

1993010983

493

9P

52-43



N93-20172
P-9

Data Compression in Remote Sensing Applications

Khalid Sayood

Department of Electrical Engineering and
Center for Communication and Information Science
University of Nebraska - Lincoln
Lincoln Nebraska 68588-0511

1 Introduction

With the current and future availability of an increasing number of remote sensing instruments, the problem of storage and transmission of large volumes of data has become a significant and pressing concern. For example, the High-Resolution Imaging Spectrometer will acquire data at 30 meter resolution in 192 spectral bands. This translates to a data rate of 280 Mbps! The Spaceborne Imaging Radar - C (SIR-C) will generate data at the rate of 45Mbps per channel with four high data rate channels [1]. To accommodate this explosion of data there is a critical need for data compression. One can view the utility of data compression in two different ways. If the rate at which data is being generated exceeds the transmission resources, one can use data compression to reduce the amount of data to fit available capacity. Or given some fixed capacity, data compression permits the gathering of more information than could otherwise be accommodated.

In this paper, we provide a survey of current data compression techniques which are being used to reduce the amount of data in remote sensing applications. The survey aspect of this paper is far from complete, reflecting the substantial activity in this area. The purpose of the survey is more to exemplify the different approaches being taken rather than to provide an exhaustive list of the various proposed approaches. For more information on compression techniques the reader is referred to [2, 3, 4].

Compression techniques in remote sensing applications can be broadly classified into three (non distinct) categories. These are

1. Classification/Clustering
2. Lossless Compression
3. Lossy Compression

The rationale behind the classification approaches is that in a given dataset, the end user is generally interested in particular features in the data. The 'dimensionality' of these features is generally substantially less than the dimensionality of the data itself. Thus, rather than transmitting the data in its entirety, if the features are extracted on-board and transmitted this can result in a significant amount of compression. Lossless compression techniques provide compression without any loss of information. That is, the raw data can be

exactly reconstructed from the compressed data. This is used when the data, or some subset of the data, is needed in exact form. In many cases, the data such as remotely sensed images, will be viewed by a human (as opposed to a machine). In these cases, distortions which are not perceptually significant can be tolerated, and lossy compression, which entails the discarding of some of the information, can be used. The utility of this approach is closely related to the amount of distortion incurred and the importance of fidelity in the particular application. The classification approaches can be viewed as a form of lossy compression. The three approaches are not mutually exclusive. For example, one may use classification as the first step with the feature vectors being losslessly encoded.

2 Classification

If we assume an image to be composed of a small number of objects, then the most efficient form of data compression is to assign each pixel in the image to one of the objects, and then simply transmit the object labels to the ground. This idea is behind several high compression schemes which attempt to classify the pixels based on different features, and then transmit the classification map.

A technique called BLOB was introduced by Kauth et. al. [5] which uses proximity information along with spectral information for unsupervised clustering. The use of proximity information allows for greater ease in the classification of boundary pixel values, which otherwise could be classified to a set different from the adjacent regions. BLOB would be most useful in situations where objects have relatively well defined boundaries.

Another object oriented unsupervised classification scheme is described in [6]. They use what they call the *path hypothesis* for object classification. The path hypothesis assumes spatial contiguity, and spectral nearness for different pixels belonging to the same object. The spectral features of the different objects are then extracted and used to classify the object. They report an increase in classification accuracy along with a decrease in the amount of data required.

Hilbert [7] proposed a more general clustering algorithm. He proposed dividing the data into blocks, and then clustering them using an unsupervised procedure. The cluster centroids were then transmitted, along with a feature map describing the cluster to which each block belonged. This approach does not depend on the existence of well defined boundaries. Hilberts technique is a precursor to current day Vector Quantization algorithms which are discussed later.

A common precursor to classification is the transformation of the data using the Karhunen-Loeve Transform. The Karhunen-Loeve transform is used to linearly transform data

Supported by the NASA Lewis Research Center under grant NAG 3-806 and The Goddard Space Flight Center under grant NAG 5-1612.

into uncorrelated coordinates. This then makes the classification task easier, as the coordinates can be clustered in a multi-dimensional space, and then classified based on their location in this space. The rows of the KL transform matrix are the eigenvectors of the correlation matrix of the data. These vectors will often be related to physical parameters. For example in [8] the first and second eigenvectors correspond to the response of the dominant surface covers. Chen and Landgrebe [8] also show that it is sufficient to send only clipped (hard limited to $+ - 1$) eigen functions along with only a fraction of the coefficients to obtain significant classification accuracy. They therefore propose the use of this scheme aboard the HIRIS instrument.

