

Spatial Compression of Seasat SAR Imagery

C. Y. CHANG, RONALD KWOK, MEMBER, IEEE, AND JOHN C. CURLANDER, MEMBER, IEEE

Abstract—This paper summarizes the results of a study into techniques for spatial compression of SAR imagery. The requirements for an algorithm to perform image compression have been established by the NASA ground processing systems supporting the Shuttle Imaging Radar (SIR-C Processor) and the European Remote Sensor (ERS-1 Alaska SAR Facility). The objective is to reduce the volume of the image data for archive and on-line storage applications while preserving the image resolution and radiometric fidelity. A quantitative analysis of various techniques, including vector quantization (VQ) and adaptive discrete cosine transform (ADCT), is presented. Various factors such as compression ratio, algorithm complexity, and image quality are considered in determining the optimal algorithm. The paper establishes the compression system requirements for electronic access of an on-line archive system based on the results of a survey of the science community. The various algorithms are presented and their results evaluated considering the effects of speckle noise and the wide dynamic range inherent in SAR imagery. The conclusion is that although the ADCT produces the best signal-to-distortion noise ratio (SDR) for a given compression ratio, the two-level tree-searched VQ technique is preferred due to the simplicity of the decoding and the near optimal performance.

I. INTRODUCTION

THE SUCCESS of the Seasat-A SAR and the Shuttle Imaging Radar (SIR-A and SIR-B) missions has stimulated considerable interest in spaceborne synthetic aperture radar (SAR) as a remote-sensing tool [1], [2]. As a result, a number of national space agencies, in addition to NASA, are planning SAR missions in the early 1990's [3]. However, the large volume of data collection planned for future spaceborne SAR missions, such as: the NASA Shuttle Imaging Radar (SIR-C) and Earth Observing System (EOS SAR), the ESA European Remote Sensor (E-ERS-1), the NASDA Japanese Earth Resources Satellite (J-ERS-1), and the Canadian Radarsat poses a severe problem for existing data handling, archiving, and distribution systems. Efficient coding of image data to reduce the data volume would significantly decrease both the transmission and archive costs [4].

The primary requirement on the SAR image compression system is to provide remote users with a large readily accessible data base. The system should provide good reconstructed image quality, short transfer delay, low transfer cost, and minimal decoding complexity. For both the SIR-C and the Alaska SAR Facility (ASF) ground pro-

cessing systems, an on-line image archive for browse and data distribution is planned. These systems will respond to user requests for electronic transfer of low-resolution (8 by 8 pixel averaged) browse image files to the remote scientists through NASA data networks (e.g., the Space Physics Analysis Network (SPAN)).

The application of data reduction for the primary (off-line) archive will not be specifically addressed in this paper except to state that although the image degradation specification will be much more strict, the inherent speckly noise in the imagery may allow a fairly high compression. The issue of loss of the phase information in the reconstructed imagery has not yet been addressed and will be the subject of a further study. The primary emphasis of this study is for compression of on-line imagery for browse of survey applications.

As an example of an operational SAR ground data system, Fig. 1 shows the functional block diagram of the Alaska SAR Facility (ASF). The ASF SAR processor system is custom hardware designed to process the E-ERS-1, J-ERS-1, and Radarsat data. The daily production of the ASF SAR processor will be on the order of 150 images/day over a period of three to five years. The standard image product of the ASF processor is a high-resolution (4-look) image of approximately 8192 by 8192 pixels (64 Mbytes). In order to reduce the data volume to be stored in the on-line disk storage, a reduced resolution product is generated by spatially averaging (8 by 8 pixels) the 4-look image data. This image product is known as low-resolution browse imagery (which is approximately 75 looks). It consists of a 1024 by 1024 pixel image frame (1 Mbyte/frame), which covers approximately 100 km by 100 km at a pixel spacing of 100 m. Even though the data volume is reduced by a factor of 64 times, it is estimated that the data volume of the browse images will be about 55 Gbytes per year, which results in extremely high on-line storage and data handling costs.

The key function of an on-line data system is for the users to have the capability of browsing the imagery routinely produced. Upon user's requests, images would be electronically transferred to members of the remote science working team. At a typical data rate of 9.6 Kbps (e.g., SPAN), a 1-Mbyte browse image would require 15 min for transfer without data compression. Since there is a single archive system shared by many remote users, the actual response time could be considerably longer depending on the loading of the archive system and data network (see Table I).

Manuscript received November 25, 1987; revised April 28, 1988. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

The authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109.

IEEE Log Number 8822298.

0196-2892/88/0900-0673\$01.00 © 1988 IEEE

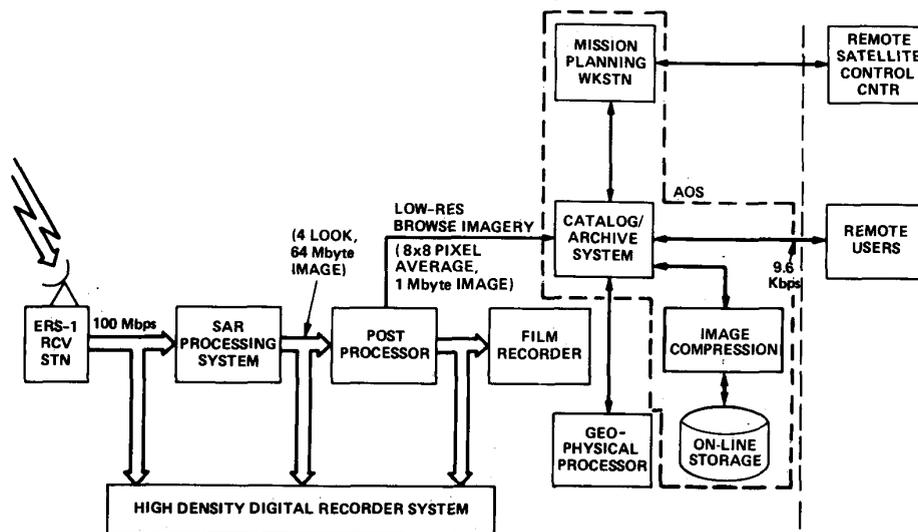


Fig. 1. The functional block diagram of the Alaska SAR Facility (ASF) Ground Processing System.

TABLE I
THE AVERAGE RESPONSE TIME FOR A: UNCOMPRESSED DATA; B:
COMPRESSED DATA WITH ENCODING TIME, $T_e = 1$ min; C: COMPRESSED
DATA WITHOUT ENCODING TIME, $T_e = 0$ min (COMPRESSED OFF-LINE); D:
UNCOMPRESSED DATA WITH FIVE SERIAL PORTS AS A FUNCTION OF ACCESS
FREQUENCY λ (IMAGES/HOUR) AND COMPRESSION RATIO R

T	R	1		5		10		15		20		30		D
		A	B	C	B	C	B	C	B	C	B	C		
1		17.85	4.40	3.49	2.94	1.97	2.46	1.48	2.22	1.23	2.02	.99	14.64	
2		22.72	4.55	3.56	2.99	1.99	2.49	1.49	2.25	1.24	2.04	.99	14.72	
3		37.07	4.72	3.66	3.05	2.01	2.53	1.50	2.27	1.24	2.06	.99	14.74	
4		1012.94	4.91	3.76	3.12	2.03	2.57	1.50	2.30	1.25	2.08	.99	14.91	
6		∞	5.42	4.01	3.26	2.08	2.66	1.52	2.37	1.26	2.13	.99	15.16	
10		∞	8.08	4.78	3.81	2.19	2.95	1.56	2.57	1.28	2.23	1.01	15.94	
12		∞	10.12	5.44	3.98	2.26	3.03	1.59	2.63	1.29	2.32	1.01	16.60	
15		∞	92.90	7.31	4.91	2.37	3.43	1.63	2.89	1.31	2.42	1.02	18.64	
20		∞	∞	52.00	7.56	2.64	4.15	1.70	3.29	1.34	2.74	1.03	63.15	
30		∞	∞	∞	∞	3.91	27.49	1.93	6.89	1.44	4.22	1.06	∞	
60		∞	∞	∞	∞	∞	∞	17.67	∞	2.20	∞	1.21	∞	

The SIR-C processor throughput is similar to the ASF (approximately 50 images/day for 2.5 years) and will require an on-line archive. The most obvious solution to save on-line disk storage and minimize the transfer delay time is data compression to reduce the data volume without further loss in spatial resolution.

