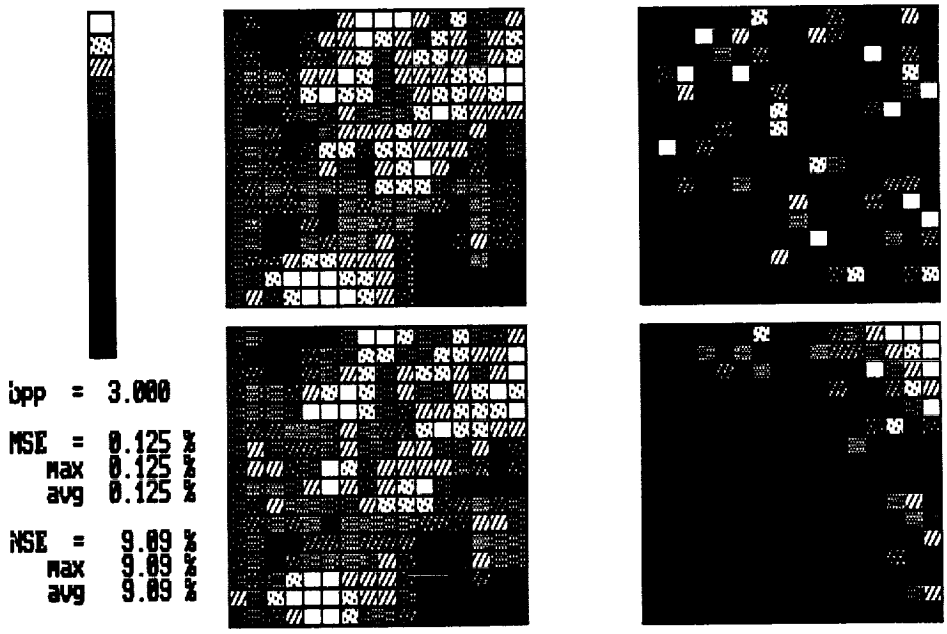


Figure 28: Program output for a random pattern

Upper left: First block, 16 x 16, of selected image
Upper right: 16 x 16 block of transform coefficients. The coefficients are ordered by frequency, with the lowest frequency term (DC) in lower right corner, increasing moving up, and increasing moving to the left.
Lower right: The truncated set of transform coefficients.
Lower left: The image, reconstructed from the truncated set of transform coefficients.

The table in the lower left of the figure lists the following results:

bpp = 3.00 (The average number of bits per pixel in the compressed image.)

MSE = 0.125 % (The Mean Square Error for the compressed block.)
max 0.125 % (Maximum average block error. Used where the block size is smaller than the image size, requiring more than one block to complete the image.)
avg 0.125 % (Combined average error for the number of blocks in the image. Used where the block size is smaller than the image, requiring more than one block to complete the image.)
NSE = 9.09% (Normalized Mean Square Error. Error Normalized with respect to the square of the maximum intensity level.)
max 9.09 % (Maximum NSE for block. Used where the block size is smaller than the image.)

avg 9.09 % (Combined average NSE for the number of blocks used in the image, where the block size is smaller than the image.)

Figure 29 shows the output display for a stored image of a bubble superimposed on a background of random pattern of intensities. The input, in response to the program prompt, is shown below:

```
Is the image in a file  Y
Enter the filename:  BUBBLE.IMG
```