3 Lossless Compression

Lossless compression, as the name implies, consists of reduction in the amount of data without sacrificing the fidelity of the data. The earliest known lossless compression technique of the technological age is probably the Morse code. In the Morse code, letters that occur often such as E are coded using short symbols, while letters that occur relatively infrequently such as Z, are represented by long symbols (a single dot for E and dash dash dot dot for Z). This idea (albeit in more sophisticated form) is at the heart of most lossless compression schemes. In 1948 Claude Shannon defined the amount of information contained in the event X as $\log_a \frac{1}{P(X)}$ [9], where $P(X)$ is the probability of the event X and a is the base of the logarithm. If $a = 2$ the unit of information is bits. If we define X^n to be the sequence of observations $(X_0, X_1, \dots, X_{n-1})$, then the entropy of the source generating the sequence is defined as

$$H(S) = \lim_{n \rightarrow \infty} G_n$$

where

$$G_n = \sum P(X^n) \log_2 \frac{1}{P(X^n)}$$

Shannon [9] showed that the minimum average rate at which the output of the source S can be encoded is $H(S)$ bits/symbol. If the source outputs $\{X_i\}$ are independent then the expression for entropy reduces to

$$H(S) = G_1 = \sum P(X_i) \log_2 \frac{1}{P(X_i)}$$

Given a sequence of independent observations, Huffman [10] developed an algorithm which provides a variable length code which gives an average coding rate R , where $H(S) \leq R \leq H(S) + 1$. The algorithm assigns shorter codewords to more probable symbols and longer codewords

to less probable symbols *a la* Morse. Another technique which operates on sequences rather than individual letters is *Arithmetic Coding*. The Arithmetic coding algorithm guarantees an average coding rate R where $H(S) \leq R \leq H(S) + \frac{2}{n}$, n being the length of the sequence. If the statistics of the sequence change with time, these techniques will suffer some degradation. To combat this several adaptive coding techniques have been proposed including dynamic Huffman coding [11], adaptive arithmetic coding [12] and the Rice algorithm [13]. The Rice algorithm has been shown to be optimal under some widely available conditions [14], and has been implemented in a VLSI chip which can process 20 M-Bytes per second [15].

If the observations are not independent then the code designed using the first order probabilities $P(X_i)$ is only guaranteed to be within one bit of G_1 which may be substantially greater than $H(S)$. Because of this fact lossless compression consists of two steps; decorrelation, and coding. The first step can be seen as an 'entropy reduction' step in which the redundancy or correlation of the data is removed (reduced). This results in another sequence which has a first order entropy G_1 which can be significantly lower than the first order entropy of the original sequence. Now if a variable length code is designed using the first order probabilities of the decorrelated data, this will result in a lower rate/higher compression. Consider for example the following sequence

12345432123212345432345

estimating the first order probabilities from the sequence we obtain

$$P[1] = P[5] = \frac{3}{23}; P[2] = P[3] = \frac{6}{23}; P[4] = \frac{5}{23}$$

which gives a value for G_1 of 2.25 bits/sample. It is obvious from looking at the data that it possesses some definite structure. Some of this structure can be removed by storing consecutive differences. The original data can be reconstructed (without loss) by simple addition. The difference data is

11111 - 1 - 1 - 1 - 111 - 1 - 11111 - 1 - 1 - 1111

The difference can be represented using a binary alphabet, so the coding rate can immediately be lowered to one bit/sample. To see what the value of G_1 is we first compute the first order probabilities as $P[1] = \frac{14}{23}$, $P[-1] = \frac{9}{23}$, which gives an entropy of .96 bits/sample. In this particular case the gain of 0.04 bits per sample may not be worth the additional complexity required for a variable length code. Notice that in this case the compression was obtained mainly



due to the decorrelation step. Because of this, research in lossless compression is focusing more and more on the development of better decorrelation algorithms. An idea of how much decorrelation gain is available can be obtained by looking at the conditional entropy.

In the example given above, the data was one dimensional so the *prediction* used to generate the difference or *residual* data was also one dimensional. In the case of remotely sensed images, the data is generally three dimensional: two spatial dimensions and a spectral dimension. In these cases, it would seem reasonable to use prediction based on all three dimensions. Chen et. al. [16] compute the theoretical advantages to be gained from using prediction based on all three dimensions. They show that while there is some advantage to be gained from using more than one dimensional prediction, the increase in compression is small. However, if the increase in complexity of going from one to two or three dimensions is acceptable (and it can be argued that the increase in complexity is minimal), it would seem reasonable to use multi-dimensional prediction to decorrelate the data.

A somewhat different approach is adopted by Memon et. al. [17, 18]. They reason that in an image the correlations may be maximum in the vertical, horizontal, or diagonal direction depending on the object being imaged. Therefore, one should use whichever pixel gives the most decorrelation for prediction. They therefore develop the concept of prediction (or scanning) trees for performing the decorrelation. The drawback with this approach when coding single images is that the cost of encoding the prediction tree may eat up any savings due to better decorrelation. In the case of multi-spectral images, because the same prediction tree can be used to code a large number of bands, the relative cost of encoding the prediction tree is small enough not to overwhelm the savings obtained via this approach [19].

In all that we have discussed above, we have taken a rather general view of the lossless compression problem. When faced with a specific problem, one can often come up with a simpler more efficient solution. Consider the problem of encoding the output of a spectrometer. A general algorithm such as the Rice algorithm will do a nice job of encoding the output of the spectrometer. However, given the very special structure of the data (the data looks like a noisy decaying exponential) one can come up with simpler techniques as in [20] which are simpler and give better performance. Similarly Stearns et. al. [21] develop a lossless compression scheme tuned to the peculiarities of seismic data. When using application specific algorithms, the user should be aware of the fact that if the data sequence deviates from the assumed structure, this may result in performance loss.