A user survey was conducted to assist in specifying the system requirements of SAR image data compression. This survey, along with sample image products, was distributed to members of the science community. The questionnaire requested both quantitative and qualitative evaluations regarding the trade-offs among image quality, compression ratio, estimated transfer delay and transfer cost. Users' response to the survey indicated that a transfer delay of less than 5 min/image would be satisfactory. It was estimated that for 50 investigators, the peak access frequency to the on-line archive will be on the order of 10 to 15 images/h. The high access frequency and required quick response time over a 9.6-Kbps data line necessitates a compression ratio of at least 10:1. Among the sample image products distributed, the reconstructed im-

age quality with compression ratio between 10:1 and 20:1 was verified as satisfactory for the browse application.

Various image data compression algorithms have been evaluated using standard Seasat [5] 4-look SAR images (for the primary archive) and 8 by 8 pixel averaged browse images. These images were processed by the NASA/JPL digital SAR processor system [6]. The data compression algorithms evaluated include the predictive coding [7], adaptive transform coding [8], [9], vector quantization [10], [11], and various other *ad hoc* techniques [12]–[14]. Among these algorithms, the transform coding and the vector quantization techniques are the two algorithms that best provide good image quality at high compression ratios. These two algorithms have been compared based on several factors, including the compression ratio, reconstructed image quality, performance controllability, the compression ratio flexibility, and computational complexity. Since small decoding complexity at the users' site is critical for wide distribution of the on-line archive data, the tree-searched vector quantization, which requires minimal decoding complexity, appears to outweigh the high compression ratios achievable with the adaptive transform coding. Thus, the tree-searched vector quantization is recommended as the image data compression algorithm for the on-line data archive system.

Section II will present a quantitative analysis regarding the compression ratio, algorithm complexity, transfer delay, and access frequency. The requirements of the on-line data archive system are also specified. Section III describes the effect of the SAR unique image data characteristics, namely the speckle noise and the large dynamic range, on data compression results. Section IV discusses the adaptive transform coding and vector quantization algorithms. Section V presents the tests results and comparison study and Section VI makes final recommendations.

II. REQUIREMENTS AND LIMITATIONS OF THE ON-LINE DATA ARCHIVE SYSTEM

This section will evaluate the quantitative relationship among compression ratio, algorithm complexity (based on the encoding and decoding time), transfer delay, and access frequency by modeling the data distribution system as a queueing system [18]. The analysis results will be utilized to specify the requirements on data compression for the on-line data archive system.

A. Queueing Analysis of the On-Line Archive System

For simplicity of analysis, we assume that access to the archive system can be modeled by a Poisson process. It is also assumed that each access requires transfer of one image data file and that only one serial port is shared by all remote users. The archive system can then be modeled as a $M/D/1$ queueing system [18], where M assumes the access is Poisson distributed, D assumes a fixed amount of time to encode and transmit a given size image, and 1 assumes a single resource (encoder and serial port) shared by all the users.

The following notations are used: λ : the mean value of the Poisson process in number of images per unit time; $L_1 \times L_2$: the uncompressed image size in pixels; K : number of bits to represent the radiometric resolution of each uncompressed pixel; R : the compression ratio; W : the average waiting time to access the archive system; T_e : the time to encode the image; T_t : the time to transmit the compressed image through the data line; C : the data rate in bits/second; T_d : the time to decode the image; T : the average total response time. Both T_e and T_d depend upon the image size and the algorithm complexity. It can be shown that T , the average response time in the long run, can be expressed as [18]

$$T = W + T_e + T_t + T_d \quad (1)$$

where the average waiting time W to get hold of the resource can be shown to be

$$W = \frac{\lambda(T_e + T_t)^2}{2(1 - \lambda(T_e + T_t))}. \quad (2)$$

$T_e + T_t$ is known as the service time of the queueing system. Let T_c be the time to transmit the uncompressed data through the data line, then

$$T_c = \frac{KL_1L_2}{C}. \quad (3)$$

Hence

$$T_t = \frac{KL_1L_2}{RC}. \quad (4)$$

Also let

$$\rho = \lambda(T_e + T_t) \quad (5)$$

where ρ is known as the utilization factor of the queueing

system. Equation (1) can then be rewritten as

$$T = (T_e + T_t) \left[1 + \frac{\rho}{2(1 - \rho)} \right] + T_d. \quad (6)$$

Note that if all the image data files are compressed as part of the image formation process, there is no encoding time required following each data request; that is, $T_e = 0$. This mode of operation is important since it presents a means of significantly reducing the amount of storage needed for the browse images, as well as lowering the computer loading of the archive system and the overall response time of delivering an image to the remote user. Since all encoding is performed at the central site, it is a reasonable expense to implement the algorithm in hardware.

The above queueing model assumes that there is only one serial port shared by all the remote users. In order for the data compression to be practically useful, it must outperform the multi-serial-port system, where many serial ports are shared by the remote users to transfer the uncompressed image data files. From the queueing theory it can be shown that the average response time of this multi-serial-port system has a lower bound of

$$T \geq \frac{T_c}{R} \left[R + \frac{\lambda \frac{T_c}{R}}{2 \left(1 - \lambda \frac{T_c}{R} \right)} \right] \quad (7)$$

where R is now an integer representing the number of serial ports. Consider an image data compression capability of ratio R . Also assume that the image data files are compressed off-line and stored in the data archive so that no encoding time would be required for each data transfer. We have

$$T = \frac{T_c}{R} \left[1 + \frac{\lambda \frac{T_c}{R}}{2 \left(1 - \lambda \frac{T_c}{R} \right)} \right] + T_d. \quad (8)$$

From (7) and (8), we can see that the image data compression will outperform the multi-serial-port system if: 1) The image data is compressed off-line; 2) the decoding time is much shorter than the transmission time of uncompressed data; 3) the overhead data is small in volume relative to the uncompressed image data; and 4) the decoding complexity at the remote sites is within the practical constraints of most computing systems.

An additional benefit of the image data compression not offered by a multi-serial-port system is that the image data compression can provide a large inexpensive on-line data base by reducing the data volume. For instance, without data compression the archive system would annually require 55 Gbytes of disk storage for the ASF ground processing system, which is a significant cost to maintain an on-line archive system. With a compression ratio of 15:1,

less than 4 Gbytes of storage is required to archive the data produced in a year. The transfer cost, which is a function of the transmit time T_t , is also reduced by the same amount due to the reduced data volume.

As an example of the performance of image data compression, consider the following: Given an image size of 1024×1024 pixels (1 Mbyte) and a data rate of 9.6 Kbps. Table I shows the average response time T , in minutes, as a function of the average access frequency λ , in images/h, and the compression ratio R . The lower bound of the average response time of the multi-serial-port system is also shown for $R = 5$. The cases with encoding time and without encoding time are also shown. It is assumed that $T_e = 1$ min and $T_d = 0.5$ min. Notice that if the image is not compressed ($R = 1$) and the average access is one image per 15 min ($\lambda = 4$), which is considered to be lower than a typical access frequency, the average response time will eventually exceed 1000 min! This is because the access frequency is very close to the limit that can be handled by the uncompressed data distribution system. Alternatively, with a compression ratio of 15:1, the average response time reduces to only 2.57 min including the encoding time and 1.50 min not including the encoding time. We can also see that the multi-serial-port system can only prevent the response time from quickly growing out of bound. It cannot reduce the transfer delay of each image below 14.56 min, which is the time required to transmit the 1-Mbyte image through the 9.6-Kbps data line. This example clearly demonstrates the benefits of data compression for a successful on-line archive and data distribution system.

B. User Survey Results

A survey of the SAR user community selected primarily from the SIR-B science team was conducted to assist in specifying the requirements of SAR image data compression for the on-line data archive system. The user survey included sample compressed and reconstructed image products. The questionnaire requested quantitative and qualitative evaluation of these images regarding the trade-offs among image quality, compression ratio, estimated transfer delay and transfer cost.

Based on the user response, the peak access frequency of all the users is estimated to be between 10 and 15 images/h. The users' requirements for the transfer delay were quite varied. Approximately half considered the delay a noncritical issue. However, the other half considered a small transfer delay necessary for their research studies. To compromise these different requirements, a transfer delay of less than 5 min/image is expected to meet most users' requirements. The users stated that the transfer cost should be less than 1 dollar/image. Most of the users also recommended that the compressed images be stored in the archive to reduce the amount of disk storage and computer time. Most of the users see no application of imagery with compression ratios higher than 50:1 (see Figs. 9 and 10 later).

Among the sample image products, the image quality

of compression ratio 10:1 has been deemed satisfactory by all users and considered to be acceptable for quick-look applications. Some of the users stated that the image quality degradation between compression ratios 10:1 and 30:1 appears acceptable. Based upon the users' requirements on access frequency, transfer delay, and image quality, we conclude the compression ratio should be at least 10:1.

