MEMORANDUM October 1, 1985

To: Robert Duquet

From: Jeffrey A. Brooks

Subject: Coding and Compression of Radio Images

In the summer of 1985 I was asked to assess techniques for radio image coding and compression which might be applicable to inter-system transfer of VLA images. This memo presents a summary of my findings.

INTRODUCTION

With the growth of supercomputer use in VLA data reduction, questions of host/remote processor communication have become increasingly important. The ability of an astronomer to review at a remote node the intermediate steps of host processing is often critical in guiding the data reduction process. Unfortunately, the digital images which typically comprise the intermediate results of data reduction operations have exceptionally large bandwidth requirements. This memo reviews a range of techniques for digital image coding and compression which may be of use in transmission of radio images across communications links.

Image coding can be considered as a process which converts a stored digital image to a sequence of binary values from which the original image can be reconstructed, with the possible addition of some acceptable amount of distortion. The transmission of this bit sequence itself is assumed to be performed by appropriate communications hardware and/or software, with error checking provided. Although error-correcting coding techniques will not be considered here, methods for minimizing the effects of errors will be presented.

All images considered here will be assumed to be rectangular arrays of integer or floating-point values. All of the techniques presented here can be generalized where appropriate to provide for the coding of spectral lines cubes and hypercubes.

CODING TECHNIQUES FOR RADIO IMAGES

Coding techniques can be broadly divided into two classes: **point coding** or **causal** methods, which encode each image pixel in some specified pixel order, and **transform coding** or **noncausal** methods, which encode entire areas of the image at a time. Coding techniques can also be classified by the type of image statistics they use, and by the extent to which they are **adaptive**; that is, the extent to which the coding technique "adapts" to the local statistics of the image being coded.

Causal methods treat an image as the output of a statistical source of pixels, with each pixel "determined" by zero or more previous pixels, with the addition of noise in the process. Such techniques require a good predictor of the next pixel value given zero or more previous pixel values. Prediction can take place in one, two, or three dimensions. Non-causal methods, on the other hand, transform an image into a set of uncorrelated values which concentrate most of the image information in a small number of transform components. These uncorrelated values are not associated with any particular pixel.

Causal methods can be classified by the type of image statistics (local or global) that they use, and the order of these statistics. For example, some schemes require accurate knowledge of the pixel value distribution or covariance structure. Theoretically, the statistical values to be used in coding are the statistics of the "image source", but in practice the statistics are either those of the specific image to be coded, or are estimated from a priori information or from a portion of the image [1].

Non-adaptive coding schemes process each region of the image in exactly the same fashion. Adaptive schemes vary processing parameters to match the characteristics of the image region being coded. For example, the number of bits used to code each pixel or the number of transform components used may vary in the course of coding the image. One-pass coding techniques may acquire during coding additional information about the image which alters the way the remainder of the image is to be coded [2]

Simple Point Coding

In **point coding**, each pixel of the image is coded in order (typically by row and column) by assigning each pixel value a particular bit sequence to be transmitted. This bit sequence may be derived immediately from the pixel value or may be found by a table look-up in a **codebook**.

The simplest point coding technique is that which transmits for each pixel the binary representation of the pixel value in the original units. This binary representation may be either fixed-length, using the maximum number of bits needed to encode any pixel value in the image, or may vary to accommodate the number of significant bits in a region of the image. Such adaptive schemes subdivide the image into subregions and transmit the coded pixel values in each region preceded by a count field indicating the number of bits to be used in encoding the pixel values in that subregion.

Simple point coding is the scheme most often used for mass storage of images; it assumes no knowledge of the image pixel values other than the pixel value range to be covered, and allows sequential processing of the entire image. This method is in fact the most efficient coding technique for stationary image sources whose pixel values are uniformly distributed. (A stationary image source is one whose image statistics do not vary across the image. A white noise source, for example, has stationary statistics.) However, radio images are often coded much more efficiently by methods which utilize the redundancy contained in the image. Such techniques are described below.

Huffman Coding

Huffman coding, developed in the early 1950's [3], attempts to improve on simple point coding by utilizing knowledge of the image histogram. The most common pixel values are assigned shorter codes, in such a way that the variable length codes can be uniquely decoded. Huffman coding is optimal for stationary image sources; if the pixel values are uniformly distributed it reduces to simple fixed-length pixel coding. However, the codebook must be sent along with the transmitted bit sequence, and a good knowledge of the image histogram is necessary if the Huffman coding scheme is to work correctly. If the wrong image statistics are used, Huffman coding can perform more poorly than simple fixed-length point coding.

