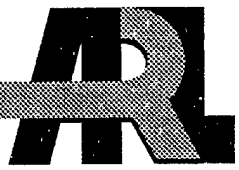
ARMY RESEARCH LABORATORY



A Survey of Image Compression Techniques and Their Performance in Noisy Environments

by Lisa M. Marvel
and George W. Hartwig, Jr.

ARL-TR-1380 June 1997

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer need. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-1380 **June 1997**

A Survey of Image Compression Techniques and Their Performance in Noisy Environments

Lisa M. Marvel, George W. Hartwig, Jr.
Information Science and Technology Directorate, ARL

Abstract

In the effort to digitize the battlefield, one of the most difficult capabilities to provide is the reliable transmission of imagery across the data links available. The most severe obstacles occur at low echelons, where the communication link is provided by combat net radios. The low-bandwidth, noisy nature of this physical channel represents the most serious challenge to implementation of the digital battlefield of the future. These channels make the compression of any imagery a requirement for timely transmission. In this paper, we explore the effects of noise on a variety of image compression techniques - namely, fractal, wavelet, DPCM, and the most recent compression standard for still imagery, JPEG version 6. Methods for minimizing the effects of the noisy channel on algorithm performance are also considered.

Table of Contents

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The objective of this report is to familiarize the reader with the aspects of image compression and transmission over the battlefield channel. Information exchange on the battlefield is enhanced by the transmission of imagery between troops in combat, unmanned scout vehicles, remote medical facilities, and the command post. Image integrity as well as timely transmission is imperative to all of these scenarios. The images must be transmitted over tactical channels, which are typically narrow bandwidth radio channels exposed to the noisy battlefield environment. It is therefore necessary to compress the image for the narrow channel and protect it from corruption in these hostile surroundings. Limitations of field hardware and time constraints prohibit the use of complex algorithms for battlefield communications. Subsequently, the desired process should be of low computational complexity and provide reliable image transmission in near real time.

It is imperative that image integrity be preserved as much as possible during transmission. For instance, corruption of an image used for remote diagnosis could be a life or death situation. Of equal importance in wartime is a decision made from an image transmitted for identification of friend or foe to prevent an incident of friendly fire. The use of both compression and error protection is needed for image transmission over narrow bandwidth noisy channels.

Timeliness is also a key factor. It is also desirable for the algorithm that processes the images to be of minimal computational complexity. In the application of mobile communications, this is particularly vital due to the constraints of the field transmitter/receiver. A less intensive algorithm allows real-time or near real-time processing and transmission of imagery. This aspect is very desirable in an image transmission system.

1.1 Combat Net Radios (CNR)

In the rush to digitize the battlefield, one of the most difficult problems to be overcome is how to transport imagery across the data links available. The difficulty of this task reaches an apex when the available communications link is via combat net radios (CNR). The low bandwidth and noisy nature of this physical channel represents the most serious challenges to implementation of the digital battlefield of the future.

The primary means of communication at low echelon fighting units has historically been and, in many cases, continues to be voice data as transmitted by CNR. Gradually, a requirement for digital data transmission is being inserted into the mission profile. Digital transmissions allow for compression and forward error correction and provide the ubiquitous computer with the information it requires.

Modern CNR are typically line-of-sight FM low-power instruments designed specifically for use at short range. For this device the bandwidth is very limited, although recent improvements have yielded an increase as high as 25 MHz. Table 1 shows where CNR fits into the modern communication world. In use these radios are commonly assembled into a single hop network of 6–12 users. Their effective use to date is testimony to the redundancy

of human language and the ability of the human neural net to extract meaningful data from a noisy signal.

TABLE 1.—*Common Broadcast Bandwidths*

Broadcast Media	Frequency	Bandwidth
Satellite Channel	6 GHz	500000 kHz
Broadcast Television		
channels 2–6	54-88 MHz	6000 kHz
channels 7–13	174-216 MHz	6000 kHz
channels 14–83	470-890 MHz	6000 kHz
FM Stereo Radio	88-108 MHz	200 kHz
Cellular Telephone	800-900 MHz	60 kHz
Combat Net Radio		
PRC-77 (old)	30-75 MHz	50 kHz
SINCGARS (current)	30-80 MHz	25 kHz
AM Radio	540-1600 kHz	10 kHz

1.2 Imagery

Images contain a large amount of information. Analog television requires 6 MHz of bandwidth per channel for voice and video, approximately 5 MHz of which is dedicated to full motion video. By contrast, the digital radios used by the Army, such as SINCGARS, allow a maximum of 16 kilobits per second (kbps), as indicated in the previous subsection. Digitizing a single gray scale image requires many bits to describe the image. In order to put into perspective the actual amount of data that an image contains, the following example is provided. A digital image may contain 512 $\times$ 512 pixels. Each pixel has a dynamic range of 0–255, symbolizing gray scale values. This translates to 8 bits/pixel $\times$ 262,144 pixels for a total of 2,097,152 bits or approximately 2 megabits (mb). For a narrow bandwidth channel of 16 kbps it would take a minimum of 2 minutes to transmit this image. In many applications, this delay is neither practical nor acceptable. Therefore, it is necessary to compress the image data before transmission. Digital techniques allow processing that can compress the data. The fact that images contain large amounts of data causes compression to be an important objective.

1.3 The Channel

This battlefield channel exhibits characteristics not normally found in traditional image transmission systems. The first important channel characteristic is narrow bandwidth. Battlefield communications need special considerations, such as encryption and the ability to transmit at low power to minimize the probability of detection. Since extra information must be added during the encryption process, the channel becomes smaller (more narrow),

leaving less room for the actual image data. The second important channel characteristic is noise. The battlefield generates much electronic noise in its own right due to shock waves from blasts, various types of communications equipment, and engine electronic noise. Other sources of noise could be intentional jamming by enemy forces.

An image can become corrupted when transmitted over a noisy channel. Depending upon the form of the data during transmission, a small amount of noise can be catastrophic, causing the received image to be distorted such that it is rendered useless. Consequently, an effort must be made to prevent or minimize the effects of channel noise on the image data.

Due to the constraints of the battlefield channel and the situational time constraints, several steps must be taken to effectively transmit imagery on the battlefield.

1.4 Source and Channel Coding

The traditional method of digital image transmission requires a multistage process. The first stage is source coding, or the removal of redundancy, to compress the image for the narrow bandwidth channel. The second stage requires coding for the channel by adding redundant characters to protect the information from noise.

C. E. Shannon wrote his definitive paper on the mathematical theory of communication in 1948 [1]. In this paper, he defined the capacity of a channel to be the maximum rate, bits/symbol, at which information could be transmitted reliably through a channel. Shannon's source coding theorem states that if a channel is noiseless it is possible to compact the information to the size of the source entropy, the minimum average word length in bits/symbol, by coding infinitely long extensions of the source. If the channel is noisy, the interest shifts from representing information as compactly as possible to encoding it so that reliable communication is possible. By sending information in a redundant form over a noisy channel, the probability that an error will occur can be reduced. Rate distortion theory states that the distortion incurred by the information is a decreasing function of the information rate. In addition, the rate is upper bounded by the capacity of the channel. The capacity is a function of the channel bandwidth, noise properties of the channel, and the intended signal. Therefore, one must compromise between rate and distortion over the noisy channel. This balance between source coding to minimize rate and channel coding to minimize distortion is dependent upon the application.

A source and channel coding system is depicted in Figure 1. In this example, $\mathbf{x}$ is the original image. It is compressed by the source encoder, which outputs $\mathbf{y}$, then protected from errors by the channel encoder, which produces $\mathbf{z}$. During transmission over a narrow bandwidth noisy channel, $\mathbf{z}$ is exposed to various forms of noise and enters the receiver as $\hat{\mathbf{z}}$. The channel decoder exploits the redundancy in $\hat{\mathbf{z}}$ to produce an estimate, $\hat{\mathbf{y}}$, of $\mathbf{y}$. The source decoder then constructs $\hat{\mathbf{x}}$, a possibly distorted version of $\mathbf{x}$.

It has been a longstanding practice to treat source and channel coding separately. This practice was motivated by Shannon's joint source channel coding theorem [2]. This theory states that source coding followed by channel coding can be made to perform as well as any single-stage source/channel coding procedure. In this scenario of separation, the source

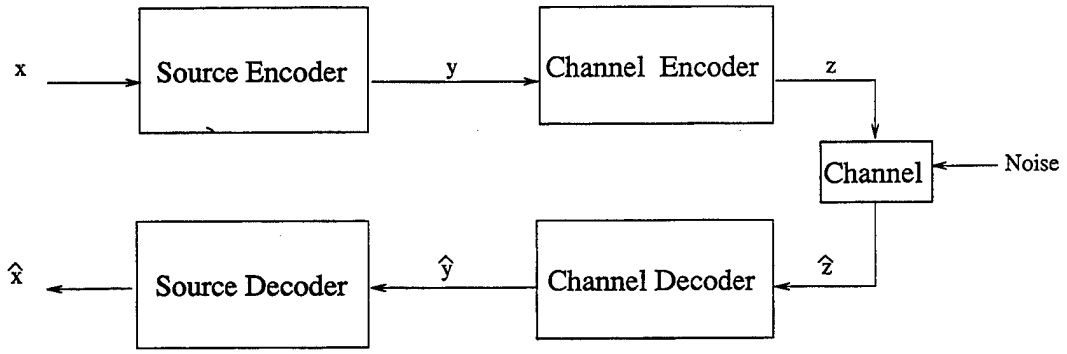


FIGURE 1.—*Traditional coding system.*

coding can be designed to suit the source and the channel coding can be designed for good performance in a particular channel. The drawback to this method is high computational complexity and possible delay. In contrast, joint source channel coding lessens the computational burden by performing both as a single process [3], as depicted in Figure 2.

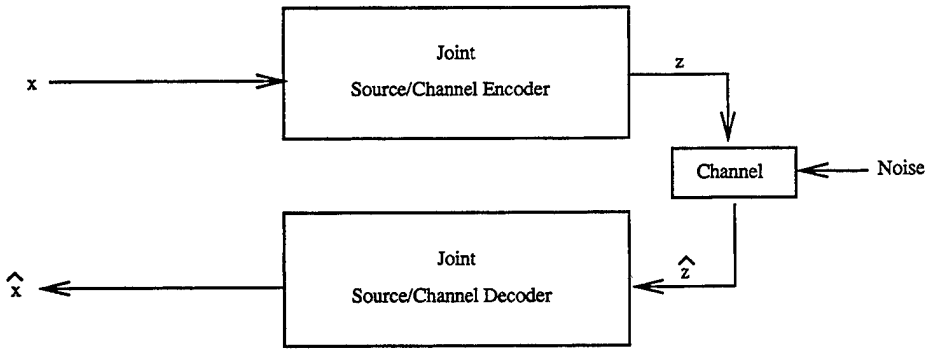


FIGURE 2.—*Joint source/channel coding system.*

In many ways, source and channel coding form a duality. Source coding removes unstructured redundancy from the data, data compression, while channel coding introduces structured redundancy to the data to combat errors caused by noise, error correction. If the source coding method is lossy, the cost for the compression is distortion. The expense for the protection provided by channel coding is more data and less distortion. The tradeoff one is willing to make between distortion and rate depends primarily on the application. As an example, little compression and high data protection may be desired for medical images transmitted on a wide bandwidth channel. Conversely, compression may be of the utmost concern for the transmission of satellite imagery over a narrow bandwidth channel. The ideal situation is efficient compression with low distortion, along with resilience to channel noise.

1.5 Report Composition

It is the intention of the authors that this report be used to familiarize the reader with current image compression techniques and their performance in noisy environments. Of

particular interest is the wireless environment of CNR. Chapter 2 will review numerous types of compression for both lossless and lossy methods. In Chapter 3, after discussing the implementation details of these compression techniques, the performance of each will be demonstrated for the noise channel as well as the noiseless channel through simulations. Chapter 4 will address methods by which these techniques can be aided to cope in a noisy environment and therefore improve performance. Finally, in Chapter 5, the significant findings will be reiterated and recommendation will be made for future research in this area.

2. Image Compression Techniques

The performance of a compression algorithm is measured in two ways: compression ratio and distortion incurred during compression. Compression ratio, C_r, is the relation of the number of bits input to the source coder, to the number of output bits of the source coder as shown in equation (1).

$$C_r = \frac{source\ input\ size}{source\ output\ size}. \tag{1}$$

The distortion measure or fidelity criterion for comparison of the original image to the reconstructed image are the mean squared error (MSE) and the peak signal to noise ratio (PSNR), as indicated in (2) and (3), respectively. In these equations, x_i is the original pixel from the original image and $\hat{x}_i$ is the corresponding reconstructed pixel. Accompanying these error measurements, a subjective qualitative comparison is also used to assess performance.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{2}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{3}$$

Figure 3 provides a diagram of the various types of image, video, and audio compression methods [4]. There are two categories of waveform-based compression: lossless and lossy. Lossless compression implies that the reconstructed image will be an exact replica of the original. This type of compression typically yields a lower compression ratio than its lossy counterpart. Lossy compression discards some nonredundant information to provide a greater reduction of data or greater compression ratio. This is accomplished at the cost of distortion in the reconstructed image. In many applications, a certain amount of distortion is acceptable, in return for greater compression performance.

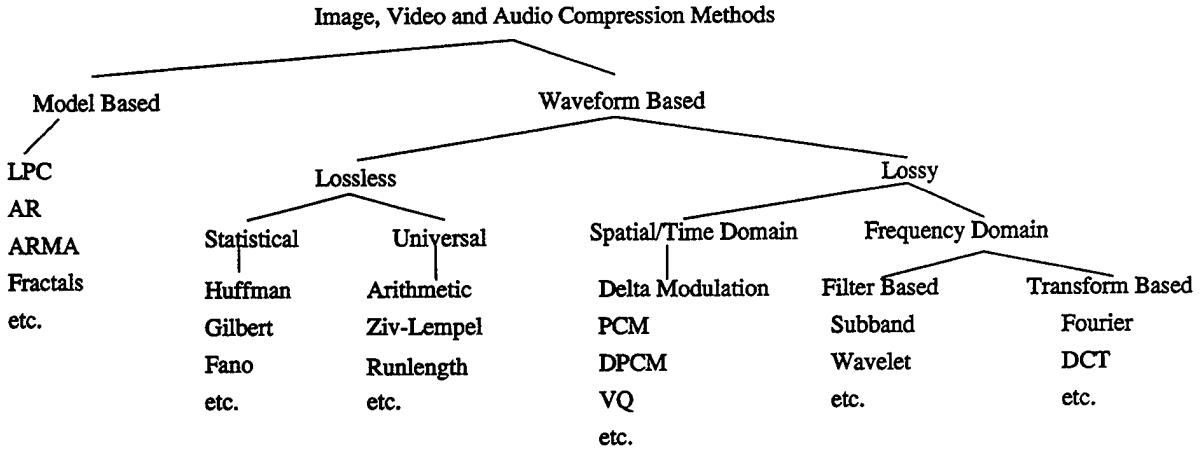


FIGURE 3.—*Compression methods.*

2.1 Lossless Compression Techniques

There are many types of lossless compression. We will explain one of the most commonly used methods: Huffman coding. Huffman coding is an entropy code. In layman's terms, entropy is the average minimum number of bits necessary to fully describe an aspect of a particular source (i.e., pixel within an image). Entropy is the basis of "Information Theory" developed by C. E. Shannon in 1948. Entropy is denoted as $H(X)$ and takes into account the probability of occurrence for all items within a source. The equation for $H(X)$ is shown in (4).

$$H(X) = \sum_{i=0}^{N} p(x_i) log_2 \frac{1}{p(x_i)} \tag{4}$$

As a simple example, suppose there are three possible values as output for a source X: $\{x_1, x_2, x_3\}$. Each of these values is equally likely; therefore the probability for each item of the source is $p(x_i) = \frac{1}{3}$. For our example, the entropy of source X is:

$$H(X) = \frac{1}{3} log_2 \frac{3}{1} + \frac{1}{3} log_2 \frac{3}{1} + \frac{1}{3} log_2 \frac{3}{1} = 1.58 \; bits. \tag{5}$$

This indicates that the average minimum number of bits necessary to fully describe this source, X, is 1.58 bits.

Consider a different source, Y, which also has three possible items that are *not* equally likely. The probability of y_1 is 0, and the probability of y_2 and y_3 are each: $p(y_i) = \frac{1}{2}$. The entropy of the source, Y, would be different from source X, as shown in (6).

$$H(Y) = 0 + \frac{1}{2} log_2 \frac{2}{1} + \frac{1}{2} log_2 \frac{2}{1} = 1 \; bits \tag{6}$$

Of course, in practice, we cannot encode a source using a fraction of a bit. Therefore, we must round up to the next whole number of bits. Subsequently, it would require 2 bits to describe source X and 1 bit to describe source Y.

Entropy coding strives to reach the value of entropy as the average number of bits used to describe the source. One of the best known entropy codes is the Huffman code. For this code, values of the source that occur frequently are assigned codewords with a small number of bits, where source values that occur less frequently are assigned codewords of longer length. This allows for an overall smaller average length codeword and thus approaches entropy.

2.1.1 Huffman Coding

As an example of Huffman coding, we digitize and compress (encode) an image of the U.S. flag. To determine the probabilities of the items within the source, F, for flag, one would take the total number of pixels denoting one color and divide by the total number of pixels in the flag image as shown in (7). This would be repeated for all possible items within the source, F.

$$p(color_i) = \frac{number\ of\ color_i\ pixels}{total\ number\ of\ pixels\ in\ image} \tag{7}$$

Assume the source F has the following source probabilities: $p(white) = .6$, $p(red) = .25$, $p(blue) = .15$. The entropy of the source F is 1.35 bits. Given the these probabilities, we can now construct the coding tree using Huffman's technique:

- List all possible symbols with the probabilities.
- Locate the two symbols with the lowest probabilities.
- Replace these symbols with a single node whose probability is the sum of the individual probabilities.
- Repeat until there is a single node whose probability is one.

To form the codewords for the symbols, traverse the tree from the root, probability 1, to the leaf labeling the left branch with a **0** and the right branch with a **1**. This process has been performed with the flag source and is depicted in Figure 4. The codewords for the colors within the U.S. flag are displayed in Table 2: The Huffman encoded flag will now require 1.4 bits/symbol on average, which is very close to the source entropy of 1.35 bits.

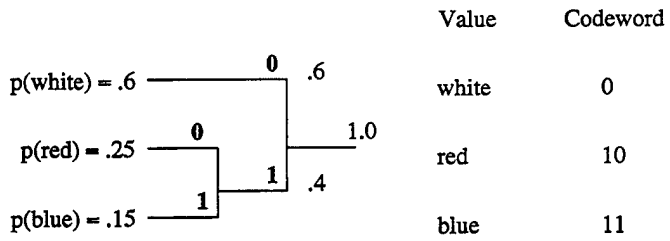


FIGURE 4.—*Example of Huffman coding.*

TABLE 2.—*Codewords for the U.S. Flag*

Color	Codeword
white	0
red	10
blue	11

As is evident in the previous example, Huffman coding produces a variable length code. Therefore, the decoder must never lose synchronization during the decoding process. A loss of synchronization would more than likely cause all subsequent codewords to be decoded incorrectly. This is called a catastrophic error. As will be evident in future sections, traditional variable length codes elicit catastrophic failures over a noisy channel.

2.1.2 Runlength Encoding

Another form of lossless encoding is runlength encoding. This method is effective for bitonal images, those in which pixels take on one of two possible values. These values can be represented with 1 bit per pixel, *0* or *1*. In this instance, *runs* of the symbols are encoded. This works particularly well for situations in which large regions of a particular symbol occur. Subsequently, facsimile transmissions use *0's* to represent white background and *1's* to represent black foreground. Each pixel is mapped to run and value. As an example, a scan line of a bitonal image is shown in Figure 5.

FIGURE 5.—*Scan line of a bitonal image.*

The original values for this scan line before encoding are {*0 1 1 1 1 1 1 0 1 0 1 1*}, representing the 12 pixels with 12 bits. We encode the black pixels with runlength encoding and limiting runs to a maximum of 7 pixels. Runlength encoding would result in no more than 4 bits necessary to describe the 6-pixel run. An example of possible output would be {*110 001 010*}. Here, only the runs of black pixels are encoded. (It is assumed that the first pixel is white, as are the pixels between runs.) To decode this runlength encoding output, begin with a white pixel and decode the number of black runs. The result is {*0 1 1 1 1 1 1 1 0 1 0 1 1*}. This simple example yields a compression ratio of $\frac{12}{9}$.

2.2 Lossy Compression Techniques

In order to obtain an increased compression ratio, we accept distortion in the reconstructed image for better compression. Depending upon the application, an exact replica of the original image may not be necessary to serve the purpose of target identification, target recognition or terrain mapping. We can save bandwidth by further compressing the image and it will still retain functionality.

2.2.1 Quantization

The simplest type of lossy compression is *quantization*. Quantization maps a range of values to a smaller range of values. As an example, consider a uniform source as a range of values of $0, \ldots, 9$. If all of these values are equiprobable the entropy of the source is 3.32 bits, indicating it would require 4 bits/symbol to describe the source. If these values were mapped into even numbers with possible values of $0, 2, 4, 6$, and 8, the entropy is 2.32 bits, making it necessary to use 3 bits/symbol to encode the source. However, by mapping the source to all even numbers, distortion is incurred because the reconstructed values are not an exact replica of the original source values. As an aside, the graphic interchange format (GIF) utilizes this technique by quantizing all pixels values to eight bits.