C. Summary of Requirements on Data Compression for the On-Line Data Archive System

The requirements on the data compression for on-line data archive and distribution system are compiled as follows: 1) The daily production of the ground processing system can be as high as 150 browse images/day; 2) each browse image contains 1024 by 1024 pixels (100 km by 100 km ground coverage); 3) the data archive will be shared by over 25 to 50 remote scientists; 4) the data rate of the data network is 9.6 Kbps; 5) the encoding computational capability will require a hardware implementation or a multiple vector processor approach; 6) the decoding computational capability is assumed at least a small workstation; 7) the peak access frequency to the data archive is between 10 and 15 images/h; 8) the total image transfer delay time is limited to 5 min/image; 9) the image quality should meet the users' requirement for browsing; and 10) the compression ratio should be at least 10:1.

III. EFFECTS OF SAR IMAGE DATA CHARACTERISTICS ON DATA COMPRESSION

In order to attain good data compression results for SAR imagery, the SAR image data characteristics must be considered [15]. In this section, effects of two SAR image data characteristics, the speckle noise [16], [17], and the wide dynamic range [1], [2], [5], on data compression are addressed.

A. Observation of Speckle Noise Effects on Data Compression

The speckle noise contamination of SAR imagery is an important data characteristic that needs to be considered in selecting a SAR image data compression algorithm. Speckle is a phenomenon in which the radar returns from the individual scatterers in a ground resolution cell combine coherently to give the resultant observed return. Speckle is a multiplicative noise in which the amplitude of the echo return is exponentially distributed and the phase is uniformly distributed. The existence of speckle makes it difficult to calibrate the radiometric characteristics of an individual SAR resolution element in terms of backscatter ratio. Speckle is reduced by a technique known as multiple-look averaging [17] where the data is processed in azimuth over several independent segments of the Doppler bandwidth and the resultant images are incoherently summed to give the final SAR image. Although the radiometric resolution is improved, the spatial resolution is degraded.

As a result of the speckle noise, there is lack of inter-

pixel correlation among the adjacent resolution cells in the SAR images as compared to the optical images. Hence, the image data compression techniques that are primarily based on inter-pixel correlation (i.e., the predictive coding) do not perform as effectively for the SAR images, especially for the images having small look number.

An example of the inter-pixel correlation function for the 4-look image and 8 by 8 pixel average browse image is shown in Fig. 2. The inter-pixel correlation decreases quickly for the 4-look image and slowly for the browse image. For instance, the correlation coefficients of the browse image at 8 and 16 pixel shift are 0.446 and 0.282, respectively. The coefficients of the 4-look image reduce to 0.180 and 0.079, respectively. This shows that in terms of signal to distortion ratio, the image data compression techniques should perform better for the low-resolution images (large look number, low speckle noise) than the high-resolution images (small look number, high speckle noise) since the low-resolution images contain more inter-pixel correlation due to its smaller speckle noise.

However, subjectively the degradation in the visual quality for the high-resolution imagery is less noticeable than for the low-resolution images due to the speckle noise. This can be explained as follows: let n_s represent the speckle noise power and n_d the distortion noise power induced by the data compression. The degradation in the image quality in terms of decibels is

$$\text{degradation} = 10 \log \frac{n_s + n_d}{n_s} \text{ dB}. \quad (9)$$

It is generally true that as the number of looks increases, the speckle noise n_s reduces by the square root of the number of looks while the distortion noise n_d reduces much more slowly. The result is that the degradation in visual quality at the same compression ratio is subjectively less noticeable for high-resolution images.

Spatial averaging of pixels is often used as a technique to reduce the speckle noise in addition to the data volume. The drawback of this approach is significant degradation in spatial resolution. For the data compression to be practically useful, it must have much smaller degradation in spatial resolution as compared to the spatial average method. The averaging resulting from the data compression/reconstruction procedure outlined in the following sections will produce a better speckle reduction per loss in resolution factor than the spatial average method with far greater reduction in data volume.

B. Effects of Large Dynamic Range on Data Compression

The dynamic range of SAR images is much greater than that of optical images. This results from the fact that a pixel in a Seasat SAR image is the result of pulse compression of approximately 1024 samples by 4096 samples, where each input sample is represented by 5 bits (32 gray levels). For an ideal point target, the processing can produce a gain of as much as 132 dB. Although most

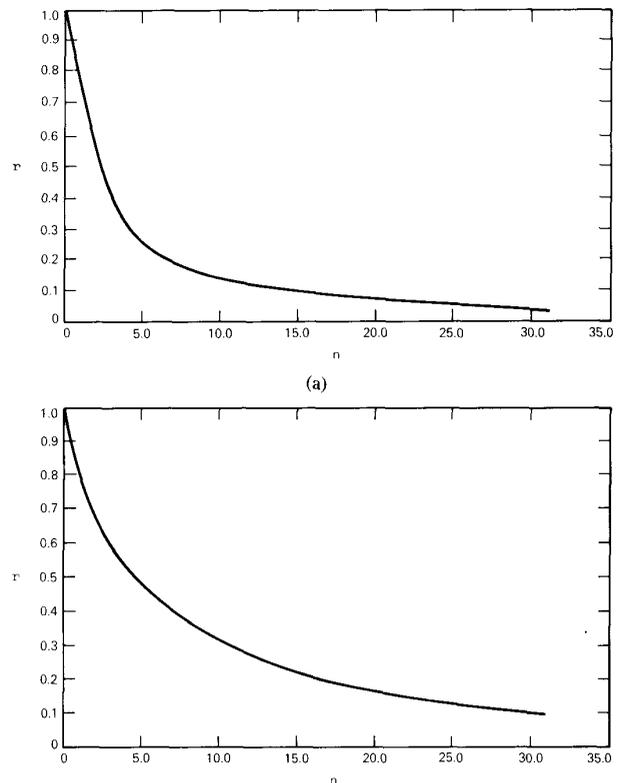


Fig. 2. The interpixel correlation r as a function of pixel shift n . (a): Example of a 4-look SAR image. (b): Example of a 8 by 8 spatial average browse image.

of these point-like scatterers characteristically exhibit a much smaller compression gain, the increase in the dynamic range places special requirements on the data compression algorithm.

The large dynamic range requires that the algorithm should be adaptive to the local image statistics. The overhead information of each algorithm used to characterize the source data statistics should be updated according to different local image statistics. This phenomenon is clearly demonstrated during the evaluation of adaptive transform coding and vector quantization techniques. For the adaptive transform coding, a blurry or blocky effect will occur at relatively small compression ratios if the block adaptivity is not properly chosen. Similarly, for the vector quantization, the edge degradation is very severe if no adaptation is allowed for different scene characteristics.

IV. IMAGE DATA COMPRESSION ALGORITHMS

In general, the image data compression algorithms can be classified into several categories, including predictive coding, transform coding, vector quantization, and other *ad hoc* techniques [4]. Several of these techniques were implemented and tested using Seasat 4-look and 8 by 8 pixel average browse images [19]. This section presents two of the image data compression algorithms: adaptive transform coding and vector quantization. In the next sec-

tion, test results and comparison study will be discussed. The test results of the other algorithms were significantly inferior to the ADCT and VQ. These results are summarized in [19].

A. Adaptive Transform Coding Algorithm

The adaptive transform coding [8], [9] is a technique that can compress the image data to any user specified volume given that the associated image quality degradation is tolerable. It generally yields better image quality than the predictive coding at the same compression ratio. Its major disadvantage is that it is computationally intensive in both encoding and decoding, requiring a large number of two-dimensional transforms.

The adaptive transform coding technique partitions the image data into small blocks (16 pixels by 16 pixels per block is typical). The image blocks are then transformed block by block by employing two-dimensional energy packing transformation, such as Fourier transform, discrete cosine transform, Hadamard transform, etc. Following the transformation, each transform block is classified into one of several classes (four classes are typical) on the basis of a block activity parameter (e.g., the ac energy of each block). The activity of each transform coefficient within a class is computed to form a bit allocation map for the class, such that more bits are assigned to those areas of high activity and less bits to those of low activity. Based upon this information (i.e., class map and bit allocation maps), the transform coefficients are then normalized, quantized, and coded. Enhanced performance is achieved by optimizing block size, number of classes, type of quantizer, bit assignment procedure, and the parameter used to characterize the block activity. The functional block diagram of the adaptive transform coding technique is shown in Fig. 3, where the discrete cosine transform is employed. The details of the algorithm are shown in [9]. This algorithm is commonly known as the adaptive discrete cosine transform (ADCT). A different compression ratio is easily achieved by adjusting a distortion factor in the algorithm. Ideally any positive real number can be achieved.