The construction of a Huffman code is based on the probability of occurrence of each pixel value, which is either estimated a priori or derived from the pixel value histogram of the image to be coded. The best-known published algorithm for the construction of Huffman codes [4] runs in time proportional to N**2, where N is the total number of pixel values to be assigned Huffman codes. However, I have developed a Huffman algorithm which runs in time proportional to N if a sorted histogram of the image is available, or proportional to (N log N) if the histogram must be sorted. Note that the histogram used for Huffman code construction must be exact, in the sense of giving a probability of occurrence for each pixel value to be coded rather than a probability of occurrence for a bin containing several pixel values.

The histogram information needed for Huffman coding need not necessarily be computed from the image itself. If a good estimator of the histogram can be found, or if the sequence to be coded can be well modelled, some combination of model and image-derived estimator may be used to produce the Huffman code, or select one from a set of "standard" source probability histograms. In this case, however, this same information must be passed to the receiver -- either the information needed to reconstruct the Huffman code used, or an indication of which standard code is to be used.

The transmission of the Huffman code book is a perplexing problem, since for images involving a skewed histogram one may need to send more bits for a codebook entry than are used to encode the single instance of the corresponding pixel value. Fred Schwab [5] discussed a technique called **Huffman shift coding** is which a smaller coding table is used, at the expense of transmitting occasional "shift-up" and "shift-down" codewords in the encoded bit sequence. A simpler and apparently more efficient method is that proposed by Hankamer [6] in which all pixel values with a higher probability than some fixed value are Huffman coded, along with a "dummy pixel value" representing the combined probability of occurrence of the remaining pixel values. Pixels are coded by transmitting the Huffman code for the higher-probability pixel values, or the Huffman code for the dummy pixel value followed by the full precision binary representation of the pixel value itself in the case of the lower-probability pixel values. This variant is related to one proposed by Murakami [7], and appears to reduce the codebook size dramatically with a near trivial increase in average codeword length.

Several authors have studied the application of Huffman coding to text storage and related problems [8] [9]. These papers also suggest several efficient schemes for transmitting the Huffman codebook.

Gallager [10] discusses the redundancy inherent in the Huffman coding process, and also suggests an adaptive Huffman coding technique which does not require the advance preparation of a histogram, instead using accumulated statistics acquired during the encoding of the initial segment of the image to encode the next portion of the image. The receiver could duplicate this process, potentially eliminating the need to transmit the Huffman codebook at all. Johnsen [11] improves on Gallager's bounds for Huffman code redundancy.

Huffman coding is an efficient coding technique for sequences of pixels when only the pixel value histogram is known. This technique can be used alone, or in combination with other coding techniques which produce uncorrelated pixels, with Huffman coding used to code the uncorrelated pixels themselves. Unfortunately, due to the inherently variable-length nature of Huffman codes, little can be done to minimize the effect of transmission errors.

Predictive Coding

In many situations, however, we know more than just the pixel value histogram. Values of adjacent pixels in radio images are almost always positively correlated, so some function of the "preceding" pixel values in a pixel sequence can be used to predict the next pixel value. If this is done, the difference between the predicted and actual pixel values can be coded and transmitted; these differences are typically smaller and more amenable to statistical analysis than are the pixel values themselves. This is the essence of **predictive coding.**

In the simplest form of predictive coding, the first pixel in a row is transmitted, followed by the differences between adjacent pixels along the row. Each pixel value is used as a predictor of the next pixel value along the same row. This technique is known as differential pulse code modulation or **DPCM.** To achieve better results, more complicated predictors can be used; for example, one could use a linear fit to the two succeeding pixels or the mean of the three pixels above and to the left. A variety of linear and non-linear predictors have been studied [12][13][14]. The same predictor is then used in the decoder to predict the next pixel from previously decoded pixel values. The received and decoded pixel value difference is then added to the predicted value to produce the next pixel value.