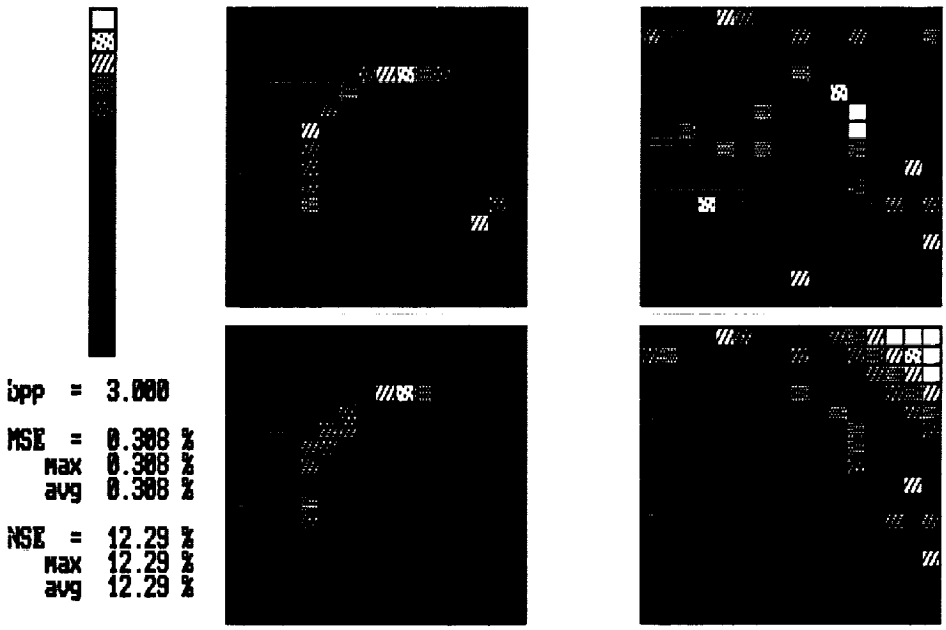


Figure 29: Program output for a bubble pattern

Figures 30 and 31 show successive output displays for the file BUBBLE.IMG, where the image size is 16 x 16 and the block size is 8 x 8. There are 4 blocks required to complete the 16 x 16 image. The first block is shown in Figure 30. Three successive displays for the three remaining blocks are obtained by depressing the space bar. Note the change in error statistics between the two figures.

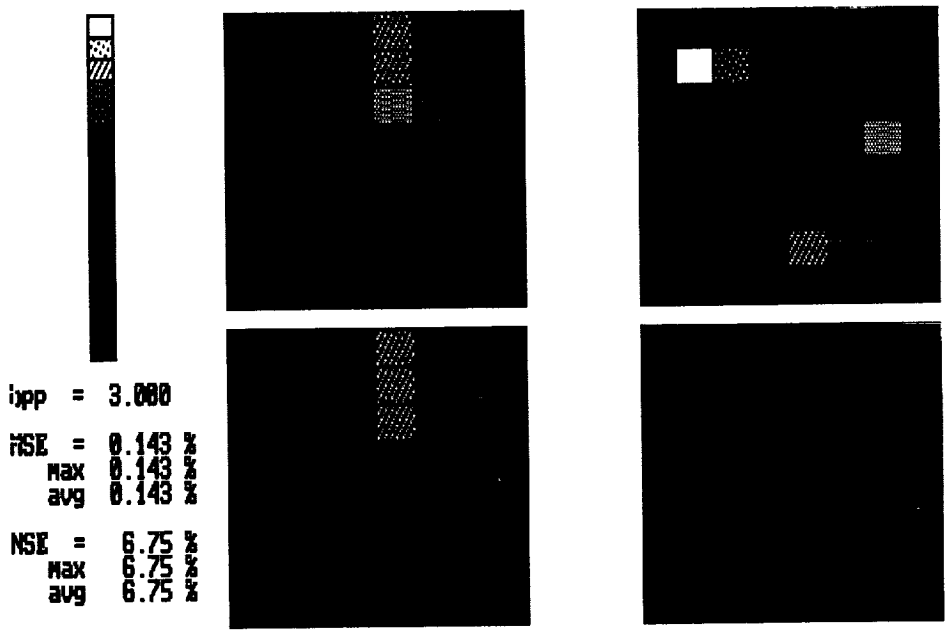


Figure 30: Program output for a bubble pattern using an 8 x 8 block DCT

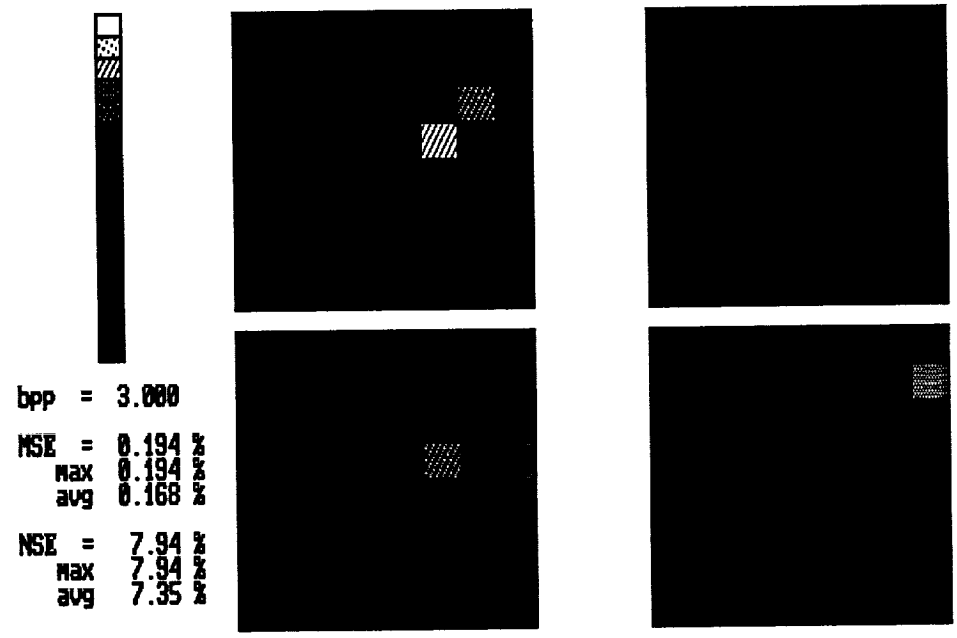


Figure 31: Program output for the second quarter of bubble pattern image using an 8 x 8 block DCT

Figure 32 shows the output display for the file EDGE1508.IMG. The display is for an image size of 8 x 8; a block size of 8 x 8; using the DCT, 3 bpp, and the Max quantizer.