Let the image size be L_1 by L_2 pixels and the block size be s_1 by s_2 pixels. Hence, there are $n_1 (= L_1/s_1)$ by $n_2 (= L_2/s_2)$ image blocks. Let the original pixel intensity be represented by K bits. Assume that after the transformation, the image blocks are divided into N classes. Let $l_{jk}^{[i]}$ denote the jk^{th} entry of the i^{th} bit allocation map. Then the compression ratio is

$$R = K \left\{ \frac{\sum l_{jk}^{[i]}}{Ns_1s_2} + \left[\frac{n_1n_2 \log N}{L_1L_2} + \frac{Ns_1s_2 \log K}{L_1L_2} \right] \right\}. \quad (10)$$

The first term in the denominator represents the average number of bits used to code the transform coefficients.

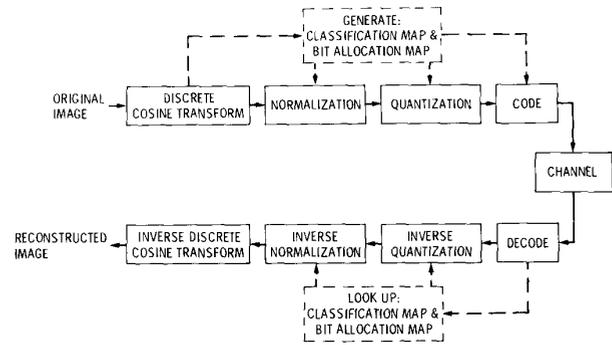


Fig. 3. The functional block diagram of the adaptive discrete cosine transform (ADCT) algorithm.

This can be obtained by an iterative operation. The second term in the denominator represents the overhead information. The first term in the bracket represents the data volume of the class map normalized by the total number of pixels, while the second term represents the data volume of N bit allocation maps normalized by the total number of pixels. For a 1024 by 1024 pixel image frame, the overhead information is minimal (one to two percent) compared to the total image data.

Let $s_1 = s_2 = s$. The encoding complexity per pixel of the ADCT algorithm requires $2s$ multiplications and additions for the two-dimensional discrete cosine transform and $(n_1n_2)/(2s^2)$ operations for the sorting (linear sort) of transform blocks to obtain the class map. Notice that the sorting complexity grows as the image size increases and the block size decreases. Table II shows the number of operations per pixel for two-dimensional transform and sorting as a function of image size (L by L pixels) and block size (s by s pixels). The decoding complexity is high since it requires a two-dimensional inverse cosine transform (no sorting is required).

From Table II, for both 512 by 512 and 1024 by 1024 pixels image, the block size requiring the fewest operations per pixel is 16 by 16 pixels. This is also a commonly used block size to reduce the inter-pixel correlation. Notice that the sorting complexity is extremely high when the block size is as small as 4 by 4 pixels. To reduce the sorting complexity, a more efficient sorting algorithm such as the quick sort algorithm could be employed. Another approach would be to use a simplified procedure to classify the image blocks. To reduce the two-dimensional transformation complexity, a fast two-dimensional transformation algorithm could also be employed.

B. Vector Quantization Algorithm

Vector quantization (VQ) [10], [11], [20], [21] is a data compression technique that provides high compression ratio with good reconstructed image quality. In addition, the decoding procedure for the vector quantization requires only a table look-up, which is very efficient for the single-encoder multiple-decoder data system. Its major disadvantage is that the encoding complexity is higher than most other image data compression techniques. In

TABLE II
THE ENCODING COMPLEXITY OF ADCT ALGORITHM FOR A: TWO-DIMENSIONAL DISCRETE COSINE TRANSFORM AND B: LINEAR SORTING, IN NUMBER OF OPERATIONS/PIXEL. THE IMAGE SIZE IS L BY L PIXELS AND THE BLOCK SIZE IS s BY s PIXELS

$s \backslash L$	256		512		1024	
	A	B	A	B	A	B
4	8	128	8	512	8	2048
8	16	8	16	32	16	128
16	32	1/2	32	2	32	8
32	64	1/32	64	1/8	64	1/2

addition, its edge degradation can be more severe than other techniques, such as transform coding, if there are large radiometric variations.

Vector quantization is a generalization of scalar quantization. In vector quantization, the input data is divided into many small blocks (vectors). The transition levels (codewords) are vectors of the same dimension as the input data vectors. Each input data vector is compared with each codeword and the codeword of the smallest distortion is chosen as the quantization vector to represent the input data vector. The indices of the codewords corresponding to the data vectors are transmitted through the data channel. The codebook is transmitted with the coded data. At the receiving end, the image is reconstructed according to these received indices by lookup using the codeword table (codebook). The functional block diagram of the vector quantization is shown in Fig. 4. The data compression is achieved since fewer bits are used to represent the codeword indices than the input data. The codebook is generated by training a subset of the source data. The performance of the codebook depends highly on the similarity between the training data and the coded data. Ideally, the encoding procedure involves computing the distortion between each input data vector and the codewords. Although this is a straightforward but computationally intensive process, the decoding procedure is as simple as a table look-up.

Let the codebook consist of $2^n (= M)$ codewords, known as an n -bit codebook, where each codeword contains a block of s_1 by s_2 pixels. The original image size is L_1 by L_2 pixels and each pixel contains K bits to represent its intensity. Assuming that each scene has its own codebook, the compression ratio is

$$R = K \left/ \left\{ \frac{n}{s_1 s_2} + \frac{2^n K s_1 s_2}{L_1 L_2} \right\} \right. \quad (11)$$

The second term of the denominator represents the amount of overhead information (codebook). Table III shows an example of the achievable compression ratios for $s_1 = s_2 = 4$ and $K = 8$. We can see that the vector quantization also has flexibility in the achievable compression ratio. However, to switch from one compression ratio to another requires the regeneration of the codebook, which is computationally intensive.

There are three problems pertaining to the vector quantizer that need to be resolved: 1) large encoding complexity, 2) scene dependence, 3) large edge degradation.

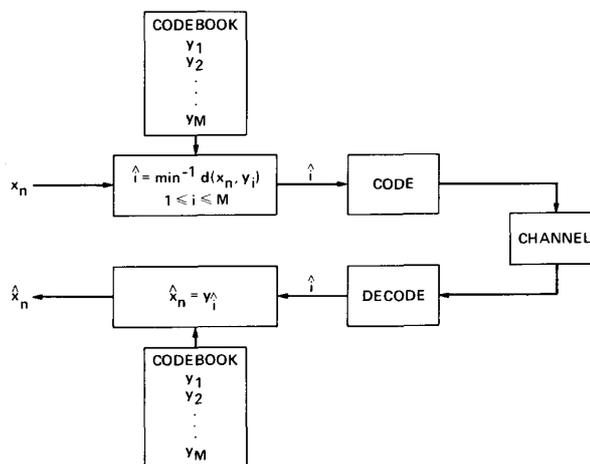


Fig. 4. The functional block diagram of the vector quantization (VQ) algorithm. $\min^{-1} d(x_n, y_i)$ means to select the i that yields the minimum distortion between x_n and y_i .

TABLE III
THE COMPRESSION RATIOS OF THE VECTOR QUANTIZER FOR 4 BY 4 PIXELS/BLOCK. THE IMAGE SIZE IS L BY L PIXELS, THE CODEBOOK SIZE IS 2^n CODEWORDS

$L \backslash n$	6	7	8	9	10
	256	16.0	11.6	8.0	5.1
512	19.7	16.0	12.8	9.8	7.1
1024	20.1	17.7	15.1	12.8	10.7
∞	21.3	18.3	16.0	14.2	12.8

The first problem is especially severe for the ideal full-searched scheme where the encoding complexity grows exponentially as the index length (n) of the codebook grows only linearly. The encoding complexity of the vector quantizer includes training of the codebook and coding of the data. Assume that a fraction F of the source image data is used and it requires m iterations in the codebook training. For the full-searched scheme of an n -bit codebook, the training requires $m2^n F$ operations per pixel, while the coding requires 2^n operations per pixel. Hence, the total computation requires

$$(mF + 1)2^n \text{ operations/pixel (full-searched)}. \quad (12)$$

In order to reduce the encoding complexity, the codebook can be divided into several levels (typically two levels are adequate). The input data vector is compared with the first level codebook. Based on the selected codeword (first level), the input data vector is compared with the codewords of the corresponding second level subcodebook. The codeword (second level) of the minimum distortion is then chosen as the quantization vector to represent the input data vector. This technique is known as a two-level tree-searched vector quantizer. Fig. 5 illustrates the encoding procedure.