The differences can be more coarsely quantized than the actual pixel values, providing a high degree of compression; this can be done while placing an upper bound on the distortion in any given pixel value. However, to prevent these distortions from accumulating in the decoder and causing instability, the predictor in both the coder and decoder should use a prediction based on the previous _transmitted_ differences. That is, the coder should use the transmitted pixel differences to reconstruct the pixel sequence as the decoder will reconstruct it, then use these reconstructed values to predict the next pixel. This prevents undesirable feedback due to the quantization errors made in quantizing and coding the pixel differences.

The effect of transmission errors can be minimized by restarting the predictor occasionally, perhaps at the start of every row for a one-dimensional predictor or after several rows for a two-dimensional predictor. This prevents the effects of most single-bit errors from propagating past the restarting point for the predictor. If variable bit allocation is used for the differences, errors can still affect the entire image; however, if a quantization level for the pixel value differences is used for the entire image, no error can propagate past the next point at which the predictor is restarted.

Recently a new prediction technique called **anisotropic nonstationary image estimation**, originally developed for image reconstruction work, has shown great promise for fast, efficient predictive image coding [15][16]. This technique does not rely on stationary image statistics and adapts well to local image information.

Transform Coding

In **transform coding**, sometimes known as **block quantization**, causal techniques are used to produce a sequence of uncorrelated values from which the original image can be reconstructed. In one-dimensional transform coding, a selected unitary transform is applied to a row of the image to produce a vector of uncorrelated values which pack the maximum energy into a small number of transform components. Multidimensional transform coding can be achieved by considering an N by M subregion of the image to be an NM element vector which is then coded using a one-dimensional transform.

The Karhunen-Loeve transform is provably optimal in the sense that it packs the most energy into the smallest number of transform components; unfortunately, the KL transform has no fast algorithm associated with it. For stationary random sequences there are a number of fast sinusoidal transforms which approach the efficiency of the KL transform. Of these, the best for images with high inter-pixel correlation is probably the discrete cosine transform (DCT), which can be computed via the fast Fourier transform. This transform uses fixed basis matrices, freeing the coding system from the overhead of computing and transmitting them, and provides a very close approximation of the KL transform's performance [12][17].

A variety of non-sinusoidal transforms have also been used, such as the Haar, Hadamard and Slant (all square-wave transforms). These transforms are typically faster than the sinusoidal transforms and require less intensive computation for certain types of data, but have poorer energy-packing capability.

In practice, to reduce time and space requirements, an image is broken up into rectangular blocks which are independently transformed and coded. For example, if a 256 by 256 image is divided into 16 by 16 subimages, storage requirements are reduced by a factor of 256 and speed is increased by a factor of 2 for a fast sinusoidal transform.

However, simply transforming an image block does not provide image compression; the transform components require as many bits to transmit completely as the original image. But since most of the image energy is contained in a small number of transform components for each block, we can allocate bits to each transform component based on the amount of energy likely to be contained in that component. If the covariance of a typical image can be modelled (by an exponential function in two dimensions, for example), one can derive estimates of the number of bits needed for each transform component in order to achieve reasonable distortion levels [12]. In such a scheme, the lowest frequency components may by coded to full precision, while the highest frequency components may be allocated no bits at all, and effectively omitted. Since the number of bits allocated to each component can be decided in advance, no additional information need be passed between coder and decoder. This method is usually called **zonal coding.**

Other schemes can improve on transform coding efficiency and stability at the cost of increased complexity. The method described above may omit some transform components entirely, based on "average case" estimates of their usefulness. One method in particular, called **threshold coding,** eliminates this problem by coding and transmitting the transform components of highest amplitude, either by selecting a fixed number of components ranked by amplitude or by selecting all components whose amplitude exceeds some threshold value. Although address information must be transmitted with the transform components, thus increasing the overhead, the adaptive nature of threshold coding makes it attractive for images whose properties cannot be predicted in advance or whose statistics change rapidly across the image.

Coding of images in subblocks can cause visible edge effects along subblock boundaries when zonal or threshold coding is used. These can be eliminated if necessary by interpolation near the boundary; on the other hand, their presence provides a visible check on the potential size of residual error near block borders.

Both zonal coding and threshold coding can provide very high performance image compression; this is particularly true if the bit allocation is made adaptive or is closely matched to the type of image being coded. Effective transform component selection and bit allocation are crucial to a good transform coding scheme.

Transform coding is subject to transmission errors in the same general way that predictive coding is. If a fixed bit allocation scheme is used, then bit errors can only affect the current block. If highly adaptive schemes involving frequently changing bit allocation are used, bit errors can corrupt the remainder of the decoded image.