2.2.2 Differential Pulse Code Modulation (DPCM)

Predictive coding is one of the many methods of compression. As with most types of compression, it can be lossy or lossless. The principle of predictive coding is based on eliminating the inter-pixel redundancy that exists between adjacent pixels. The redundancy is removed by taking the difference between the predicted value of the pixel and its actual value. This difference is the prediction error, or the *residual*, which constitutes the new information in the pixel. Only this new information is encoded in the compressed image.

An image is a highly spatially correlated source due to the physical system used to acquire it. For example, a charge coupled device (CCD) camera, which is used to obtain the image, consists of an array of photo-cell elements that leak energy into each other. This physical phenomenon is evident in the gradual change of pixel values throughout the image. To illustrate this fact, an example is provided in by extracting a row of pixels from an image, shown in Figure 6. Here it is evident that the change from one pixel to the next is subtle.

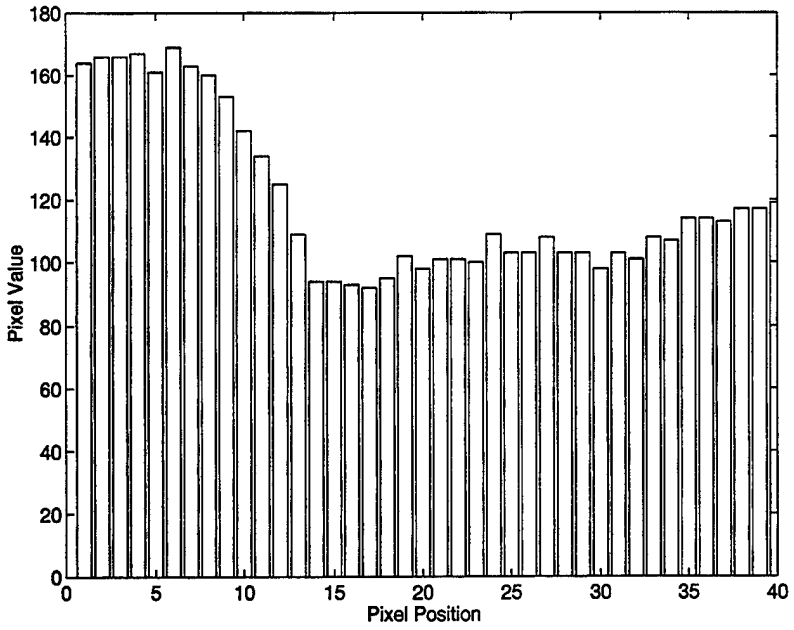


FIGURE 6.—*A row of pixels extracted from an image.*

As an example of the compression obtainable through lossless predictive coding, the most basic linear predictor, the difference filter, is used to demonstrate. Therefore, the difference between pixel n and pixel $n-1$ becomes the residual. In Figure 7a, the histogram represents the distribution of the Lena (256×256) image, with mean $\mu = 123$, standard deviation $\sigma = 47.92$ and an entropy rate of 7.45 bits/pixel. After applying the linear prediction utilizing the difference filter, the distribution becomes more compact with a smaller dynamic range, with $\mu = .002$, $\sigma = 19.12$, and an entropy rate of 5.66 bits/pixel, as shown in Figure 7b.

The model for lossy predictive coding is similar to that of the lossless model with the addition of a *Quantization* step and feedback, as shown in Figure 8

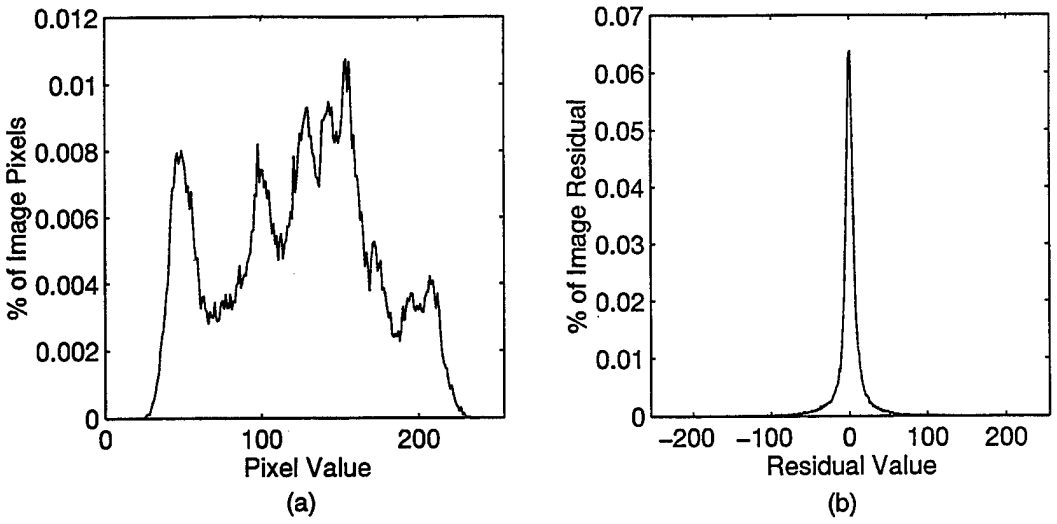


FIGURE 7.—*(a) Histogram of Lena image; (b) Histogram of residual.*

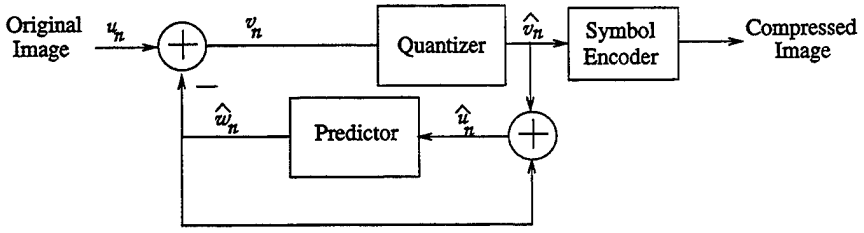


FIGURE 8.—*Lossy predictive encoder.*

As for all lossy compression schemes, there is a compromise between distortion and compression. The quantization in the lossy scheme maps the prediction error to a limited number of outputs, denoted $\hat{v}$. Along with the predictor, the quantizer defines the amount of compression and distortion associated with this lossy system.

In this system, u_n is the input pixel and $\hat{w}_n$ is the prediction of u_n as determined by the prediction filter. The prediction error v_n is the difference between the actual pixel value u_n and the predicted value $\hat{w}_n$. The prediction error is then quantized to $\hat{v}_n$ and added to the prediction $\hat{w}_n$ to form $\hat{u}_n$, the estimated value of u_n. Then $\hat{u}_n$ is used as the input to the prediction filter which produces $\hat{w}_{n+1}$, the predicted value of u_{n+1}. The feedback loop is added so the inverse operation can be performed at the decoder stage and to compensate for the distortion induced by the quantizer. This closed loop configuration also prevents error buildup at the decoder's output [5].

The decoder for lossy predictive coding is identical to that of the lossless scheme is shown in Figure 9. Assuming a perfect noiseless channel, $\hat{v}_n$ is received and $\hat{w}_n$ is added producing $\hat{u}_n$, the reconstructed pixel. The only distortion in this noiseless system is that caused by the quantizer.

The following prediction filters contain a relatively robust set of coefficients, which provide

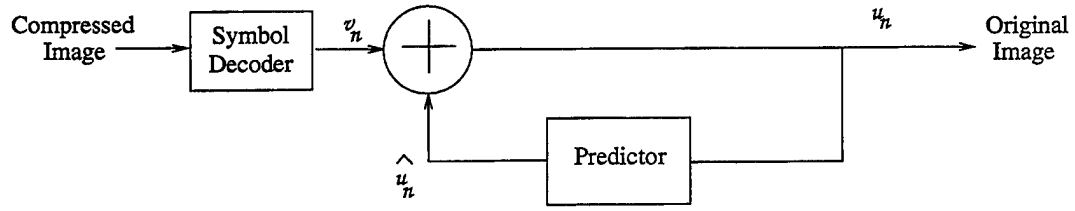


FIGURE 9.—*Predictive decoder.*

satisfactory performance over a wide range of images [5]. The filters are specified in (8)–(10).

$$\hat{u}_{i,j} = 0.97u_{i,j-1} \tag{8}$$

$$\hat{u}_{i,j} = 0.5u_{i,j-1} + 0.5u_{i-1,j} \tag{9}$$

$$\hat{u}_{i,j} = 0.75u_{i,j-1} + 0.75u_{i-1,j} - 0.5u_{i-1,j-1} \tag{10}$$

Predictive coding utilizing the optimal filter method, as described in [5], is commonly known as DPCM. It is a simple, easy way to implement compression technique for imagery.

2.2.3 Discrete Cosine Transform (DCT)

Transform coding is a reversible linear process where the pixel values are transformed from the spatial domain to the frequency domain. These transformations are performed in order to *compact* the energy into a few transform coefficients. For most natural images, a significant number of these coefficients have small magnitude, which can be coarsely quantized or discarded entirely with little image distortion [5]. An example of a transform encoder and decoding is shown in Figure 10.

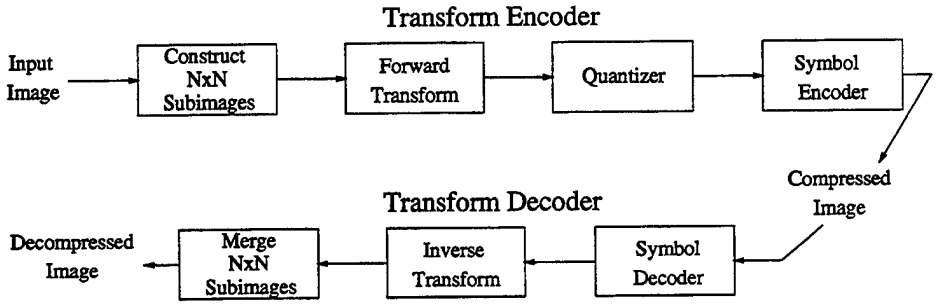


FIGURE 10.—*Transform coding diagram.*

The DCT is currently the most widely used transformation in image transform coding. It is the basis for current image compression standards such as Joint Photography Experts Group (JPEG), Motion Picture Experts Group (MPEG) and MPEG-2. The first step of this process is to divide the image into $N \times N$ blocks or subimages. A significant factor in DCT coding is this subimage size. For reasons such as computation complexity and the typical pixel correlation within a pixel neighborhood, the standards specify the subimage size to be $N = 8$. For instance, a 256×256 digital image would contain 32×32 subimages, where each 8×8 pixel block, or subimage, contains 64 pixels.

Each of these subimages is processed by the transformation which converts the pixel values to transform coefficients. These coefficients are then quantized. Compression is actually achieved during the quantization stage and not the transformation stage. This quantization stage selectively eliminates coefficients or more coarsely quantizes the coefficients that carry the least information and have the lowest impact on the quality of the reconstructed image. The final stage of the compression process is the symbol encoder. This stage is performed using a lossless method similar to the Huffman or runlength coding detailed in the previous section. To decompress the image, all the stages but the quantization step are reversed; the quantization stage is irreversible. All distortion incurred in the reconstructed digital image via the system depicted in Figure 10 is generated by this stage.

Utilizing the cosine waveform as the basis function, the DCT is used to decompose each block of pixels into a series of waveforms, each with a particular spatial frequency [4]. The result of the transformation is 64 coefficients, y_{kl} where $(k, l = 0, 1, \ldots, 7)$, of these waveforms as shown in (11). Here, k designates the row number and l indicates the column number within the 8×8 block.

$$y_{kl} = \frac{c(k)c(l)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} x_{ij} \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, \tag{11}$$

where,

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}} & if\ k = 0, \\ 1 & otherwise. \end{cases} \tag{12}$$

A diagram of the transformation process is demonstrated in Figure 11. In the frequency coefficients portion of this figure, the lighter blocks represent coefficients with high magnitude and the darker blocks represent coefficients with smaller values. The block in the upper left corner represents the 0^{th} or DC coefficient. The transformed data of Figure 11 is ordered in a zigzag pattern, as shown in Figure 12. This allows for the lower spatial frequency coefficient to be grouped with the DC coefficient, while the higher frequency coefficient are grouped at the lower right corner. The DC coefficient is related to the average value of the 8×8 block and typically has the greatest value of all the transform coefficients. The magnitude of the remaining coefficients usually decreases as the frequency increases. During compression, some of these high-frequency coefficients are truncated and quantized. This allows for easier coding of the transformed data; the higher frequency coefficients are typically close to zero and can be truncated without causing great adverse effect on the reconstructed image.

In order to transform the frequency coefficient back into the spatial domain to obtain image pixels, the *inverse* discrete cosine transform (IDCT) is used (13).

$$x_{ij} = \sum_{k=0}^{7} \sum_{l=0}^{7} y_{kl} \frac{c(k)c(l)}{4} \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}. \tag{13}$$

As is evident from equations (11) and (13), the DCT and its inverse are very similar. Therefore the same hardware used to implemented the DCT can also be used to implement

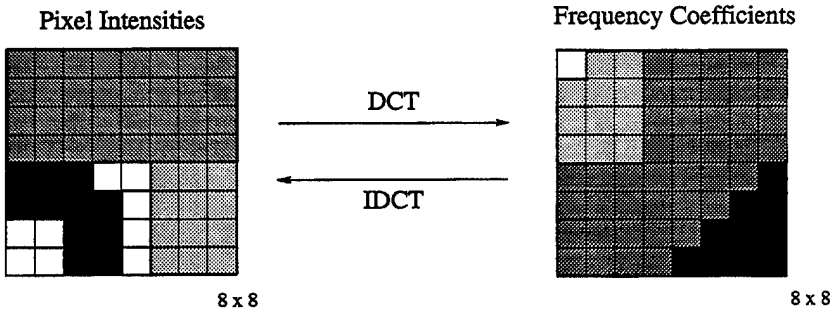


FIGURE 11.—*DCT method.*

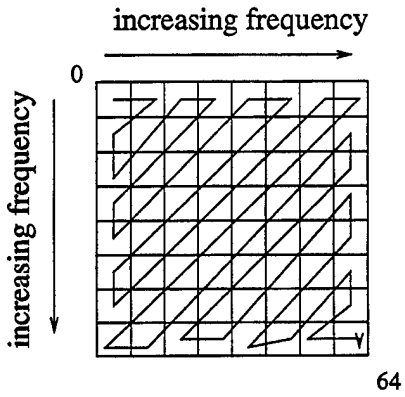


FIGURE 12.—*Zigzag coefficients.*

the IDCT. In addition, fast algorithms have been designed to perform the transform process in real time.

2.2.4 Discrete Wavelet Transform (DWT)

Recently, wavelets are the new trend in image compression. Compression by wavelets is similar to that of the DCT, where image pixels are transformed to spatial frequency coefficients. However there are some major differences. The first is the basis function. The DCT uses the cosine waveform as its basis function where wavelets can have one of many basis functions that meet certain criteria. The other difference between traditional DCT image compression and wavelets is that, for the DCT, the image is divided into 8×8 subimages before transformation, but for the wavelets transform, the whole image is used as a single input.

In order to describe the application of wavelets to image compression we must first address a few preliminaries. We will begin by describing what a wavelet actually is, in a very basic sense. We will then discuss the major contributions to wavelet image compression: the Daubechies wavelets and their implementation. Daubechies wavelet filter implementation necessitates some information be provided on the subject of filter banks. An example of a wavelet image coding system will then be presented along with the decomposition of an actual image.

A wavelet is a waveform of effectively limited duration — "localized in time," it is also localized in frequency and has an average value of zero. Sine waves, which are the basis of the DCT and Fourier methods, have infinite duration and are smooth and predictable. But wavelets tend to be irregular and asymmetric as shown in Figure 13.

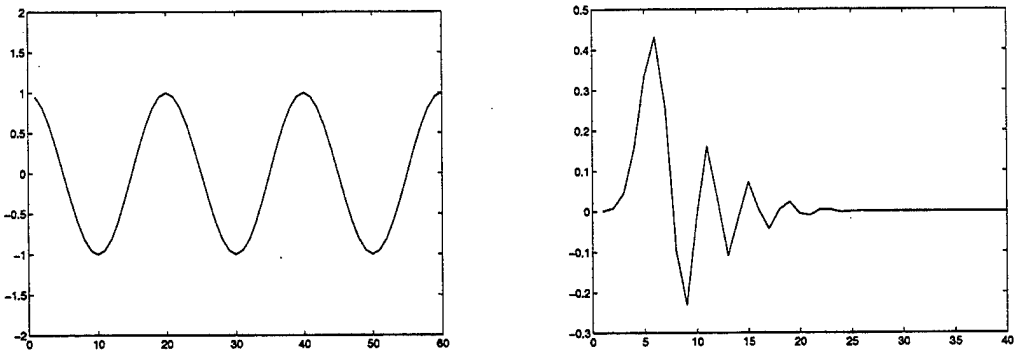


FIGURE 13.—*Comparison of the cosine waveform and D20 wavelet.*

Fourier analysis, and therefore the DCT, consists of breaking up a signal (an image is a two-dimensional signal) into sine waves of various frequencies. Similarly, wavelet analysis is the decomposition of a signal into shifted and scaled versions of the original, or mother, wavelet. By viewing Figure 13, one can see that signals with sharp changes might be better analyzed by the wavelet than the smooth sinusoid. Also local features, those restricted to a certain location, can better be described with wavelets whose duration is finite.

There is much mathematical theory within the subject of wavelets. Basically, a wavelet must adhere to three conditions. The first is a lowpass property, the second is orthogonality,

and the third is a degree of smoothness. All of these conditions can be specified by the basic wavelet equations. Interestingly, one elegant property of the wavelet-type bases is a self similarity of the wavelet basis functions, which are all obtained from a single prototype mother wavelet using scaling and translation.

A significant contribution to wavelet image compression was made by Ingrid Daubechies [6]. The outcome of this research was the explicit construction of a family of wavelets that were maximally flat, compactly supported, and orthonormal. These wavelets are easily extended to finite impulse response (FIR) filter banks: the digital implementation for the DWT.

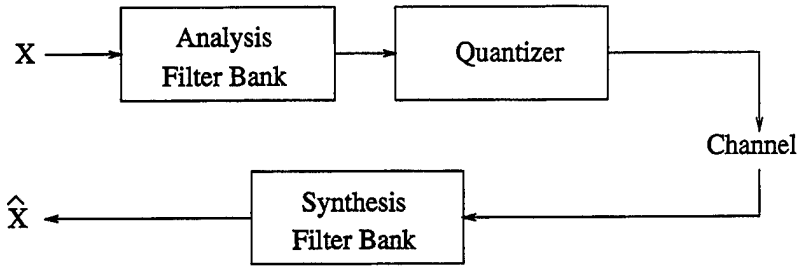


FIGURE 14.—*Wavelet image compression system.*

The basic components of the DWT image coding system are shown in Figure 14. These components consist of an analysis filter bank system used to decompose the image into wavelet series coefficients. The next stage is the quantizer, which provides the actual compression as with image coding via the DCT. The information is then passed through the channel and received at the receiver. The final stage in this system, the synthesis filter bank, converts the wavelet coefficients back to the spatial domain to represent pixels.

A filter bank is an implementation of subband decomposition, where various filters are used to decompose a signal into various frequency bands. This decomposition often has a successive approximation property: reconstruction based on an appropriate subsets of the basis function leads to a good approximation of the signal [7]. This is, of course, a good characteristic of a signal compression system.

Daubechies wavelets also have "perfect reconstruction properties," meaning $X = \hat{X}$. Perfect reconstruction is a form of subband decomposition where the reconstructed signal is a perfect replica of the input signal. These systems are also called quadrature mirror filters (QMF).

A two-channel filter bank/QMF is shown in Figure 15. The resulting frequency spectrum decomposition is depicted in Figure 16. The decomposition filter bank, the first stage: h_0 and g_0 in Figure 15, is called the analysis filter bank. Here h represents the analysis lowpass filter and g is defined as the analysis highpass filter. The reconstruction filter bank, h_1 and g_1 above, is called the synthesis filter bank. These analysis and synthesis systems together constitute a way of implementing a structured, orthogonal expansion of the signal by means of a filter bank, this process yields a perfect replica of the input signal.

Subband decomposition can be performed uniformly or by octave. Typical subband coding involves a uniform splitting of the frequency bands. For example, if an M channel

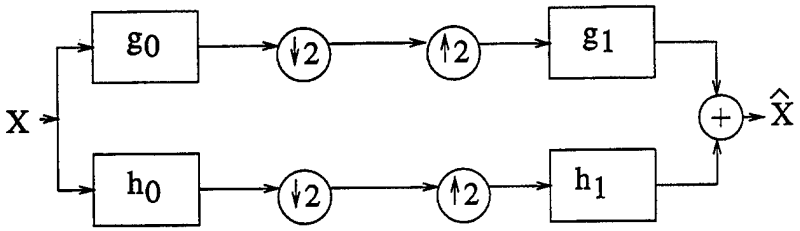


FIGURE 15.—*Perfect reconstruction filter bank/QMF.*

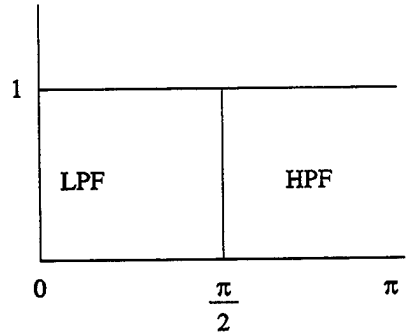


FIGURE 16.—*Frequency spectrum of perfect reconstruction filter bank.*

filter bank is implemented, the resulting frequency band decomposition for each band is of width π/M: all frequency bands are equal in size. The filter bank implementation of the DWT for image compression uses another technique of octave splitting of the frequency bands. This technique requires an iterative system of filter banks using a tree structure. The operation is performed by simply iterating a two-channel filter bank on the output of the previous lowpass channel. An octave band filter bank system with scale = 3, the number of iterations, is shown in Figure 17. Here the highpass filter (HPF) and lowpass filters (LPF) are indicated by HPF and LPF, respectively. The DWT equation for the low-resolution coefficients, $C_{j-1,n}$, is illustrated in (14) as a generic analysis system. The low-resolution coefficient, $C_{j-1,n}$, is derived from the lowpass filter of the higher scale, $C_{j,n}$. The DWT equation for the detail coefficients, $d_{j-1,n}$, is shown in (15). The detail coefficients are computed from highpass filtering the lowpass of the higher scale, again $C_{j,n}$.

