

# **VIDEO COMPRESSION TECHNIQUES**

**Author: Tom Cilke  
LORAL/CONIC  
9020 Balboa Avenue  
San Diego, California 92123**

## **ABSTRACT**

This paper will attempt to present algorithms commonly used for video compression, and their effectiveness in aerospace applications where size, weight, and power are of prime importance. These techniques will include samples of one-, two-, and three-dimensional algorithms. Implementation of these algorithms into usable hardware is also explored but limited to monochrome video only.

## **INTRODUCTION**

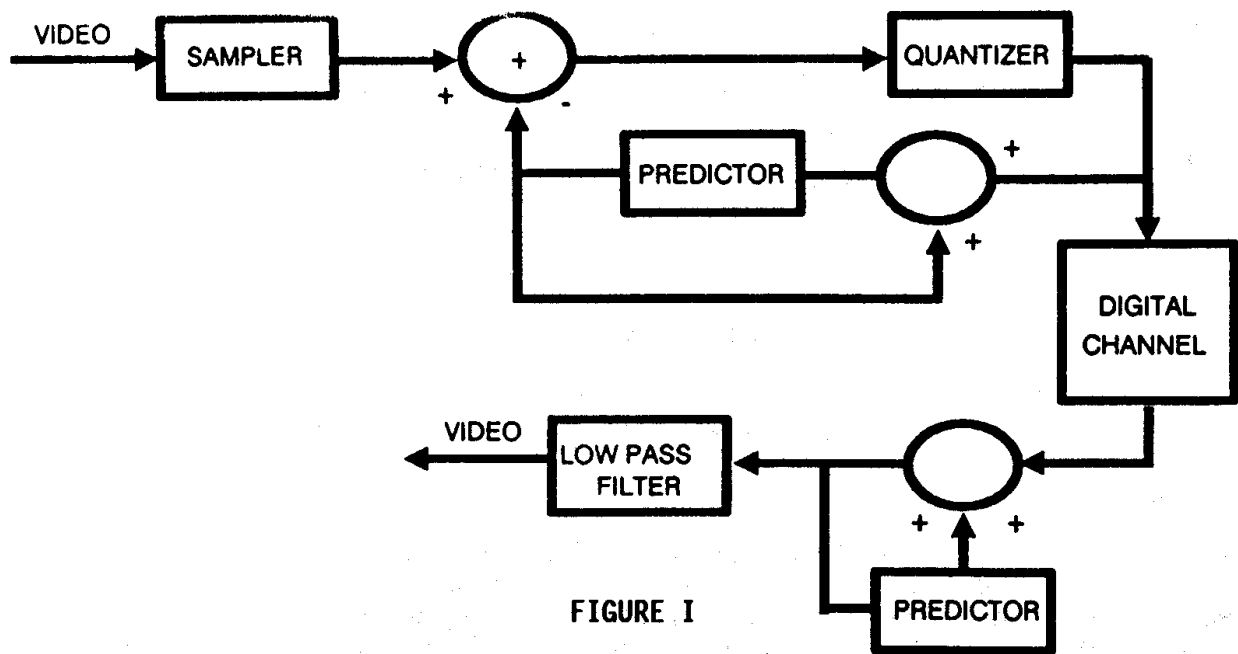
Compression is possible in video images because of pixel redundancy in those images. This redundancy is, in a way, a measure of entropy. The more entropy a video image has, the less it can be compressed. There are several useful methods for compressing images with one-, two-, and three- dimensional algorithms--each method more complex than the last. One of the largest problems facing the future of digital video in aerospace applications is the lack of a recognized standard. Typical examples of the most common of the many available techniques are presented in the following sections. The last sections will cover how these techniques relate to realizable aerospace hardware.

### **One-Dimensional Spatial Redundancy Reduction Techniques**

Differential Pulse code Modulation (DPCM), is sometimes referred to as predictive encoding and is by far the most popular and simple spatial redundancy algorithm technique used today. Though apparently very simple this technique, when implemented to achieve large compression ratios and high image quality, is more complicated than it would first appear. Compression ratios of 4:1 are quite common with 8:1 being about the pragmatic limit. The DPCM algorithm is often combined with entropy or Huffman encoding to achieve these higher compression ratios.