Hybrid Coding

Transform and predictive coding can be combined to produce **hybrid coding** schemes. Typically, a two-dimensional image is transformed along one dimension to obtain a sequence of one-dimensional vectors. The sequence can then be coded using a predictive method such as DPCM. This type of method can easily be adapted to images with slowly varying statistics by updating the predictor parameters used.

Note that since the sinusoidal transforms such as Fourier, sine and cosine all transmit what is essentially spectral information, any method of spectral extrapolation or interpolation which can be applied to radio images can also provide better image coding performance using these transforms, since the extrapolation or interpolation scheme can be used as a predictor for the second phase of a hybrid coding scheme. Thus, deconvolution methodology and experience could be profitably applied to image coding strategies.

Feature Coding

Another non-causal coding technique which holds great potential for radio image compression is **feature coding**. Pattern recognition techniques are used to identify expected features such as gaussian sources or easily-modelled extended structures; a best fit of model parameters is made and the model with these parameters is subtracted from the image. By repeating this process, we obtain a set of model parameters to be transmitted, plus a residual map. This method obviously works best on cleaned images and has a basic philosophy similar to that of maximum entropy deconvolution methods.

The type of structures to be recognized are obviously application dependent; in the case of radio images they would probably include point sources, jets, general geometric structures such as ellipses of varying amplitude, and beam response patterns.

Unfortunately, the pattern recognition techniques involved are computationally hard, and are unlikely to be solved in the near term. However, any advances made in radio source recognition could be applied to improved image coding techniques, if only by removing structures from an image to simplify the task of encoding the underlying semi-random background and random noise.

Other Coding Methods

A variety of image coding techniques do not fall into the above categories; these methods are in general less promising than those previously described but may be useful in special circumstances.

Bit-Plane Coding: Instead of transmitting pixel values one pixel at a time, one can quantize the entire image to a fixed length binary representation and transmit the first bit of each pixel value in pixel order, followed by the second bit, and so on. Compression is achieved either by halting transmission before all bits have been send (see "Progressive Transmission" below) or by using run-length coding to compress regions with identical values in the same bit.

The high-order bits over the image will obviously be highly correlated, with correlation decreasing as one moves to the lower-order bits. Unless the image is very smooth or the number of quantization levels is small, however, bit-plane coding will typically result in only factor of two compression in the average case [18].

Run-Length Coding: Run-length coding, often used for binary (two-level) images such as facsimile, breaks an image into blocks of consecutive identical pixel values, coding and transmitting only the length of the block and the value of the pixels in the block. This technique can be very useful for sending bit-map images such as those produced by bit-plane coding, or for indicating areas of a an image (such as those areas of an image which are to be blanked). Images with a large number of quantization levels (greater than about 16) are seldom coded efficiently by this method.

Singular Value Decomposition: Fred Schwab [5] has proposed the use of the singular value decomposition (SVD) representation of a matrix for the transmission of digital images, as it provides many of the advantages of transform coding. O'Leary and Peleg have considered image compression methods using the SVD with basis vectors limited to values of -1, 0, and 1 [19]. However, as pointed out by Wilson [17], the computational expense of the SVD method and the requirement that the SVD basis vectors be recomputed and transmitted for each image to be coded make it less desirable than transform coding for most applications.

SUPPLEMENTARY COMPRESSION TECHNIQUES

A variety of pre-processing and transmission techniques can improve image coding performance, generally by reducing the amount of information to be transmitted beforehand or by transmitting data only until the user's need are satisfied. This section describes several such techniques.

Filtering and Smoothing

Both predictive and transform coding techniques work best with smooth images; the information content of such images is lower, and inter-pixel correlation is accordingly higher. If a filtered or smoothed image satisfies the needs of the user who is transmitting the image, then this decrease in transmitted information allows an increase in overall transmission efficiency.

Many simple kinds of smoothing can be performed at the same time as coding, requiring only local information and a small amount of computation. (For example, predictive coding techniques and transform coding bit allocation schemes can both serve as low-pass filters applied to the coded image.) Smoothing can also decrease the effect of noise on any resampling or requantization to be performed.

Resampling

In many cases, the user may wish to examine an entire image without requiring the full spatial resolution available. In other cases, the user may wish to examine only a portion of the image, but at full resolution. The term **resampling** is sometimes used to describe the general process of adapting spatial coverage and resolution to the user's needs.

