

AD-A219 590

178000-30-X

Final Technical Report

A SURVEY OF IMAGE COMPRESSION TECHNIQUES WITH APPLICABILITY TO SAR IMAGERY

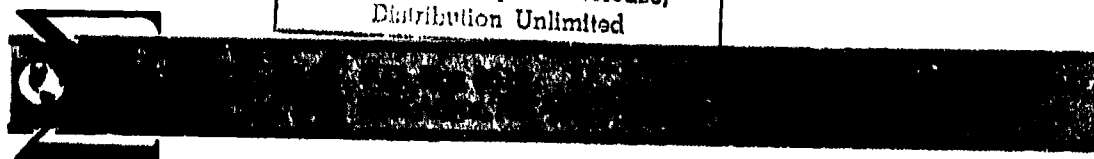
K.W. DANIEL

DECEMBER 1987

Prepared for:
Joint-STARS PMO
Fort Monmouth, NJ 07703-5306
Contract DAAK 20-84-C-0892

DTIC
ELECTE
MAR 22 1990
S E D

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited



90 03 22 008

A SURVEY OF IMAGE COMPRESSION TECHNIQUES
WITH APPLICABILITY TO SAR IMAGERY

Kenneth W. Daniel

December 1987

Environmental Research Institute of Michigan

Accession For	
□	□
□	□
□	□
form 50 <i>Mr</i>	
Distribution/	
Availability Code <i>4</i>	
Special	
A-1	



TABLE OF CONTENTS

1.0	INTRODUCTION AND BACKGROUND	1
1.1	INTRODUCTION	1
1.2	UNIQUE ASPECTS OF SAR	1
2.0	COMPRESSION AND CODING	5
2.1	REMAPPING	5
2.2	LOG COMPRESSION	5
2.3	SPATIAL-DOMAIN ENCODING	6
2.4	BLOCK CODING	7
	2.4.1 Transform-Domain Techniques	8
	2.4.2 Block Truncation Coding (BTC)	9
3.0	BTC RESULTS	13
3.1	COMPARISON OF BTC WITH DPCM	23
3.2	VECTOR QUANTIZING	24
4.0	RECOMMENDATIONS	29
	REFERENCES	31

LIST OF FIGURES

2-1	BTC Technique	10
3-1	Full Resolution Sar Image	15
3-2	4 x 4 BTC Image	17
3-3	8 x 8 BTC Image	19
3-4	16 x 16 BTC Image	21
3-5	One-Dimensional Vector Quantization Example	25
3-6	Two-Dimensional Vector Quantization	26
3-7	Code Table Example	28

1.0 INTRODUCTION AND BACKGROUND

1.1 INTRODUCTION

This paper is intended as a primer on image compression techniques as they apply to compressing Synthetic Aperture Radar (SAR) imagery. Particular attention will be focused on two techniques which are thought to be particularly suitable to this particular application because of their simplicity and speed, namely Vector Quantization (VQ) and Block Truncation Coding (BTC). More well-known techniques such as Differential Pulse Code Modulation (DPCM) and Discrete Cosine Transform (DCT) will also be discussed. Recommendations for experiments on SAR imagery are made.

1.2 UNIQUE ASPECTS OF SAR

SAR imagery differs from conventional visual band photographic or electro-optical imagery in two significant respects: first by having a large dynamic range (as great as 96 dB), and second, by the presence of speckle noise, the sort of noise seen in optical holography, (Ref's 1-8). Since SAR images are produced by coherent processing of scattering-type signal reflections, speckle effects are produced by what has been described as intra-pixel interferometry: the coherent signal interaction between separate scatterers within a single pixel. This speckle phenomenon is always present and varies from pulse to pulse for a given pixel. Speckle contains information pertaining to the size and location of terrain scatterers and perhaps to sub-pixel textural information (Tomiyasu [8]), but for most common SAR imaging applications this information is of no concern, and in fact can be described as signal-dependent (multiplicative) noise. In holography this degradation has been estimated as causing a reduction in resolution such that an aperture 2.6 times as large as that required for a speckle-free image would be necessary for equivalent resolution (Kozma, [2]). Speckle noise likely contributes to a reduction in

adjacent cell correlation according to a Rayleigh model described by Wu [3]. Spatial lowpass filtering can be used to smooth the speckle noise, but edge sharpness is reduced by such filtering. Multiple-look averaging will improve the image quality without blurring edges, but at the expense of requiring collection of the additional information. Numerous approaches to the speckle-reduction problem have been studied. Lim and Nawab [4] observed that grey scale modification could be used to improve the image. This is of note since grey scale modification represents a dynamic range adjustment (DRA); thus it may be expected that a logarithmic type of DRA will decrease the speckle effect. The same authors found some improvement using homomorphic filtering such as described by Oppenheim [14]. A homomorphic Wiener filter was found effective by Jain and Christensen [6]. Tur, Chin, and Goodman [7] have observed that the multiplicative noise model is not always valid for speckle, specifically, when the object imaged contains fine details which cannot be resolved by the imaging system. Lee [9] has proposed a filtering approach called Sigma Filtering, in which each pixel is replaced by an average of those neighboring pixels which have their amplitude levels within two standard deviations of that pixel. Lee's approach minimizes degradation of fine detail, as does the recent approach of Crimmons called Geometric Filtering [10], [11]. Geometric Filtering is an iterative non-linear geometric approach using the convex hull description of a pixel set. Crimmons defines a Speckle Index as the ratio of deviation to mean of a block of pixels, and demonstrates superior reduction of speckle so defined using geometric filtering as compared with look averaging.

Speckle reduction is considered relevant to image compression because: (1) in the case of the Discrete Cosine Transform (DCT), speckle very likely causes the algorithm to assign bits to transform domain regions (such as the high spatial frequencies) which typically should receive very few bits, thus forcing an unwanted reallocation of bits; (2) in the case of Block Truncation Coding (BTC), the algorithm

codes and transmits block standard deviation, and since speckle represents a non information-bearing component (for typical exploitation applications) which contributes to the block variance, it may well contribute to the coding of unnecessary bits for the block variance.