The terms “predictive quantizing” and “differential pulse code modulation” (DPCM) are used interchangeably and will describe a special kind of predictive communications system

coding. A predictive communication system is one in which the difference between the actual signal and an estimate of the signal, based on its past, is transmitted. Both the transmitter and the receiver make an estimate or prediction of the next pixel value based on the previously transmitted pixel(s) using a set of previously defined rules. The transmitter subtracts this prediction from the true value of the pixel and transmits the difference. The receiver adds its pixel prediction to the received difference signal, yielding the true pixel. Highly redundant signals, such as television, are well suited for predictive transmission systems because of the accuracy possible in the prediction thus yielding a small difference. A one dimensional DPCM system only computes or transmits differences based on previous pixels occurrences from the same line. A block diagram of systems of this type is shown in **Figure 1**.



**FIGURE I**

### **ADAPTIVE OR VARIABLE LENGTH DPCM**

The most prevalent shortcoming of traditional DPCM techniques is slope overload. Slope overload occurs when a predictive quantizer encounters a large unexpected change of the data that cannot be closely predicted. These large prediction errors cannot be properly quantized by traditional DPCM fixed length encoders. Properly implemented VL DPCM can effectively eliminate the slope overload limitation of DPCM.

In traditional DPCM, a fixed number of bits is used to encode the value of the current pixel by quantizing it into one of  $2^n$  possible quantization levels centered around the predicted pixel value. Extensions of DPCM schemes in which the predictor and/or quantizer are adaptive to the local characteristics of the image are sometimes called Adaptive DPCM (ADPCM) or Variable Length DPCM. Substantial compression gains can be achieved by

making the number of bits sent for each pixel adaptive on the basis of information from the neighborhood. In smooth regions of the image, fewer quantization levels are needed to encode most pixels. A large change in the intensity level is always a possibility and in traditional DPCM would cause slope overload. A means of estimating the number of bits required for a given pixel, and of encoding an unexpected change if it occurs, would be useful for keeping the image quality high while keeping the number of bits transmitted per pixel low.

One effective solution to this problem is to use an error-bit to specify whether or not the estimated number of bits required to code the next pixel is sufficient. This pixel-by-pixel adaptive approach is simple and efficient. To estimate the number of bits required to encode a given pixel, an estimate of the degree of smoothness around the pixel must be made. The estimate of the smoothness in the area of the predicted pixel is made from the deltas of previous pixels. If the maximum error between the predicted value and the actual value for a number of local preceding pixels is used to estimate the maximum expected error this can be used as a measure of smoothness.

Smoothness = MAX value of (error pixel 1,...,error pixel n)

Smoothness is then used to select the estimated number of bits required to encode a given pixel. The value of smoothness determines a range of expected quantization levels around the predicted pixel value. If the actual intensity value is within this range the predicted number of bits (possibly 0) is used. The table below illustrates the allowed levels for various values of smoothness. Q1 to Qn are quantizer levels.

| Smoothness value | Additional bits/pixel | Quantizer Levels | Max Prediction Error |
|------------------|-----------------------|------------------|----------------------|
| 0, 1             | 0                     | 0                | THR                  |
| 2                | 1                     | +/-1             | $(Q1 + Q2)/2$        |
| 3                | 2                     | +/-2             | $(Q2 + Q3)/2$        |
| 4, 5             | 3                     | +/-4             | $(Q5 + Q6)/2$        |
| >5               | 4                     | +/-8             | $(Q8 + Q9)/2$        |