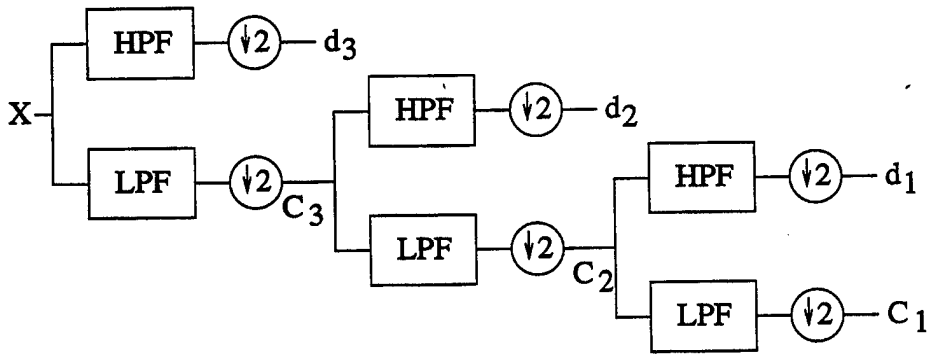


FIGURE 17.—*DWT filter bank.*

$$C_{j-1,n} = \sqrt{2} \sum_n h_{n-2k} C_{j,n} \tag{14}$$

$$d_{j-1,n} = \sqrt{2} \sum_n g_{n-2k} C_{j,n} \tag{15}$$

The octave splitting of the frequency domain by this DWT filter bank is demonstrated in Figure 18. This is sometimes called a logarithmic filter bank since the channels are of equal bandwidth on a logarithmic scale [7]. Each successive highpass output, $d_{1,2,3}$, contains an octave of input bandwidth. This lends to the successive approximation property mentioned earlier in this section. The successive approximation property can be directly related to the human visual system (HVS) where lowpass information, C_1, is improved as detailed information is provided by d_1, d_2, and d_3.

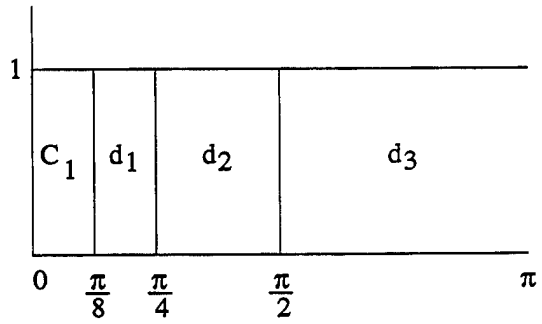


FIGURE 18.—*Example of octave splitting of the frequency domain.*

Once the wavelet has been determined and the orthonormal filters constructed, image transformation from the spatial domain to the frequency domain can be accomplished. For an example of wavelet image compression, we will use the eight tap Daubechies wavelet filter, D8, whose coefficients are shown in Table 3. For Daubechies wavelets, the filter coefficients for the highpass filter are constructed directly from the lowpass filter coefficients using equation (16).

TABLE 3.—*Daubechies D8 Filter Coefficients*

D8	Coefficient
h_0	0.230377813
h_1	0.714846570
h_2	0.630880767
h_3	-0.027983769
h_4	-0.187034811
h_5	0.030841381
h_6	0.032883011
h_7	-0.010597401

$$h_1(n) = (-1)_n h_0(-n + 2N - 1) \tag{16}$$

A two-scale analysis filter bank used for image compression is shown in Figure 19. For a two-dimensional signal, the filters are invoked first in the x dimension, the result is then decimated or subsampled in this dimension. This operation is repeated for the y dimension. Here the lowpass filter is again indicated by LPF and the highpass by HPF. The resulting

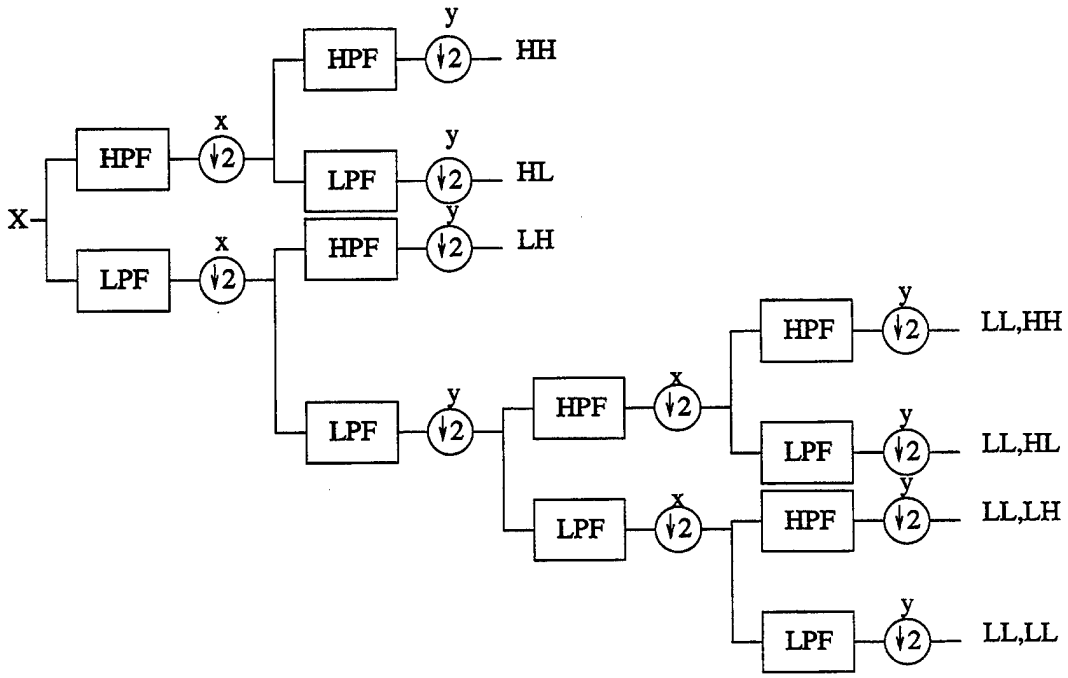


FIGURE 19.—*Wavelet analysis filter bank.*

spectral decomposition of the analysis filter bank is typically illustrated by an image pyramid (see Figure 20). This pyramid is the same size as the original image because of the decimation by two performed after each filter operation. Here the HH region contains the coefficients whose magnitude represents diagonal components in the original image. The LH represents the coefficients with strong horizontal components and the HL represent the vertical components. The remaining sections, LL-LH, LL-HL, LL-HH, and LL-LL, constitute the second scale of this wavelet transform, revisiting the diagonal, horizontal, and vertical components on the next scale from the all-lowpass result of the previous scale. The all lowpass area, LL-LL, is called the postage-stamp image. It is a small lowpass, subsampled version of the original image. One can imagine the demonstration of the successive approximation property by imagining decoding the all-lowpass image and having it gradually improve, becoming more sharp and clear, as more bands (i.e, LL-HL to -HH) are decoded. Of course, in this example we show only two scales of decomposition; most image compression schemes incorporate more than two scales.

After decompression, quantization is performed in the bands. The higher frequency detailed bands, HH on down, are greatly quantized, but the all lowpass postage-stamp image is typically finely quantized for less distortion in the reconstructed image. Again, as with the DCT, without quantization of the coefficients there is no compression. In most cases, many coefficients of the high-frequency bands are very small in magnitude and can be eliminated

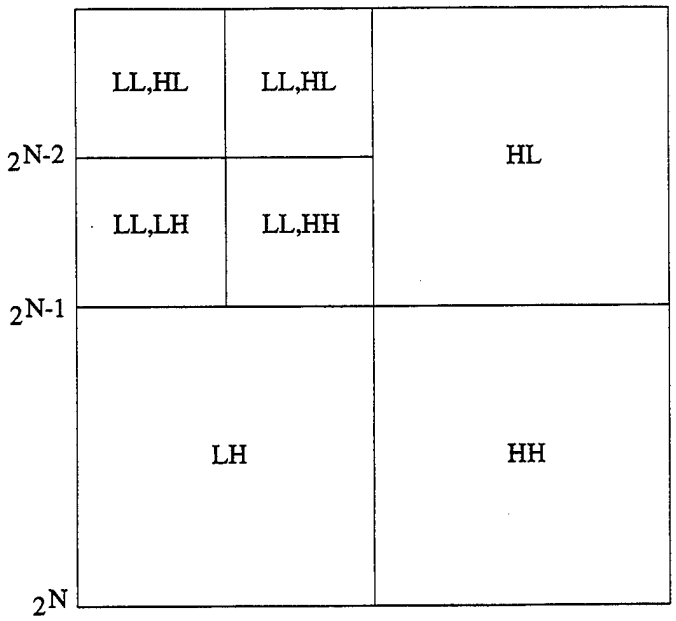


FIGURE 20.—*Wavelet image pyramid.*

altogether having little effect on the quality of the reconstructed image. This allows wavelet image compression to yield very good visual results and high compression ratios without the blocking artifacts of the DCT image compression caused by the subimage division.

To reconstruct the image, the synthesis filter bank is used. It is very much like the analysis filter bank in tree structure although the actually filters may be different. Figure 21 shows synthesis filter for the reconstruction of two scales. This filter bank is the converse of the analysis filter specified in Figure 19. To provide an example with a real image, we used the well-known Image Barbara (Figure 22).

Image compression with the DWT will most certainly be refined in the future. Current research focuses on the selection of the appropriate wavelet in addition to the search for techniques that will lessen the computational complexity. Ultimately, this research will result in standardized image compression techniques.

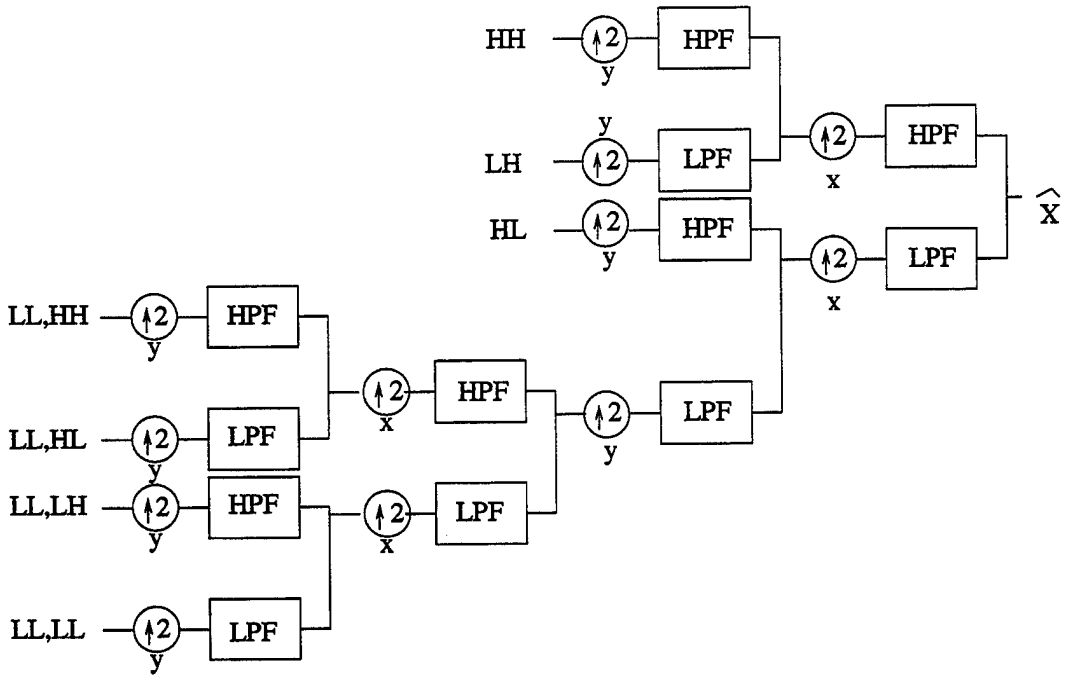


FIGURE 21.—*Wavelet synthesis filter bank.*

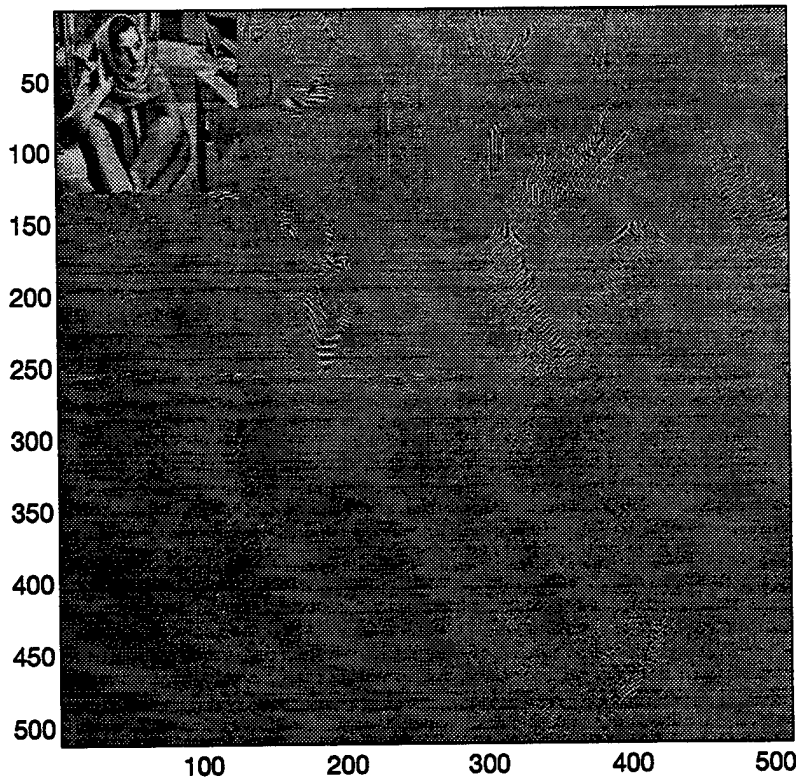


FIGURE 22.—*Two-scale decomposition of the Barbara image.*

2.2.5 Fractal Compression

Of the lossy image compression techniques, perhaps none has caused more controversy than fractal image compression. During the late 80's the use of fractal techniques for image compression was inspired by the early work of Mandelbrot [8] and refined with the iterated function system (IFS) work of Hutchinson [9] and Barnsley and Demko [10]. This was based on the premise that, since fractal mathematics are good for generating natural looking images, the inverse problem could be solved (i.e., could an IFS be created that generated a sufficiently good approximation to the original image). Initially, fractal image compression was clouded by claims for fantastic compression ratios of 1000:1 or even more. Although a few special images were in fact compressed at this rate, it was a manual process requiring not only human guidance but tens of hours of computer time. A graduate student of Barnsley's, Arnaud Jacquin, produced the first practical implementation of a fractal technique when he introduced the concept of the partitioned iterated function system (PIFS) in which an image is successively subdivided until each piece can be properly processed. Jacquin's software implementation [11], done as a part of his PhD work, successfully compressed images without human intervention, but only at modest rates in the range of 8:1 to 50:1. However, all current fractal transform implementations are based on Jacquin's work. Yuval Fisher's book Fractal Image Compression [12] and John Kominek's paper "Advances in Fractal Compression for Multimedia Applications" [13] are highly recommended for a more detailed introduction to fractal techniques.

What is Fractal Image Compression? To understand fractal image compression, one must understand the properties of an IFS, and the best way to gain this understanding is through the example of the multiple reduction copy machine (MRCM)[14][12]. Almost universally used to explain IFS, the MRCM may be thought of as a regular copy machine that has the capability to produce multiple overlapping copies of the original, with each copy required to be smaller than the original and each output image routed to the input so that the copier runs in a feedback loop.

Figure 23 shows an MRCM that produces three reduced and translated copies of the original image. If this MRCM is iterated many times, the image in Figure 24 appears. This is the well-known Sierpinski's triangle, and it is known as the attractor for our MRCM. If we consider each duplicating element in the copier as a mathematical transform, we may express the transform as (17), where the coefficients a_i, b_i, c_i, and d_i provide for rotation and scaling and e_i and f_i allow for translation. Transformations of this form are known as affine transformations of the plane and the IFS is the collection of the w_i (18).

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}, \tag{17}$$

$$W = \bigcup w_i \quad w_i : \Re^2 \to \Re^2. \tag{18}$$

As long as the transform w_i is contractive (i.e., given a metric d and two points in our image p_1, p_2), we have

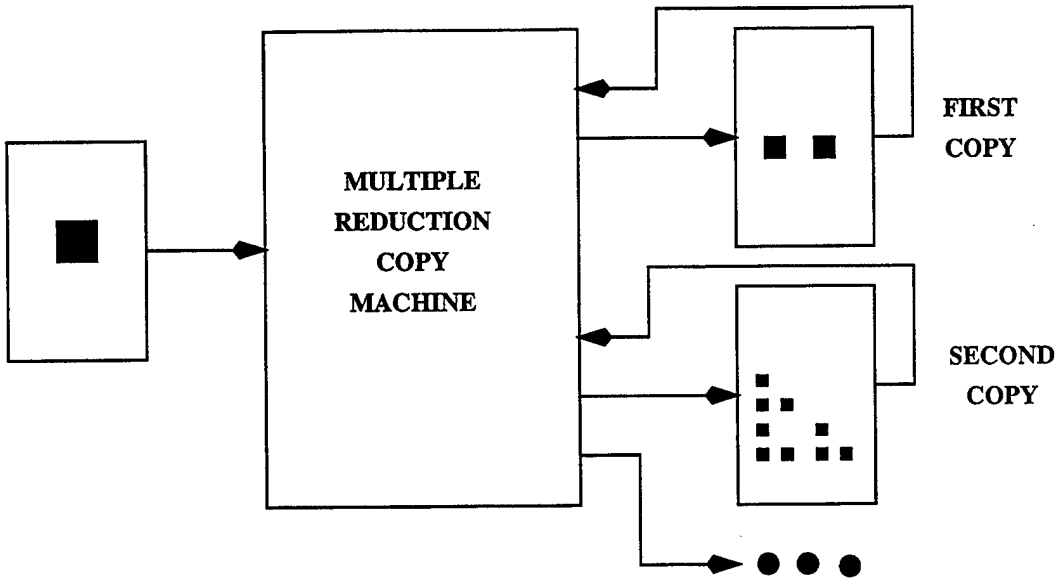


FIGURE 23.—*Multiple reduction copy machine.*

$$d(W(p_1), W(p_2)) < s * d(p_1, p_2), \tag{19}$$

where s ¡ 1), the Contractive Fixed Point Theorem allows us to say:

If X is a complete metric space and $W : X- > X$ is contractive, then W has a unique fixed point $|W|$. This final fixed point is known as the "attractor" for the set of transformations and is expressed mathematically (20).

$$|W| = \lim_{n \to \infty} W^{\circ n}(X). \tag{20}$$

The $W^{\circ n}(X)$ notation means $W(W(W(\ldots W(X))\ldots)$ done n times. Note that the final image of an IFS is independent of the starting picture. In other words, the transformations totally determine the final image.

Practical Fractal Image Compression The goal in fractal image compression is to take an image I and find a set of affine transformations that when run as an IFS have I as the attractor. Since the attractor is defined completely by the set of transformations, the image could be replaced by the transformations. Unfortunately, in practice, images are not self-similar and it is impossible to determine such an IFS for natural images. So using the techniques of Jacquin, the image is subdivided into small nonoverlapping partitions called ranges. Then another set of larger (generally twice the size of a range block), potentially overlapping, partitions called domains are created. The problem of fractal image compression is to find the "best" domain to map into a particular range partition using a simplified set of transformations.