2.0 COMPRESSION AND CODING

General discussions on compression are contained in References [12-15]. Brief discussions on Remapping and Log compression are given below, followed by descriptions of Differential Pulse Code Modulation (DPCM), Transform Domain approaches such as the Discrete Cosine Transform (DCT), and finally by discussions of Block Truncation Coding (BTC) and Vector Quantization (VQ).

2.1 REMAPPING

Remapping techniques are those which can be implemented by the use of simple lookup tables which map one image pixel amplitude into another amplitude on the basis of input amplitude. Strictly speaking, remapping may or may not result in a compression of the number of bits per pixel, but may simply expand or contract the range of output value for a corresponding input range in order to improve viewing or to match the characteristics of a display device (Matthews, et al [16]). This is sometimes performed to expand the contrast over a desired range of amplitudes where desired features are felt to lie, and is then referred to as Dynamic Range Adjustment (DRA). If the output range is re-quantized at fewer bits per pixel, a data compression results. Lipes [17] reports on compression of SAR using various quantizations. The quantizing is optimally done so as to result in minimum average error in the quantized signal by apportioning the quantizing ranges as described by Max [18].

2.2 LOG COMPRESSION

A particular remapping, the log transform, has been recommended for compression of SAR imagery by Crowe, et al [19]. An actual example of such a remapping used to achieve compression is the log compression presently in use in the ASARS, in which a 96 dB dynamic range has been achieved by incorporating a scaling parameter derived from

the system AGC into the log compression. Informal experiments by the ASARS contractor have resulted in pleasing results from using a simple square-root compression having only 48 dB of dynamic range.

2.3 SPATIAL-DOMAIN ENCODING

Differential Pulse Code Modulation (DPCM) is a compression technique that has been subject to much examination in the literature, and thus will only be dealt with briefly here. Lynch [12] and Oppenheim [14] provide good general descriptions. DPCM algorithms achieve compression by employing an embellishment on what is called "delta modulation", where only a one-bit estimate (slope up or slope down estimate of) the difference between adjacent pixels rather than their absolute value is transmitted. In the case of DPCM, a predictor is employed to make an estimate of an adjacent pixel amplitude based on the value of one or more of its neighbors. Identical predictor equations are employed at both the sending and receiving ends, thus making it possible to store or transmit only the difference between the actual and predicted values. Considering a spatial matrix of pixels with a single pixel designated as $x(m,n)$, a typical linear predictor equation might have the form:

$$x(m, n) = +0.75*x(m, n - 1) + 0.75*x(m - 1, n) - 0.5*x(m - 1, n - 1)$$

If all three adjacent pixels (left, above, and above-left) were equal, the predicted value would be equal to that value, whereas if the "above-left" pixel were low in amplitude, the predicted value would be correspondingly higher since the three pixels would appear to define a plane sloping upwards toward the pixel to be predicted. Although the coefficient terms above add to unity, this need not be the case in general. Calculation of predictor coefficients based on the statistics of the imagery will result in appropriate coefficients for that imagery. Subtracting the predicted value from the actual pixel value at the point of the original image leaves an error term which can be

quantized into fewer bits than the original pixels and then transmitted or stored. Practical DPCM systems typically achieve average compressed quantization levels of between 4 and 5 bits per pixel. The design of the quantizer lookup tables should be done according to the nature of the type of imagery to be transmitted. The error signal of SAR will most likely have a different probability distribution than the corresponding error signal for photographic or IR imagery. An optimum quantizer will have its quantization ranges apportioned so as to minimize the overall error probability on the average; different Quantizer Tables should thus be employed according to the class of imagery. DPCM systems designed in two dimensions such as the above equation are often described as 2D-DPCM to distinguish them from single dimension or temporal schemes such as used with time waveforms like speech. Adaptive DPCM (ADPCM) applies to systems which adapt the predictor or the quantizer to the statistics of the input image on a block basis. Practical DPCM systems typically achieve average compressed quantized levels of between 4 and 5 bits per pixel, although Werness [20] has described a dual rate predictor (ADPCM) which achieves coding rates ranging between 1.3 and 2.3 bits per pixel for SAR data.

2.4 BLOCK CODING

Block coding of images includes those coding or compression techniques which operate on blocks of data samples, typically square in the case of imagery. Such algorithms include the transform techniques such as DCT, and quantization techniques such as Block Truncation Coding (BTC), and Vector Quantizing (VQ). These algorithms typically process each block differently according to the local statistics of each block, so a common concern among them is the occurrence of visible block boundaries in some processing situations and at larger block sizes (say 16x16 blocks as opposed to 4x4 blocks). Habibi [21] and Gonsalves [22] provide comparisons of block techniques with DPCM.

2.4.1 Transform-Domain Techniques

Transform domain techniques (Wintz, [23]) in general achieve their effectiveness by taking advantage of the fact that the information contained in some of the transformed values of the image is greater than in others. That is, transforming a spatial image using a common transformation such as the Discrete Fourier Transform, one obtains an orthogonal or uncorrelated set of coefficient values in the spatial frequency domain; the information contained in the high spatial frequency coefficients has been shown to be of less value than those of lower frequency. It is thus possible to either dispose of the less important coefficients, or assign fewer bits to them in an encoding scheme prior to storage or transmission. A number of transforms have been investigated over the years, and it has been shown that the optimal orthogonal transform from a coefficient truncation mean-square error (MSE) sense is the Karhunen-Loeve (K-L) transform. However, no efficient algorithm exists for computing the K-L transform. Given this fact, other transforms which do have "fast" algorithms have been investigated, including the Haar, Slant, Walsh, and the DFT/FFT. It is generally concluded that, for Markov sources, the DCT gives the MSE performance closest to the optimum K-L results, as described by Ahmed, et al [24]. Gioutsos and Werness [25] have found that in the case of SAR imagery, the Sine transform rather than the Cosine transform is optimum. It should be noted that the MSE measure of performance is commonly used because it is objective and can be easily be measured; however, it is a pixel-oriented measure, and may not be truly indicative of the actual spatial feature distortion. Werness [15] has proposed a set of metrics for evaluating the performance of compression schemes that take into account the unique nature of SAR imagery; the metrics quantify two types of noise common in predictive systems such as ADPCM: slope overload and large quantizer step size noise.