Finally, lossless coding can be used in conjunction with other techniques. Several schemes in the literature use lossless compression as the second stage, where the first stage is feature extraction or lossy compression [23, 23, 24].

4 Lossy Compression

In many applications, loss of information which is not perceptually significant can be easily tolerated. In fact in certain cases, such as processing of SAR data [25], the 'information' lost may actually be the noise. In these cases, it makes sense to use lossy compression techniques which provide much higher compression than the lossless techniques. However, before we extoll the virtues of various lossy compression techniques, one should keep in mind the importance of carefully picking the distortion measure. Most of the compression schemes described here use the mean squared error (or some variant) as the distortion measure. The mean squared error is defined as

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

where x_i is the original data value and \hat{x}_i is the *reconstructed* (compressed and then decompressed) value. Note that this is an *average* measure therefore it will spread out the error effects at any one location. Under this measure, a large error in one sample value with no or little error in the other $N-1$ sample values may be equivalent to small errors in all N sample values. If the application requires that each sample value be represented within some tolerance, then the MSE is probably not the distortion measure that should be used.

4.1 Quantization

The heart (and sometimes the totality) of most lossy compression schemes is the quantization process. Quantization is a many to one mapping from a possibly infinite set to a finite set. The input to the quantizer can be a scalar, in which case the quantizer is called a scalar quantizer, or a vector in which case the quantizer is called a vector quantizer (VQ). The scalar quantizer is simply a concatenation of an A/D and a D/A. A simple A/D is shown in Figure 1. Assuming $\Delta = 1$, in this A/D if the input falls in the range $(0, 1]$, the output is the codeword 10, if the input falls in the range $(-\infty, -1]$ the output is the codeword 00, and so on. The D/A takes the codeword produced by the A/D and generates a real value corresponding to the interval represented by the codeword. In our simple example if the codeword 00 is received the D/A will put out a value of -1.5. The input/output map for this quantizer (A/D-D/A combination) is shown in Figure 2. Figures 1 and 2 describe a two bit *uniform* quantizer. If the stepsize Δ is not constant for the different intervals, the quantizer is called a *non-uniform* quantizer. Given information about the statistics of the input signal, Max [26] and Lloyd [27] have developed algorithms for the design of optimum uniform and non-uniform quantizers for memoryless sources. Kwok and Johnson [28] use a two bit quantizer designed for Gaussian data to code SAR data from the Magellan mission. The SAR data is originally at 8-bit resolution, so the

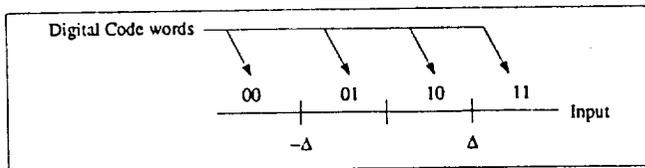


Figure 1. A two bit A/D

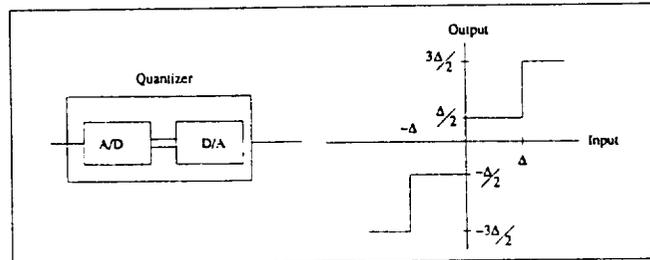


Figure 2. A two bit uniform quantizer

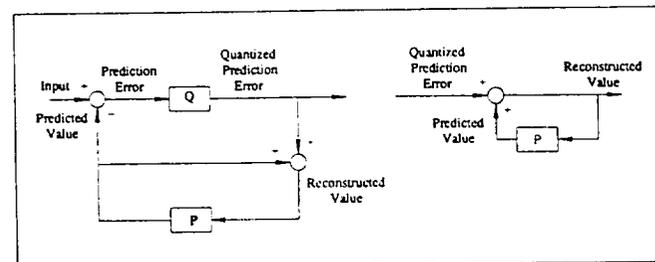


Figure 3. DPCM structure

compression ratio is 4:1. To accommodate the rather large dynamic range of the SAR data, the quantizer is adapted on a block by block basis, using the average signal magnitude. The signal magnitudes in a block are used to compute a threshold value which is used in place of Δ in Figure 1. The output of the D/A are the optimum values for a Gaussian input with variance of one multiplied by the computed threshold.

The SIR-C [1] uses 8 bit uniform quantization followed by a feature which allows it to reduce the number of bits per sample to facilitate the acquisition of more samples. Data compression thus allows the acquisition of more data at the cost of reduced resolution.

In some cases it might be more efficient to quantize some function of the data rather than the data itself. Dubois et. al. [29] compress the output of an imaging radar polarimeter by first obtaining the Stokes matrix from the scattering matrices. Four Stokes matrices from contiguous pixels are added to form one four-look Stokes matrix. The elements of the four-look Stokes matrix are then quantized. The advantage to this approach is that the elements of the Stokes matrix have certain well defined properties which can be used in the quantization process of the Stokes matrix.