Most resampling operations required by the user can be described in terms of a **window** in the image for which coverage is required, and a **resolution factor** by which the spatial resolution is to be (or can be) decreased. If the window is given in terms of corner pixels and the resolution factor is an integer, then resampling involves selecting every Nth pixel along both spatial axes. Power-of-two resolution changes are particularly useful for digitally stored and processed images. Resolution factor can be further generalized to provide different factors along each axis.

If the image is assumed to describe a smooth surface, then smoothing and resampling can be used to re-grid the image onto any desired coordinate system and resolution. However, these schemes typically require convolution or similar computationally intensive methods, whereas the simpler form of resampling described above does not. Smoothing can also be used to reduce the effect of noise on resampling by using the values of unsampled pixels to damp the effect of outliers at the sampled grid points.

An image should never be resampled in such a way as to increase the spatial resolution, unless additional information is being combined with the existing image. Increasing spatial resolution not only dramatically increases the amount of information contained in the image, and thus the costs of storing and transmitting the image, but also misleads the user by providing apparent detail that does not exist in the original image.

Requantization

By the time an analog signal has been converted to a digital image, it has been quantized to convert its range to a finite set of values. Occasionally, a user will need the whole range to full precision; for other uses, only a small range of pixel values may be of interest, or large quantization steps may be appropriate. The term **requantization** is used to describe the general process of adapting intensity range and resolution to the user's needs. Requantization is the analog of resampling, performed along the intensity axis instead of the spatial axes.

Most requantization steps can be specified in terms of an **intensity range** of pixel values to be coded and a **quantization step** which determines the smallest intensity step possible in the image after requantization. The intensity range is usually specified in units of the map; pixel values which fall outside this range are clipped to the appropriate range limit or to a blank value which distinguishes them from other pixel values. The quantization step can be specified in units of the map, in percent of the peak, or in any other convenient units. Requantization can be generalized if appropriate to nonuniform quantizers (e.g., logarithmic or exponential) which emphasize particular properties of the intensity distribution.

One particularly common nonuniform quantization is **histogram equalization**, a recoding of the pixel values in such a way that all pixel values have approximately equal probability of occurrence (i.e., the histogram is flat). This eliminates redundancy in high dynamic range data, but requires that a recoding table be sent along with the coded bit stream. Histogram equalization performs a function similar to Huffman coding, but is not reversible and sometimes has higher overhead. More general histogram modifications have also been studied [20].

Just as an image should never be resampled to a higher spatial resolution than exists in the original image, it should never be requantized to a higher intensity resolution or a smaller quantization step. This is particularly important since fine requantization of coarsely quantized data may produce large amounts of redundancy in the data which cannot be removed by linear predictive coding schemes.

Progressive Transmission

In discussing resampling and requantization, we have assumed that the user knows beforehand what spatial and intensity resolution will be needed. This is usually not the case. Users often need to see an image at low resolution, with the opportunity to refine the resolution for all or part of the image as needed. Such schemes are generally grouped under the term **progressive transmission**, and have been studied by Rots [21] and Schwab [5]. Progressive transmission methods can be used with both predictive and transform coding schemes.

In progressive predictive coding, a low resolution image is transmitted in its entirety, using any desired coding scheme. The low resolution image is then used in predicting the pixel values for a higher resolution image. The user decides whether to continue the process, using the higher resolution image as an estimator of an even higher resolution image. Powers-of-two resolution increase are commonly used, leading to a process in which rectangular blocks are successively subdivided to form a more and more accurate approximation of the final image.

In progressive transform coding, the image is divided into blocks and a few transform coefficients from each block are transmitted, beginning with the lowest-frequency components. The user decides whether to continue, adding a few more transform coefficients from each block until sufficient resolution is achieved. The number of coefficients per block and the number of bits per coefficient can both be varied. Transform coding can be made particularly stable by reconstructing the image as it will be decoded from the quantized transform coefficients, subtracting this reconstructed image from the original one, and transform coding the residual image at each step. This is analogous to the use of internal coder feedback in predictive transmission.

In these cases we speak primarily of spatial progressive transmission. However, we could instead transmit a map at low intensity resolution, then refine the intensity resolution at each step, possibly by sending the next bit plane. While less common and more I/O intensive, this approach might also be worthwhile for radio image transmission.