As compared to simple algorithms such as the BTC or simple spatial VQ, the DCT is computationally intensive, but can result in compression rates below 2 bits/pixel with little visual degradation. The heart of the scheme is a systematic apportionment of quantization bits across the set of transform coefficients, with those coefficients in the "upper left" of the coefficient matrix, corresponding to those components closer to zero spatial frequency, being assigned the greater number of bits. The highest spatial frequency components are often discarded or assigned only one bit. Each coefficient is then quantized according to the number of assigned bits, with quantization lookup tables available for each bit-assignment case. The quantizer tables must be optimized so as to minimize the quantization error, thus taking into account the probability distribution of the transform coefficients. In order to achieve consistent performance, it then becomes necessary to engage in normalization of the input blocks prior to performing the 2D transforms from the spatial domain. It is typical to do the transforms using blocks on the order of 16x16. In addition to the computation intensity of the two dimensional cosine transform, the normalization calculations and selection of bit assignment rules add significantly to the complexity of DCT types of transforms. Chen and Smith [26] describe a practical DCT implementation. Recently, a two dimensional Fast Cosine Transform (FCT) has been introduced by Lee [27] as a means of improving on the computational efficiency of the DCT, with refinements described by Haque [28].

2.4.2 Block Truncation Coding (BTC)

A recently developed spatial domain compression technique is Block Truncation Coding (BTC). References [29]-[39] describe BTC and its application to visual-domain imagery. The application to Seasat SAR imagery is described by Frost in [39]. Figure 2-1 illustrates the BTC procedure. The BTC algorithm uses a two-level (one-bit) non-parametric quantizer which adapts to preserve the first two local moments of the image. Non-overlapping blocks (typically 4 x 4 or

FIGURE 2-1. BTC TECHNIQUE

Suppose a 4 x 4 image block consists of the following pixel amplitudes:

116	120	53	49
36	199	255	249
17	0	10	170
30	3	6	249

The block mean and standard deviation are thus:

$$\begin{aligned}\bar{x} &= \text{mean} &= 97.63 \\ \sigma &= \text{std. dev.} &= 93.79\end{aligned}$$

with

$$k = 16 = \text{number of elements in the block}$$

Using the block mean as the threshold value,

$$q = 7 = \text{the number of pixels above the threshold}$$

$$\begin{aligned}\text{yields: } a &= 14 \quad (\text{to the nearest integer}) \\ b &= 203\end{aligned}$$

for the "low" and "high" values respectively.

The bit plane is then

1	1	0	0
0	1	1	1
0	0	0	1
0	0	0	1

resulting in a reconstructed block of

203	203	14	14
14	203	203	203
14	14	14	203
14	14	14	203

which preserves the block mean and standard deviation.

8 x 8 pixels) are replaced with pixels quantized to one of two levels so that the mean and variance of the block are maintained. The two levels for each block represent local "bright" and "dark". A bit plane of one's and zero's is transmitted, representing each pixel as either high (bright) or low (dark), corresponding to pixels above or below the block means. Also transmitted are quantized values of the block mean (corresponding to the average pixel brightness) and block standard deviation (corresponding to the pixel contrast within the block). The receiver or decoder reconstructs an approximation to the original block by replacing each pixel having a "high" value represented by a "one" in the bit plane by a value equal to

$$b = \bar{x} + \sigma\sqrt{(k - q)/q}$$

and points represented by a "zero" in the bit plane by

$$a = \bar{x} - \sigma\sqrt{q/(k - q)}$$

where $\bar{x}$ = block mean

σ = block standard deviation

k = number of points in block

q = number of points above the block mean

Thus each point above the threshold (the original block mean) is replaced by value equal to "b", and those below the mean by a value equal to "a". The process is illustrated in Figure 2-1. The threshold is commonly chosen a priori to be the block mean, although other more complex calculated thresholds have been proposed in order to preserve higher-order moments. As seen in Figure 2-1, the compressed block consists of pixels having a value above the original block mean (the 204-values) and pixels below the mean (the 17-values). The block mean and variance of this new block are identical to those of the original block. The transformation is repeated for every (non-overlapping) block over the image, with new high and low values calculated for each

block. The average value (or average brightness) and variance (corresponding to the contrast between pixels) are thus adaptive over the image, adjusting to fit the image at each 4 x 4 area, for example. The required number of bits to be transmitted is simply

$$m^2 + N(\text{mean}) + NSTDEV,$$

where m^2 = block size (4 x 4 = 16 for a square 4 x 4 block) and NMean, NSTDEV are the number of bits used to encode the block mean and block standard deviation. If the mean and standard deviation were each represented by 8 bits, then a 4 x 4 block would require 16 bits for the 1-bit mask and 2 x 8 = 16 bits for the mean and standard deviation, giving a total of 32 bits for 16 pixels, or 2.0 bits per pixel. If 8 x 8 blocks were similarly encoded, the result would be 80 bits for 64 pixels, or 1.25 bits per pixel. The performance is seen to be asymptotic to 1 bit per pixel with increasing block size. The general formula for the number of bits per pixel is simply:

$$\text{bpp} = 1 + (NM + NV)/(\text{Block Size})$$

where bpp = bits/pixel

NM = Number of bits used to encode the block mean

NV = Number of bits used to encode the block standard deviation.