4.2 DPCM

The relationship between the variance of the input to the quantizer and the *MSE* can be given by the following relationship,

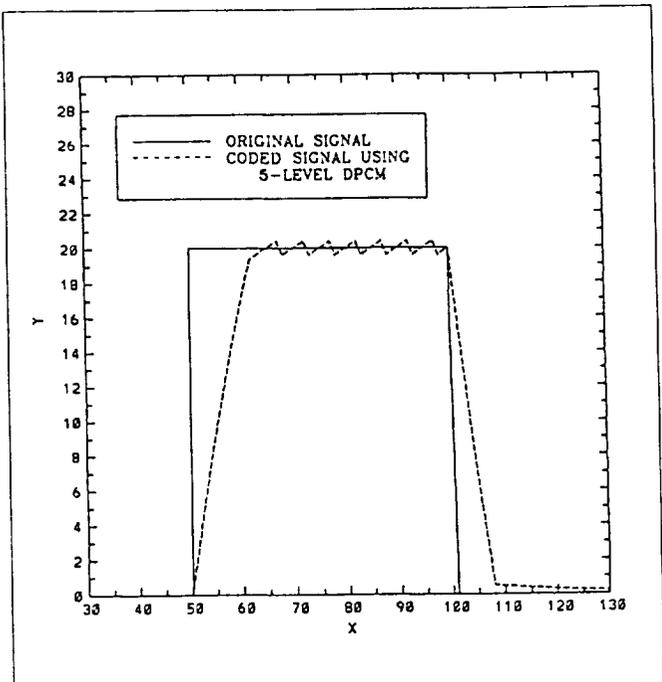


Figure 4. DPCM coding of a one dimensional edge

$$MSE = \epsilon_*^2 2^{-2R} \sigma_x^2$$

where ϵ_*^2 depends on the input probability density function, and R is the number of bits/sample. As can be seen from this expression, the *MSE* is proportional to the input signal variance. Therefore, if we could reduce the input signal variance this would lead to a reduction in the *MSE*. (It should be noted that the operations to remove the redundancy could also change the input pdf which may diminish the benefits of a reduced variance.) This is the motivation for a class of lossy compression schemes known as Differential Pulse Code Modulation (DPCM) schemes. DPCM schemes remove redundancy in the source sequence by using the correlation in the source sequence to predict ahead. The predicted value is removed from the signal at the transmitter and reintroduced at the receiver. The prediction error, which has a smaller variance than the input signal is then quantized and transmitted to the receiver. A block diagram of a DPCM system is shown in Figure 3. This technique is used in the coding of the SPOT satellite's panchromatic band.

While DPCM coding performs well in quasi-stationary regions of an image, it does a poor job in edge regions. The reason for this is that the prediction in DPCM uses the previous *reconstructed* pixels. In an edge region, the prediction error is quite large. Therefore, the input to the quantizer lands in one of the outer regions ($(-\infty, -1], [1, \infty)$ in our example). The quantization error can therefore be quite large. This is fed back via the prediction process into the coding of the next pixel, and so on causing a smearing of the edges. This process is demonstrated on a one-dimensional 'edge' in Figure 4. This problem can be overcome by using recursive-



Figure 5. Original aerial view of Omaha



Figure 6. DPCM coded Omaha image at 1.4 bpp

ly indexed quantization [30, 31] which avoids the large quantization error problem by operating the quantizer in two different modes. Whenever the input to the quantizer falls in the external regions, the quantizer switches into a recursive mode, and the quantization error is requantized until the error falls within some predetermined tolerance. This approach not only prevents large quantization errors from propagating through the coded sequence, it also guarantees that the error *per pixel* will be less than a pre-determined value. To show how well this scheme works, we code the aerial view of Omaha shown in Figure 5. The compressed (and decompressed) image coded using the DPCM scheme described above at a rate of 1.4 bits per pixel is shown in Figure 6. Note that while there is an overall increase in 'blurriness' the distortion introduced does not blur the edges.

While the DPCM structure removes substantial amounts of the redundancy from the data stream, it should be remembered that the prediction process in the DPCM structure is linear, and can therefore remove only those redundancies which are expressed as linear processes. For example, a slowly varying sequence 1 2 3 4 5 4 3 3 4 5 6 7 7 7 6 has redundancies that can be modeled by a linear process. However, we can easily come up with sequences that have redundancies that can not be characterized by a linear process such as 4 24 15 19 4 24 15 19 This fact has been used by some to improve the data compression by making use of this redundancy for code selection [32], and by others for providing error protection [33].