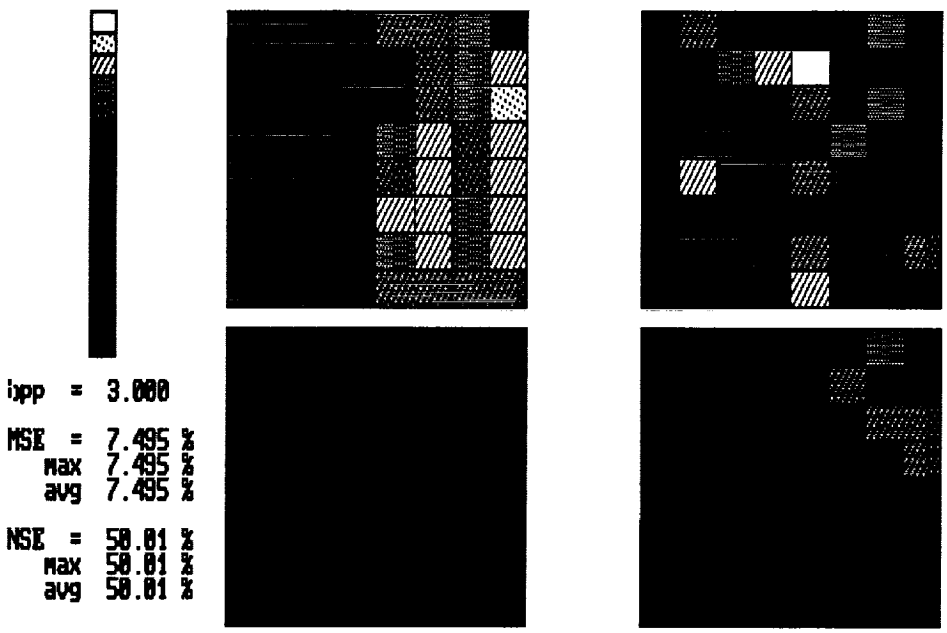


Figure 32: Program output for an edge pattern using an 8 x 8 block DCT

Figure 33 shows the File menu, selecting `Exit` ends operation of the program.