For a given smoothness, if the pixel error is less than the maximum value, the given number of bits/pixel plus a “0” error-bit are used to encode the pixel. Otherwise, a different code scheme is implemented to reach the quantization levels outside the expected range. If smoothness is less than two, the only expected quantization level relative to the predicted pixel value is 0. In all other cases, an even number of quantization levels arranged symmetrically around the value of predicted pixel value is used. When the error-bit is “1”, the predicted number of bits is inadequate. A code word is then constructed

consisting of a sign bit followed by a “0” for each quantization level skipped, and terminated by a “1” at the desired quantization level. The table below illustrates the pattern of code words for smoothness value of 4.

| Pixel error value | Additional code bits |
|-------------------|----------------------|
| +Q6               | 10001                |
| +Q5               | 1001                 |
| +Q4               | 101                  |
| +Q3               | 11                   |
| -Q3               | 01                   |
| -Q4               | 001                  |
| -Q5               | 0001                 |
| -Q6               | 001001               |

The error-bit, if left uncompressed, would set a lower bound on the DPCM coding of 1 bit/pixel. When entropy coding is used to compress the error-bits, however, coding rates well below 1 bit/pixel are possible in smooth areas of the image.

## HUFFMAN ENCODING

Huffman encoding is a method of transforming a stream of PCM data into a stream of bits containing the least amount of redundancy. This technique is sometimes referred to as entropy encoding. The principles governing the operation of this algorithm depend on knowledge of the statistics of the source that is to be encoded. In other words, more probable characters can be given shorter length codes and less probable characters can be given longer length codes. Developing a good statistical representation of the source data is done by collecting statistical data on a very large sample of data and developing a code book. Blocking the input data and calculating the probabilities of the characters in each block (Blocked Huffman), or by keeping a running count of the encoded characters and predicting that similar statistics will follow (Adaptive Huffman), are other ways to use Huffman coding. The Statistical Huffman technique is most widely used for video transmission because the statistics of the data (video) are easily gathered, and since they are required to be known before encoding, they need not be transmitted. The Blocked Huffman is a “two pass” algorithm and requires that the tree structure (or code book) be stored and transmitted along with the data. This technique is also popular in video compression schemes with the code book being transmitted every line, field, or frame.

## **Two-Dimensional Techniques**

Two dimensional techniques introduced by this paper will be 2D-DPCM, transform techniques, and fractal geometry techniques. The most promising techniques for two-dimensional video compression, both from an airborne hardware and compression ratio standpoint, are two-dimensional predictive techniques and two-dimensional transform techniques. The first method is the simpler from a hardware point of view, but the latter methods, while more complex with relation to hardware, yield much higher compression ratios.

### **Two-Dimensional Predictive or DPCM Techniques**

The two dimensional DPCM approach we describe in this section achieves both high quality and compression ratios only slightly better than other sophisticated DPCM techniques. As with most DPCM schemes, the first field is coded recursively, in raster scan order, from left to right. The decoder, when reaching the location of the current pixel, has access to all intensity values of the reconstructed image including history pixels above and to the left. The values of the reconstructed intensity of previous pixels around and above (from previous lines) could be used to construct the intensity of the current pixel.

In traditional DPCM, a fixed number of bits,  $n$ , is used to encode the value of the current pixel by quantizing it into one of  $2^n$  possible quantization levels centered around this predicted value. A set of quantization levels derived from psychophysical experiments which gives excellent image quality using 5 bit DPCM is 0, 2, 5, 8, 13, 20, 27, 36, 47, 59, 72, 86, 100, 116 and 128.

This technique, though improving the accuracy of the predicted value and therefore requiring fewer bits to encode differences helps somewhat, but compression ratios more than 8:1 are beyond reasonable application of DPCM.

### **Transform Techniques**

Various transforms exist that can be computed with “fast” algorithms. Some of these, the Walsh-Hadamard transform (WHT), discrete Fourier transform (DFT), Haar transform (HT), and the slant transform (ST), are all orthogonal transforms that can be computed with “fast” algorithms, and have been considered for various compression applications. The performance of these transforms is generally compared with that of the Karhunen-Loeve transform (KLT) which is known to be optimal with respect to the following performance measures: variance distribution, estimation using the mean-square error criterion, and the rate-distortion function. Although the KLT is optimal, there is no general algorithm that enables its fast computation.