4.3 Vector Quantization

Until now we have been talking about quantization as a scalar process, however, the basic idea of quantization can easily be extended to the vector case. Scalar quantization can be viewed as a partition of the real number line, with the

A/D doing the partitioning, and the D/A providing a representative value for each partition. Similarly, vector quantization can be seen as a partitioning of multidimensional space. While conceptually the problems of scalar and vector quantization approaches are very similar, the practical problem of designing vector quantizers is significantly more difficult. Two somewhat different approaches have been taken towards the design of vector quantizers. The first is a clustering approach similar to the Hilbert technique [7]. In this approach [34], a training sequence is used to identify the regions in multi-dimensional space where the data seems to cluster. The quantizer outputs are the centroids of these clusters, and the partitions are the nearest neighbor partitions of these centroids. An example of a two dimensional vector quantizer is shown in Figure 7. The VQ in Figure 7 contains 4 output levels, or codewords. Thus the size of each codeword is two bits. But each output level corresponds to the coding of two input samples, therefore, the number of bits per sample is one. In general, given the dimension of the vector d and the number of bits per sample R , the size of the vector codebook is 2^{Rd} . Notice, that this means an exponential increase in the size of the codebook with dimensionality and rate. For example, given $d = 12$ and $R = 2$, the size of the codebook would be $2^{24} = 16777216!$ This represents an enormous expense in storage and computing resources. Thus the rate-dimension product provides a limitation on the clustered VQ designs. Fortunately, a lot can be done at low rate-dimension products. For more moderate rate-dimension products a number of somewhat more structured VQ algorithms have been developed [35]. Chang et. al. [25] report the use of a tree-structured VQ on Seasat SAR imagery with favorable results. As the codebook of the VQ is obtained by training, it is important that the data in the training set be representative of the data in the test set. If this is not the case, there can be

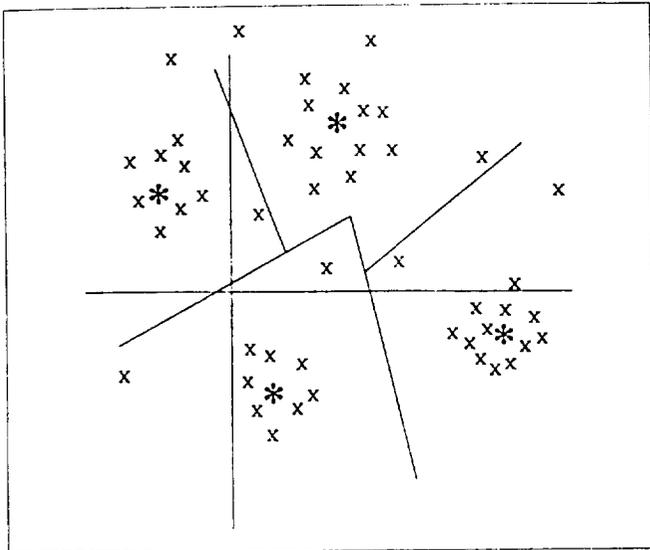
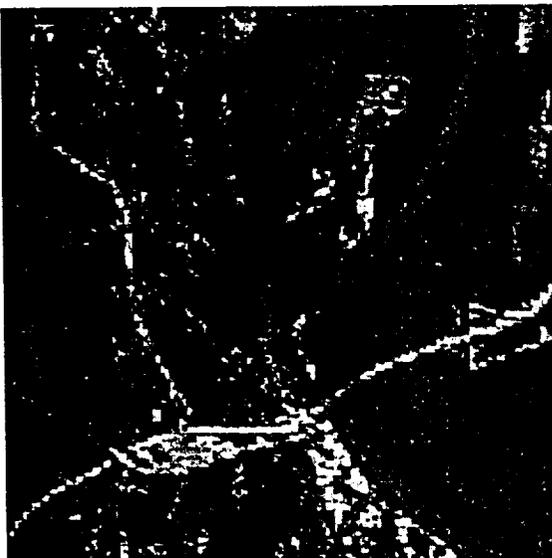


Figure 7. A four level, two dimensional vector quantizer

significant degradation in the data that is not typical of the training set [25]. The Omaha image of Figure 5 is coded using a clustered VQ at 0.5 bits per pixel. The result is shown in Figure 8. The VQ dimension was 16 (4X4 blocks) and this is evident from the coded image in Figure 8, where there is a noticeable amount of blockiness. The blocks that lie on the edges of objects in the image clearly distort the edges. The VQ codebook was obtained using another aerial image. We can improve the performance of this algorithm by increasing the rate and/or by generating the codebook from an image which more closely resembles the image being coded. In Figure 9 we have the Omaha image coded at 1 bit per pixel using a codebook generated using the Omaha image itself. There is substantial improvement in the quality, though there is still some distortion in the lower quarter of

Figure 8. VQ coded Omaha image at 0.5 bpp



the picture. It should be noted that the use of the image to generate the codebook is generally not realistic.

Vector Quantization is also used by Gupta and Gersho for the coding of Landsat TM images [36]. They use a vector DPCM system with vector quantization in the spatial domain, and predictive encoding in the spectral domain. A variation of predictive VQ is also used by Giusto [37] for the compression of multispectral images.

The rate-dimension product constraint on vector quantizers can be lifted by making the vector quantizer more and more structured. Of course, as the VQ acquires more and more structure of its own, it is less and less responsive to structure in the data. The most structured vector quantizers are those based on a multi-dimensional lattice [38]. While these quantizers do an excellent job of quantization, they cannot at the same time perform the redundancy removal operation performed by the clustered VQs. They therefore have to be used in conjunction with other techniques to provide compression [39, 40].