3.0 BTC RESULTS

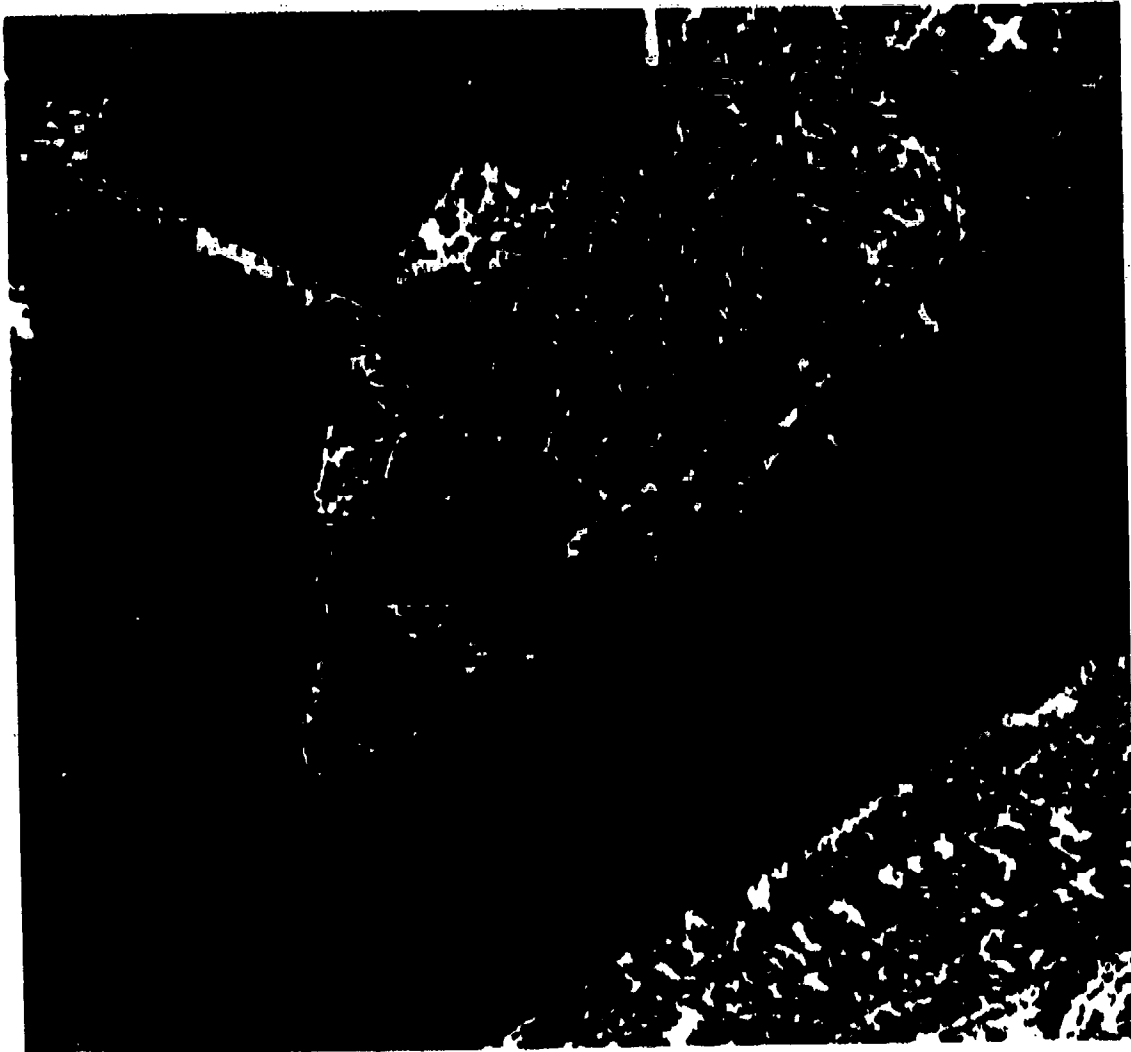
The results of Block Truncation Coding of SAR imagery are shown in Figures 3-1 through 3-4. These figures were obtained photographically using a DeAnza image processing softcopy (CRT) display having 512 x 512 display resolution, with 256 levels of intensity displayed. For this investigation, an original STAR-1 SAR image of Belle Isle and Belle Isle Bridge in the Detroit River was used (Figure 3-1). The digital image, having 15 bits/pixel, is first mapped to 8 bits/pixel using a lin/log dynamic range adjustment (DRA) as is typical for SAR data display. No transmission error effects are included. The mean and standard deviation parameters are quantized at 8 bits (per block) each. The results of the 4 x 4 BTC (2.0 bits per pixel), 8 x 8 BTC (1.25 bits/pixel) and 16 x 16 BTC (1.06 bits/pixel) are presented for the same SAR scene in Figures 3-2 through 3-4 respectively. The BTC4 results are clearly superior to those of BTC8 and BTC16; the marginal improvement in compression available by increasing block size is small as BTC asymptotically approaches 1 bit per pixel. As the BTC block size increases, the underlying blocks can be seen in the image, particularly for the 16 x 16 BTC. However, flicker comparisons between BTC4 and the original images reveal almost no differences: some bright pixels are reduced slightly in amplitude, while some darker pixels increase slightly, but shapes and edges are preserved quite well. A 4 x 4 block size is so small that it is virtually unnoticeable to an observer. Thus we conclude that SAR data compression up to 2 bits/pixel can be achieved using BTC with 4 x 4 blocks without significant loss of image quality.

BTC has sometimes been considered as a special case of Vector Quantizing, discussed below. One significant difference lies in the non-parametric or adaptive nature of BTC; the algorithm adjusts to the statistics of each individual scene on a block by block basis.



87-10489-1

FIGURE 3-1. FULL RESOLUTION SAR IMAGE



87-10459-4

FIGURE 3-2. 4 X 4 BTC IMAGE



87-10459-7

FIGURE 3-3. 8 X 8 BTC IMAGE



87-10489-10

FIGURE 3-4. 16 X 16 BTC IMAGE

3.1 COMPARISON OF BTC WITH DPCM

In comparison with adaptive DPCM (ADPCM), probably the most prevalent example of a spatial-domain compression technique, a number of aspects of BTC are noteworthy. BTC is inherently block adaptive in that all the coding parameters have the ability to change at every block. DPCM is computationally more complex than BTC: a three-step ADPCM predictor based on pixels to the left, above, and at a 45° angle would require three multiplications for every pixel for the prediction process alone. The ADPCM coder loop is more complex than the BTC coding process, requiring an ADPCM decoder in the predictor feedback loop as the cost required to achieve somewhat higher expected quality than with BTC.