For the two-level tree-searched scheme of n_1 and n_2 bit codebook, training requires $m(2^{n_1} + 2^{n_2})F$ operations per pixel. Normally, $|n_1 - n_2| = 0$ or 1 . The encoding requires $2^{n_1} + 2^{n_2}$ operations per pixel. Hence, the total

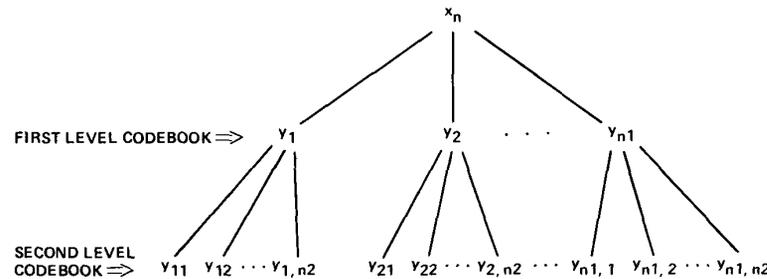


Fig. 5. The encoding procedure of the two-level tree-searched vector quantization algorithm. The first-level codebook contains n_1 codewords and each second-level subcodebook contains n_2 codewords. The complexity is reduced from $(n_1 n_2)$ operations/pixel of the full-searched scheme into $(n_1 + n_2)$ operations/pixel of the two-level tree-searched scheme.

TABLE IV
THE ENCODING COMPLEXITY OF FULL-SEARCHED (FULL) AND TWO-LEVEL TREE-SEARCHED (TREE) VECTOR QUANTIZER. M: NUMBER OF CODEWORDS, F: PERCENTAGE OF TRAINING DATA, S: TOTAL NUMBER OF OPERATIONS/PIXEL, T: NUMBER OF OPERATIONS/PIXEL IN TRAINING THE CODEBOOK, E: NUMBER OF OPERATIONS/PIXEL IN CODING THE DATA.

F	M	128		256		512	
		full	tree	full	tree	full	tree
1	S	896	168	1792	224	3584	336
	T	768	144	1536	192	3072	288
	E	128	24	256	32	512	48
1/4	S	320	60	640	80	1280	120
	T	192	36	384	48	768	72
	E	128	24	256	32	512	48
1/16	S	176	33	352	44	714	66
	T	48	9	96	12	192	18
	E	128	24	256	32	512	48

computation requires

$$(mF + 1)(2^{n_1} + 2^{n_2}) \text{ operations/pixel} \cdot (2\text{-level tree-searched}). \quad (13)$$

An example, shown in Table IV for $m = 6$, shows that it is very costly to train the codebook by using all the image data. A good choice for F is one-fourth or one-sixteenth of the image data. To avoid degradation in the reconstructed image quality, the training data needs to be selected uniformly over the entire image. Notice that for $F = 1/4$ or $F = 1/16$, the computations required by the codebook training (using a two-level codebook scheme) are comparable to or even less than those required by the encoding. This implies that the processing cost to have a unique codebook for each image frame is reasonable.

The second problem (scene dependence) is due to the wide radiometric variations of SAR image data. A fixed codebook shared by all the image frames may not perform well. The degradation resulting from radiometric variations can be reduced by: 1) a unique codebook for each image; 2) uniformly selecting training data; 3) revising the algorithm for adaptivity to local image statistics. A unique codebook for each image requires the codebook generation for each image frame. In addition, the codebook must be transmitted with the coded data to the receiving end. Since the image quality is critical and the real-time data transfer is not a requirement for the browse application, this approach appears to be more advanta-

geous than the fixed codebook scheme. More details will be discussed in Section V-B.

To reduce the effects of the third problem (edge degradation), a large codebook is normally employed. For the SAR images of 1024 by 1024 pixels, a codebook of more than 256 codewords yields little degradation in the reconstructed image quality. Another approach to reduce the edge degradation is by classifying the input data into several classes according to its edge characteristics [22].

V. TEST RESULTS OF ADAPTIVE TRANSFORM CODING AND VECTOR QUANTIZATION

Several 4-look and 8 by 8 averaged Seasat browse images were used as test data sets. The browse images were obtained from the 4-look images by spatially averaging a square block of 64 pixels. Both the signal to distortion ratio (SDR) and the visual quality were used to evaluate the coding performance. The SDR is defined as the average intensity of the reconstructed image data divided by the mean square error between the original and reconstructed image data. Let I_{jk} represent the magnitude of the pixel intensity in the original image and let \hat{I}_{jk} represent the magnitude of the pixel intensity in the reconstructed image.

$$\text{SDR} = 10 \log \left(\frac{\sum \hat{I}_{jk}^2}{\sum (I_{jk} - \hat{I}_{jk})^2} \right) \text{ dB}. \quad (14)$$

A. Test Results of Adaptive Transform Coding

The adaptive transform coding demonstrates the best rate distortion performance among the tested algorithms. Its achievable compression ratio is much higher than that for the predictive coding and ideally any compression ratio can be achieved assuming the induced distortion still yields a useful product.

The compression results of Seasat 4-look images of the Los Angeles urban scene and Beaufort Sea ice scene using the ADCT algorithm are shown in Figs. 6 and 7. Each figure contains the original image together with the reconstructed images at compression ratios of 20:1, 40:1, and 80:1. The test results show that the reconstructed image contains no noticeable artifacts at a compression ratio of 20:1 for both scenes. The blockiness of the reconstructed

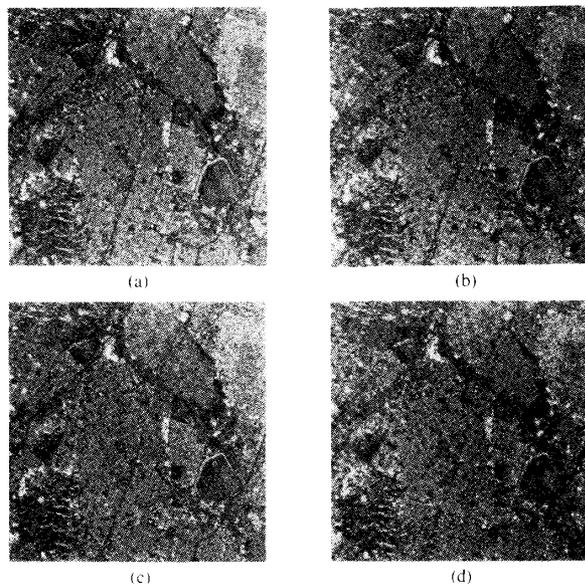


Fig. 6. Compression results of Seasat 4-look SAR imagery by the adaptive discrete cosine transform algorithm. (a) Original image of Los Angeles, CA, 1024×1024 pixels. (b) Compression ratio = 20:1, SDR = 11.56 dB. (c) Compression ratio = 40:1, SDR = 9.11 dB. (d) Compression ratio = 80:1, SDR = 8.83 dB.

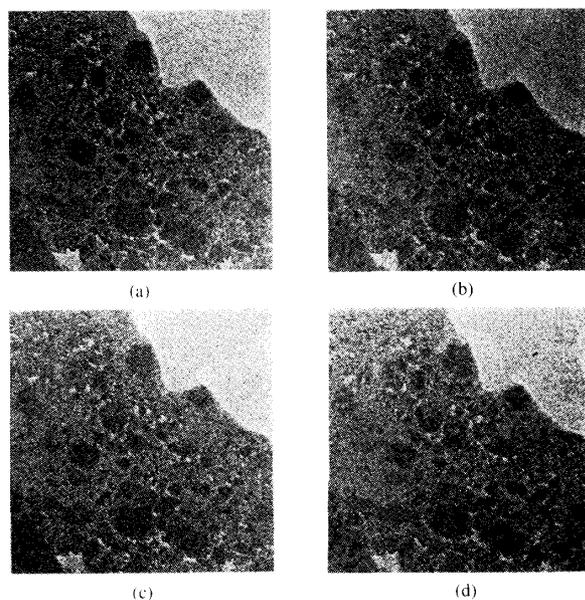


Fig. 7. Compression results of Seasat 4-look SAR imagery by the adaptive discrete cosine transform algorithm. (a) Original image of Beaufort Sea, Alaska, 1024×1024 pixels. (b) Compression ratio = 20:1, SDR = 16.6 dB. (c) Compression ratio = 40:1, SDR = 14.2 dB. (d) Compression ratio = 80:1, SDR = 12.7 dB.

images occurs at a compression ratio of 40:1 for the urban scene and at a compression ratio of 80:1 for the sea ice scene. This is in part due to a shorter inter-pixel correlation distance exhibited in the urban scene. The test

results also show that the low activity regions of the image are more susceptible to blockiness.