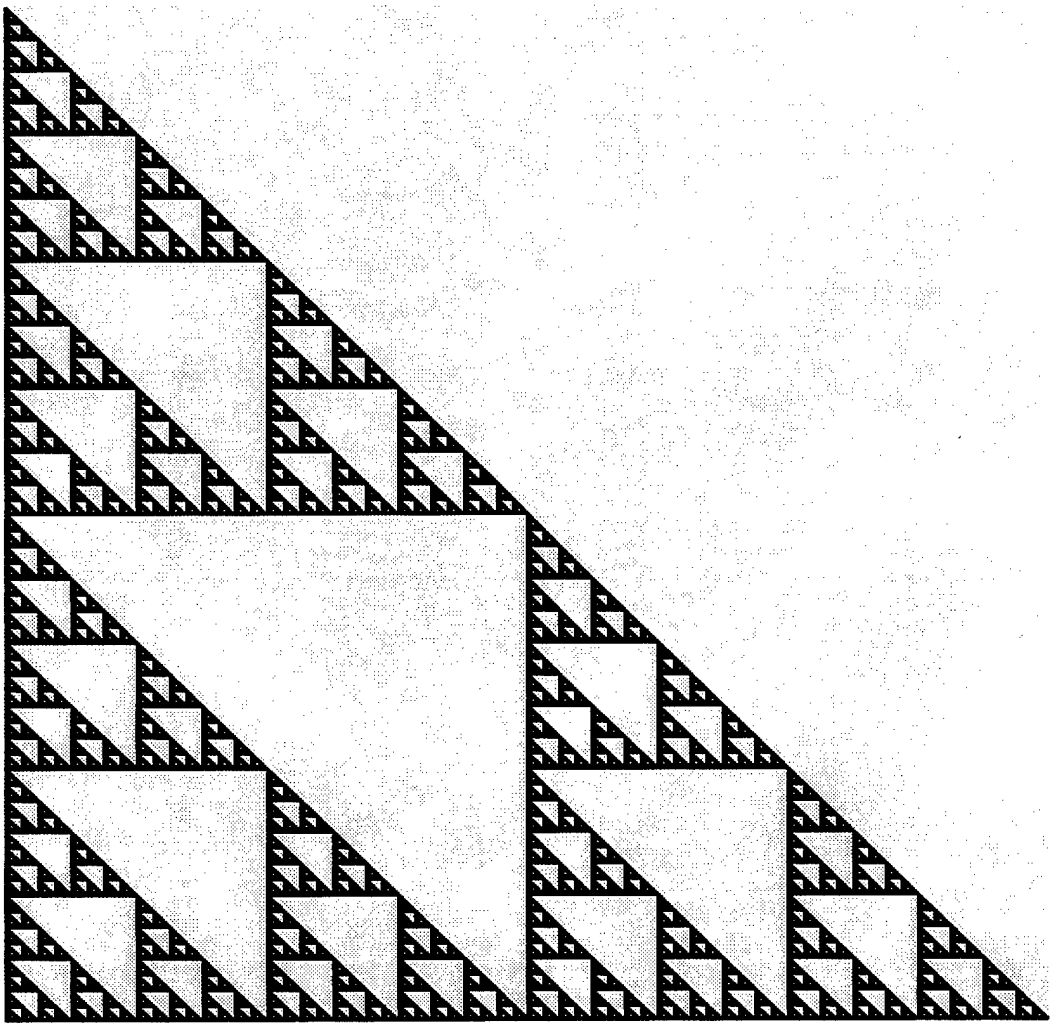


FIGURE 24.—*Sierpinski's triangle.*

The Metric To determine this "best fit," some measure of distortion between images is needed. The metric most commonly used is simply the *rms* error between the transformed domain and the range partition where $f(x, y)$ is the transformed domain value at the particular pixel (x, y), $g(x, y)$ is the range value at the same (x,y), and the summation to n counts all the pixels in the range block (21).

$$E_{rms} = \frac{1}{n^2}\sqrt{\sum_{x=0}^{n}\sum_{y=0}^{n}(f(x,y) - g(x,y))^2}. \tag{21}$$

Image Partitioning The partitioning of the original image is one of the key decision points in designing a fractal image compression implementation since the selection of the range blocks will to, a large degree, determine the fidelity of the restored image with the smaller range partitions giving the best results. However there is a tradeoff in that the more range blocks the more comparisons that must be done between range blocks and do-

main blocks to determine the optimum match. Although we will only discuss fixed-size range blocks here, the reader should be aware that much work is being done on hierarchical partitioning schemes such as the quadtree partitioning shown in Figure 25.

For other partitioning techniques, the reader should look to [13][12].

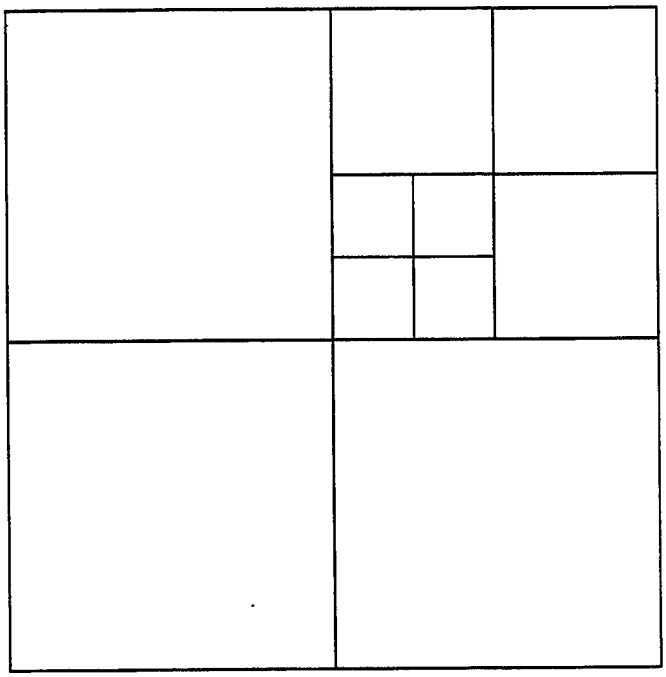


FIGURE 25.—*Quadtree partitioning of an image.*

Domain Determination The selection of the domains for the pool of domains that will be mapped into the ranges is fairly straightforward. If a fixed-range size is used, then the domain blocks are generally twice the size of the range blocks, and the collection is all the subdivisions of the image with overlapping allowed that are that size. Figure 4 shows some of the mappings that may occur.

If the ranges are determined by a scheme that generates a variety of different size range blocks, such as a quadtree scheme, then a technique must be used that generates a corresponding variety of larger domain block sizes. Whatever method is used, the more domain blocks in the domain pool, the better the mapping will be. Conversely, the more domain blocks there are to compare to the various range blocks, the longer the encoding will take. Much of the current work in fractal compression is focused on how the domain search space may be pruned to speed the search for "best match" between the domain and range blocks.

The Transforms Since we will be working with gray scale images (which may be easily extended to a color image), a slightly more complex affine transformation is necessary to handle the gray value (22).

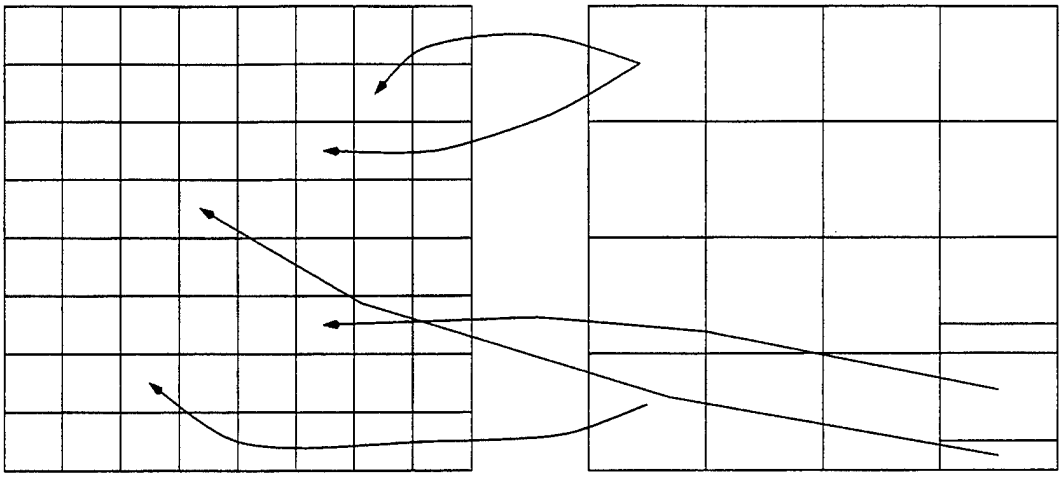


FIGURE 26.—*Domain block to range block mapping.*

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}, \tag{22}$$

where z is the gray level at pixel (x,y), and s_i is the contrast control and o_i is the brightness control.

Although theoretically any contractive affine transformation could be used in the mapping of domains to ranges, time and computational constraints demand that a smaller set of transforms be used to perform the mapping applied to domain blocks. Generally these transforms may be grouped into three categories: geometric, massic, and isometry [15] [16].

Geometric

The fact that the domains are chosen to be physically larger than the ranges implies that mapping from a domain block to a range block will be contractive. A geometric transformation may consist of subsampling the domain block so as to match the size of the range block, or more often, averaging the pixels of the domain block.

Massic

Massic transformations adjust the pixel value at each point. The two massic parameters are the contrast and brightness. To compute these it is necessary to minimize the quantity R by finding appropriate s and o. So given a_i pixel values from the transformed domain D_i and b_i from the range R_i, find s and o so that (23) is a minimum.

$$R = \sum_{i=1}^{n} (s * A_i + o - B_i). \tag{23}$$

Isometries

Isometries are transformations that do not distort the object being transformed. For a four-cell box, the isometries are shown in Figure 27. These transformations are

0. original image
1. identity
2. reflection about mid-vertical axis
3. reflection about mid-horizontal axis
4. reflection about first diagonal
5. reflection about second diagonal
6. rotation through $+90°$
7. rotation through $+180°$
8. rotation through $-90°$

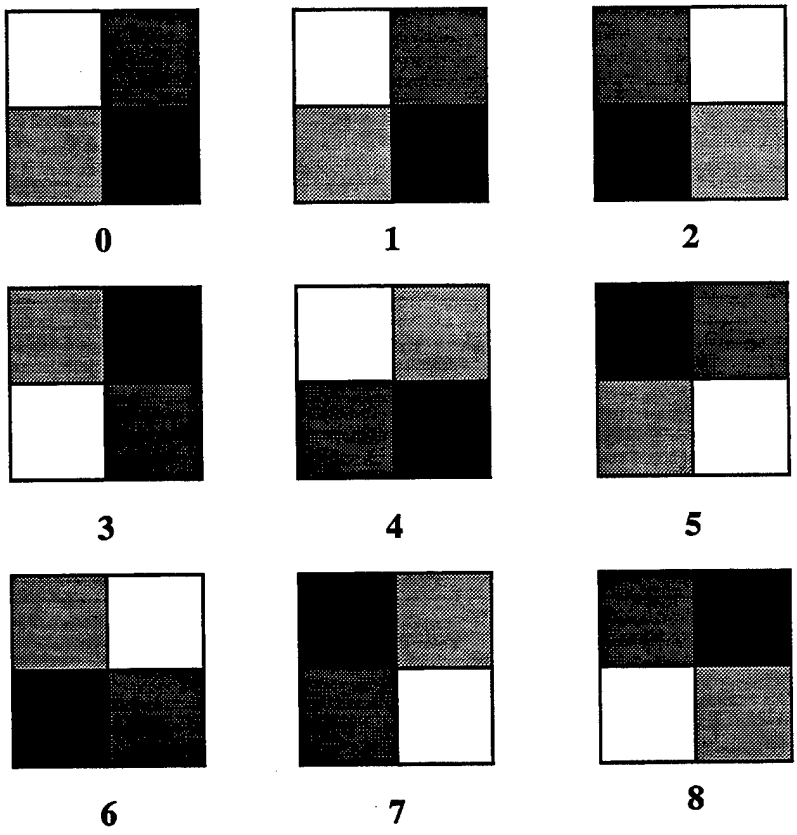


FIGURE 27.—*Isometry transformations of domain blocks.*

Storing the Compressed Image Once the transform parameters have been determined, the information must be stored. The information to be stored includes the size and location of the range blocks, the size and location of the domain blocks, and the transformation information. The fractal encoder used in this study was created by Fisher and uses the following typical values. s_i, the contrast, is quantized to 5 bits, and o_i, the brightness, is stored in 7 bits. For the quadtree partitioning used in this implementation, we use 1 bit to denote whether this block is subdivided or not. The range information is then stored as a side effect of storing the transformations. Domain information must include both size and position; however, for a given domain pool for a particular range partitioning, it is possible

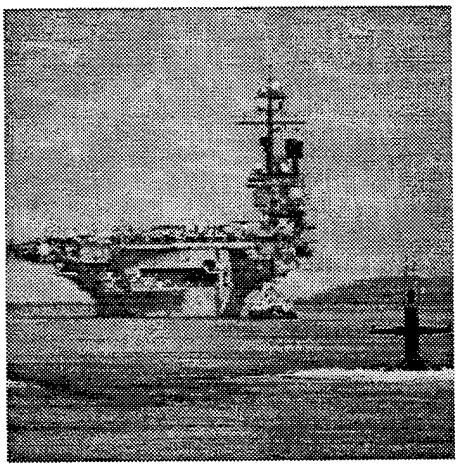
to index the domains and just store the index for the domain being mapped to this range block. Finally, to store the massic transformation information consisting of rotation and flip, 3 bits are used. Note that if the s_i is zero no domain or massic transformation information need be stored at all. All of this results in an average total of 31–34 bits per w_i. Contrast and brightness have a nonuniform distribution, so entropy coding is beneficial and the quantized values for s_i and o_i should be used when computing the error during encoding to obtain maximum fidelity. Finally, a brief header is included giving such information a final image size, maximum number of quadtree recursions, etc.

Decoding Decoding is simply a matter of reading in the stored information and running the IFS through a number of iterations. Generally, 7 or 8 iterations serve to bring the image to convergence. Although the choice of starting image will not affect the final image, it does have an effect on the speed of convergence. Typically, a uniform gray initial image will converge slightly faster than some arbitrary image and a low-resolution version of the compressed image will converge even faster [13]. Sometimes it is desirable to run a smoothing function over the image to reduce edge effects occurring at range block boundaries [12].

Conclusions - Fractal image compression has gotten off to a slow start. The exaggerated claims for fantastic compression ratios followed by patents on the enabling technology have suppressed the amount of research devoted to the development of fractal compression techniques. Recent surges in ongoing research, nearly 300 publications in 1996 [17], provide assurance that fractal image compression will have a place among modern image compression techniques.

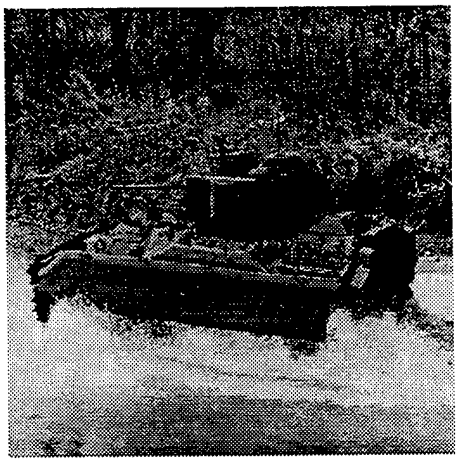
3. Performance of Image Compression Techniques

In this chapter we will compare the various image compression techniques outlined in Chapter 2. For this comparison, we will use the three images shown in Figure 28.



America

Lena



Bradley Infantry Fighting Vehicle (BIFV)

FIGURE 28.—*Original images.*

The America image is a grayscale image of the USS America, a Kitty Hawk Class Aircraft Carrier CV-66, that was decommissioned this year. The Lena image is an image that has been used as a standard image compression test image for the image compression industry. The last image is one of the Bradley Infantry Fighting Vehicle (BIFV) as seen in the field. These images were selected as test images because they exhibit various image characteristics. The America and BIFV images are typical images one might find in military applications. The BIFV images contains a considerable amount of high-frequency content, and the America image contains many low-frequency areas. The Lena image is used as a standard test image due the the variety of composition, namely the smooth areas of the background and shoulder and the high-frequency areas of the feathers on her hat.

All of these images are of size 256 × 256 pixels. Each of these images are grayscale, indicating that each pixel value is represented by a number from 0–255, 0 representing black and 255 representing white. Grayscale corresponds to 256 or 2^8 gray levels. Therefore, 8 bits are necessary to represent these 256 levels ($\log_2 256 = 8$). Subsequently, each image contains 256^2 pixels of 8 bits each or $256^2 * 8 = 524,288$ bits, approximately one-half megabit. Considering that these are rather small images and that they require .5 Mb of storage for a single picture, one can imagine how essential compression is for low bandwidth radio links not to mention the processing of video sequences that contain 30 single image frames per 1 second of viewing.

We will begin by using each compression technique to compress the image to 2 bits per pixel (bpp), 1 bpp and .5 bpp, corresponding to compression ratios of 4:1, 8:1, 16:1, respectively. The performance of the various schemes will be demonstrated at these bit rates by compressing the image to the desired compression ratio and then reconstructing the compressed image. This is analogous to transmitting the compressed image over a noiseless channel. We will then expose each compressed image to channel noise of varying amounts in a systematic manner. Channel noise was described in section 1.3. This will allow the reader to determine the amount of resiliency each technique has to image transmission in a noisy environment. Accompanying the qualitative comparison of the images, quantitative error measurements will also used to assess algorithm performance. These fidelity criteria for comparison are the MSE and PSNR, as indicated in (24) and (25), respectively. Here the variable x_i represents a pixel from the original image and $\hat{x}_i$ represents the same pixel from the reconstructed image. The ideal situation is to have an MSE of 0 and a PSNR of infinity.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{24}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{25}$$

3.1 Implementation Details

In order to provide a sampling of the state-of-the-art image compression methods, we have selected DPCM, JPEG, Wavelets, and Fractals, described in Sections 2.2.2–2.2.5, as the image compression techniques of choice.

3.1.1 DPCM

The DPCM program used for these simulations was written by the authors using MATLAB, a fourth generation mathematical simulations tool. The optimal linear prediction filter was determined for each of the three test images utilizing the procedure found in [5]. For the quantizer, optimal quantizers designed for a Laplacian source were used. This program performs only the DPCM compression. No subsequent entropy code was used. The reader is reminded that DPCM is a pixel-based method and images cannot be compressed lower than 1 bpp using DPCM.

3.1.2 DCT - JPEG6

To represent a DCT algorithm, version 6 of the Independent JPEG Group software was selected. The compression and decompression programs are cjpeg and jpeg, respectively. The code used in these examples was obtained from the "official" archive site for this software, which is ftp.uu.net (Internet address 192.48.96.9). The most recent released version can always be found there in directory graphics/jpeg. This particular version is archived as graphics/jpeg/jpegsrc.v6.tar.gz. More complete descriptions of the JPEG algorithm and the various options available to the user may be found in documentation found in the distribution and in Wallace [18], Nelson [19], and Pennebaker and Mitchel [20].

As mentioned in section 2.2.3, the JPEG algorithm consists of dividing the image into 8×8 blocks, transforming each of these subimages utilizing the DCT, storing the transform coefficients in "z-order" and quantizing the coefficients. The DPCM technique is used to encode the the DC coefficient of all the blocks. A different quantizer is used for the quantization of the high-frequency coefficients. The data is then encoded utilizing Huffman encoding, also described in chapter 2.

Using this JPEG program, it is not possible to specify an exact output file size but instead the user is furnished a "quality factor" that allows tradeoffs between compressed image size and image quality. The "quality" argument varies from 0 to 100 with the best image quality being obtained at high values (75 is an often chosen default). Of the many command line arguments available, the only one specified was the quality factor for our simulations. Default values were used for all other program options. The values used for the quality factor and the resulting file sizes are shown in Table 4.

TABLE 4.—*JPEG6 Quality Factor and Bit Rates*

Image	Desired Rate	Quality Factor	Resultant Rate
America	2.0	89	1.99
America	1.0	60	0.86
America	0.5	22	0.42
Lena	2.0	86	1.99
Lena	1.0	56	1.00
Lena	0.5	18	0.51
BIFV	2.0	54	2.00
BIFV	1.0	20	1.00
BIFV	0.5	8	0.51

3.1.3 Wavelets - Set Partitioning in Hierarchical Trees (SPIHT)

Wavelet image compression is represented by the SPIHT algorithm of Amir Said and William Pearlman [21]. This algorithm combines wavelet techniques with the Zerotree encoding of Shapiro [22] to achieve "state-of-the-art" image compression. Additional information

may be obtained from the SPIHT home page on the World Wide Web at http://ipl.rpi.edu pointer SPIHT. The code used in these demonstrations, "EW_code", was obtained from ftp://ipl.rpi.edu/pub/EW_Code. The compressor, codetree, and the decompressor, decdtree, both offer interactive and batch user interfaces. For this work, the batch interfaces were used. Encoding was done using arguments naming the input and output files, x and y sizes, and a target bit error rate. Zerotree encoding techniques allow compression to specified bit rates. This program utilizes the DWT as described in Chapter 2. The coefficients of the DWT are then quantized and encoded via the SPIHT algorithm. This data is then further compressed by an arithmetic encoder, which compresses the information losslessly.

3.1.4 Fractals

To represent fractal image compression, an implementation by Yuval Fisher, ENC/DEC, was chosen. This program is described in detail in Fisher's book [12], and the code used was obtained from his home page at http://inls.ucsd.edu/y/Fractals. These programs, ENC for encoding and DEC for decoding, implement a straightforward fractal compression scheme using quadtree partitioning. Of the large number of command line arguments available to the user of this code, only the tolerance, or "-t," flag was used. The tolerance is typically selected in the range of 2–15, with lower values resulting in larger better looking images. Since this tolerance flag does not allow the user to exactly specify the size of the compressed file, it was necessary to experimentally determine tolerances so that the desired bit per pixel rates could be approximated. Table 5 displays the actual tolerance and corresponding bit rate used for the simulations.

TABLE 5.—*Fractal Tolerance and Bit Rates*

Image	Desired Rate	Tolerance	Resultant Rate
America	2.0	0	1.58
America	1.0	4	0.92
America	0.5	10	0.52
Lena	2.0	1	1.58
Lena	1.0	6	1.03
Lena	0.5	17	0.51
BIFV	2.0	0	1.54
BIFV	1.0	24	1.00
BIFV	0.5	46	0.51

3.2 The Noiseless Channel

In this section we will assume the following system shown in Figure 29. The compressed image incurs no errors as if the image were stored on a disk or passed over a noiseless channel. The reader will find all reconstructed images compiled in Appendix A for easy reference.