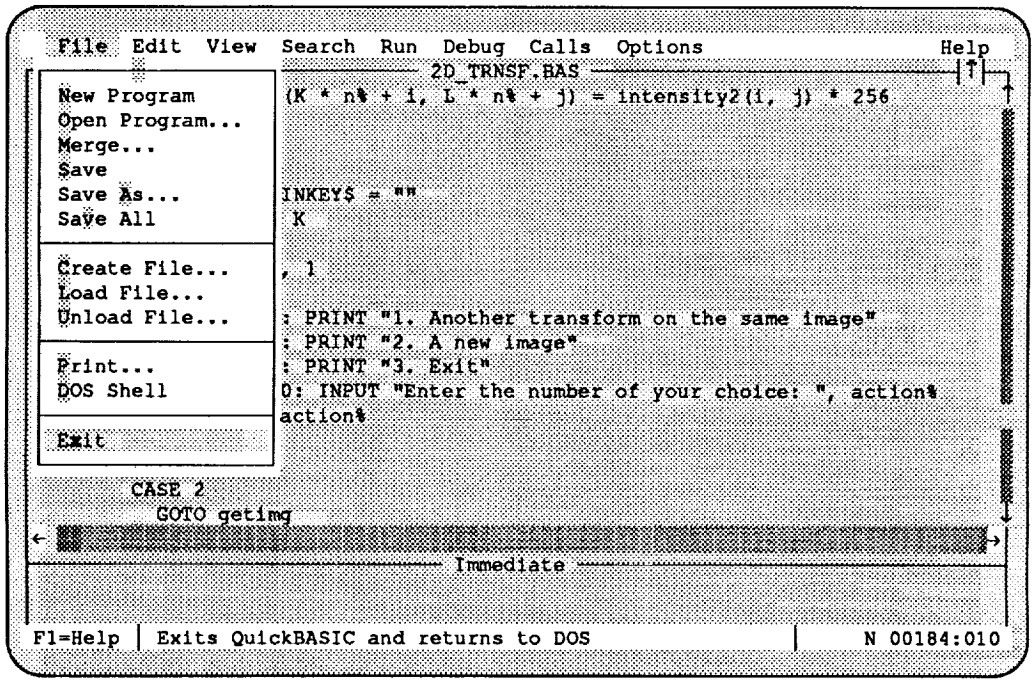


Figure 33: File menu, EXIT

The second program for evaluation of video compression is used to evaluate the use of one dimensional transforms with DPCM. There are two program files, XFRMDPCM.BAS which can be run using QuickBASIC, and an executable file which can be run by typing *xfrmdpcm* at the DOS prompt. The program will operate with image sizes of 8 x 8 and 16 x 16. The block size is restricted to 8 x 8.

The third program, EDGEDET.BAS, is a simple edge detection algorithm. It operates with an image size of 16 x 16 using BUBBLE.IMG, and EDGE.IMG using an image size of 8 x 8. There is not an executable file for this program, therefore it must be run using QuickBASIC.

The enclosed disk contains all of the files, listed in Figure 34, in a directory named QBFILES.

1%2B16F .DAT	530	11/17/88 14:11	1%2B8F .DAT	138	11/29/88 15:32
1B16F .DAT	530	11/17/88 13:28	1B8F .DAT	138	11/29/88 15:45
2D_TRNSF.BAS	13168	02/06/89 13:36	2D_TRNSF.EXE	88910	12/02/88 14:04
2D_TRNSF.MAK	77	02/06/89 13:36	2D_TRNSF.RSP	67	12/02/88 14:04
3B16E .DAT	530	11/18/88 11:35	3B16F .DAT	530	11/16/88 12:27
3B8E .DAT	138	11/18/88 11:37	3B8EH .DAT	26	11/22/88 17:21
3B8F .DAT	138	11/18/88 11:58	3B8FH .DAT	26	11/22/88 17:23
BUBBLE .IMG	1263	11/28/88 16:16	COSINE .BAS	1510	02/06/89 16:43
COSINE .BI	158	11/29/88 15:19	CQ .BI	76	12/07/88 10:06
CQ1-D .BAS	2695	11/22/88 12:02	CQ1-D .BI	41	11/22/88 12:26
CQGAUSS .BAS	3457	02/06/89 16:43	CQLINEAR.BAS	1225	11/15/88 11:07
DCT .BAS	2462	12/13/88 10:45	DCTDPCM .BAS	2346	11/28/88 12:31
DCTDPCM .BI	115	11/28/88 11:37	EDGE1508.IMG	288	12/07/88 09:09
EDGE50_8.IMG	320	12/07/88 09:08	EDGEDET .BAS	2719	11/30/88 10:01
FASTWHT .BAS	1757	02/06/89 16:43	FASTWHT .BI	48	11/29/88 13:55
HADAMARD.BAS	2080	11/16/88 16:52	HADAMARD.BI	48	11/29/88 13:55
HALFTONE.BAS	2706	02/06/89 16:43	HALFTONE.BI	34	11/14/88 15:20
IMAGE .BAS	3304	12/07/88 09:09	ORIGINAL.IMG	1263	02/07/89 11:43
QBFILES .SCT	2896	02/13/89 15:14	QBFILES_.SCT	1136	02/13/89 15:16
RECEIVED.IMG	1265	02/07/89 11:43	SLANT .BAS	1410	02/06/89 16:43
SLANT .BI	109	11/30/88 09:42	SOBEL .BAS	1116	11/18/88 16:04
THATONE .BAS	13412	02/06/89 16:43	THATONE .MAK	76	02/06/89 16:43
WHT-DPCM.BAS	2497	11/22/88 16:08	WHTDPCM .BAS	2497	11/28/88 10:51
WHTDPCM .BI	115	11/28/88 11:37	XFRMDPCM.BAS	11638	12/07/88 10:48
XFRMDPCM.EXE	81650	11/28/88 17:12	XFRMDPCM.MAK	54	12/07/88 10:48
XFRMDPCM.RSP	77	11/28/88 16:58			

Figure 34: List of files in QBFILES directory

The files include executable files for 2D_TRNSF and XFRMDPCM, and corresponding files which can be run in QuickBASIC. In addition, there is the The EDGEDET.BAS which can be run in QuickBASIC. The files also include all of the necessary FILENAME.BAS subroutine files, IMAGEFILE.IMG files for the bubble and edge images, XFRM.BI files for the transforms, and the TRUNC.DAT files used in compressing the transform coefficient arrays.