Several parameters in the adaptive transform coding such as block size, quantization, number of classes, and normalization factors were varied during the evaluation. The results showed that there is about a 1-dB gain in SDR for a block size increase from 8 by 8 pixels/block to 16 by 16 pixels/block. An additional gain of about 0.4 dB is realized from an increase of 16 by 16 pixels/block to 32 by 32 pixels/block. The quantizer of the transform coding is used to quantize the transform coefficients. The optimal quantizer is the one which matches with the statistics of the transform coefficients. The Max quantizers [23] corresponding to Laplacian distribution and Gaussian distribution [24] were tested separately. The results showed that the Max quantizer of the Laplacian distribution performs slightly better than that of the Gaussian distribution when the rate is below 1 bit/pixel. When the rate is above 1 bit/pixel, the Laplacian model is slightly worse than the Gaussian model. The performance is also a function of the number of classes. Subjectively, using a small number of classes, such as 1 or 2 classes, results in more degradation in dynamic range than a large number of classes, such as 4 or 8 classes. However, if there are highly varying local statistics, a large number of classes tends to make the low activity regions become blocky, even at small compression ratios. Different normalization factors were also tested without significant gain in the SDR.

In evaluating the Seasat browse image data sets, a key result was that in low activity regions, some of the reconstructed images became blocky at a compression ratio of 10:1. This initially was unexpected since the browse image has a higher inter-pixel correlation than the 4-look image. Two reasons are postulated as to why the reconstructed browse image was qualitatively inferior to the 4-look image: 1) The two types of images used in the testing are different in spatial resolution and ground coverage and 2) The 4-look image does not have low activity regions due to the variance from the speckle noise.

To compensate for the blockiness, we constrained the block adaptivity by revising the original bit assignment of the adaptive transform coding algorithm. The bit assignment is chosen such that a specified percentage of the bits (e.g., 50 percent) are assigned with no block adaptivity and the remainder of the bits are then assigned according to the block activity for a given number of classes. In other words, a minimum number of bits are assigned to all the image blocks such that the low activity regions are guaranteed to receive sufficient bits to avoid the early blocky effect.

The difference between the original algorithm and the modified algorithm is illustrated in Fig. 8. Notice that there is clearly blocky effect in the smooth ocean areas by using the original ADCT algorithm as shown in Fig. 8(a) and (c). The blockiness is removed by the new bit assignment procedure as shown in Fig. 8(b) and (d). This revised bit assignment procedure offers more flexibility to

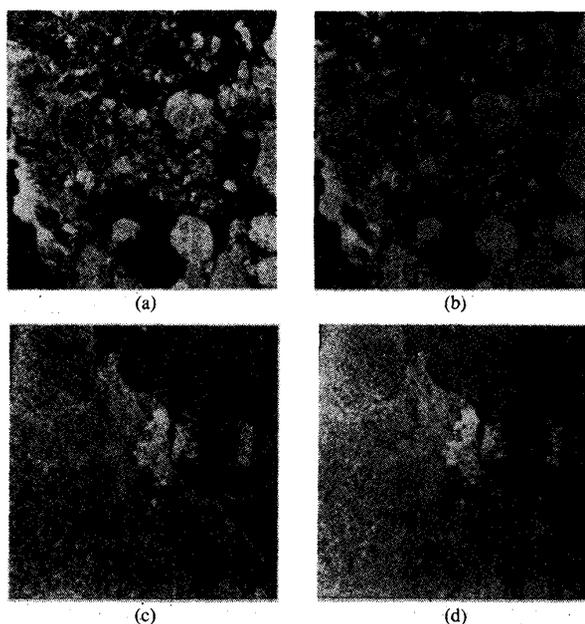


Fig. 8. Comparison between the original ADCT algorithm and the revised ADCT algorithm at compression ratio 10:1. (a) Compression result of Beaufort Sea, Alaska by the original ADCT algorithm, SDR = 18.33 dB. (b) Compression result of Beaufort Sea, Alaska by the revised ADCT algorithm, SDR = 18.11 dB. (c) Compression result of Detroit, MI, by the original ADCT algorithm SDR = 18.30 dB. (d) Compression result of Detroit, MI by the revised ADCT algorithm, SDR = 18.40.

adjust for different scene characteristics and is considered an improvement when compressing SAR images with a large number of looks. It may also improve the quality of those data sets that exhibit highly varying local image statistics. Additional results are shown in Figs. 9 and 10 of Kennewick, Washington, and Detroit, Michigan for compression ratio 10:1, 30:1, and 50:1. At compression ratio of 30:1, the image shown in Fig. 10(c) still exhibits a reasonably good reconstructed image quality. However, for the image shown in Fig. 9(c), it appears blurry and blocky. At a compression ratio of 50:1, the degradation in image quality is very clear.

The Hadamard transform [25] was also used in place of the discrete cosine transform in the algorithm of Fig. 3. This was done to evaluate the performance of the Hadamard transform, which is less computationally intensive since the Hadamard transform does not require multiplication. The image quality was found to be very close to the one resulting from use of the discrete cosine transform.

The major defect of the transform coding technique is its high encoding and decoding complexity. Unlike most of the image coding techniques in which the decoding complexity is usually much simpler than the encoding complexity, the transform coding has about the same order of computation in both encoding and decoding since its computational complexity is dominated by the two-dimensional transformation. The transform coding would appear more promising for a single-encoder single-de-

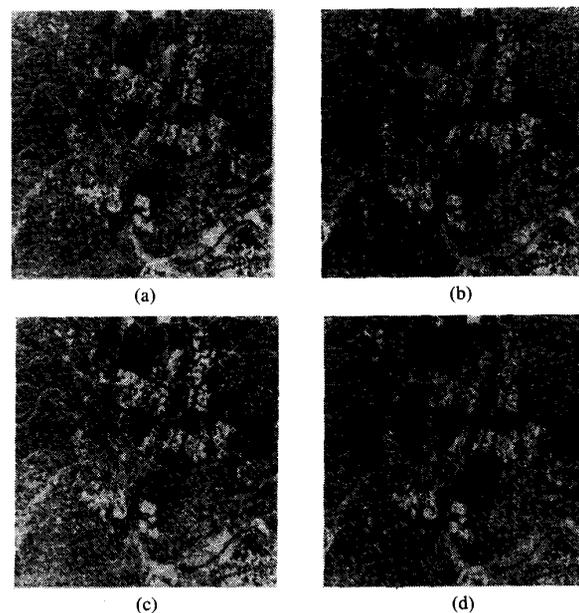


Fig. 9. Compression result of Seasat 8 by 8 spatial average browse imagery by the adaptive discrete cosine transform algorithm. (a) Original image of Kennewick, WA, 896×896 pixels. (b) Compression ratio = 30:1, SDR = 15.62 dB. (c) Compression ratio = 30:1, SDR = 12.86 dB. (d) Compression ratio = 50:1, SDR = 11.80 dB.

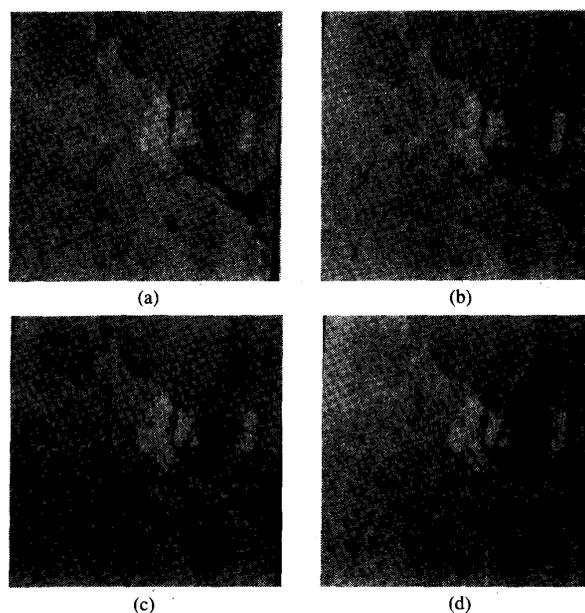


Fig. 10. Compression results of Seasat 8 by 8 spatial average browse imagery by the adaptive discrete cosine transform algorithm. (a) Original image of Detroit, MI, 896×896 pixels. (b) Compression ratio = 10:1, SDR = 18.40 dB. (c) Compression ratio = 30:1, SDR = 16.00 dB. (d) Compression ratio = 50:1, SDR = 15.13 dB.

coder data system where high compression ratio and good image quality are demanded such as for a primary archive. It becomes less attractive for the single-encoder multiple-decoder type system since it places additional requirements on the remote computational facilities.

B. Test Results of Vector Quantization

The vector quantizer based on a two-level tree-searched codebook structure was implemented on a VAX/785 computer equipped with a FPS/5205 array processor. Codebooks of different number of codewords from 32 to 512 codewords were tested. For each case, one-fourth of the original image data was used as the training data set. The result was that about 6 min is required to train the codebook (8-bit) and encode an image of 1024 by 1024 pixels (1 Mbyte). The codebook is considered as the overhead information in the calculation of compression ratio as described in Section IV-B. As the number of codewords increases, the SDR improves only slightly. However, there is dramatic improvement in the visual quality. The improvement in the reconstructed image quality from 32 to 128 codewords is clearly noticeable. Improvement beyond 128 codewords is not visually apparent.