4.4 Transform Coding

Most of the techniques we have talked about operate in the data domain, i.e. without any transformation. There is a large class of compression techniques that operate on a transformed version of the data. They are called transform coding techniques. The idea behind transform coding is to transform the data in such a way as to compact most of the energy (and information) into a few coefficients. These coefficients can then be coded, while other coefficients can be discarded thereby achieving data compression. The most efficient transform from the compaction point of view is the Karhunen-Loeve [2] transform. However, the Karhunen-Loeve transform is data dependent which makes it impractical for most compression applications. The best alternative to the

Figure 9. VQ coded Omaha image at 1.0 bpp



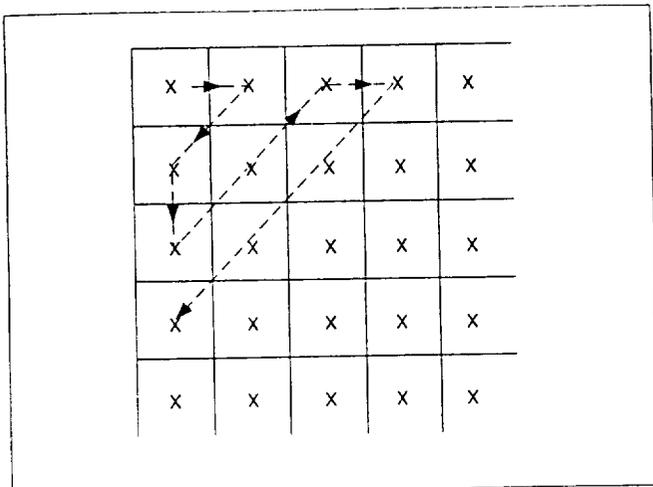
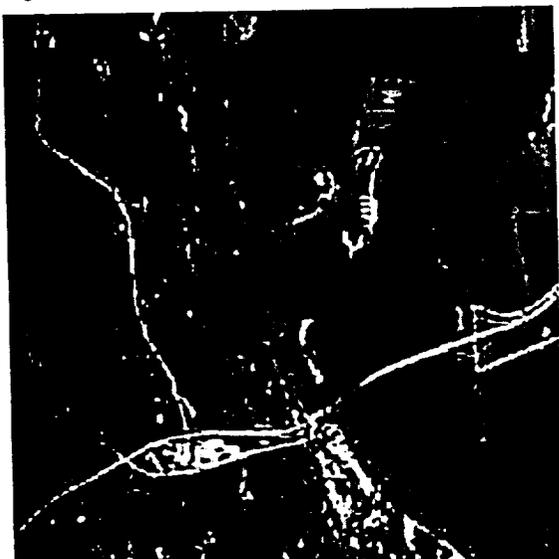


Figure 10. Zig zag ordering

Karhunen Loeve transform is the Discrete Cosine Transform (DCT). This is a real, separable, unitary transform that is the basis for an image compression standard [41]. Because of its popularity in image compression various fast algorithms have been proposed for its implementation [42, 43].

The ESA Huygens Titan Probe to be launched by the Cassini Orbiter will use the DCT for compressing the image data acquired during its descent through Titan's atmosphere. The images of size 256X256 will be divided into 8X8 blocks. These blocks will be transformed and the transform coefficients reordered using the zigzag ordering shown in Figure 10. The ordered coefficients will then be blocked into substrings of four coefficients each. Substrings with all coefficient values below a specified threshold will be deleted while the remainder will be quantized using scalar quantizers. Details can be found in [44, 45].

Figure 11. DCT coded Omaha image at 0.5 bpp.



To see the artifacts introduced by DCT coding we have coded the Omaha image at 0.5 bits per pixel and 1 bit per pixel as shown in Figures 11 and 12. Note the substantial block artifacts in Figure 11 which have been reduced to a large extent in Figure 12. However even in Figure 12 one can see significant distortion in edge regions.

An adaptive version of DCT was also considered by Chang et. al. [25] for the compression of Seasat SAR imagery. They compare the DCT technique with a VQ technique and decide in favor of the VQ technique based on complexity issues. With the wide acceptance of the DCT as an image compression standard, the complexity issue may no longer be relevant, as more and more manufacturers are bringing hardware implementations of the DCT to the market.

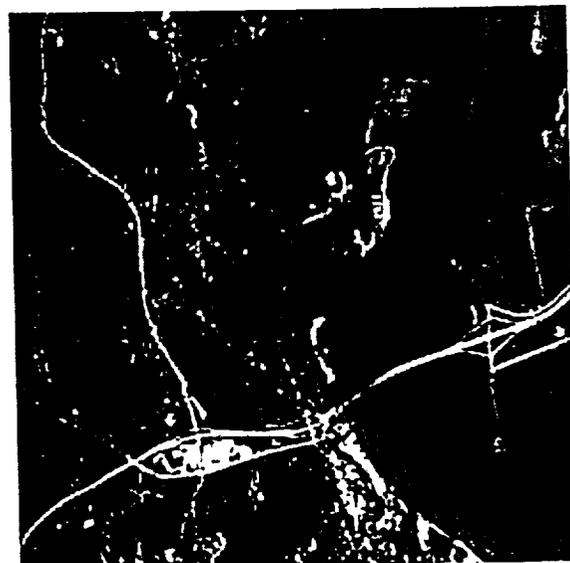
5 Conclusions

As can be seen from this discussion, there is a substantial amount of on-going activity in the area of data compression for remote sensing applications. This will only increase as there is more and more need for data compression. However, there are several areas of research which have not been addressed in any significant way.