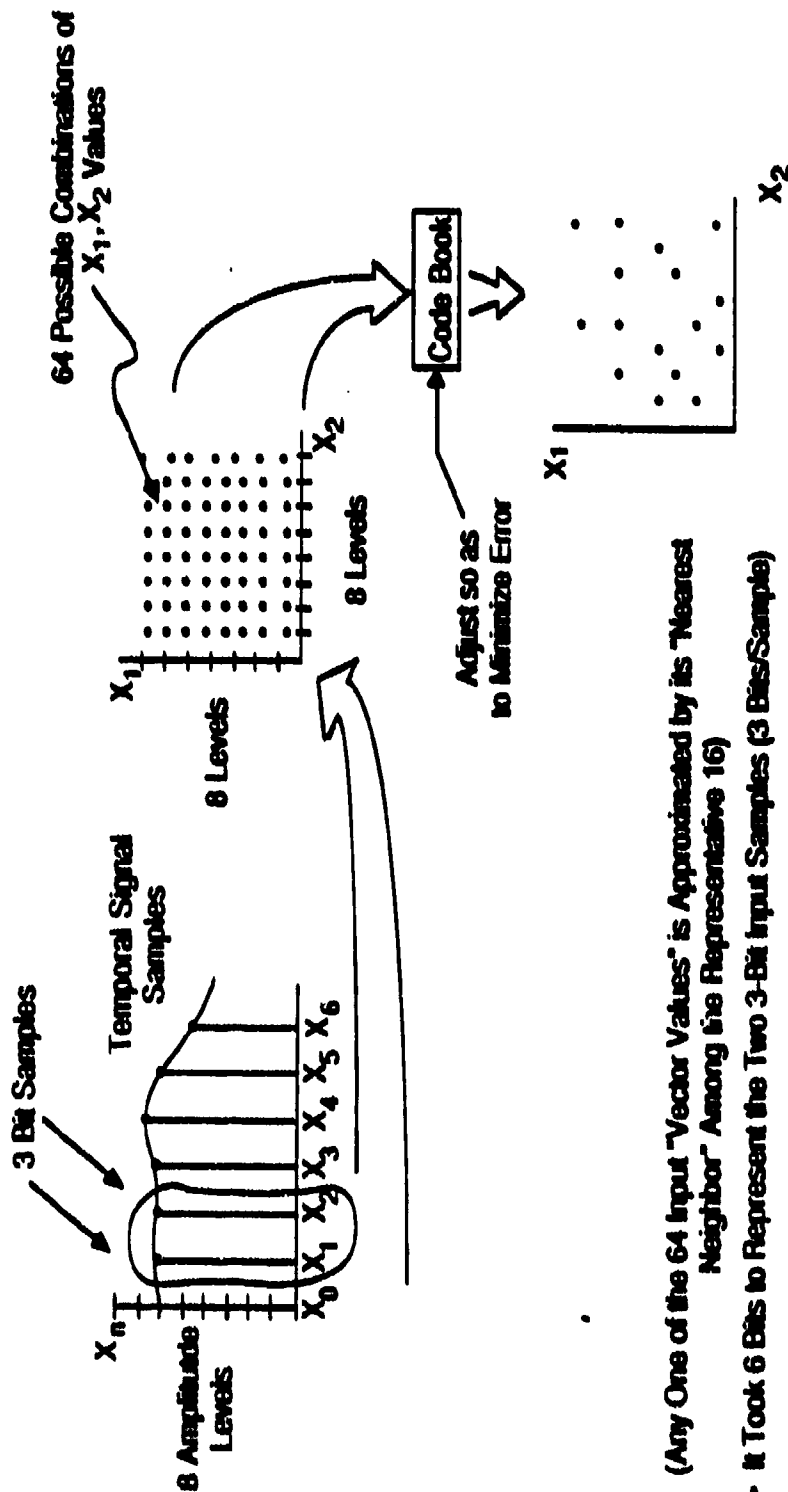
Channel errors are observed in the reconstructed image differently in BTC and DPCM. In DPCM the channel errors appear as streaks in the image. If the DPCM predictor is not adaptive or new initial conditions are not sent after a few rows of the image, then these streaks will propagate throughout the entire image. The amount of propagation is determined by the type of predictor used. If a Moving Average (MA) predictor is used, there will be no streaking. For BTC the errors appear as bad blocks of data (small and localized in the case of 4×4 pixel blocks).

Reference [31] describes a DPCM algorithm capable of compressing photographic images from 11 to 3 bits/pixel with indiscernible (subjective) visual degradation. A conservative BTC approach (4×4 , with 11 bits per input pixel and 11 bit mean and standard deviation) would achieve 2.375 bits/pixel, by comparison. An interesting combination of DPCM and BTC has been suggested in Reference [32], in which the DPCM difference signal is quantized to one bit using BTC. A photographic image was quantized to 1.18 bits/pixel using this hybrid technique, compared to a 1.63 bits/pixel rate for BTC alone for the same image.

3.2 VECTOR QUANTIZING

Vector Quantizing (VQ) as first applied to temporal signals is described by Gray [40], and by Gersho [41] as applied to imagery. An appreciation for the possibilities inherent in VQ can be gained by recalling a classical result from communication theory, namely that the error performance achieved in encoding a sequence of data samples may be decreased to an arbitrarily small level by coding the data in increasingly large blocks of samples. This result is known as the Block Orthogonal Coding case. The application of this principle to data compression can be appreciated by considering that the same technique of block data encoding could be applied, not to reduce error with increasing block length, but holding the error rate constant with increasing block length, and compensating for the potentially improved error performance by allowing quantizing noise to increase by coarser quantizing. The net would be to encode with fewer bits, thus achieving compression. In general then, encoding the samples in blocks or vectors rather than individually (as scalar amplitudes) can be used either to achieve compression or to reduce error rate. Reducing the average number of bits per encoded sample while encoding blocks is called Vector Quantization.