2D_TRNSF.BAS

```
DECLARE SUB plot (n%, x!(), lgrthm%)
DECLARE SUB transform (n%, its!(), c!(), LUT!(), Xform%, dir$)
DECLARE FUNCTION NumRemBlanks$ (n%)
DECLARE SUB stats (n%, i2!(), i!(), dc!, bittotal%)
DECLARE FUNCTION linquant! (x!, K%)
DECLARE FUNCTION del! ()
DECLARE SUB picture (imagesiz%, image%())
'$INCLUDE: 'COSINE.BI'
'$INCLUDE: 'FASTWHT.BI'
'$INCLUDE: 'HALFTONE.BI'
'$INCLUDE: 'CQ.BI'
'$INCLUDE: 'SLANT.BI'

' This program demonstrates the effects of image compression using
'   two-dimensional transform techniques.
' Written by Marc S. Neustadter, Analex Corporation, 1988.
        DIM SHARED tile$(15)
' This statement should be removed to produce the identical image every time.
        RANDOMIZE TIMER
' This subroutine call produces the tiles for the gray scale.
' See the documentation on the subroutine 'halftone' for details.
        CALL halftone(tile$())
' This section prompts for the image size (# of pixels in each dimension)
'   and the transform block size (# of pixels).  The arrays are then
'   dimensioned appropriately.
getimg: SCREEN , , 1, 1
        CLS 0
        LOCATE 7, 10: INPUT "Enter the image size: ", imagesiz%
        LOCATE 8, 10: INPUT "Enter the block size: ", n%     ' The block size
is n% x n%.
        REDIM image%(imagesiz% - 1, imagesiz% - 1), B%(n% - 1, n% - 1),
image2%(imagesiz% - 1, imagesiz% - 1)
        REDIM intensity(n% - 1, n% - 1), coeff(n% - 1, n% - 1)
        REDIM intensity2(n% - 1, n% - 1), LUT(n% - 1, n% - 1), ilut(n% - 1, n%
- 1)
' This section offers the choice of using a previously stored image
'   or producing a new one.
       LOCATE 10, 10: INPUT "Is the image in a file"; YorN$
       IF UCASE$(LEFT$(YorN$, 1)) = "Y" THEN
         LOCATE 11, 10: INPUT "Enter the filename: ", imfile$
         OPEN imfile$ FOR INPUT AS #3
         FOR i = 0 TO imagesiz% - 1
           FOR j = 0 TO imagesiz% - 1
                INPUT #3, image%(i, j)
           NEXT j
         NEXT i
         CLOSE #3
       ELSE
          CALL picture(imagesiz%, image%())       ' Form the original image.
       END IF
```

```
menu:   CLS 0
        LOCATE 6, 10: PRINT "Which transform would you like to use?"
        LOCATE 7, 15: PRINT "1. Discrete Cosine Transform (DCT)"
        LOCATE 8, 15: PRINT "2. Walsh-Hadamard Transform (WHT) [8 or 16 only]"
        LOCATE 9, 15: PRINT "3. Slant Transform [8 only]"
        LOCATE 10, 15: PRINT "4. No Transform"
        LOCATE 11, 15: PRINT "5. Exit"
        LOCATE 13, 10: INPUT "Enter the number of your choice: ", Xform%
        CLS 0
        LOCATE 6, 10: PRINT "Which bit allocation would you like to use?"
        LOCATE 7, 15: PRINT "1. 3 bpp, uniformly distributed"
        LOCATE 8, 15: PRINT "2. 3 bpp, sub-optimally distributed"
        LOCATE 9, 15: PRINT "3. 1 bpp, sub-optimally distributed"
        LOCATE 10, 15: PRINT "4. 0.5 bpp, sub-optimally distributed"
        LOCATE 11, 15: PRINT "5. Exit"
selal:  PRINT : LOCATE , 10: INPUT "Enter the number of your choice: ", alloc%
' The files of bit allocations must exist where they can be found.
        SELECT CASE alloc%
        CASE 1
         file$ = "3b" + NumRemBlanks$(n%) + "e"
        CASE 2
         file$ = "3b" + NumRemBlanks$(n%) + "f"
        CASE 3
         file$ = "1b" + NumRemBlanks$(n%) + "f"
        CASE 4
         file$ = "1%2b" + NumRemBlanks$(n%) + "f"
        CASE 5
         GOTO getout
        CASE ELSE
         PRINT "Invalid choice"
         GOTO selal
        END SELECT
        bittotal% = 0
        ON ERROR GOTO Handler
' Loads the bit allocation.
        OPEN file$ + ".dat" FOR INPUT AS #1
        ON ERROR GOTO 0
        FOR i = 0 TO n% - 1
        FOR j = 0 TO n% - 1
          INPUT #1, B%(i, j)
          bittotal% = bittotal% + B%(i, j)
        NEXT j
        NEXT i
        CLOSE #1
        LOCATE 13, 10: PRINT "Which quantization scheme would you like to
use?"
        LOCATE 14, 15: PRINT "A. Linear (uniform) quantization"
        LOCATE 15, 15: PRINT "B. Max quantizer based on Gaussian distribution"
        PRINT : LOCATE 17, 10: INPUT "Enter the letter of your choice: ",