There is a need for the development of better distortion measures which can be then used to develop more sophisticated compression algorithms. It is possible that rather than a single distortion measure, a set of distortion measures will be needed for different applications. The development of such measures, and algorithms utilizing these measures, require close cooperation between data compression specialists and the scientists and engineers who are the end-users of the data obtained through remote sensing.

The multi-dimensional (spatial and spectral) nature of the data has not really been thoroughly explored (except in the

Figure 12. DCT coded Omaha image at 1.0 bpp.



classification approaches). With the development and deployment of high spectral resolution instruments, this particular aspect of remotely sensed data will become more important. Compression schemes which take advantage of this fact need to be developed. An analogy could be drawn with the development of compression algorithms for video as opposed to still images. However, the algorithms developed for video cannot be directly applied to high spectral resolution image data sets, as the differences that occur between frames of a video sequence are not the same as the differences that occur between different spectral images. It would seem that VQ approaches such as [36] would provide possible solutions. The rate-dimension constraints in clustering VQ could be avoided by the use of Lattice VQ techniques. Another approach described in [46] is to use a two step strategy, in which the first step is used to model the data in the spectral direction. The resulting models are then treated as a vector image for compression in the spatial directions. Beyond this, however, there is a need for the development of three dimensional approaches, both to model the data, and develop compression algorithms.

6 Acknowledgement

The author would like to thank the reviewers for their careful reading of the manuscript and for their suggestions. The DCT coded images were obtained using software from the Independent JPEG foundation. The VQ images were provided by Dr. Nasir Memon.

References

- [1] B.L. Huneycutt. Spaceborne Imaging Radar-C Instrument. *IEEE Transactions on Geoscience and Remote Sensing*, pages 164-169, March 1989.
- [2] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, 1984.
- [3] M. Rabbani and P.W Jones. *Digital Image Compression Techniques*, volume TT7 of *Tutorial Texts Series*. SPIE Optical Engineering Press, 1991.
- [4] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.
- [5] R.J. Kauth, A.P. Pentland, and G.S. Thomas. BLOB: An Unsupervised Clustering Approach to Spatial Pre-Processing of MSS Imagery. In *Proceedings Int Symp Remote Sensing Environ*, pages 1309-1317, 1977.
- [6] H. Ghassemian and P.A. Landgrebe. Object Oriented Feature Extraction Method for Image Data Compaction. *IEEE Control Systems Magazine*, pages 42-48, June 1988.
- [7] E.E. Hilbert. Cluster Compression Algorithm - a Joint Clustering Data Compression Concept. Technical Report JPL Publication 77-43, NASA, 1977.
- [8] C-C.T. Chen and D.A. Landgrebe. A Spectral Design System for the HIRIS/MODIS Era. *IEEE Transactions on Geoscience and Remote Sensing*, 27:681-686, November 1989.
- [9] C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:398-403, July 1948.
- [10] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098-1101, 1951.
- [11] R.G. Gallager. Variations on a theme by Huffman. *IEEE Transactions on Information Theory*, IT-24(6):668-674, November 1978.
- [12] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons Inc., 1991.
- [13] R.F. Rice. Some practical universal noiseless coding techniques. Technical Report JPL Publication 79-22, NASA, 1979.
- [14] P-S. Yeh, R.F. Rice, and W. Miller. On the Optimality of Code Options for a Universal Noiseless Coder. Technical Report JPL Publication 91-2, NASA, February 1991.
- [15] J. Venbrux and N. Liu. A Very High, Speed Lossless Compression/Decompression Chip Set. In *Data Compression Conference*, 1991.
- [16] T.M. Chen, D.H. Staelin, and R.R. Arps. Information Content Analysis of Landsat Image Data for Compression. *IEEE Transactions on Geoscience and Remote Sensing*, GE-25:499-501, July 1987.
- [17] N.D. Memon, S.S. Magliveras, and K. Sayood. Prediction trees and lossless image compression - an extended abstract. In Reif J.H. Storer J.A., editor, *Proceedings of the Data Compression Conference*, pages 83-92. IEEE Computer Society Press, 1991.
- [18] N.D. Memon, K. Sayood, and S.S. Magliveras. Lossless Image Compression Using Efficient Scans. In *26th Annual Conference on Information Sciences and Systems*, March 1992.
- [19] N.D. Memon. *Image Compression Using Efficient Scanning Patterns*. PhD thesis, University of Nebraska - Lincoln, August 1992.
- [20] K. Sayood and M.C. Rost. A Robust Compression Scheme for Low Bit Rate Telemetry - Test Results with Lunar Data. In *Scientific Data Compression Workshop*, pages 237-250, May 1988.
- [21] S.D. Stearns, L. Tan, and N. Magotra. A Technique for Lossless Compression of Seismic Data. In *Proceedings IGARSS '92*, pages 681-683. IEEE, 1992.
- [22] I. Asanuma, Y. Kuroda, K. Yoshifumi, G. Kai, K. Muneyama, and S.-I. Isshii. Development of Data Compression and Transfer System. In *Oceans '89. Part 3: Navigation; Remote Sensing; Underwater Vehicles/Exploration*. IEEE, 1989.