An important consideration is the information made available to the user in deciding whether to continue. At a minimum, this information should include the current resolution, the maximum possible resolution, and an estimate of the distortion contained in the current version of the image. This could be either mean-square-error distortion, for example, or the maximum amount by which a pixel value is in error.

A useful feature is the ability to select a subwindow in the image which is to be refined. Rather than refining that subimage within the context of the lower resolution image, it might be desirable to write the complete lower resolution image to mass storage, refine the smaller window to the desired resolution, then write that smaller image to mass storage.

The type of transmission medium used becomes more important when progressive schemes are considered. Progressive transmission only achieves compression when the coded image is read in real time over the communications link, waiting to transmit higher resolution information until the user has requested it. If some subset only of the coded stream is needed (e.g., if a subwindow is requested), some real time computation may be required at the transmitting end; otherwise, simple access to a remote file containing the coded stream may be sufficient. The coded stream must contain information up to the highest resolution that may be desired by the user, to avoid recoding if the first few levels of resolution are insufficient.

Pyramidal Transmission

A particularly attractive progressive predictive scheme is the **pyramidal coding** method, studied by Tanimoto [22] and Knowlton [23] for facsimile transmission and by Rots [21] for transmission of radio images. In this scheme, a single value representing the entire image is transmitted first, followed by values which successively define pixel values in power-of-two expansions until full spatial resolution is reached. Pyramidal schemes generally require that the entire pyramid of reduced-resolution images be constructed before transmission begins.

Rots' scheme transmitted the sum of all pixels over the map, followed by pixel value differences which required the transmission of only three numbers to expand a pixel at one level into four pixels at the next lower level. This scheme, while ingenious, added two bits at each new level of the image, with the number of levels being the log (base 2) of the length of the longer side of the image. I have developed a scheme which eliminates this disadvantage by transmitting successive maxima over the map, as follows:

1. The original image is divided into four-pixel squares. The maximum of each square is placed in the appropriate pixel location of a lower resolution image of one-quarter size. Each pixel in the original image is replaced by the difference between that pixel and the maximum over the square. All differences are positive.

2. The previous step is repeated, producing a new image of one-quarter the size of the previous image at each step, until a one-pixel image representing the maximum over the entire image is reached. This single value is transmitted to full precision.

3. Each level of the difference pyramid is transmitted is reverse order of their construction, allowing the receiver to construct a new image at twice the spatial resolution of the previous one. The differences can generally be transmitted in fewer bits than the original pixel values, providing compression at each level. The receiver decides at each level whether another level in needed, allowing additional compression by declining to call for transmission of unneeded information.

With the exception of the first value, only positive integer pixel values are transmitted, and their bit length can be bounded in advance. The differences at each level can be transmitted using adaptive point coding at each level, using the minimum number of bits required. Since the maximum over a region is available at each step, no bright source in the field will be missed. Alternatively, the minimum may be used instead of the maximum, providing slightly faster convergence and an earlier look at noise in the field.

A SOFTWARE TOOL FOR IMAGE CODING

In the summer of 1985, I implemented a software tool under AIPS for the analysis of image coding strategies. This tool consists of a pair of AIPS tasks for converting an AIPS image to and from an "external" format which can be easily transported from one AIPS system to another. These tasks, XPORT and MPORT, were intended to allow for implementation and testing of various image coding schemes, and to allow the convenient transfer of images between the VLA AIPS systems and the Los Angeles Cray.

Design Considerations

The following design goals were made before tasks XPORT and MPORT were written.

1. They should provide a simple means for implementing and testing new image compression algorithms.
2. They should be transportable as possible, and should use an external file format which would allow image transfer between any two AIPS systems.
3. They should meet the needs of a wide range of users. Unlike "real-world" image coding, astronomical image coding often requires the analysis of "noise" data during postprocessing. A user is just as likely to be interested in this "noise" as in the peaks of the image, particularly during image deconvolution. The tasks should allow the user as much choice as possible as to what information in the image will be coded.
4. They should not depend explicitly on the type of communication hardware and software used to transfer the image file, with the exception that for progressive schemes, an MPORT task running on one AIPS system should be able to read an external file created by XPORT on another AIPS system without actually transferring the entire external file to the local system.
5. The user should have the ability to display the image as it is received.
6. The tasks' user interface should be straightforward and should fit into the AIPS "scheme of things."