quant$
        quant$ = UCASE$(quant$)
' Initializes the display.
        SCREEN 8, , 2, 2                        ' EGA 640 x 200 with tiling.
        CLS 0
        WINDOW
        VIEW
```

```
' This loop puts the gray scale on the screen.
        FOR K = 0 TO 15
          LINE (52, 7 * K)-(68, 7 * K + 7), 3, B
          PAINT (59, 7 * K + 5), tile$(K), 3
        NEXT K
        WINDOW (0, 0)-(n%, n%)
        FOR K = 0 TO imagesiz% \ n% - 1
        FOR L = 0 TO imagesiz% \ n% - 1
         FOR i = 0 TO n% - 1                          ' Get the current block.
           FOR j = 0 TO n% - 1
             intensity(i, j) = image%(K * n% + i, L * n% + j) / 256
           NEXT j
         NEXT i
         VIEW (140, 96)-(333, 0)
         CALL plot(n%, intensity(), 0)                ' Plot the block.
' This section calls the subroutines to produce look-up tables for the
'  forward and inverse transforms.  The fast WHT doesn't need a look-up table.
        SELECT CASE Xform%
        CASE 1
          CALL mklutcos(n%, LUT(), ilut())
        CASE 2
        CASE 3
          CALL mklutslant(n%, LUT(), ilut())
        CASE 4
          FOR i = 0 TO n% - 1
           FOR j = 0 TO n% - 1
             intensity2(i, j) = linquant(intensity(i, j), B%(i, j))
           NEXT j
          NEXT i
          GOTO recon
        CASE ELSE
          GOTO getout
        END SELECT
' Transform the block.
        CALL transform(n%, intensity(), coeff(), LUT(), Xform%, "for")
        VIEW (407, 96)-(600, 0)
        CALL plot(n%, coeff(), 1)                     ' Plot the coefficients.
        IF quant$ = "B" THEN
          CALL cquant(n%, coeff(), B%())              ' Quantize the coefficients.
        ELSE
          CALL clquant(n%, coeff(), B%())
        END IF
        VIEW (407, 199)-(600, 103)
        CALL plot(n%, coeff(), 1)                     ' Plot the quantized coeffs.
' Inverse transform
        CALL transform(n%, coeff(), intensity2(), ilut(), Xform%, "inv")

        FOR i = 0 TO n% - 1                           ' Quantize the reconstructed
          FOR j = 0 TO n% - 1                         '   image to 8 bits.
            intensity2(i, j) = linquant(intensity2(i, j), 8)
          NEXT j
        NEXT i
```

```
recon:    VIEW (140, 199)-(333, 103)
          CALL plot(n%, intensity2(), 0)          ' Plot the reconstructed
block.
          CALL stats(n%, intensity2(), intensity(), coeff(0, 0) / n%,
bittotal%)

' Stores the current block in the reconstructed image.
          FOR i = 0 TO n% - 1
            FOR j = 0 TO n% - 1
              image2%(K * n% + i, L * n% + j) = intensity2(i, j) * 256
            NEXT j
          NEXT i
          DO
          LOOP WHILE INKEY$ = ""
        NEXT L: NEXT K
        SCREEN , , 1, 1
        CLS 0
        LOCATE 7, 15: PRINT "1. Another transform on the same image"
        LOCATE 8, 15: PRINT "2. A new image"
        LOCATE 9, 15: PRINT "3. Exit"
        LOCATE 11, 10: INPUT "Enter the number of your choice: ", action%
        SELECT CASE action%
        CASE 1
          GOTO menu
        CASE 2
          GOTO getimg
        CASE ELSE
          OPEN "original.img" FOR OUTPUT AS #2
          FOR i = 0 TO imagesiz% - 1
            FOR j = 0 TO imagesiz% - 1
              WRITE #2, image%(i, j)
            NEXT j
          NEXT i
          CLOSE #2
          OPEN "received.img" FOR OUTPUT AS #2
          FOR i = 0 TO imagesiz% - 1
            FOR j = 0 TO imagesiz% - 1
              WRITE #2, image2%(i, j)
            NEXT j
          NEXT i
          CLOSE #2
          SCREEN 0
        END SELECT
getout: END
Handler:        'Error handling routine.
        errnum = ERR
        IF errnum = 53 THEN     'file not found
          CLOSE #1
          PRINT "This choice is not available for the current block size."
          PRINT "Please make another choice."
          RESUME selal
        ELSE
          ERROR errnum
        END IF
```

```
FUNCTION del
                ' This function returns a value to be used as
                '   the increment between succesive pixels.