Figure 3-5 illustrates the basic principle of VQ with a very simple one-dimensional case encoding signal samples by pairs. Figure 3-6 extends the technique to the two-dimensional image case. The key step of the process is the selection of the optimum mapping from an input sample block or vector to one of a limited set of representative codevectors. The set of possible codevectors must be chosen based on the probability densities of the input blocks so as to minimize the overall average error or image distortion. The most popular means of determining this codebook mapping is by employing an optimization algorithm known as the LBG algorithm, in reference to the three authors (Linde, Buzo, and Gray, [42]) who developed it. A codebook produced by such a process for imagery data is represented by a set of



(Any One of the 64 Input "Vector Values" is Approximated by its "Nearest Neighbor" Among the Representative 16)

- It Took 6 Bits to Represent the Two 3-Bit Input Samples (3 Bits/Sample)
- The 16 Output Pairs Can Be Represented by 4 Bits, for an Average of 2 Bits Per Sample, Incurring Some Quantization Noise
- Easily Extended to Two-Dimensional Image Input Sample Blocks, Typically 3×3 or 4×4 Blocks

Example: Determine the 16 Most Representative Points in the 2-Dimensional Space to Represent the 64 Possibilities

FIGURE 3-5. ONE-DIMENSIONAL VECTOR QUANTIZATION EXAMPLE

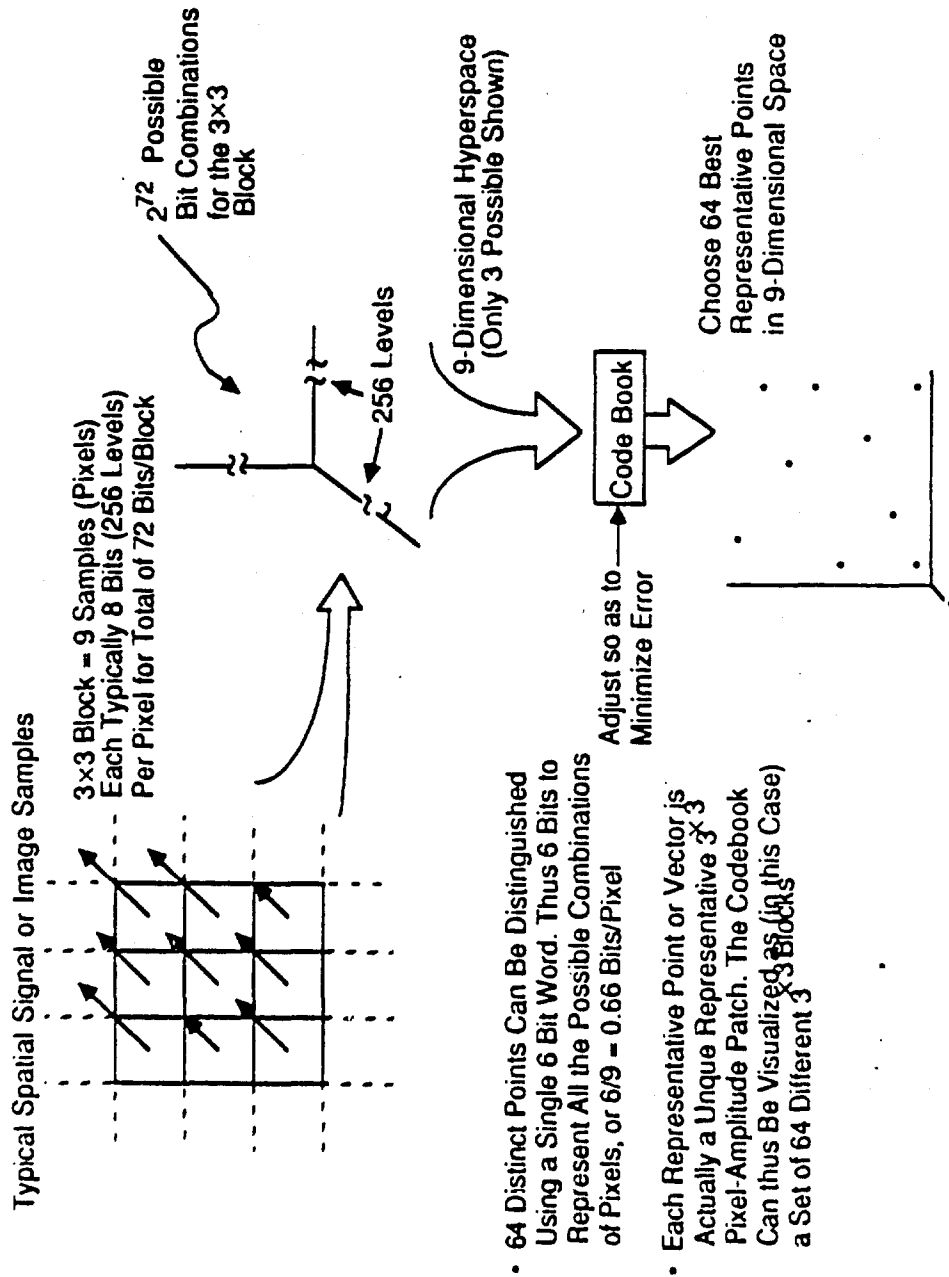


FIGURE 3-6. TWO-DIMENSIONAL VECTOR QUANTIZATION

representative pixel patches as shown in Figure 3-7. In this example case, the 64 representative pixel blocks are arrayed in an 8×8 matrix (the matrix presentation is arbitrary, only the number of unique blocks is significant). Any 3×3 (in this case) input data block is replaced by the most appropriate representative block from the codebook matrix. Thus as one would expect, the codebook consists of blocks or image patches having various intensities and gradients, with some of the blocks having typical edge patterns.

Compression results of 0.66 bit/pixel can be achieved using 3×3 input blocks with a library of 64 codevector blocks. Such VQ techniques are found to encode visual imagery with very little perceptual degradation except at sharp edges, where a staircase effect is found. This latter distortion has been attacked by using a Classified VQ (CVQ) to first classify the block as to type in order to separate out the blocks having edges and encode them from a special code table for edge blocks. Ramamurthi and Gersho describe CVQ in [43]. A number of further extensions to VQ are given by Nasrabadi in [44], including predictor techniques and vector coding of transform coefficients.