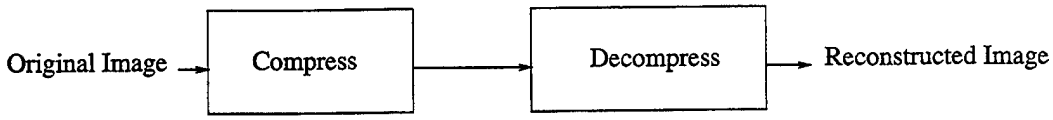


FIGURE 29.—*Noiseless channel test scenario.*

Figures A-1–A-3 display the performance of the DPCM algorithm, JPEG, SPIHT, and fractals at 2 bpp or 4:1 compression ratio. As one can see, the difference between the respective compression methods at such a high rate of 2 bpp is miniscule.

Figures A-4–A-6 depict the performance of the compression techniques at a rate of 1 bpp or a compression ratio of 8:1. Performance differences between the algorithms are becoming more apparent at this rate.

Figures A-7–A-9 demonstrate compression performance at .5 bpp, a rate of 16:1. DPCM is unable to compress the image to the level because it is pixel based; therefore a reconstructed image does not exist for DPCM at rates that are a fraction of a pixel. The reader is able to discern the differences between the techniques as the bit rate is lowered and higher compression is obtained. JPEG6 begins to show some blocking artifacts at this rate: this phenomenon is particularly noticeable in the America image. The fractal method produces images that appear to be smoothed or smeared; this is evident in all fractal test images. By far the best performer is the SPIHT implementation of wavelet compression. Even at a low bit rate of .5 bpp the SPIHT algorithm produces results that are very acceptable.

Tables 6 and 7 contains the data for the quantitative portion of the comparison. A *lower* MSE indicates that the reconstructed image is most like the original. The PSNR is inversely proportional to the MSE; therefore, a *high* PSNR indicates that the reconstructed image is most like the original image.

TABLE 6.—*MSE Results for Noiseless Channel*

Image	Bit Rate	DPCM	JPEG	SPIHT	Fractals
America	2.0	38.3	8.55	3.65	49.70
	1.0	145.7	26.38	11.64	52.90
	0.5	—	50.16	24.52	66.05
Lena	2.0	79.6	12.31	4.05	68.85
	1.0	285.6	37.27	14.57	73.83
	0.5	—	83.21	41.12	134.66
BIFV	2.0	244.5	169.21	47.32	729.24
	1.0	890.5	533.84	267.31	773.96
	0.5	—	836.57	595.84	1164.36

TABLE 7.—*PSNR Results for Noiseless Channel*

Image	Bit Rate	DPCM	JPEG	SPIHT	Fractals
America	2.0	32.3	38.81	42.51	31.17
	1.0	26.5	33.92	37.47	30.90
	0.5	—	31.13	34.24	29.93
Lena	2.0	29.1	37.23	42.06	29.75
	1.0	23.6	32.42	36.50	29.45
	0.5	—	28.93	31.99	26.84
BIFV	2.0	24.2	25.85	31.38	19.50
	1.0	18.6	20.86	23.86	19.24
	0.5	—	18.91	20.38	17.47

3.3 The Noisy Channel

The overall test scenario for the noisy channel is shown in Figure 30. The original image is compressed and then transmitted over a noisy channel where errors are incurred. The corrupted, compressed image is received and then decompressed to result in the reconstructed image.

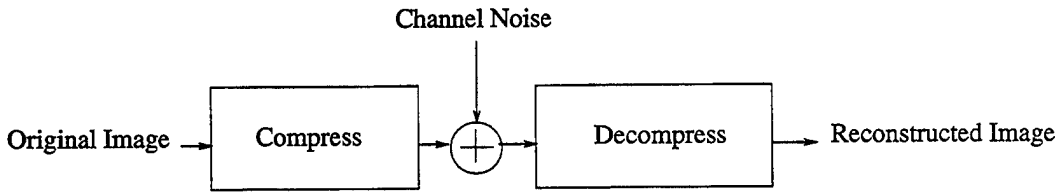


FIGURE 30.—*Noisy channel test scenario.*

The simple channel model that will be used is known as the Binary Symmetric Channel (BSC), which is shown in Figure 31. Here, each bit is independently exposed to a channel error; the bit is changed from 0 to 1 or 1 to 0 based on the probability of error ϵ. The probability that the bit will remain unchanged, 0 to 0 or 1 to 1, takes on the value of 1 - ϵ. In order to put this probability of error, also known as the bit error rate (BER), into perspective we inform the reader that a fiber optic cable, which is nearly error free, has a BER of approximately 10^{-12}. Radio communications are typically in the range of 10^{-5} to 10^{-1} BER, depending on the quality of channel. Consequently, as the BER increases the channel becomes more noisy and therefore the data is more prone to error.

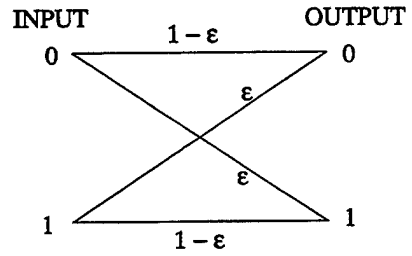


FIGURE 31.—*Binary symmetric Channel.*

We have included the pseudo code used to implement this binary symmetric channel in order to clarify the process of adding noise to the compressed image.

```
/*Generate random matrix with a uniform distribution and initialize noise matrix*/
randmat = rand(row,col);
noise = zeros(row,col);

/*Where random number less that epsilon - error - set noise matrix to 1*/
rr,rc = find(randmat < epsilon);
numerrors = size(rc);
for i=1:numerrors
```

```
noise(rr(i),rc(i)) = 1;
end

/*Exclusive Or operation for input and noise to "flip bits" when errors occur*/
newoutput = xor(input,noise);
```

The BSC is a very simple way of introducing bit error into a transmission stream. It does not contain all of the characteristics of the wireless battlefield environment, such as fading, burst errors, multipath and cochannel noise, where bit error probabilities may relate to one another. However, it is an accepted method that adds noise to a bit stream to determine an algorithms resiliency.

Noisy environments are the primary concentration of this report. Therefore, we will consider error rates of 10^{-5} up to 10^{-1} on boundaries of powers of 10. All images and techniques will be evaluated at these noise levels. A single bit rate of 1 bpp has been chosen for this evaluation, allowing all techniques, particularly DPCM, to be included in the assessment. In some cases, as the noise level was increased the decoders could not decode the corrupted, compressed image. This situation, where a reconstructed image does not exist nor are there quantitative values of error, will be duly noted. Appendix B will be used to demonstrate the performance of each technique through the reconstructed images. However, the quantitative results will be shown in graph form for ease of comparison.

Figures B-1–B-3 display each algorithm's resiliency to noise when the BER is 10^{-5}. Even at this low rate of error, some images are showing signs of corruption. The DPCM image has hardly changed from the noiseless DPCM 1 bpp version. The JPEG algorithm shows corruption on the last row of blocks within the Lena image. Corrupted blocks are also noted within the JPEG BIFV image. In all three test images, the Wavelet algorithm shows degradation: this is grossly exhibited by the fuzzy look of the Lena image. The fractal method shows some degradation also in the Lena image, where small segments of the image are corrupted.

The BER is increased to 10^{-4} for Figures B-4–B-6. At this noise level, the DPCM image has again been slightly altered. The JPEG images show block corruption more frequently, with the Lena image totally corrupted after half of the image has been received. This type of corruption can be primarily attributed to the Huffman entropy encoding. Because an entropy code is uniquely decodable and codewords have variable length, when an error occurs, it could cause incorrect decoding. In this case, the decoder loses synchronization of the code and all codewords after this point could be decoded incorrectly until the decoder is resynchronized. The SPIHT algorithm shows further degradation as the noise is increased. The same is true for the fractal algorithm.

The BER is again incremented to 10^{-3} in Figures B-7–B-9. The DPCM algorithm is still providing recognizable images with a few noticeable errors. The JPEG method eventually loses synchronization on the America and Lena images, and the JPEG BIFV is totally corrupted. The wavelet algorithm provides some nice abstract art at this error level, providing totally useless images. Finally, the fractal images are also entirely corrupted.

Figures B-10–B-12 demonstrate algorithm performance at 10^{-2} BER. The DPCM images are still quite good. The JPEG decoder has ceased to function entirely; a reconstructed image does not exist. The SPIHT wavelet algorithm provides more abstract art of higher frequency. And the fractal method again results in unrecognizable images.

The highest error rate of our simulations, 10^{-1}, where 10% of all received bits are in error, is displayed in Figures B-13–B-15. In this noisy environment, the only operational algorithm is the DPCM algorithm. The JPEG, SPIHT, and fractal methods have all ceased to function.

For the quantitative results, graphs have been constructed. Figure 32 displays the MSE for each test image over the range of BER. From these graphs, one can see that the MSE approaches infinity for BER greater than 10^{-4} - 10^{-3} for the JPEG, SPIHT, and fractal algorithms. However, the DPCM algorithm retains reasonable performance up to very high BER levels of 10^{-1}.

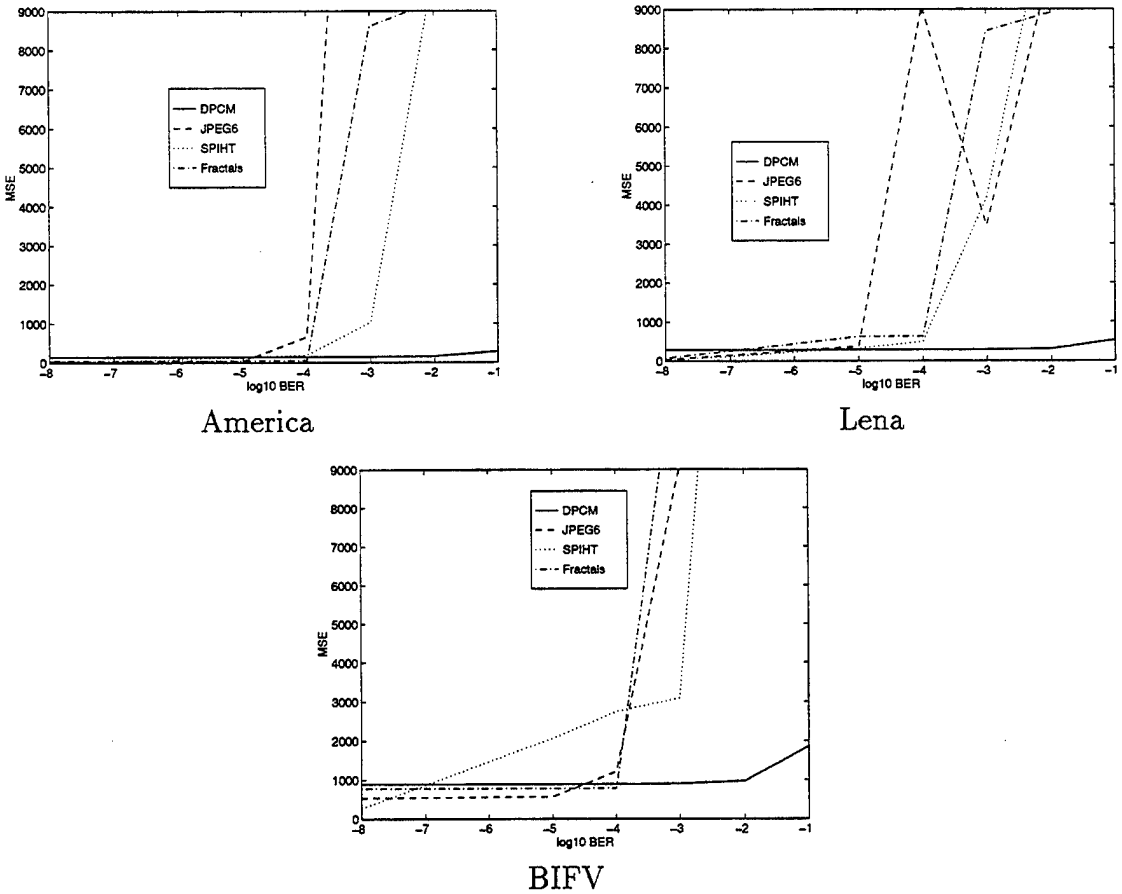
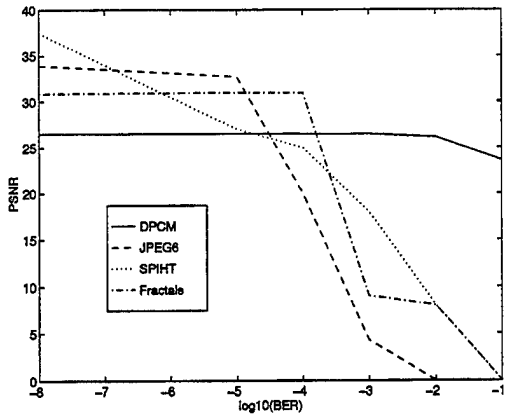


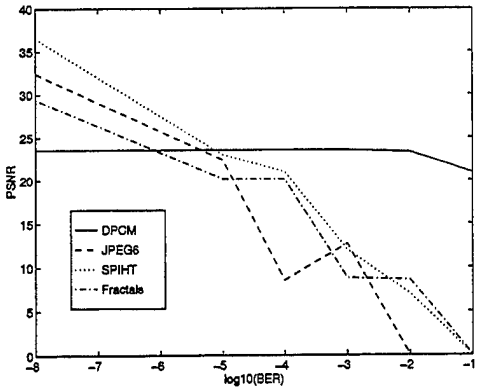
FIGURE 32.—*MSE performance.*

Figure 33 depicts the PSNR results. A PSNR value above 35 dB typically depicts a reconstructed image that is very much like the original; therefore, these results are easier to generalize than the MSE. At low error rates, the SPIHT/wavelet algorithm performs at this level for the America and Lena images. This algorithm also surpasses the others in

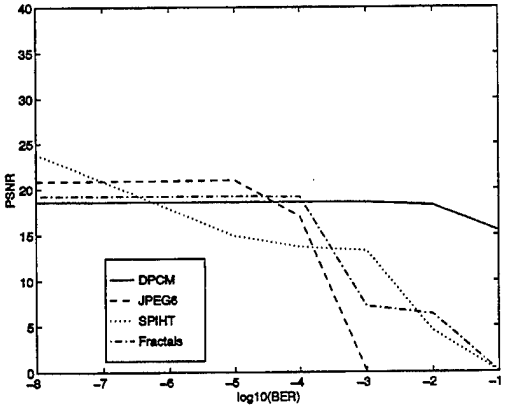
performance at low BER. Consequently, when the BER is increased, the DPCM method provides continuous performance where the PSNR of the other algorithms quickly fall to zero.



America



Lena



BIFV

FIGURE 33.—*PSNR performance.*

4. Methods of Coping in a Noisy Environment

The previous chapter demonstrated the resiliency to noise, or lack thereof, of the current state-of-the-art image compression techniques. From these simulations, it can be deduced that to achieve *productive* image transmission in a noisy environment, certain concessions must be made.

A few of the subject compression techniques utilize an entropy code to obtain extra compression. As mentioned in chapter 3, an entropy code is uniquely decodable and codewords have variable length. Errors incurred on entropy encoded data may cause the decoder to lose synchronization. By eliminating the entropy code altogether or adding resynchronization flags to the entropy code, it is possible to obtain better performance in a noisy environment. The resynchronization technique will be demonstrated in the next section.

One of the most obvious ways to protect image data from the effects of noise is to add a forward error correction (FEC) code. FEC is a channel coding technique that adds structure redundancy to the encoded data so that it may be reconstructed if damaged by transmission noise. This topic was introduced in subsection 1.4, where the tradeoff between source and channel coding was described. FEC is used for one direction transmission, from transmitter to receiver, and automatically corrects errors detected at the receiver [23]. FEC is generally used in those situations where retransmission schemes are difficult or impossible to implement, such as compact discs and noisy wireless channels.

By utilizing compression, we attempt to compact the data in order to adhere to some bandwidth constraint. To add an FEC, the data must be further compressed to allow for the additional bandwidth of the FEC and still meet the total bandwidth constraint. FECs are capable of correcting a fixed number of channel errors. For the very noisy channel, e.g. BER of 10^{-2} and above, most FECs cannot correct this many errors and their presence can actually cause more harm to the data it is trying to protect. However "low-rate" FECs, like the 255,4 Reed Solomon Code can correct a large amount of errors. This code produces 255 output bits for every 4 input bits. These very low-rate codes are capable of correcting errors at a BER well above 10^{-1} but utilize a significant amount of bandwidth in return [24]. An example of the result obtainable from the use of an FEC will be shown within this chapter.

Another method that allows for compression without the need for channel coding is called Robust Source Coding [25]. This method uses a nonlinear prediction filter with pixel based compression methods similar to DPCM. Joint source channel coding processes are also a viable alternative [26], [27], [28]. Soft decision decoding [29], power control schemes [30], and many other techniques are also used for transmission in noisy environments. However, there is no single, widely accepted method to provide reliable image transmission in noisy environments.

4.1 Modifications to Current Techniques

In this section, we will provide two examples of possible modification to the subject image compression techniques that will allow for better performance over noisy channels.

The reconstructed images provided are compressed at rate 1 bpp and exposed to channel noise of 10^{-3} BER.

4.1.1 JPEG6 Modifications

JPEG version 6 [18] offers the restart option to resynchronize the entropy code as an effort to improve performance over noisy channels. This option is invoked by the command line argument, "-restart n", and causes a restart marker to be inserted every n MCU (minimum coded unit) rows or MCU blocks. An MCU is an interleaved set of blocks of size determined by the sampling factors, or a single block in a noninterleaved scan. A block is an 8×8 group of frequency coefficients. These restart markers allow the entropy coding scheme to regain synchronization after noise is encountered, provided that the restart markers themselves are not corrupted by error. Of course the penalty paid for this more robust coding is the increased file size.

As an aside: the MCU is a set of interleaved blocks in the case of a multicomponent image and a single block otherwise. The MCU concept is used to simplify the process of decompression particularly when display and decompression are occurring simultaneously.

Table 8 shows resulting file sizes for a variety of restart configurations. Keep in mind that the larger file size will occupy additional bandwidth on the channel.

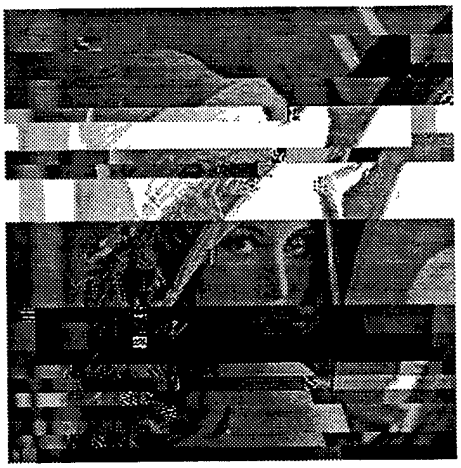
TABLE 8.—*JPEG6 File Size With Restart Markers*

Restart Interval	File Size (bytes)
NONE	8206
2 rows	8246
1 row	8274
10 blocks	8471
5 blocks	8723
2 blocks	9499
1 block	10821

Although the restart markers will help the entropy coding scheme avoid catastrophic failure, they are not always successful in ensuring successful decompression. Figure 34 shows the effect adding these restart markers has on image reconstruction. As the figure demonstrates, "more robust coding" is not always more robust.

One should note that JPEG6 performance without restart markers can be viewed in Figures B-7–B-9, in Appendix B, under the same channel conditions. By placing the restart markers within n rows, the performance of the algorithm has markedly improved. By increasing the number of markers within a row to every n MCU blocks, the performance continues to improve and the file size increases along with the need for bandwidth.

The JPEG version 6 also supports a "-progressive" flag, which could be useful in a noisy environment. For instance, if the initial portion of the file is received without error, a low



Every 5 Rows

Every Row

Every 10 Blocks

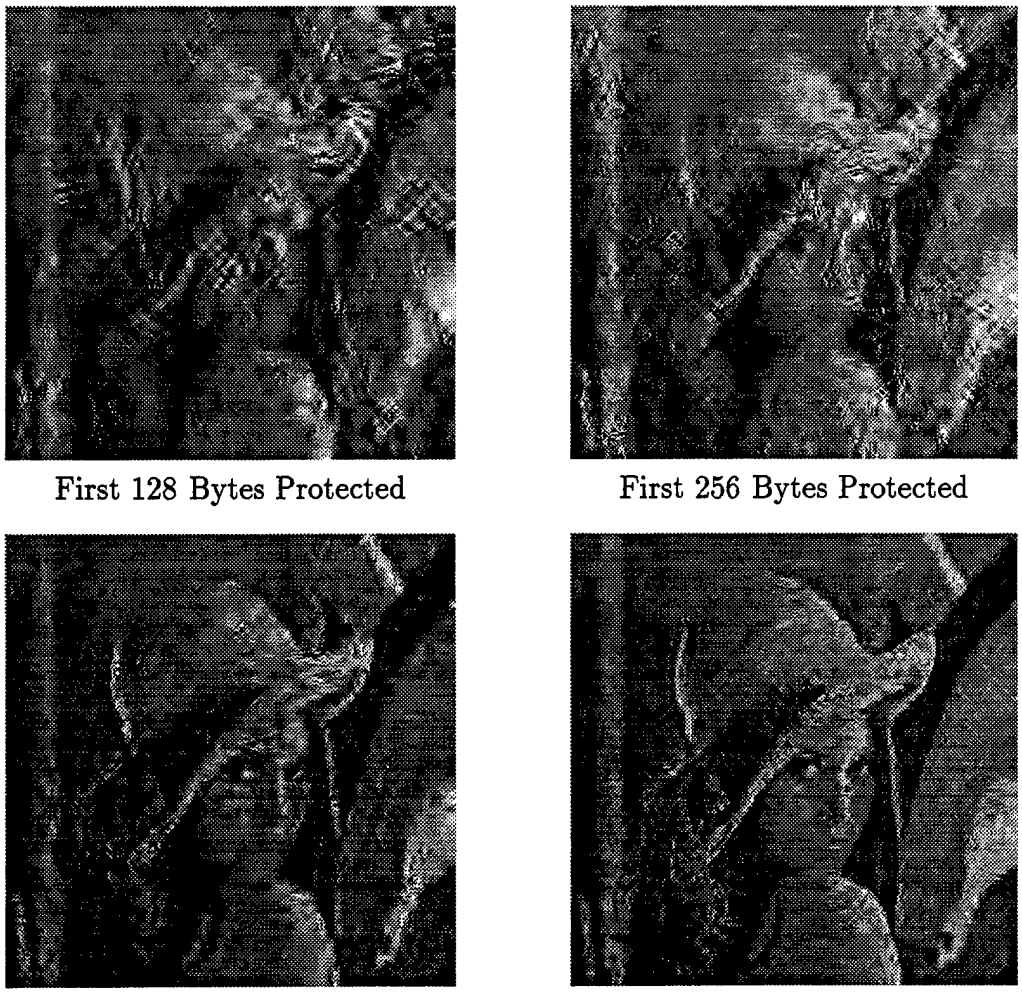
Every 5 Blocks

FIGURE 34.—*Reconstructed images with JPEG6 restart markers.*

resolution image could be displayed. Unfortunately, in the trials attempted by the authors, every file was catastrophically damaged by the addition of noise and failed to decompress.