                ' It approximates a Gaussian distribution.
r = 2 * RND(1) - 1
SELECT CASE 10 * RND(1)
CASE IS <= 1#
  d = .0315 * r
CASE 1# TO 2#
  d = .0318 * r + SGN(r) * .0315
CASE 2# TO 3#
  d = .0332 * r + SGN(r) * .0633
CASE 3# TO 4#
  d = .0345 * r + SGN(r) * .0965
CASE 4# TO 5#
  d = .038 * r + SGN(r) * .131
CASE 5# TO 6#
  d = .042 * r + SGN(r) * .169
CASE 6# TO 6.5#
  d = .023 * r + SGN(r) * .211
CASE 6.5# TO 7#
  d = .025 * r + SGN(r) * .234
CASE 7# TO 7.5#
  d = .029 * r + SGN(r) * .259
CASE 7.5# TO 8#
  d = .032 * r + SGN(r) * .288
CASE 8# TO 8.5#
  d = .04 * r + SGN(r) * .32
CASE 8.5# TO 9#
  d = .051 * r + SGN(r) * .36
CASE 9# TO 9.5#
  d = .079 * r + SGN(r) * .411
CASE 9.5 TO 9.75
  d = .07 * r + SGN(r) * .49
CASE 9.75 TO 9.875
  d = .065 * r + SGN(r) * .56
CASE ELSE
  d = .375 * r + SGN(r) * .625
END SELECT
'  Adjust for the amount of variation desired in the image.
del = d * 256 / 4
END FUNCTION
FUNCTION linquant (x, K%)
' This function returns the linearly quantized (to K% bits) value of x.
L& = 2 ^ K%
linquant = FIX(L& * x) / L&
END FUNCTION
FUNCTION NumRemBlanks$ (n%)
' This function converts a number to a string with all blanks stripped off.
        num$ = STR$(n%)
        NumRemBlanks$ = LTRIM$(RTRIM$(num$))
END FUNCTION
SUB picture (imagesiz%, image%())
' This subroutine produces the input image.
```

```
' First column
        image%(0, 0) = INT(256 * RND(1))
          FOR j = 1 TO imagesiz% - 1
t0:         temp = image%(0, j - 1) + del
            IF temp < 0 OR temp >= 256 THEN GOTO t0
            image%(0, j) = INT(temp)
          NEXT j
' All the rest
        FOR i = 1 TO imagesiz% - 1
          image%(i, 0) = (image%(i - 1, 0) + image%(i - 1, 1)) \ 2
          FOR j = 1 TO imagesiz% - 1
t:          temp = (image%(i - 1, j) + image%(i, j - 1)) / 2 + del
            IF temp < 0 OR temp >= 256 THEN GOTO t
            image%(i, j) = INT(temp)
          NEXT j
        NEXT i
END SUB
SUB plot (n%, x(), lgrthm%)
' This subroutine plots the array x on the screen.
'  lgrthm% should be 1 for a log plot, otherwise the plot will be linear.
DIM y%(n% - 1, n% - 1)
PALETTE 4, 4                        ' Plot the outlines in red (non-gray).
FOR i = 0 TO n% - 1
 FOR j = 0 TO n% - 1
   IF lgrthm% = 1 THEN
    y%(i, j) = INT(LOG(ABS(x(i, j) + 1E-20)) / LOG(1.5) + 16)  ' The argument
of the second log function is the base of the log.
   ELSE
    y%(i, j) = INT(x(i, j) * 16)
   END IF
   IF y%(i, j) < 0 THEN y%(i, j) = 0
   IF y%(i, j) >= 16 THEN y%(i, j) = 15
   LINE (j, i)-(j + 1, i + 1), 4, B              ' Outline the box & tile it.
   PAINT (j + .5, i + .5), tile$(y%(i, j)), 4
 NEXT j
NEXT i
PALETTE 4, 0                        ' Change the outlines to black.
END SUB
SUB stats (n%, i2(), i(), dc, bittotal%) STATIC
' This subroutine computes and displays the statistics
'   for evaluation of the compression method.
' bpp = bits per pixel
' MSE = mean square error
' NSE = normalized square error (square error / AC energy)
        msen = 0: msed = 0
        count = count + 1
        FOR i = 0 TO n% - 1                     ' Compute the MSE.
        FOR j = 0 TO n% - 1
          msen = msen + (i2(i, j) - i(i, j)) ^ 2
          msed = msed + (i(i, j) - dc) ^ 2
        NEXT j
        NEXT i
        VIEW PRINT 16 TO 25
        bpp = bittotal% / n% ^ 2
        mse = 100 * msen / n% ^ 2
        IF mse > msemax THEN msemax = mse
```

```
        msetot = msetot + mse
        nse = 100 * msen / msed
        IF nse > nsemax THEN nsemax = nse
        nsetot = nsetot + nse
        PRINT "bpp  =  "; : COLOR 13: PRINT USING "#.###"; bpp: COLOR 15
        PRINT
        PRINT "MSE  =  "; : COLOR 14: PRINT USING "#.### _%"; mse: COLOR 15
        PRINT "   max  "; : PRINT USING "#.### _%"; msemax
        PRINT "   avg  "; : PRINT USING "#.### _%"; msetot / count
        PRINT
        PRINT "NSE  = "; : PRINT USING "###.## _%"; nse
        PRINT "   max "; : PRINT USING "###.## _%"; nsemax
        PRINT "   avg "; : PRINT USING "###.## _%"; nsetot / count
END SUB
SUB transform (n%, its(), c(), LUT(), Xform%, dir$)