An Overview of Task XPORT

Task XPORT reads an AIPS image file, codes it using a user-selected coding scheme, and writes the resulting coded bit stream into an external file. This external file consists of a FITS header with coding-specific information contained in the AIPS history section, followed by the coded bit stream. The file consists of 80-byte records with the coded bit stream starting in the next record after the FITS END statement and stored in the low-order seven bits of each byte.

Resampling and smoothing are requested through the specification of an image window, resolution factors along the first three axes (expressed as pixel increments), and the type of sampling to be used. Types of sampling include the selection of the pixel value at the sample point and the computation of the maximum, minimum or mean over the sampling cell for each pixel. This allows the user to highlight bright sources, noise or overall behavior, respectively, in a low-resolution transmitted version of the original image.

Requantization is requested through the specification of a pixel range to be coded and a quantization step to be used, expressed as a percentage of the image intensity peak or in units of the image. This allows the user to select an appropriate intensity resolution for his or her purposes.

Finally, the CODETYPE adverb is used to select the type of image coding to be used. The currently available coding methods are simple fixed-length point coding and a pair of pyramidal coding schemes. New coding methods can be added by the addition of an entry in the initialization portion of task XPORT, the addition of an entry in the FITS header processing portion of task MPORT, and the addition of subroutines to code and decode the image.

The image to be exported is converted to scaled integer format, resampled and requantized, then coded and written to the specified output file. This output file can then be transported to another AIPS system or read through a communications link by an MPORT task.

Output to the external file is handled by a group of ZPUTx routines. These routines, which process the external data file at the bit level, should be the only system-dependent routines in the task.

An Overview of Task MPORT

Task MPORT reads an external file created by task XPORT and decodes it to produce an AIPS image file, which may optionally be written to the television screen as it is received.

The coding method used to encode the image is determined from the FITS header, and the appropriate routine is called to decode the image. As each row is decoded, it is passed to routine TVDATA which writes the row to the television. If a progressive coding scheme is being used, the television image may be refreshed several times at higher resolution, with the user given the opportunity to halt processing and write the current version of the image to mass storage. The user can specify a window within the received image which is to be displayed as the image is decoded, as well as the range of pixel values to be displayed and the type of labelling to be used. History information describing the source of the file is added to the new AIPS image.

Input from the external file is handled by a group of ZGETx routines. These routines, which process the external data file at the bit level, should be the only system-dependent routines in the task.

Problems Encountered

Since the difference of two I*2 values can exceed the range of an I*2 variable, I*4 variables were used at various points in processing. The integer range constraint should be considered whenever an integer map is being coded.

A restriction on the use of A-type formatting in VAX-11 FORTRAN prevented all eight bits per external file byte from being used.

Suitability of AIPS for Image Coding

Of the VLA computer systems, AIPS was certainly the most suitable for the development of a set of image coding tools. Its user interface is excellent, and contributed to a number of features in XPORT and MPORT otherwise impossible.

However, its internal structure, particularly that relating to image I/O, is convoluted in the extreme and makes implementing a coding technique much more difficult than necessary. The AIPS I/O system should be made more accessible, particularly to the programmer who has a series of operations to perform on an entire image. Image update in place would have also been most helpful during this project.

Another weakness of AIPS relevant to image coding is that of text file I/O. If would simplify image coding immensely if 80-byte files could be read and written via AIPS subroutine calls, particularly if those routines could also handle the bit-level I/O required by most image coding methods.

Similarly, utility routines for such tasks as image sampling and quantization and histogram production would simplify the XPORT task immensely.

Finally, it should be noted that while XPORT and MPORT use integer scratch maps exclusively, these scratch maps could be real-valued if AIPS is re-organized to process floating-point images only. However, more care will need to be taken to control quantization and roundoff error during intermediate processing and coding.

CONCLUSIONS

The XPORT and MPORT tasks are only a beginning. The following projects would increase our ability to code and transfer radio image efficiently between processing systems.

1. Add support for the following coding techniques to the XPORT and MPORT tasks:

    - Huffman coding, preferably in a generalized form which allows it to be used to postprocess output sequences from other image coding schemes
    - Adaptive predictive coding, using a variety of linear and nonlinear predictors
    - Discrete cosine transform coding using zonal coding

    These methods, once implemented, should be compared to determine the most appropriate use for each.