4.1.2 SPIHT Modifications

The wavelet coding techniques implemented in the SPIHT algorithm allow for demonstration of another technique for noisy channel compression. In this algorithm, the DC levels are coded at the beginning of the compressed file, so that if we protect the beginning of the file using some type of FEC, a usable image may be recovered even in a very noisy environment. Figure 35 shows the effects of protecting the first N bytes of the SPIHT encoded file. In this case, the effects of protecting the initial bytes were simulated by *not* inserting noise into this portion of the compressed image file as if an FEC code had detected and corrected all errors encountered within the first N bytes.



First 128 Bytes Protected

First 256 Bytes Protected

First 512 Bytes Protected

First 1024 Bytes Protected

FIGURE 35.—*Reconstructed SPIHT images, first N bytes protected.*

We cite a few examples of FEC codes that could be used to protect the data resulting

in an effective BER of approximately 10^{-5}, the BER at which all test algorithms performed well. One can note that protecting the first 1024 bytes provides the best quality image under these channel conditions of those in Figure 35. Table 9 shows the types of codes and their respective overhead for protecting a block size of 1024 bytes containing the compressed image data. It should be emphasized that the addition of an FEC is not free but costs not only in bandwidth but also in computation. For reference, the original file is 8192 bytes in length.

TABLE 9.—*Estimated Coding Overhead for Modified SPIHT Algorithm*

Code	Overhead (bytes)	Final File Size (bytes)
NONE	0	8192
Reed Solomon(127,115)	108	8300
Reed Solomon(255,235)	88	8280
Reed Solomon(511,481)	66	8258
Reed Solomon(1023,973)	57	8249
BCH(2047,1992;10)	35	8227

The zerotree type of hierarchical coding works well with most wavelet algorithms which partition the data into data sets, each with a different contribution to the reconstructed image. These algorithms also automatically lend themselves to progressive transmission because of the partitioned data sets. Utilizing this method, a low-resolution image could be first received and then the image could be improved in resolution and detail as more data sets were received and decoded.

5. Conclusions

5.1 Summary

Through this report, we hope to have familiarized the reader with the current state-of-the-art image compression techniques and the issues associated with the transmission of imagery in the battlefield. To accomplish this, an overview of these issues was provided, the technical details of several types of compression techniques were discussed, and the performance of the current image compression algorithms, namely DPCM, JPEG6, SPIHT, and fractals, was demonstrated. The performance was divided into two primary categories, the noiseless channel and the noisy channel. For the noisy environment, intended to simulate the battlefield, the compressed images were subjected to several noise levels in order to fully demonstrate the effects of this type of channel. It was shown that for very noisy channels only the DPCM algorithm provided a reconstructed image and that the JPEG6 program was the first of the transform techniques to cease functioning. Modifications to the techniques were also presented and yielded a slight improvement in the quality of the reconstructed image. All of these efforts bring us to the conclusion that more effective, resilient image compression and transmission schemes should be developed in order to provide the capability of reliable image transmission on the wireless digital battlefield.

5.2 Future Work

Military imagery tends to fall into two general categories: the first being low-resolution images for threat identification or general monitoring of an area; and the second being high-resolution imagery for battle damage assessment, telemedical applications, etc. Therefore, a general solution to natural image compression for battlefield applications should be able to fulfill both needs. In addition, the compressed image should be robust in the noisy environment of combat net radio. From this survey, the authors deduce that the wavelet image compression technique provides the best possible compression results.

As an example, Figure 36 shows the reconstructed image of Lena produced using the SPIHT algorithm, with the compression ratio noted below each image. Even at relatively high compression ratios, such as 64:1, the wavelet method performs very well. Since such a large amount of compression can be obtained with good performance, an FEC could be added to provide protection for a certain amount of noise. However, this is not a complete solution. It may not be possible to obtain *enough* compression in order to provide the bandwidth necessary for the FEC and still produce a recognizable image. Additionally, at some noise level (i.e, burst of errors), the FEC cannot correct all of the errors and can actually cause more distortion to the reconstructed image than by just transmitting the image without it. This occurs because now the image and the FEC together contain more bits; therefore, there are more bits which can be in error. A more effective scheme must be devised to deal with the transmission of imagery in this unstable wireless environment.

Compression Ratio 32:1, .25 bpp

Compression Ratio 64:1, .125 bpp

Compression Ratio 128:1, .0625 bpp

Compression Ratio 256:1, .03125 bpp

FIGURE 36.—*High compression of the Lena image.*

6. References

[1] C. E. Shannon.
A mathematical theory of communication.
The Bell Systems Technical Journal, XX VII(3), July 1948.

[2] T. M. Cover and J. A. Thomas.
Elements of Information Theory.
John Wiley & Sons, Inc., New York, NY, 1991.

[3] E. Ayanoğlu and R. M. Gray.
The design of joint source and channel trellis waveform coders.
IEEE Transactions on Information Theory, 33(6):855–865, November 1987.

[4] V. Bhaskaran and K. Konstantinides.
Image and Video Compression Standards - Algorithms and Architectures.
Kluwer Academic Publishers, Norwell, MA, 1995.

[5] R. C. Gonzalez and R. E. Woods.
Digital Image Processing.
Addison-Wesley Publishing, New York, NY, 1992.

[6] I. Daubechies.
Orthonormal bases of compactly supported wavelets.
Communications in Pure and Applied Mathematics, 41:906–996, 1988.

[7] M. Vetterli and Kovačević J.
Wavelets and Subband Coding.
Prentice-Hall, Inc., Englewood Cliffs, NJ, 1995.

[8] B. Mandelbrot.
The fractal geometry of nature.
W. H. Freeman, San Francisco, CA, 1983.

[9] J. Hutchinson.
Fractals and self-similarity.
Indiana University Mathamatics Journal, 3(5):713–747, 1981.

[10] M. Barnsley and S. Demko.
Iterated function systems and the global construction of fractals.
Proceedings of the Royal Society of London, A399:243–275, 1985.

[11] A. Jacquin.
Image coding based on a fractal theory of iterated contractive image transformations.
IEEE Transactions on Image Processing, 1(1):18, 1992.

[12] Y. Fisher, et.al.
Fractal Image Compression - Theory and Application to Digital Images.
Springer-Verlag, New York, NY, 1995.

[13] J. Kominek.
Advances in fractal compression for multimedia applications.
Technical report, http://links.uwaterloo.ca/TOC4, 1996.

to appear in MultiMedia Systems Journal.

[14] J. Kominek.
Subject 17 - What is the state of fractal compression - tal kubo and subject 77 Introduction to fractal compression.
Technical report, http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/part1/faq.html, 1995.

[15] T. Hemmingsen C. Frigaard, J. Gade and T. Sand.
Image compression based on a fractal theory.
Technical report, ftp://ftp.informatik.uni-freiburg.de/papers/fractal/, 1994.

[16] A. Jacquin.
Fractal image coding based on a theory of iterated contractive image transformations.
Proceedings of the SPIE Visual Communications and Image Processing, pages 227–239, 1990.

[17] D. Saupe H. Hartenstein and R. Hamzaoui.
Fractal image compression an introductory overview.
Technical report, ftp://ftp.informatik.uni-freiburg.de/papers/fractal/, 1996.

[18] G. Wallace.
The JPEG still picture compression standard.
Communications of the ACM, 34(4):30–44, April 1991.

[19] M. Nelson.
The Data Compression Book.
M&T Books, Redwood City, CA, 1991.

[20] W. Pennebaker and J. Mitchel.
JPEG Still Image Data Compression Standard.
Van Nostrand Reinhold, New York, NY, 1993.

[21] A. Said and W. Pearlman.
A new fast and efficient implementation of an image codec based on set partitioning in hierarchical trees.
IEEE Transactions on Circuits and Systems for Video Technology, 6(3):243–250, June 1996.

[22] J.M. Shapiro.
Embedded image coding using zerotrees of wavelets coefficients.
IEEE Transactions on Signal Processing, 41:34–45, December 1993.

[23] S. Lin and C. J. Costello Jr.
Error Control Coding: Fundamentals and Applications.
Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[24] Charles T. Retter.
Decoding binary expansions of low-rate Reed-Solomon codes far beyond the bch bound.
Proceedings of the International Symposium on Information Theory, Whistler, British Columbia, page 276, September 1995.

[25] L. M. Marvel, A. S. Khayrallah, and C. G. Boncelet, Jr.
Robust source coding of images for tactical channels.
Proceedings of the 20th Army Science Conference, Norfolk, VA, June 1996.

[26] M. Wang and T. R. Fischer.
Trellis-coded quantization designed for noisy channels.
IEEE Transactions on Information Theory, 40(6):1792–1802, November 1994.

[27] A. Goldsmith.
Joint source/channel coding for wireless channels.
Proceedings of the 1995 IEEE 45th Vehicular Technology Conference, Chicago, IL, 1995.

[28] G. Davis and J. Danskin.
Joint source channel coding for internet image transmission.
Proceedings of the SPIE Conference on Wavelet Apllication of Digital Image Processing XIX, Dever, CO, August 1996.

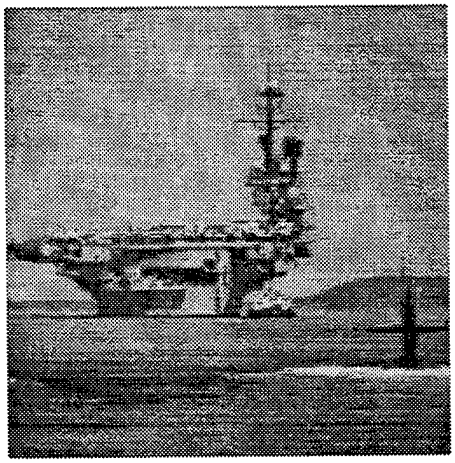
[29] O. Olaniyan.
Implementable soft decision decoding schemes.
International Journal Electron, 66(3):321–332, March 1989.

[30] A. Jalali and P. Mermelstein.
Effects of diversity, power control, bandwidth on the capacity of microcellular CDMA systems.
IEEE Journal on Selected Areas in Communications, 12(5):952–961, June 1994.

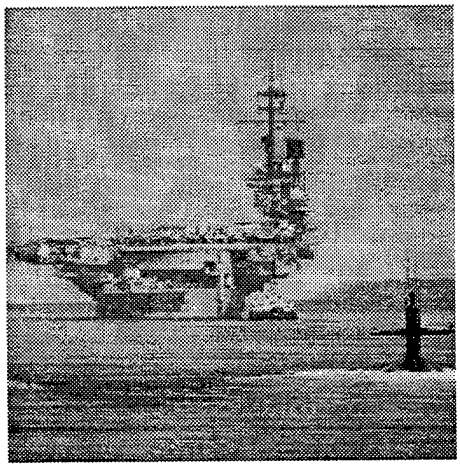
Appendix A:

RECONSTRUCTED IMAGES FOR NOISELESS CHANNEL

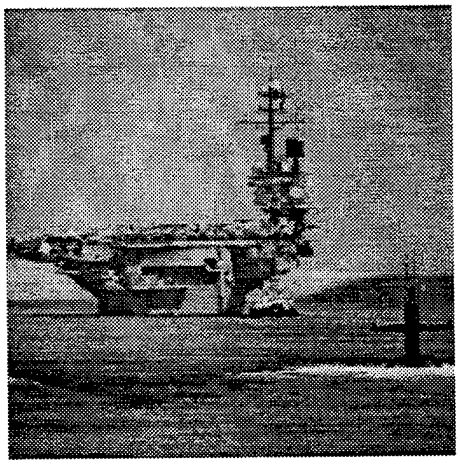
INTENTIONALLY LEFT BLANK.



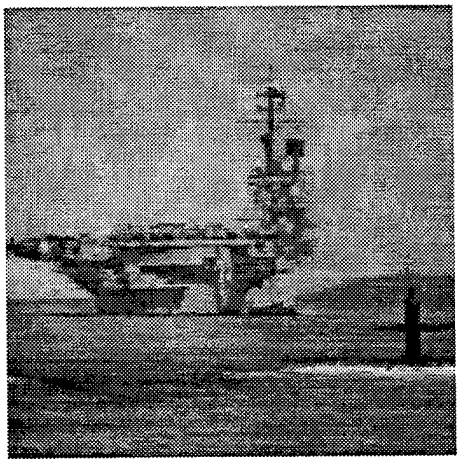
DPCM



JPEG6



SPIHT



Fractals

FIGURE A-1.—*America image, 2 bpp, noiseless channel*

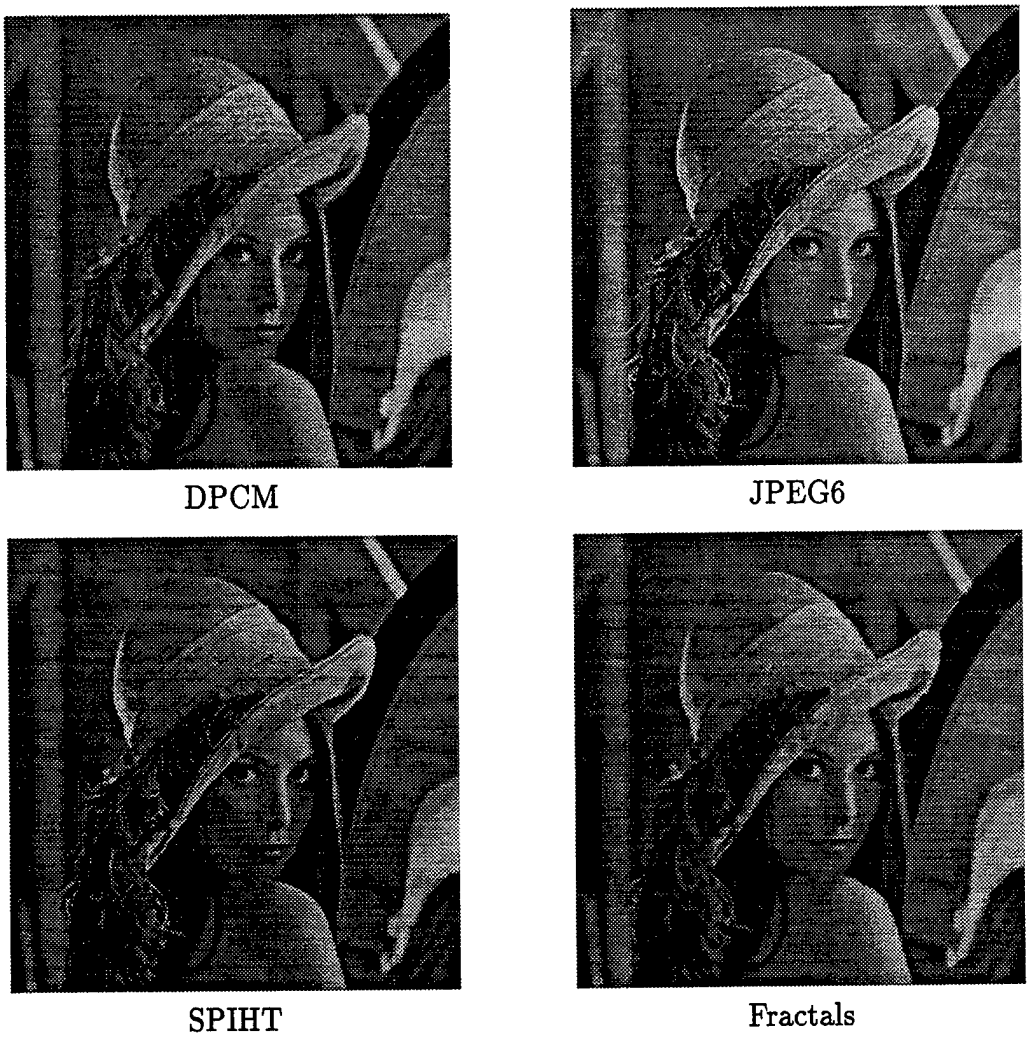
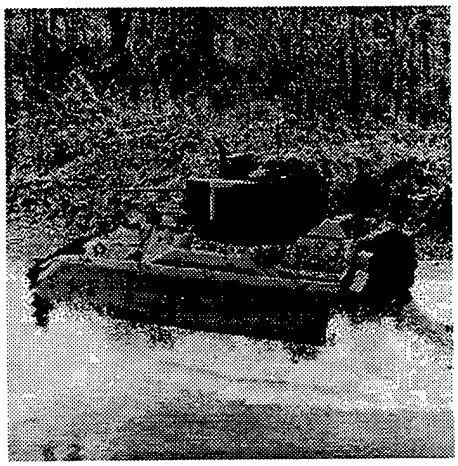
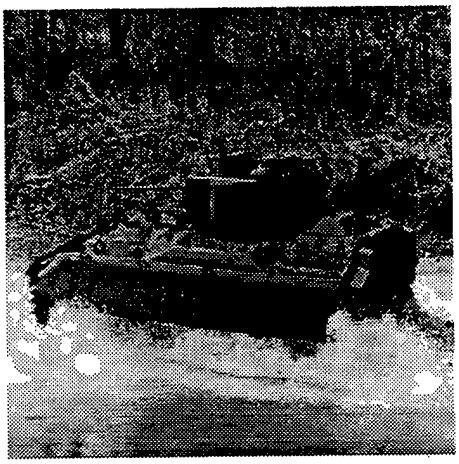


FIGURE A-2.—*Lena image, 2 bpp, noiseless channel*

DPCM



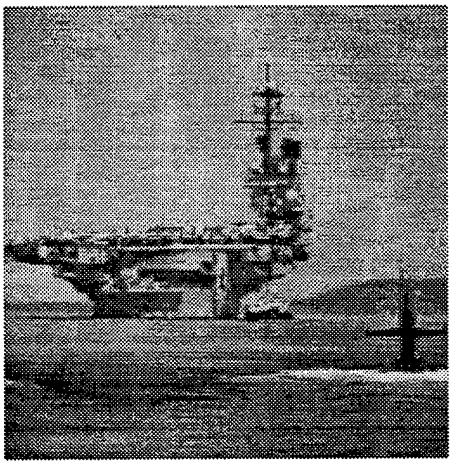
JPEG6



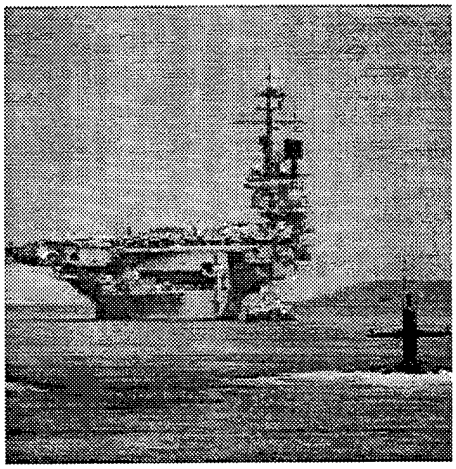
SPIHT

Fractals

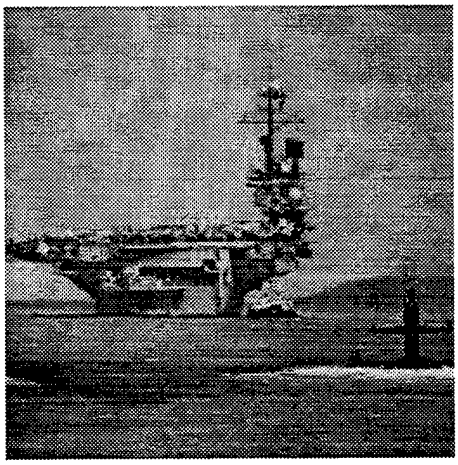
FIGURE A-3.—*BIFV image, 2 bpp, noiseless channel*