DIM temp(n% - 1, n% - 1), rowi(n% - 1), rowo(n% - 1)
' This subroutine computes the two-dimensional (forward or inverse)
'   transform of the n% x n% array its.  The result is
'   returned in the array c.
' Perform one-dimensional transform on each row.
FOR i = 0 TO n% - 1
  FOR j = 0 TO n% - 1
    rowi(j) = its(j, i)                          'Includes matrix transpose.
  NEXT j
  GOSUB pick
  FOR u = 0 TO n% - 1
    temp(i, u) = rowo(u)
  NEXT u
NEXT i
' Transform each column.
FOR u = 0 TO n% - 1
  FOR i = 0 TO n% - 1
    rowi(i) = temp(i, u)                         'Includes matrix transpose.
  NEXT i
  GOSUB pick
  FOR v = 0 TO n% - 1
    c(u, v) = rowo(v)
  NEXT v
NEXT u
GOTO done
                '***** subroutine *****
pick:
  SELECT CASE Xform%
  CASE 1
    IF dir$ = "for" THEN
      CALL DCT1D(n%, rowi(), rowo(), LUT())
    ELSEIF dir$ = "inv" THEN
      CALL IDCT1D(n%, rowi(), rowo(), LUT())
    END IF
  CASE 2
    CALL WHT1D(n%, rowi(), rowo())
  CASE 3
    CALL SLANT1D(n%, rowi(), rowo(), LUT())
  END SELECT
  RETURN
done: END SUB
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-4263		

4. Title and Subtitle	5. Report Date
Data Compression Techniques Applied to High Resolution High Frame Rate Video Technology	December 1989
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
William G. Hartz, Robert E. Alexovich, and Marc S. Neustadter	None (E-5126)
	10. Work Unit No. 694-03-03

9. Performing Organization Name and Address	11. Contract or Grant No.
Analex Corporation NASA Lewis Research Center Cleveland, Ohio 44135	NAS3-24564
	13. Type of Report and Period Covered Contractor Report Final

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	

15. Supplementary Notes

Technical Manager: William K. Thompson, Electronic and Control Systems Division, and
Project Manager: Marlene S. Metzinger, Space Experiments Division, NASA Lewis Research Center.

16. Abstract

This report details an investigation of video data compression applied to microgravity space experiments using High Resolution High Frame Rate Video Technology (HHVT). An extensive survey of methods of video data compression, described in the open literature, was conducted. The survey examines compression methods employing digital computing. The results of the survey are presented. They include a description of each method and assessment of image degradation and video data parameters. An assessment is made of present and near term future technology for implementation of video data compression in a high speed imaging system. Results of the assessment are discussed and summarized in a tabular listing of implementation status. The results of a study of a baseline HHVT video system, and approaches for implementation of video data compression, are presented. Case studies of three microgravity experiments are presented and specific compression techniques and implementations are recommended. The results of the investigation conclude that video data compression approaches for microgravity space experiments are experiment peculiar in requirements and no one single approach is universally optimum. It is shown, for the experiments studied, that data compression required is separable into two approaches: the first to limit data rates for storage, and the second to reduce data rates for transmission. For high resolution and/or high frame rate experiment requirements and real time compression, hardware implementations are currently limited, by technology, to methods that can be implemented using parallel processing and simple compression algorithms. Although theoretically attractive, no approach could be identified for focal plane processing alone, that could be implemented with state of the art hardware.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
High resolution video High frame rate video Data compression Imaging algorithms	Unclassified – Unlimited Subject Category 32

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	140	A07

*For sale by the National Technical Information Service, Springfield, Virginia 22161