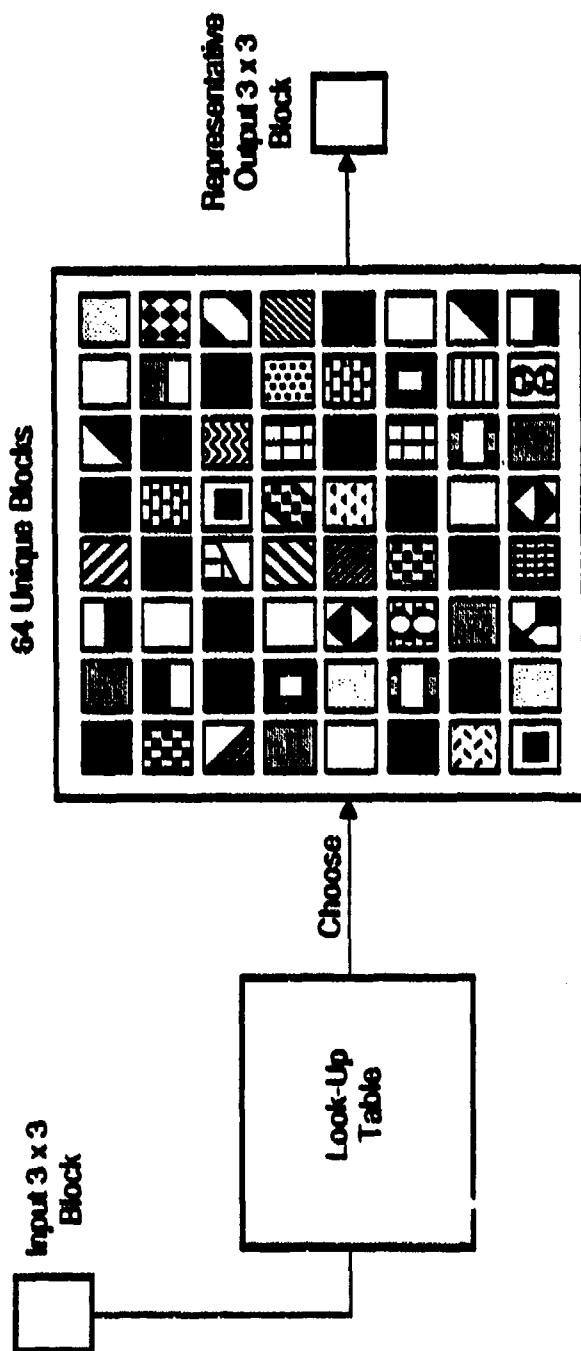


FIGURE 3-7. CODE TABLE EXAMPLE

4.0 RECOMMENDATIONS

No results of the application of VQ to SAR imagery are available at present. Considering the nature of SAR imagery as described earlier, it may be possible suggest areas for investigation as to the best means to employ VQ with SAR imagery:

- To what degree does the edge fidelity criterion apply to SAR imagery.
- The effect that Speckle-reduction algorithms would have on simplifying codebook selection operations.
- The inherent degradation of target patterns which would be incurred using VQ (jagged edges for example) may affect image utility different for human Image Analysts (IAs) than it would affect current generation Automatic Target Recognition (ATR) algorithms; the relative effects of such degradations should be investigated by a comparative experimental effort.

It is thus recommended that:

- A codebook optimization algorithm be implemented (such as the LBG algorithm) and run against representative SAR imagery containing tactical targets of interest.
- In house trained IA's be employed to assess the degradation of target signatures for simple VQ (as opposed to Classified VQ, or CVQ). This assessment would include a recommendation as to whether a more complex multistage VQ approach such as CVQ be employed to retain edge fidelity.
- The utility of speckle reduction for pre-VQ processing or post-VQ edge smoothing should be investigated.
- A modest performance single stage VQ having small blocks (3 x 3 recommended) be used for baseline, with codebook sizes of 64 and 256 be employed.

REFERENCES

NATURE OF SAR IMAGERY AND SPECKLE REDUCTION

1. Sieber, A.J., "Statistical Analysis of SAR Images," Int. Journal of Remote Sensing, Vol. 6, No. 9, 1555-1572, 1985.
2. Kozma, A. and Christensen, C.R., "Effects of Speckle on Resolution," J. Optical Society Amer., Vol. 66, No. 11, November 1976.
3. Wu, C., "Considerations on Data Compression of Synthetic Aperture Radar Images," SPIE Vol. 87, Advances in Image Transmission Techniques, 1976.
4. Lim, J.S. and Nawab, H., "Techniques for Speckle Removal," SPIE Vol. 243, Applications of Speckle Phenomena, 1980.
5. Arsenault, H.H. and April, G., "Properties of Speckle Integrated with a Finite Aperture and Logarithmically Transformed," J. Optical Soc. Amer., Vol. 66, No. 11, November 1976.
6. Jain, A.K. and Christensen, C.R., "Digital Processing of Images in Speckle Noise," SPIE Vol. 243, Applications of Speckle Phenomena, 1980.
7. Tur, M., Chin, K.C., and Goodman, J.W., "When is Speckle Noise Multiplicative?," Applied Optics, Vol. 21, No. 7, April 1982.
8. Tomiyasu, K., "Spectral Analysis of Complex Pulse Response History of a Synthetic Aperture Radar Pixel," Proc. of the 1983 IEEE International Symposium on Geoscience and Remote Sensing, Ann Arbor, Michigan, p. 9.1.
9. Lee, J.S., "A Simple Speckle Smoothing Algorithm for Synthetic Aperture Radar Images," IEEE SMC-13, No. 1, Jan/Feb. 1983.
10. Crimmins, T.R., "Geometric Filter for Speckle Reduction," Applied Optics, Vol. 24, No. 10, 15 May 1985.
11. Crimmins, T.R., "Geometric Filter for Reducing Speckle," Optical Engineering, Vol. 25, No. 5., May 1986.