2. Make a study of the statistical structure of a set of "typical" VLA images, including but not limited to a detailed survey of their covariance structure; this information could be used to develop a good predictor for radio images produced by the VLA [13].

3. Survey transmission techniques for coded radio images, covering particularly the possibilities for progressive coding as an image transmission and review mechanism.

4. Consider the following question: what information in a radio image is important? How can we extract that information as efficiently as possible? An answer to this question would be the ideal guiding principle for the coding of VLA radio images.

ACKNOWLEDGEMENTS

I would like to thank Pat Moore and Al Braun for general assistance with AIPS and the VLA computer systems; Tim Cornwell, Fred Schwab and Ron Ekers for helpful conversations concerning deconvolution, filtering and coding; Rick Perley, Frazer Owen, Pat Crane and Arnold Rots for suggestions toward the design of the XPORT and MPORT tasks; and Bob Duquet, without whose advice and guidance I would not have been able to complete this project.

REFERENCES

1. R.N. Strickland, "Estimation of local statistics for digital processing of nonstationary images", IEEE Trans. Acoust., Speech, Signal Processing ASSP-33 (1984), 465-469.

2. A. Habibi, "Survey of adaptive image coding techniques", IEEE Trans. Commun. COM-25 (1977), 1315-1322.

3. D.A. Huffman, "A method for the construction of minimum redundancy codes", Proc. IRE 40 (1952), 1098-1101.

4. J.M. Yohe, "Algorithm 428: Hu-Tucker minimum redundancy alphabetic coding method", Collected Algorithms from ACM, Association for Computing Machinery, New York, 1972.

5. Fred Schwab, "More on image compression", VLA Computer Memo. No. 174, February 1985.

6. M. Hankamer, "A modified Huffman procedure with reduced memory requirement", IEEE Trans. Commun. COM-27 (1979), 930-932.

7. H. Murakami, "Algorithm for construction of variable length code with limited word length", IEEE Trans. Commun. COM-32 (1984), 1157-1159.

8. M. Pechura, "File archival techniques using data compression", Comm. ACM 25 (1982), 605-609.

9. F. Rubin, "Experiments in text file compression", Comm. ACM 19 (1978), 617-623.

10. R.G. Gallager, "Variations on a theme by Huffman", IEEE Trans. Inf.Theory IT-24 (1978), 668-674.

11. O. Johnsen, "On the redundancy of binary Huffman codes", IEEE Trans. Inf. Theory IT-26 (1980), 220-222.

12. A.K. Jain, "Image data compression: a review", Proc. IEEE 69 (1981), 349-389.

13. A.K. Jain, "Advances in mathematical models for image processing", Proc. IEEE 69 (1981), 502-528.

14. P.A. Maragos, R.W. Schafer and R.M.Mersereau, "Two-dimensional linear prediction and its application to adaptive predictive coding of images", IEEE Trans. Acoust., Speech, Signal Processing ASSP-32 (1984), 1213-1229.

15. H.E. Knuttson, R. Wilson and G.H. Granlund, "Anisotropic nonstationary image estimation and its applications. Part I--Restoration of noisy images", IEEE Trans. Commun. COM-31 (1983), 388-397.

16. H.E. Knuttson, R. Wilson and G.H. Granlund, "Anisotropic nonstationary image estimation and its applications. Part II--Predictive image coding", IEEE Trans. Commun. COM-31 (1983), 398-406.

17. Steve Wilson, letter to Bob Burns, July 21, 1985.

18. W.K. Pratt, Digital Image Processing, Wiley, New York, 1978.

19. D.P. O'Leary and S. Peleg, "Digital image compression by outer product expansion", IEEE Trans. Commun. COM-31 (1983), 441-444.

20. Kautsky, Nichols, Jupp, "Smooth histogram modification for image processing", Computer Vision, Graphics and Image Processing 26 (1984), 271-291.

21. Arnold Rots, "Transmission of digital images over slow communication lines", VLA Computer Memo. No. 173, October 1984.

22. S.L. Tanimoto, "Image transmission with gross information first", Computer Graphics and Image Processing 9 (1979), 72-76.

23. K. Knowlton, "Progressive transmission of gray-scale and binary pictures by simple, efficient and lossless encoding schemes", Proc. IEEE 68 (1980), 885-896.