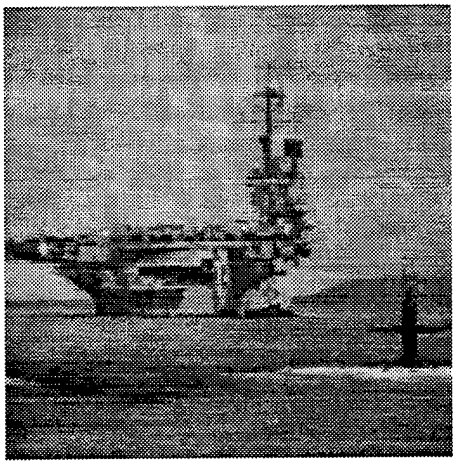
DPCM



JPEG6



SPIHT



Fractals

FIGURE A-4.—*America image, 1 bpp, noiseless channel*

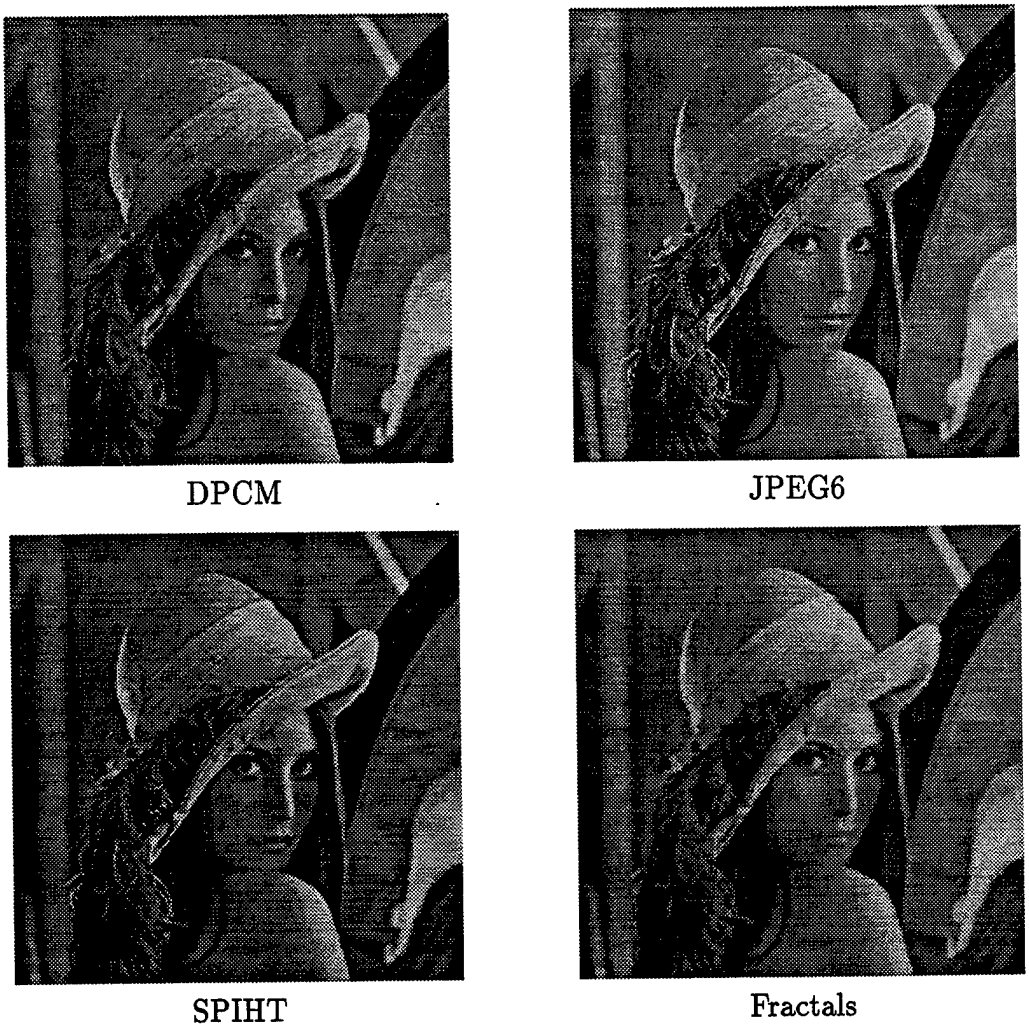
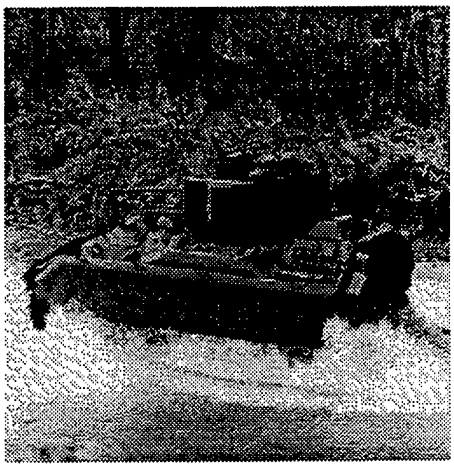
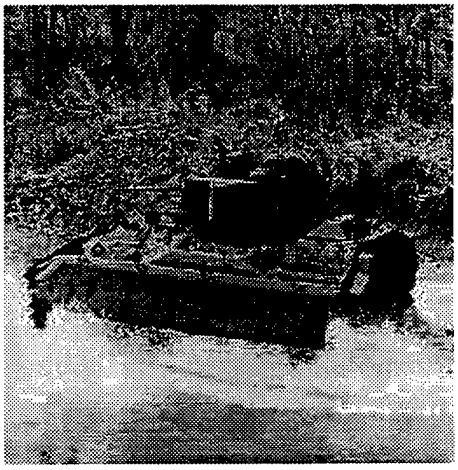


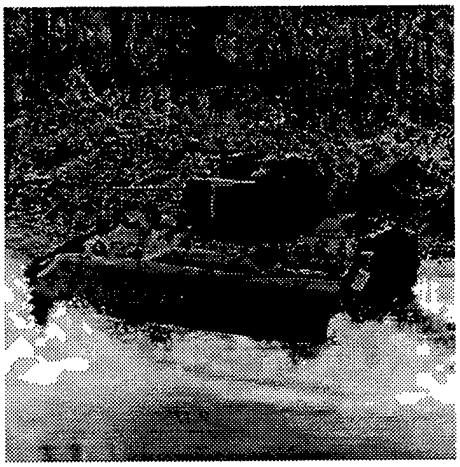
FIGURE A-5.—*Lena image, 1 bpp, noiseless channel*



DPCM



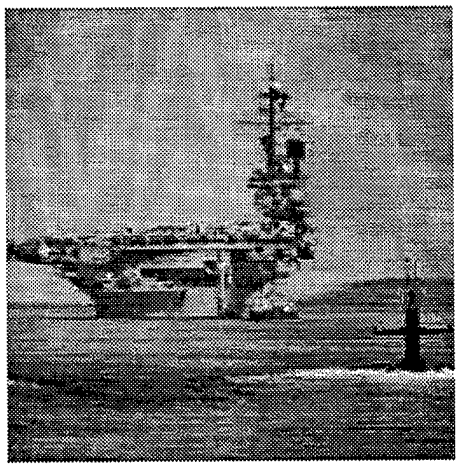
JPEG6



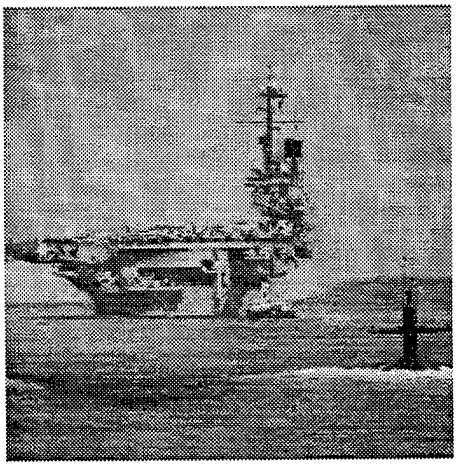
SPIHT

Fractals

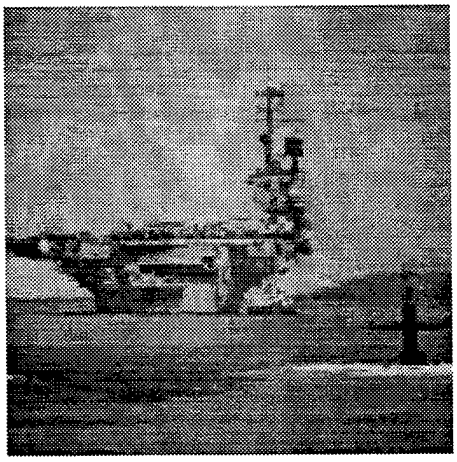
FIGURE A-6.—*BIFV image, 1 bpp, noiseless channel*



JPEG6



SPIHT



Fractals

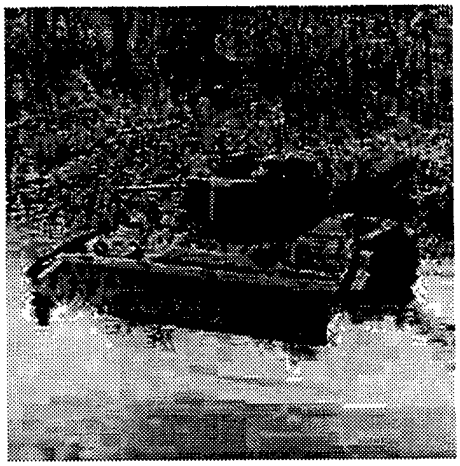
FIGURE A-7.—*America image, 0.5 bpp, noiseless channel*

JPEG6

SPIHT

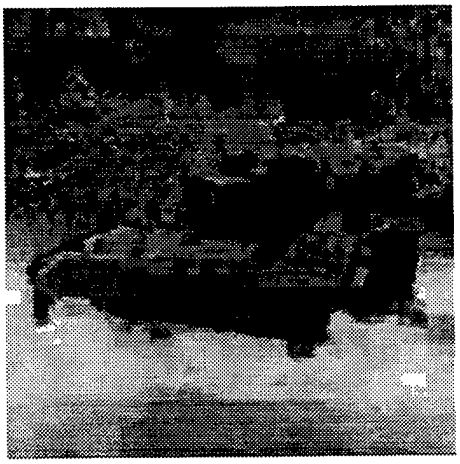
Fractals

FIGURE A-8.—*Lena image, 0.5 bpp, noiseless channel*



JPEG6

SPIHT



Fractals

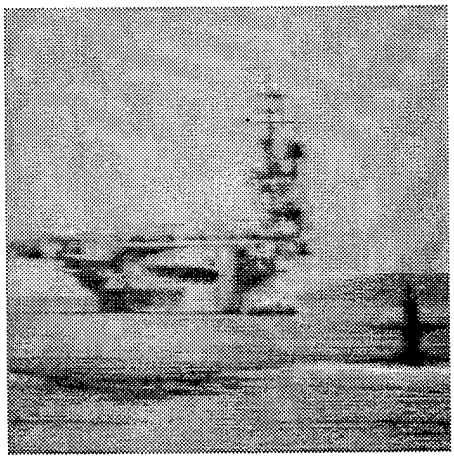
FIGURE A-9.—*BIFV image, 0.5 bpp, noiseless channel*

INTENTIONALLY LEFT BLANK.

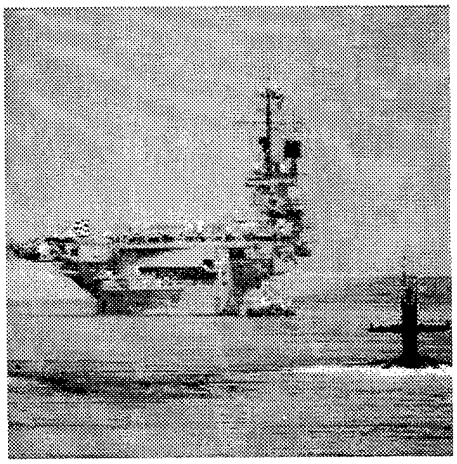
Appendix B:

RECONSTRUCTED IMAGES FOR NOISY CHANNELS

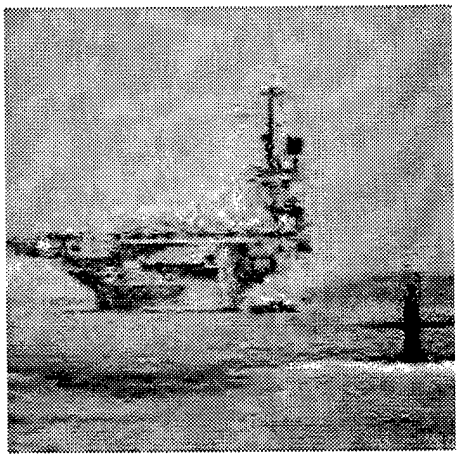
INTENTIONALLY LEFT BLANK.



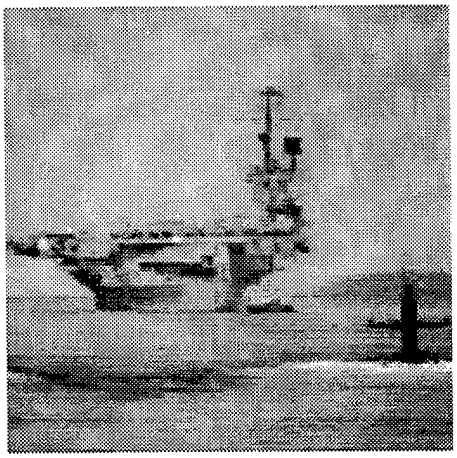
DPCM



JPEG6



SPIHT



Fractals

FIGURE B-1.—*America image, 1 bpp, BER* 10^{-5}

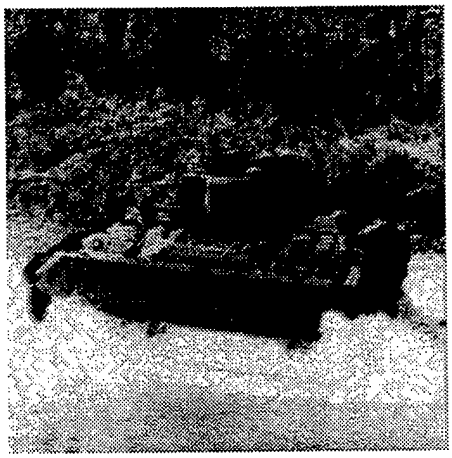
DPCM

JPEG6

SPIHT

Fractals

FIGURE B-2.—*Lena image, 1 bpp, BER* 10^{-5}



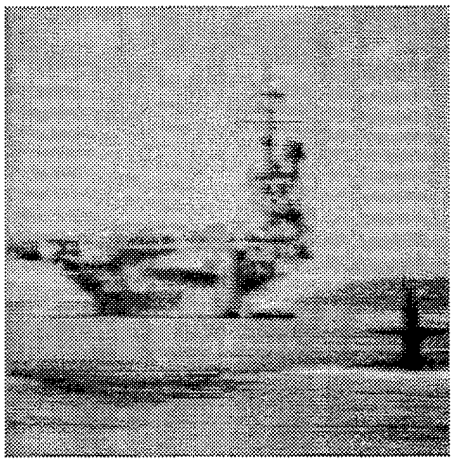
DPCM

JPEG6

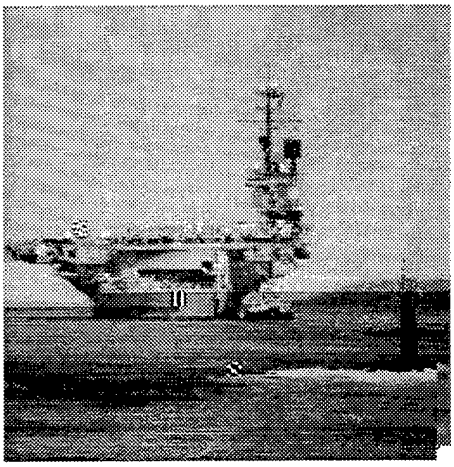
SPIHT

Fractals

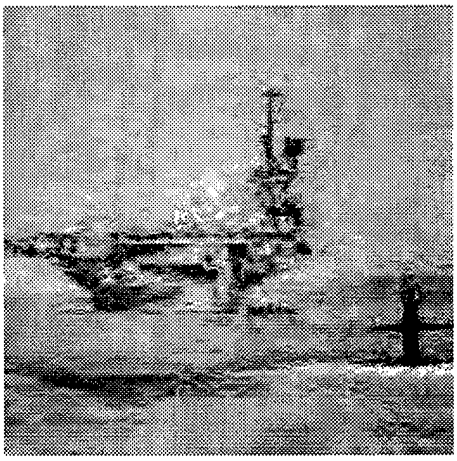
FIGURE B-3.—*BIFV image, 1 bpp, BER* 10^{-5}



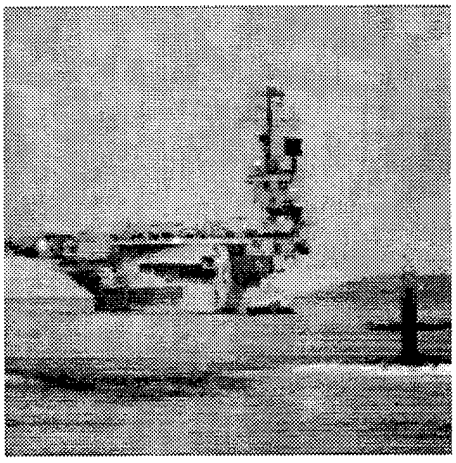
DPCM



JPEG6



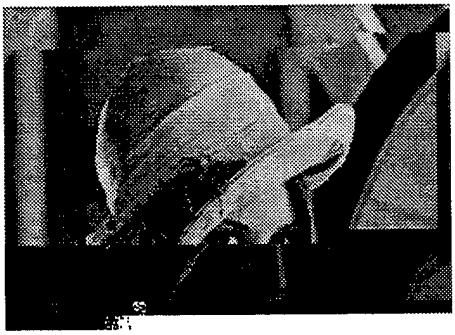
SPIHT



Fractals

FIGURE B-4.—*America image, 1 bpp, BER* 10^{-4}

DPCM

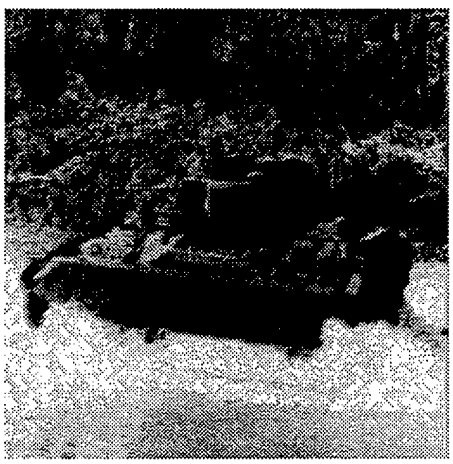


JPEG6

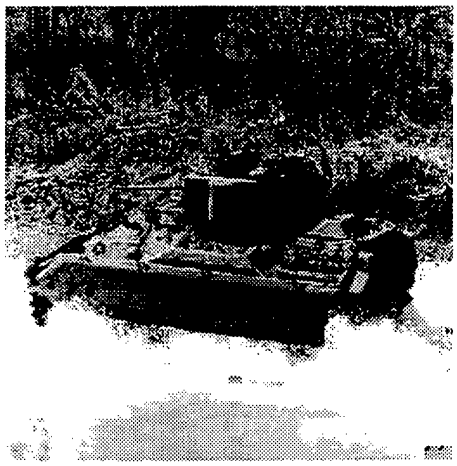
SPIHT

Fractals

FIGURE B-5.—*Lena image, 1 bpp, BER* 10^{-4}



DPCM

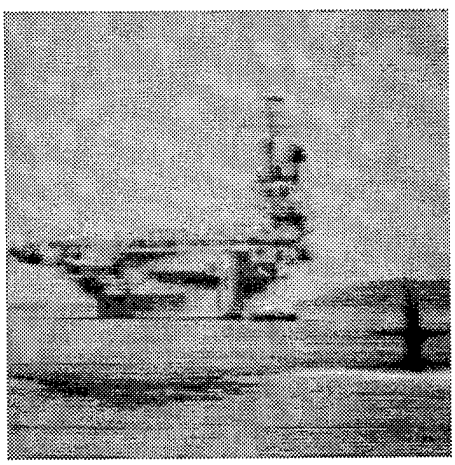


JPEG6

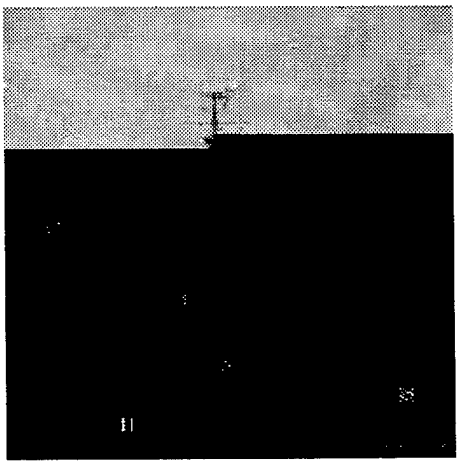
SPIHT

Fractals

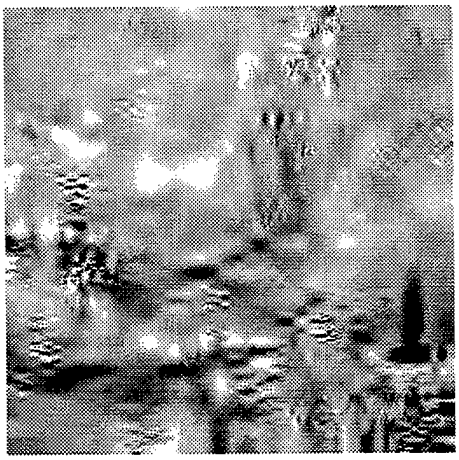
FIGURE B-6.—*BIFV image, 1 bpp, BER* 10^{-4}