COMPRESSION, GENERAL

12. Lynch, T.J., Data Compression, Lifetime Learning Publ., Belmont, CA, 1985.

13. Davisson, L.D., "The Theoretical Analysis of Data Compression Systems," Proc. IEEE, Vol. 56(2), pp. 176-186, 1968.
14. Oppenheim, A.V. (ed.), Applications of Digital Signal Processing, Chapt. 4, Section 4.3, Prentice-Hall, 1978.
15. Werness, S.A., "Statistical Evaluation of Predictive Data Compression Systems," IEEE ASSP-35, No. 8, August 1987.

QUANTIZING AND REMAPPING TECHNIQUES

16. Matthews, N.D., Kaupp, V.H., Waite, W.P., and MacDonald, H.C., "An Improvement in SAR Image Interpretability Provided by Post-Correlation Signal Processing," Proc. of the 1983 IEEE International Symposium on Geoscience and Remote Sensing, Ann Arbor, Michigan, p. 5.1.
17. Lipes, R.G. and Butman, S.A., "Bandwidth Compression of Synthetic Aperture Radar Imagery by Quantization of Raw Radar Data," SPIE Vol. 119, Application of Digital Image Processing (IOCC 1977), p. 107.
18. Max, J., "Quantizing for Minimum Distortion," Data Compression, L.D. Davisson and R.M. Gray (Editors), Dowden, Hutchinson, and Ross, pp. 169-174, 1976.
19. Crowe, D.G., Zelenka, J.S., and Granrath, D.J., Logarithmic Compression of SAR Imagery, SAIC Tech. Rpt., prepared for USAICS, ATSI-CD-SPAD under contract DAEA/8-73-A-0127/0018, 16 July 1981.

DPCM TECHNIQUES

20. Werness, S.A., "Application of Predictive Compression Methods to Synthetic Aperture Radar (SAR) Imagery," to be published in Journal of Optical Engineering, Vol. 26, No. 12, December 1987.
21. Habibi, A., "Comparison of nth Order DPCM Encoder with Linear Transformations and Block Quantization Techniques," IEEE COM-19, 1971.
22. Gonsalves, R.A., Shea, A., Evans, N., Huang, T.S., and Delp, E., "Fixed-Error Encoding for Bandwidth Compression," SPIE 22nd International Technical Symposium, San Diego, CA, 1978.

DCT AND OTHER TRANSFORM-DOMAIN TECHNIQUES

23. Wintz, P.A., "Transform Picture Coding," Proc. IEEE, Vol. 60, No. 7, p. 809, July 1972.
24. Ahmed, N., Natarajan, T., and Rao, K.R., "Discrete Cosine Transform," IEEE Transactions on Computers, p. 90, January 1974.
25. Gioutsos, A. and Werness, S.A., "Transform Coding of SAR Images," Proceedings of IEEE ICASSP, 1987.
26. Chen, W-H and Harrison Smith, C., "Adaptive Coding of Monochrome and Color Images," IEEE COM-25, No. 11, November 1977.
27. Lee, B.G., "FCT - A Fast Cosine Transform", Proc. IEEE ICASSP, pp. 28A.3.1-28A.3.4, 1984.
28. Haque, M.A., "A Two-Dimension Fast Cosine Transform," IEEE ASSP-33, No. 6, December 1985.

BLOCK TRUNCATION CODING (BTC)

29. Tasto, M. and Wintz, P.A., "Image Coding by Adaptive Block Quantization," IEEE COM-19(1), pp. 957-972, 1971.
30. Delp, E.J. and Mitchell, O.R., "Image Compression Using Block Truncation Coding," IEEE Trans. on Communications COM-27, No. 9, September 1979.
31. Mitchell, O.R., et al., "Image Coding for Photoanalysis," Proc. of the SID, Vol 21/3, 1980.
32. Mitchell, O.R. and Delp, E.J., "Use of Moment Preserving Quantizers in Differential Pulse Code Modulation (DPCM) Image Coding," SPIE Vol.249, Advances in Image Transmission II, 1980.
33. Mitchell, O.R. and Delp, E.J., "Multilevel Graphics Representation Using Block Truncation Coding," Proc. IEEE, Vol. 68, No. 7, July 80.
34. Delp, E.J., Kashyap, R.L., and Mitchell, O.R., "Image Data Compression Using Autoregressive Time Series Models," Pattern Recognition, Vol. 11, pp. 313-323, 1979.
35. Delp, E.J. and Mitchell, O.R., "Some Aspects of Moment Preserving Quantizers," Proc. of the IEEE International Conference on Communications (ICC), Boston, MA, June 1979.

36. Mitchell, O.R., et al., Coding of Aerial Reconnaissance Images for Transmission Over Noisy Channels, Technical Report RADC-TR-78-210, Rome Air Development Center, 1978.
37. Mitchell, O.R., et al., Improvements in Some Image Compression Techniques for Aerial Reconnaissance Analysis, RADC-TR-79-99, May 1979.
38. Halverson, D.R., Griswold, N.C., and Wise, G.L., "A Generalized Block Truncation Coding Algorithm for Image Compression," IEEE ASSP-32, No. 3, June 1984.
39. Frost, V.S. and Minden, G.J., "A Data Compression Technique for Synthetic Aperture Radar Images," IEEE AES-22, No. 1, January 1986.

VECTOR CODING TECHNIQUES

40. Gray, R.M., "Vector Quantization," IEEE ASSP Magazine, April 1984.
41. Gersho, A., "On the Structure of Vector Quantizers," IEEE IT-28, No. 2, March 1982.
42. Linde, Y., Buzo, A., and Gray, R.M., "An Algorithm for Vector Quantizer Design," IEEE COM-28, pp. 84-95, January 1980.
43. Ramamurthi, B. and Gersho, A., "Classified Vector Quantization of Images," IEEE COM-34, No. 11, November 1986.
44. Nasrabadi, N.M., "Use of Vector Quantizers in Image Coding," IEEE ICASSP, pp. 125-128, 1985.