[23] M.P. Cough, R. German, J.A. Thompson, and L.J.C. Wooliscroft. Simulation of Satellite Geophysical Data. *International Journal of Remote Sensing*, 8:1219-1227, 1987.

[24] B.R. Epstein, R. Hingorami, J.M. Shapiro, and M. Czigler. Multispectral Image Compression by Wavelet/Karhunen-Loeve Transformation. In *Proceedings IGARSS '92*, pages 672-674, 1992.

[25] C.Y. Chang, R. Kwok, and J.C. Curlander. Spatial Compression of Seasat SAR Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, pages 673-685, September 1988.

[26] J. Max. Quantizing for Minimum Distortion. *IRE Transactions on Information Theory*, IT-6:7-12, March 1960.

[27] S.P. Lloyd. Least Squared Quantization in PCM. *IEEE Transactions on Information Theory*, IT-28:129-137, March 1982.

[28] R. Kwok and W.T. Johnson. Block Adaptive Quantization of Magellan SAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, pages 375-383, July 1989.

[29] P.C. Dubois and L. Norikane. Data Volume Reduction for Imaging Radar Polarimetry. In *IGARSS '87*, pages 691-696, May 1987.

[30] K. Sayood and K. Anderson. A differential lossless image compression algorithm. *IEEE Transactions on Signal Processing*, 40(1):236-241, January 1992.

[31] M.C. Rost and K. Sayood. An Edge Preserving Differential Image Coding Scheme. *IEEE Transactions on Image Processing*, 1:250-256, April 1992.

[32] M.J. Shalkhauser and W.A. Whyte Jr. Digital CODEC for Real Time Signal Processing at 1.8 bpp. In *Global Telecommunication Conference*, 1989.

[33] K. Sayood and J.C. Borkenhagen. Use of Residual Redundancy in the Design of Joint Source/Channel Coders. *IEEE Transactions on Communications*, 39:838-846, March 1991.

[34] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, COM-28:84-95, Jan. 1980.

[35] H. Abut, editor. *Vector Quantization*. IEEE Press, 1990.

[36] Smita Gupta and Allen Gersho. Feature Predictive Vector Quantization of Multispectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 30:491-501, May 1992.

[37] D.D. Giusto. On the Compression of Multispectral Images. In *Proceedings IGARSS '90*, pages 1651-1654. IEEE, 1990.

[38] J.D. Gibson and K. Sayood. Lattice Vector Quantization. In *Advances in Electronics and Electron Physics*, volume 72, pages 259-330. Academic Press, 1988.

[39] K. Sayood, J.D. Gibson, and M.C. Rost. An Algorithm for Uniform Vector Quantizer Design. *IEEE Transac-*

Biography

Khalid Sayood (S'78-M'79-S'80-M'82) was born in Pakistan in 1956. He received his undergraduate education at the Middle East Technical University, Ankara, Turkey, and the University of Rochester, Rochester, NY. He received the B.S. and M.S. degrees from the University of Rochester, and the Ph.D. degree from Texas A&M University College Station, TX, in 1977, 1979, and 1982 respectively, all in electrical engineering.

He joined the Department of Electrical Engineering at the University of Nebraska-Lincoln, in 1982, where he is currently serving as a Professor. His current research interests include data compression, joint source/channel coding, communication networks, and biomedical applications.

Dr. Sayood is a member of Eta Kappa Nu and Sigma Xi.



tions on Information Theory, IT-30:805-814, November 1984.

[40] T.R. Fischer. A Pyramid Vector Quantizer. *IEEE Transactions on Information Theory*, IT-32:568-583, July 1986.

[41] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):31-44, April 1991.

[42] K.R. Rao and P. Yip. *Discrete Cosine Transform - Algorithms, Advantages, Applications*. Academic Press, 1990.

[43] V. Cappelini and E. Del Re. Some Efficient Methods for Image Data Compression. In A. Luque, A.R. Figueiras Vidal, and J.M.R. Delgado, editors, *Proceedings of MELECON '85*, pages 473-475. IEEE, 1985.

[44] P. Ruffer, F. Rabe, and F. Gliem. A DCT Image Data Processor for use on the Huygens Titan Probe. In *Proceedings IGARSS '92*, pages 678-680. IEEE, 1992.

[45] H. Michalik, P. Ruffer, and F. Gliem. Versatile On-line Image Data Compression for Spaceborne Applications. In *Proceedings IGARSS '91*, pages 1443-1447. IEEE, 1991.

[46] C. Mailhes, P. Vermande, and F. Castanie. Spectral Image Compression. *Journal of Optics*, 21:121-132, 1990.