The test results for the 4-look and browse images using an 8-bit codebook are shown in Figs. 11 and 12, respectively. The two-level 8-bit codebook consists of 16 codewords in each level. The reconstructed image quality when using 256 codewords or more is considered to be satisfactory for browse applications. The importance of properly selecting the training data set was also demonstrated in the experiment by coding the data with a codebook obtained by training data of very different characteristics as shown in Fig. 13. Only the lower right corner of the image in Fig. 13(a) was used as the training data set and the resultant codebook was used to code the entire image. The result was that the bright part of the image (Detroit city) was totally smeared as shown in Fig. 13(b). This experiment strongly suggests that: 1) each image frame should have its own codebook, 2) the training data must be selected uniformly from the entire image data, and 3) the algorithm should be revised to be adaptive to the local statistics.

The last recommendation can be implemented by storing the dc terms of local image blocks and removing these dc values from the original image data. The algorithm can then proceed by training and encoding the data without dc values. The dc values are quantized and transmitted with the coded data to the receiving end. An 8-bit representation of the dc values is considered adequate to preserve the local image statistics. They are added back into the image data during the image reconstruction. Since the dc values represent additional overhead information, the image blocks usually assume larger dimension than the codeword dimension to minimize the induced overhead information. The test results showed that the performance of this revised algorithm is much less sensitive to the varying local statistics. An example is shown in Fig. 14, where the dc values of the 8 by 8 pixel image blocks are preserved.

The transform coding and the vector quantization techniques have dramatically different types of distortion. The main distortion of the transform coding is a blurry type of noise spread over the image while the main distortion of

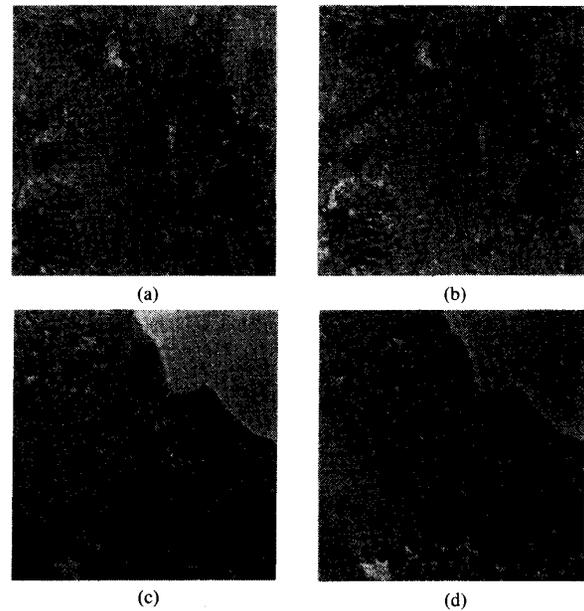


Fig. 11. Compression result of Seasat 4-look SAR imagery by the two-level tree-searched vector quantization algorithm. (a) Original image of Los Angeles, CA, 1024 \times 1024 pixels. (b) Compression result of (a) by 8-bit codebook, compression ratio = 15.1:1, SDR = 11.67 dB. (c) Original image of Beaufort Sea, Alaska, 1024 \times 1024 pixels. (d) Compression result of (c) by 8-bit codebook, compression ratio = 15.1:1, SDR = 17.21 dB.

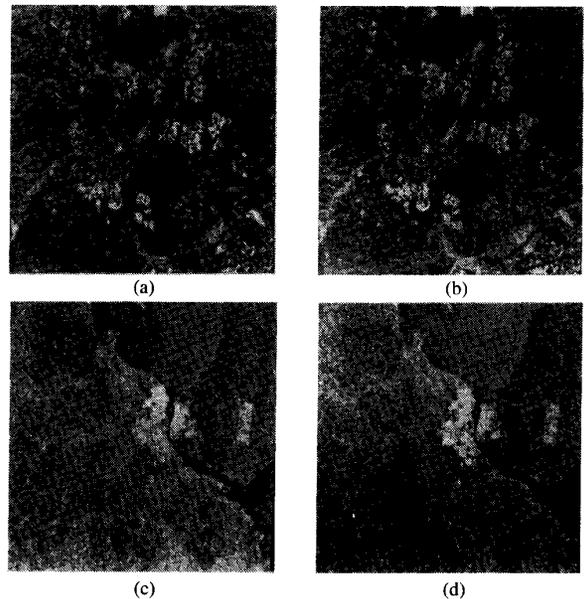


Fig. 12. Compression result of Seasat 8 by 8 spatial average browse imagery by the two-level tree-searched vector quantization algorithm. (a) Original image of Kennewick, WA, 896 \times 896 pixels. (b) Compression result of (a) by 8-bit codebook, compression ratio = 14.8:1, SDR = 14.28 dB. (c) Original image of Detroit, MI, 896 \times 896 pixels. (d) Compression result of (c) by 8-bit codebook, compression ratio = 14.8:1, SDR = 16.24 dB.

the vector quantization is edge degradation. In addition, the achievable compression of vector quantization is basically not flexible as switching to a different compression

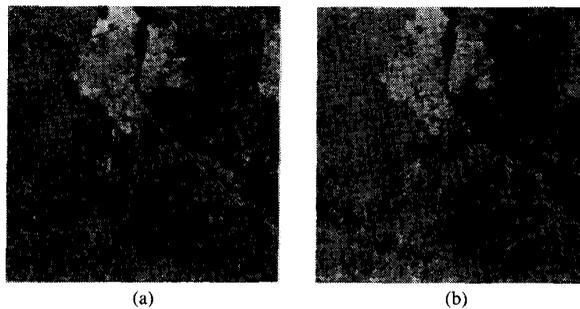


Fig. 13. The effect of mismatch between training data and coded data of the vector quantization algorithm. (a) Original image of Detroit, MI, 512×512 pixels. (b) Compression result of (a) by 7-bit codebook. The codebook was trained by 1/4 of the entire image data (the lower right corner). Compression ratio = 16:1, SDR = 14.61 dB.

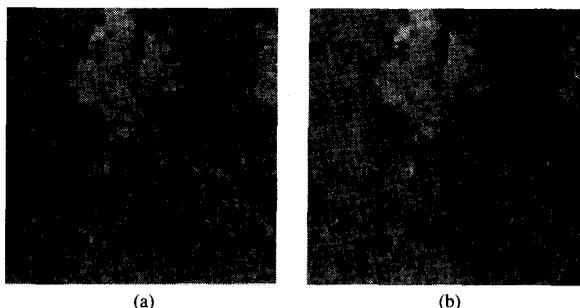


Fig. 14. Comparison between the vector quantization algorithm with and without adaptivity. Detroit, MI, 512×512 pixels. (a) Without adaptivity, CR = 12.8:1, SDR = 17.55 dB. The training data were uniformly selected over the entire image.

ratio involves the regeneration of codebook. Nevertheless, the vector quantization performs well for compression ratios between 10:1 and 20:1, which meets the users' requirements.

The performance of the vector quantization is more scene dependent than the transform coding. In addition, the generation of its overhead information (i.e., the codebook) is much more computationally intensive than that of the transform coding (i.e., the class map and bit allocation maps). Two features of the vector quantization, the scene dependence and the high complexity in updating the codebook, make it less promising for applications requiring transfer of the compressed SAR image data in real-time. However, for the on-line data archive application, to reduce the scene dependence effect, each scene could have a separate codebook since the codebook can be generated off-line and used repeatedly thereafter.

The vector quantization is considered preferable to the adaptive transform coding technique as a data compression algorithm for on-line data archive applications. The primary reason is its extremely simple decoding procedure and that it achieves a good reconstructed image quality between compression ratios of 10:1 and 20:1.

VI. CONCLUSIONS

Based on the algorithm evaluation results and inputs from the user survey, the two-level tree-searched vector

quantization technique is recommended as the SAR image data compression algorithm for the on-line data archive system. Since the performance of the vector quantizer is scene dependent, each scene requires a separate codebook of 256 codewords or more with each codeword containing 4 by 4 pixels. The codebook can be trained by one-fourth of the image data uniformly selected across the entire frame. The algorithm should also be made adaptive to large radiometric variations to minimize the scene dependence effects. We also recommend that the codebook be generated and stored as the image is received by the archive to reduce the effective pretransfer computation and disk storage space. Upon a request, the codebook can be immediately transferred with the coded data (codeword indices). At the remote site, the image will be reconstructed based on the codebook (look-up table) received with the indices.