A discrete cosine transform (DCT) is also interesting as a compression algorithm because it can be computed fairly quickly. The DCT can be computed with one of several “fast” algorithms, called Fast Cosine Transform (FCT) algorithms. One of the most efficient that has been developed is based on the FFT. It can be shown that the performance of the DCT or FCT also compares more closely to that of the KLT relative to the Performances of the DFT, WHT and HT.

The performance of the DCT with KLT, DFT, and the identity transforms, can be compared using the rate-distortion criterion. This performance criterion provides a measure of the information rate that can be achieved, while still maintaining a fixed distortion, for encoding purposes. Using this criterion it can be shown that the KLT and DCT compare more closely than the KLT and DFT.

DCT distortions are graininess, loss of edge detail and a blocked appearance. The graininess and loss of edge details are associated with the loss of or incorrect quantization of the higher frequency components through quantization. When these components are eliminated, edge details become less sharp. When the higher frequency components are incorrectly quantized, a grainy or noisy pattern occurs. The blocked appearance is associated with implementation of small transformation: sub-blocks. The boundaries between the sub-blocks become apparent because of the different quantization characteristics within each sub-block.

## **FRACTAL Geometry Techniques**

Researchers at the Georgia Institute of Technology have developed compression methods that can reduce the amount of information needed to store and reproduce digitized images by a factor of more than 10,000. The technique was developed by mathematics Professors Michael F. Barnsley and Alan D. Sloan using fractal geometry.

Using stored fractal codes to generate geometrical shapes that are compiled to simulate landscapes is not new, and has been used by cinematographers for generating special effects since large super computers used for image processing have been available to the motion picture industry. However, Barnsley and Sloan have developed the ability to start with an actual image and then find fractals that will reproduce the image to almost any level of detail required. The result is a highly compressed, image reproduction data set.

The process begins by taking a digitized image of the subject. The image is then broken up into segments using image process techniques that include edge detection and texture variation analysis. The segments are then checked against a library which contains relatively compact sets of numbers called iterated function codes that will reproduce the fractals required to develop the image segment. The library is cataloged so shapes that

look similar are located closely together with nearby codes corresponding to nearby fractals. The structure of the catalog permits automated library searches for fractals, which, when combined, will approximate the segment.

Once the iterated function codes are found for each segment the original digitized image can be discarded and the codes transmitted. To reproduce the image, the received codes are put into a random iteration algorithm. The algorithm will produce the same image repeatedly, independent of the sequence of random choices that are made.

Selection of the proper iterated function codes is currently very time intensive, and the image encoding and decoding process requires a great deal of computations. Presently, complex images require many hours to encode and decode using hardware and techniques available today.

The major draw back of fractal geometry techniques is the “cartoonish” quality of the reconstructed images. This distortion is usually considered acceptable for military aerospace application such as target recognition or surveillance.

### **Three-Dimensional Techniques or Temporal Redundancy Reduction**

**Inter-frame coding** is used to remove redundancies between consecutive frames. This is usually achieved by applying DPCM on the temporal axis. Thus, a prediction of the current frame, based on the previous frame, is used to exploit the correlation between frames. This prediction can further be improved by motion compensation. The displacements of objects in the image can be estimated and taken into consideration by the predictor to help minimize prediction errors. The resulting prediction error can then be encoded via an intra-frame technique such as transform coding or predictive coding.

**Displacement estimation** is required to compensate for moving objects in the scene. Most of the displacement-estimation algorithms fall into one of two groups: recursive algorithms and block-matching algorithms. With pixel-recursive algorithms an initial estimate is updated iteratively, while the recursion may be horizontal, vertical, or temporal. The displaced frame difference (DFD) is defined as the difference between a pixel and the corresponding previous frame pixel, shifted by the displacement estimate. The pixel-recursive algorithms minimize DFD.