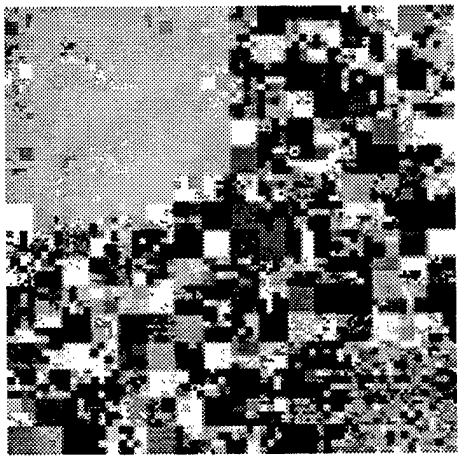
DPCM



JPEG6



SPIHT



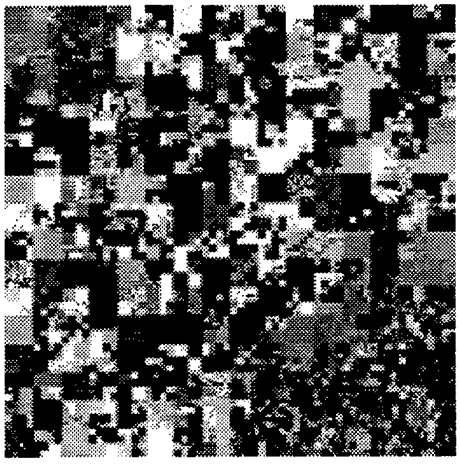
Fractals

FIGURE B-7.—*America image, 1 bpp, BER* 10^{-3}

DPCM

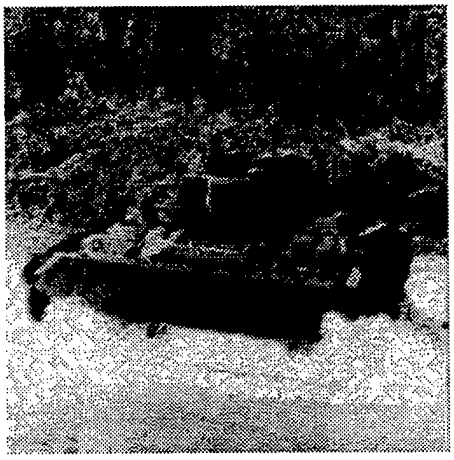
JPEG6

SPIHT

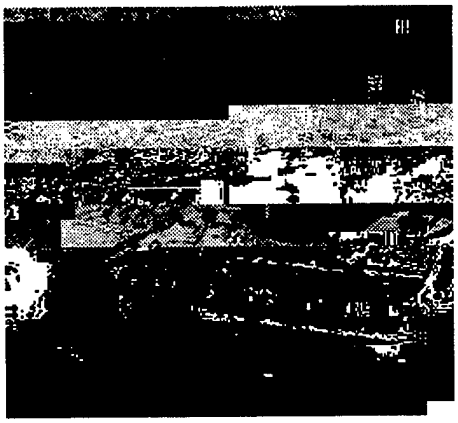


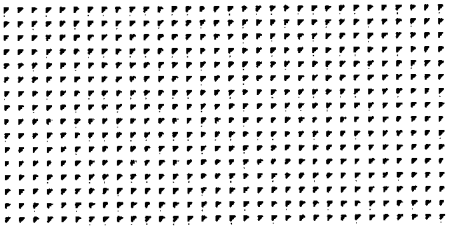
Fractals

FIGURE B-8.—*Lena image, 1 bpp, BER* 10^{-3}



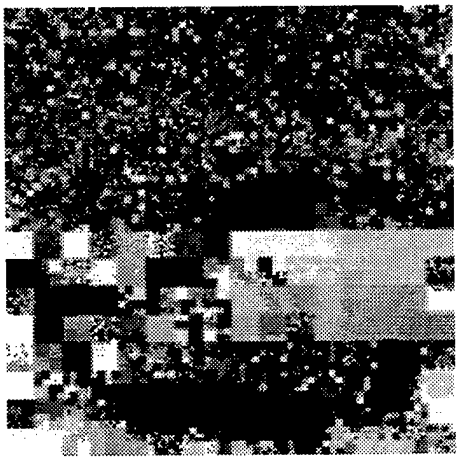
DPCM





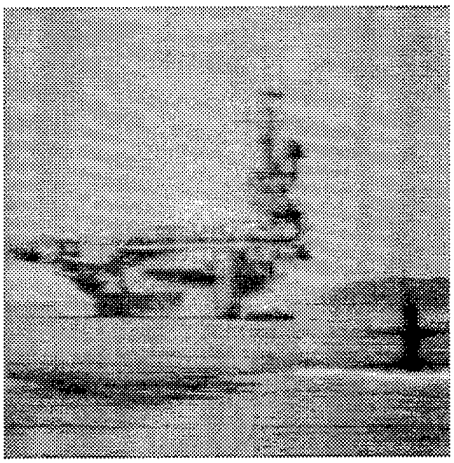
JPEG6

SPIHT

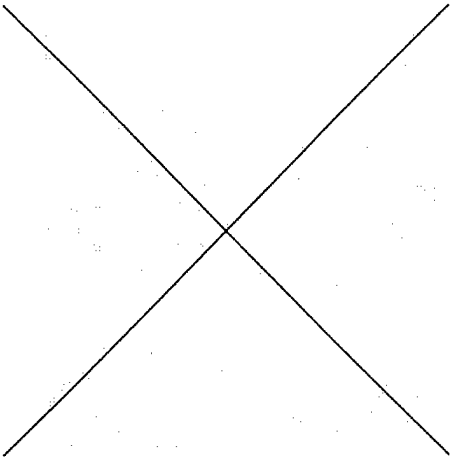


Fractals

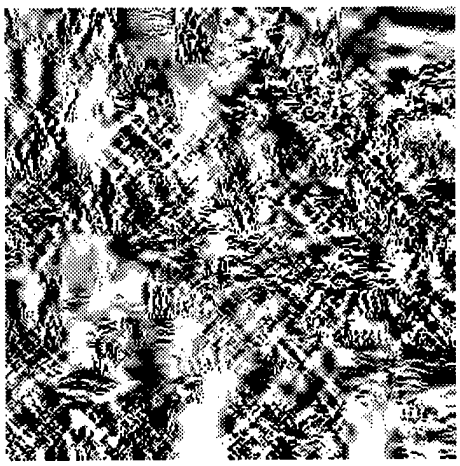
FIGURE B-9.—*BIFV image, 1 bpp, BER* 10^{-3}



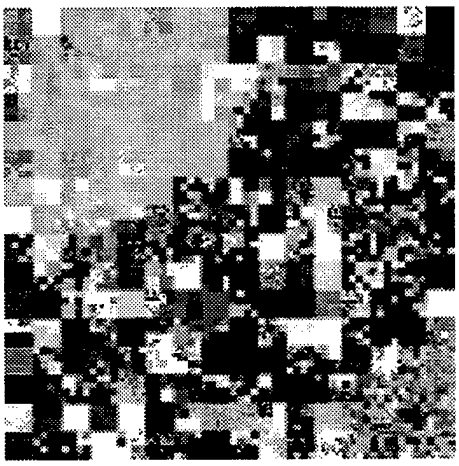
DPCM



JPEG6



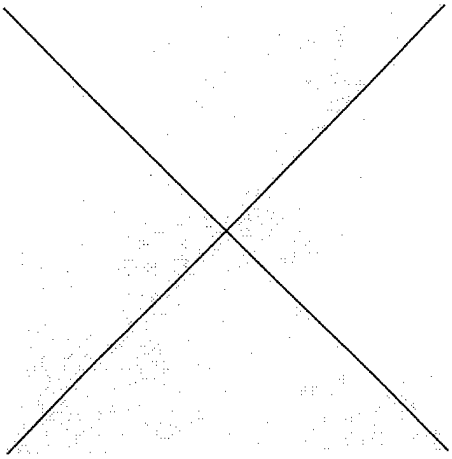
SPIHT



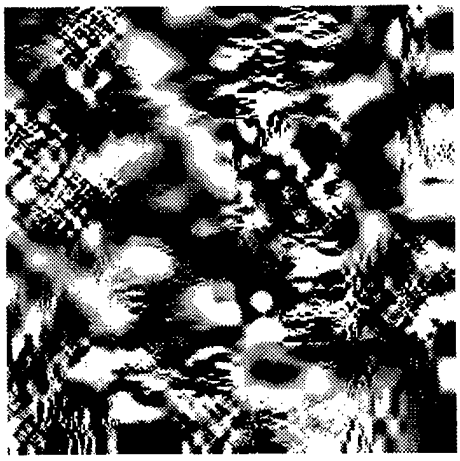
Fractals

FIGURE B-10.—*America image, 1 bpp, BER* 10^{-2}

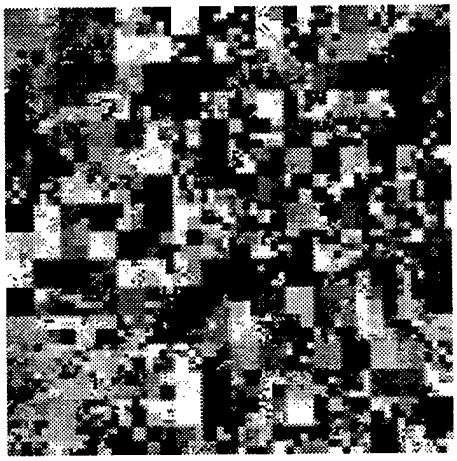
DPCM



JPEG6



SPIHT

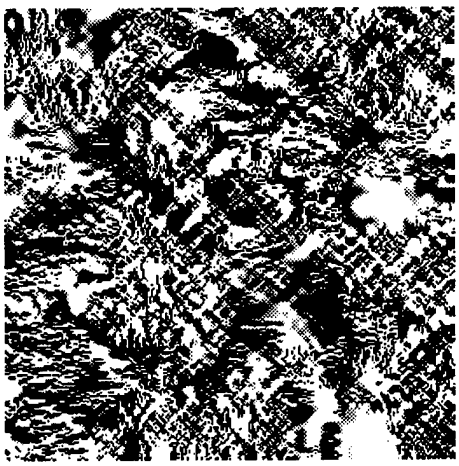


Fractals

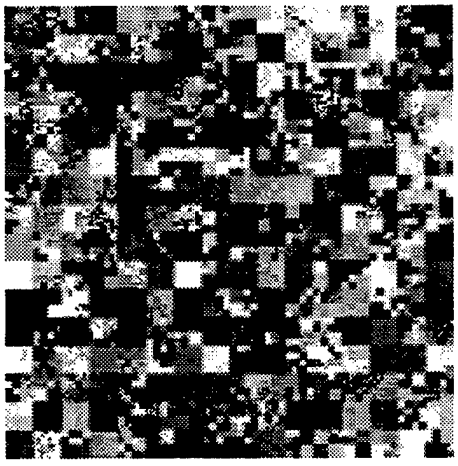
FIGURE B-11.—*Lena image, 1 bpp, BER* 10^{-2}

DPCM

JPEG6

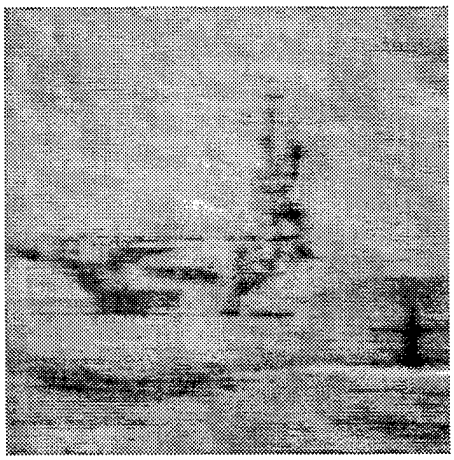


SPIHT

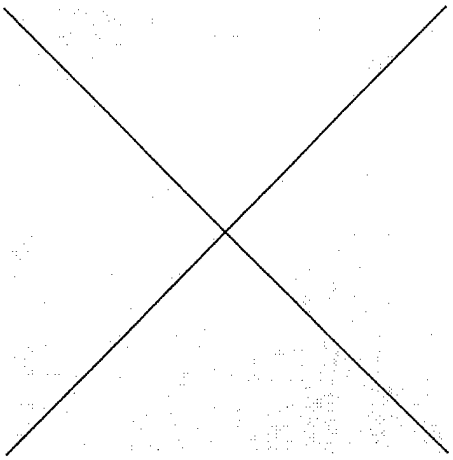


Fractals

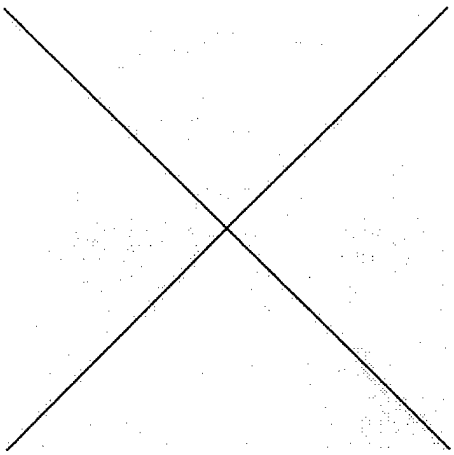
FIGURE B-12.—*BIFV image, 1 bpp, BER* 10^{-2}



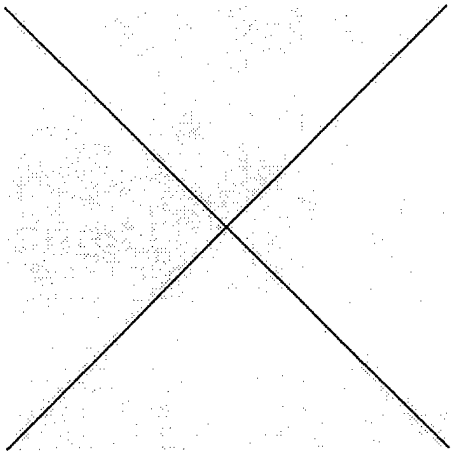
DPCM



JPEG6



SPIHT



Fractals

FIGURE B-13.—*America image, 1 bpp, BER* 10^{-1}

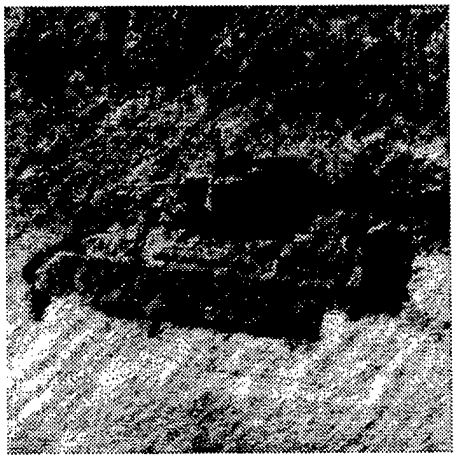
DPCM

JPEG6

SPIHT

Fractals

FIGURE B-14.—*Lena image, 1 bpp, BER* 10^{-1}



DPCM

JPEG6

SPIHT

Fractals

FIGURE B-15.—*BIFV image, 1 bpp, BER* 10^{-1}

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDQ DENNIS SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	CECOM SP & TRRSTRL COMMCTN DIV AMSEL RD ST MC M H SOICHER FT MONMOUTH NJ 07703-5203
1	PRIN DPTY FOR TCHNLGY HQ US ARMY MATCOM AMCDCG T M FISETTE 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	PRIN DPTY FOR ACQUSTN HQS US ARMY MATCOM AMCDCG A D ADAMS 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	DPTY CG FOR RDE HQS US ARMY MATCOM AMCRD BG BEAUCHAMP 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	ASST DPTY CG FOR RDE HQS US ARMY MATCOM AMCRD COL S MANESS 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001

NO. OF COPIES	ORGANIZATION
1	DPTY ASSIST SCY FOR R&T SARD TT F MILTON THE PENTAGON RM 3E479 WASHINGTON DC 20310-0103
1	DPTY ASSIST SCY FOR R&T SARD TT D CHAIT THE PENTAGON WASHINGTON DC 20310-0103
1	DPTY ASSIST SCY FOR R&T SARD TT K KOMINOS THE PENTAGON WASHINGTON DC 20310-0103
1	DPTY ASSIST SCY FOR R&T SARD TT B REISMAN THE PENTAGON WASHINGTON DC 20310-0103
1	DPTY ASSIST SCY FOR R&T SARD TT T KILLION THE PENTAGON WASHINGTON DC 20310-0103
1	OSD OUSD(A&T)/ODDDR&E(R) J LUPO THE PENTAGON WASHINGTON DC 20301-7100
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797
1	DUSD SPACE 1E765 J G MCNEFF 3900 DEFENSE PENTAGON WASHINGTON DC 20301-3900
1	USAASA MOAS AI W PARRON 9325 GUNSTON RD STE N319 FT BELVOIR VA 22060-5582

NO. OF COPIES ORGANIZATION

1 CECOM
PM GPS COL S YOUNG
FT MONMOUTH NJ 07703

1 GPS JOINT PROG OFC DIR
COL J CLAY
2435 VELA WAY STE 1613
LOS ANGELES AFB CA 90245-5500

1 ELECTRONIC SYS DIV DIR
CECOM RDEC
J NIEMELA
FT MONMOUTH NJ 07703

3 DARPA
L STOTTS
J PENNELLA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 SPCL ASST TO WING CMNDR
50SW/CCX
CAPT P H BERNSTEIN
300 O'MALLEY AVE STE 20
FALCON AFB CO 80912-3020

1 USAF SMC/CED
DMA/JPO
M ISON
2435 VELA WAY STE 1613
LOS ANGELES AFB CA 90245-5500

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MDN A MAJ DON ENGEN
THAYER HALL
WEST POINT NY 10996-1786

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TP
2800 POWDER MILL RD
ADELPHI MD 20783-1145

NO. OF COPIES ORGANIZATION

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TA
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

2 DIR USARL
AMSRL CI LP (305)

NO. OF COPIES	ORGANIZATION
1	CDR USA TACOM ATTN AMSTA TR R K ADAMS ADR MS 264 WARREN MI 48397
4	CDR USA TACOM ATTN AMSTA TA D P HANIACK MS 201B WARREN MI 48397
4	CDR USA ATCOM ATTN AMSAC R TD CC R WALL FT EUSTIS VA 23604-1104
2	CDR USA CECOM ATTN AMSEL RD C3 AC P SASS FT MONMOUTH NJ 07703
2	CDR USA CECOM ATTN NVESD DR GARN D II FT BELVOIR VA 22060-5806
2	DIR USARO ATTN AMXRO EL J FREEBERSYSER W SANDER PO BOX 1211 RESEARCH TRIANGLE PARK NC 22709-2211
1	CMDT US MILITARY ACADEMY WEST POINT NY 10996
1	CMDT US NAVAL ACADEMY ANNAPOLIS MD 21404
1	DIR NAVAL RSRCH LAB WASHINGTON DC 20375-5000
1	CDR NWC ATTN 543400D S GATTIS ADMNSTRTN CRCL CHINA LAKE CA 9355-6001
1	CMDT US AIR FORCE ACADEMY COLORADO SPRINGS CO 80840

NO. OF COPIES	ORGANIZATION
1	UNIV OF DE DEPT OF ELCTRCL ENGRNG ATTN C BONCELET JR NEWARK DE 19716
1	THE MITRE CORPORATION ATTN FRANK MCBRIDE 145 WYCKOFF ROAD EATONTOWN NJ 07724
	ABERDEEN PROVING GROUND
1	CDR USATECOM ATTN AMSTE-TA
35	DIR USARL ATTN AMSRL-IS, J. D. GANTT R. SLIFE P. EMMERMAN AMSRL-IS-TP, J. GOWENS A. B. COOPER C. RETTER S. CHAMBERLAIN D. TORRIERI B. SADLER G. CIRINCIONE G. HARTWIG (2 CPS) H. CATON M. LOPEZ F. BRUNDICK M. MARKOWSKI C. SARAFIDIS A. MARK L. MARVEL (4 CPS) AMSRL-IS-CI, B. BROOME T. MILLS (2 CPS) R. KASTE A. BRODEEN D. GWYN L. WRENCHER P. BUDULUS AMSRL-SE-RT, M. HAMILTON AMSRL-SE-SR, A. M. P. MARINELLI L. HAPP AMSRL-WM-WB, W. D'AMICO L. BURKE

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1997	Final, May 1996 - Jan 1997

4. TITLE AND SUBTITLE
A Survey of Image Compression Techniques and Their Performance in Noisy Environments

5. FUNDING NUMBERS
611102.H44

6. AUTHOR(S)
Lisa M. Marvel and George W. Hartwig, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-IS-TP
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1380

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In the effort to digitize the battlefield, one of the most difficult capabilities to provide is the reliable transmission of imagery across the data links available. The most severe obstacles occur at low echelons, where the communication link is provided by combat net radios. The low-bandwidth, noisy nature of this physical channel represents the most serious challenge to implementation of the digital battlefield of the future. These channels make the compression of any imagery a requirement for timely transmission. In this paper, we explore the effects of noise on a variety of image compression techniques - namely, fractal, wavelet, DPCM, and the most recent compression standard for still imagery, JPEG version 6. Methods for minimizing the effects of the noisy channel on algorithm performance are also considered.

14. SUBJECT TERMS	15. NUMBER OF PAGES
image compression, transmission noise	86
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1380 (Marvel) Date of Report June 1997

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

Organization

Name E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
US ARMY RESEARCH LABORATORY
ATTN AMSRL IS TP
ABERDEEN PROVING GROUND MD 21005-5067