The benefits of employing image data compression for the on-line data archive distribution system are obvious. A larger data base can be accommodated and both small transfer delay and transfer cost are achieved without large sacrifice in resolution that would result from the spatial averaging. With a compression ratio of 15:1, the archive and transfer cost can be reduced by 93 percent. In addition, given a maximum access frequency of 10 to 15 images/h with a 9.6-Kbps capacity data line, the transfer delay can be maintained at an average of only a few minutes per image frame.

This performance improvement per unit cost cannot be realized by other system modifications such as the multi-serial-port system, or by spatial averaging the data. The cost of the image data compression is in the increased computational capability requirement on the archive (encoder) and the remote users (decoder). The implementation cost at the user's site is negligible since the vector quantization decoding mainly requires a software routine for table look-up. The memory requirement to store the codebook is only 4 Kbytes for 256 codewords of 4 by 4 pixels, which is very small compared to the 1 Mbyte of original image data. Since the algorithm structure of the tree-searched vector quantization is suited for hardware implementation and the encoding can be performed in non-real time (i.e., modest performance requirements), it would be cost effective to build such an encoder.

In this paper, we have concentrated on the application of image data compression for electronic transfer of browse image data files to remote users. Another important application of data compression for SAR is the downlink data transmission of raw signal data or image data from the on-board SAR processor to the ground station. By employing efficient coding, this data volume can also be greatly reduced, thus easing the high data rate requirement on the downlink channel. Due to the different environmental constraints, the requirements on image data compression and hence the selected algorithm may change relative to the browse application.

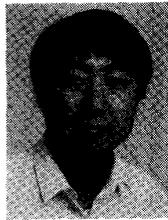
The compression of raw SAR echo data is an area that has yet to be thoroughly explored. For the SAR image

data compression, the visual quality and radiometric accuracy are usually the main concerns. For the SAR raw data compression, the compressed data still must be processed to form the image. Due to the existence of the large speckle noise and the sensitivity of the phase information to correlated image quality, preliminary analysis indicates that compression of raw signal data may not produce equivalent results as with the image data. This application will be the subject of a further study.

REFERENCES

- [1] C. Elachi *et al.*, "Spaceborne synthetic aperture imaging radars: Applications, techniques, and technology," *Proc. IEEE*, vol. 70, no. 10, pp. 1174-1209, Oct. 1982.
- [2] Special Issue on the Shuttle Imaging Radar (SIR-B), *IEEE Trans. Geosci. Remote Sensing*, vol. GE-24, no. 4, July 1986.
- [3] *Proc. Int. Geosci. Remote Sensing Symp.*, Ann Arbor, IEEE Pub. 87CN2434-9, May 1987.
- [4] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, no. 3, pp. 349-389, Mar. 1981.
- [5] R. L. Jordan, "The SEASAT-A synthetic aperture radar system," *IEEE J. Ocean. Eng.*, vol. OE-2, pp. 154-165, Apr. 1980.
- [6] J. C. Curlander, "Performance of the SIR-B digital image processing subsystem," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-24, no. 4, July 1986.
- [7] A. Habibi, "Comparison of the n th-order DPCM Encoder with Linear Transformations and Block Quantization Techniques," *IEEE Trans. Comm. Tech.*, vol. COM-19, pp. 948-956, Dec. 1971.
- [8] P. A. Wintz, "Transform picture coding," *Proc. IEEE*, vol. 60, no. 7, pp. 809-820, July 1972.
- [9] W. H. Chen and H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Comm.*, vol. COM-25, no. 11, pp. 1285-1292, Nov. 1977.
- [10] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, pp. 4-29, Apr. 1984.
- [11] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. COM-28, no. 1, pp. 84-95, Jan. 1980.
- [12] D. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Comm.*, vol. COM-27, no. 9, pp. 1335-1342, Sept. 1979.
- [13] V. S. Frost and G. F. Minden, "A data compression technique for synthetic aperture radar images," *IEEE Trans. Aero Elect. System*, vol. AES-22, no. 1, pp. 47-54, Jan. 1986.
- [14] A. E. Labonte, "Two-dimensional image coding by micro-adaptive picture sequencing (MAPS)," *SPIE Application of Digital Image Processing (IOCC 1977)*, vol. 119, pp. 99-106.
- [15] C. Wu, "Considerations on data compression of synthetic aperture radar images," *Proc. SPIE*, vol. 87, pp. 134-140, 1976.
- [16] —, "A derivation of the statistical characteristics of SAR imagery data," in *Proc. 3rd European Space Agency SAR Image Quality Workshop* (Frascati, Italy), Dec. 1980.
- [17] L. J. Porcello *et al.*, "Speckle reduction in synthetic aperture radars," *J. Opt. Soc. Amer.*, vol. 66, pp. 1305-1311, Nov. 1976.
- [18] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. New York: Wiley, 1975.
- [19] C. Y. Chang, R. Kwok, and J. C. Curlander, "Data compression of synthetic aperture radar imagery," Jet Propulsion Lab. Pasadena, CA, JPL Pub. D-5210, Feb. 1988.
- [20] R. L. Baker and R. M. Gray, "Image compression using non-adaptive spatial vector quantization," in *Proc. 16th Asilomar Conf. Circuits Syst. Computers*, pp. 55-61, 1982.
- [21] B. Ramamurthi, A. Gersho, and A. Sekey, "Low-rate image coding using vector quantization," *IEEE Global Comm. Conf. Rec.*, pp. 184-187, Nov. 1983.
- [22] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Comm.*, vol. COM-34, no. 11, pp. 1105-1115, Nov. 1986.
- [23] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, pp. 7-12, Mar. 1960.
- [24] W. Mauersberger, "Experimental results on the performance of mismatched quantizers," *IEEE Trans. Inform. Theory*, vol. IT-25, No. 4, pp. 381-386, July 1979.
- [25] W. Pratt *et al.*, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, no. 1, pp. 58-68, Jan. 1969.

*



C. Y. Chang was born in Taiwan, Republic of China on May 31, 1960. He received the B.S. degree in electrical engineering from the National Taiwan University in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles in 1983 and 1986, respectively.

He worked on systolic array processing of the convolutional decoding algorithms at the University of California. In 1986, he joined the Jet Propulsion Laboratory, California Institute of Technology. He has been engaged in the data compression of synthetic aperture radar imagery, radar signal processing, radar system design, and analysis. His research interests include channel coding, source coding, VLSI implementation, signal processing, communication systems, and radar systems.

*

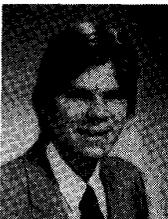


Ronald Kwok (M'85) received the B.Sc. (summa cum laude) degree from Texas A&M University, College Station, TX, in 1976 and the Ph.D. degree from Duke University, Durham, NC in 1980. He was a postdoctoral fellow at the University of British Columbia, Vancouver, BC in 1981.

After that, he was with MacDonald Dettwiler and Associates in Richmond, B.C. where he worked on algorithms for Landsat and Radarsat ground data systems. In 1985, he joined the Radar Science and Engineering Division at the Jet Propulsion Laboratory, Pasadena, CA, where he developed techniques for geometric rectification of SAR imagery and served in a radar system engineering capacity on the Magellan and Alaska SAR facility projects. He is currently group leader in the SAR Systems Development and Processing Group responsible for research and development of post-processing techniques for SAR image data.

Dr. Kwok is a member of Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu.

*



John C. Curlander (M'85) was born in Baltimore, MD, in 1954. He received the B.S.E.E. (with honors) and M.S.E.E. degrees in electrical engineering from the University of Colorado, Boulder, CO, in 1976 and 1977, respectively, and the Ph.D. degree from the University of Southern California in 1985.

From 1977 to 1978 he was with the Institute for Telecommunication Sciences in Boulder, where he worked in the area of high-speed digital communication systems. In 1980 he joined the Radar Signal Processing Research Group at the Jet Propulsion Laboratory, Pasadena, CA. There his work has centered on research in the area of synthetic-aperture radar digital image processing. He is currently involved in systems engineering for the Shuttle Imaging Radar project and Venus Radar Mapper and is group supervisor of the Radar Systems Development and Processing Group responsible for SAR signal processing research and development of ground based processing systems.

Dr. Curlander is a member of Tau Beta Pi and Eta Kappa Nu. He has received NASA Achievement Awards for his contributions to the development of the Seasat SAR Digital Processor and the Shuttle Imaging Radar projects, the 1984 IEEE Browder J. Thompson Prize Paper Award for his paper, "Location of spaceborne SAR imagery," the IEEE Geoscience and Remote Sensing Society's Centennial Award for Young Engineers and the 1988 NASA Exceptional Service Medal for his work in SAR processing and post-processing techniques.