**Block-matching** consists of defining a block of pixels in the current frame and correlating it with pixels within a search area in the previous frame. The best match is located at the point of the correlation peak. In practical applications, maximizing the cross-correlation function may be replaced by minimizing the mean square error or the mean absolute

difference. To reduce the number of steps in finding the best match, several search techniques can be used: 2D-logarithmic search, three-step search, and modified conjugate-direction search.

### **Availability of Promising Techniques as Aerospace Hardware**

Many types of compression hardware are available today using a multitude of algorithms that all claim to be the best available. Most of the available hardware is large and not rugged enough for many aerospace applications. Much of the video compression research and hardware development has been done by commercial industry in search of the “Dick Tracy” video phone which will potentially replace the standard telephone. Aerospace applications have benefitted from this work as well as also benefitting from developments in military electronics in the general field of signal processing. Aerospace hardware employing DPCM is available today with DCT hardware in development today, and more advanced 3D hardware in the algorithm and breadboard stage.

### **DPCM**

Aerospace hardware implementations of DPCM techniques are available today from more than one company. One-dimensional Variable-length DPCM has been implemented on VLSI chips, and is the heart of a real-time, video compression machine capable of compressing real time video down to 8.8 Mbps and less. This same hardware has built-in encryption and a multi-rate modem capable of bit rates down to 1.1 Mbps. These lower bit rates are achieved by decimation of video stream. This hardware is 120 cubic inches, weighs 5.5 lbs, and consumes about 20 watts of 28 Vdc power.

### **TRANSFORM CODING**

The transformation itself has always been a computational “bottleneck” in real-time image coding. General purpose digital signal processing chips are becoming more commercially available every day, and some special purpose signal processing chips optimized for DCT’s or FFT’s are presently available as standard chips. Chips available for this function today are slow in relation to real time video, or 8 megapixels per second, and must be used in tandem to process blocks of pixels in parallel. This makes this technique very expensive, power-hungry, and larger than DPCM hardware available today. Hardware of this type, if built today for aerospace applications, should achieve compression ratios of 20:1, in a package of 100 to 200 cubic inches, weighing about 6 pounds, and consuming about 75 to 100 watts of 28 Vdc power. It is estimated that military transformer chips capable of speeds usable with real time video will be available within 1-2 years.

## **INTER-FRAME CODING**

This technique is very promising since it can achieve compression ratio as high as 100:1 and would allow transmission of full frame real time video with a channel bandwidth of less than 1 Mbps. VLSI designs for displacement estimation are becoming possible with existing technologies. Special purpose motion-compensation chips may be available within 2-3 years. Algorithm simulation and optimization work is in progress today and more rudimentary algorithms are presently being breadboarded as the first step to aerospace applications. As with hardware described above initial 3D hardware will be more complex than 1D or 2D hardware when introduced, since it will more than likely be a combination of these techniques. This complexity will translate to larger boxes consuming more power. More than likely initial 3D hardware will not employ motion compensation but will be a combination of 2D with 1D added temporary. The need for this hardware is very real and is driven by the continual decrease of available RF bandwidth.

## **CONCLUSION**

In this study, one-, two-, and three-dimensional video compression techniques have been explored and hardware implementation of both current and potential techniques has been considered. Because a great deal of research and design is presently dedicated to improving methods of video compression, this paper has sought to present algorithms which are now commonly used for video compression techniques, as well as those techniques that hold promising possibilities for future application into telemetry systems.

## **REFERENCES**

1. Ahmed, N., T. Natarajan, and K. R. Rao. "Discrete Cosine Transform." IEEE Transactions on Computers, Jan. 1983, pp. 90-93.
2. Anastassiou, Dimitris, Joan L. Mitchell, and William B. Pennebaker. "A High-Compression DPCM Based Scheme for Picture Coding", IEEE, 1983, pp. 453-455.
3. Kandebo, Stanley W. "Fractals Research Furthers Digitized Image Compression", Aviation Week, Apr. 1988, pp. 91, 95.
4. O'Neal, J. B. Jr. "Predictive Quantizing Systems for the Transmission of Television Signals", Bell System Technical Journal, pp. 689.691