BOUNDED ERROR DATA COMPRESSION AT CONSTANT RATE

Robert A. Gonsalves*
Norman E. Evans
Alicia Shea
EIKONIX Corporation
103 Terrace Hall Ave.
Burlington, MA 01803

ABSTRACT

We present a technique to compress M-bit data to N-bit data with a bounded error between the original and de-compressed data. The bounded error is established by quantizing the error signal of DPCM and by entropy encoding the quantized error. The fixed data rate is maintained by non-linear adaptation of the quantization interval on an intermittent basis. Examples of image data compression are given.

INTRODUCTION

Our goal is:

(a) select a compression scheme that achieves a modest compression ratio (about 4 to 1),
(b) is easy to implement in real time, in particular, requires only one pass through the image, and
(c) minimizes the loss of information between the original and reconstructed images.

To achieve the goal we have selected DPCM with a non-linear predictor, efficient, zero-error encoding of the quantized error, and adaptive line-by-line control of the quantizing interval. The technique will be suitable for use in a digital image processing system to increase the storage capability and to speed up the transmission of images over the slower data links in the system.

It is well known that image data can be compressed to one or two bits per pixel without significant degredation of the viewed image [1]. However the question of information loss is difficult to assess. In this paper we have adopted a conservative approach and have required that the error between original and final images be bounded on a line-by-line

* Also with Northeastern University, Boston, MA.

basis. This avoids the possibility of slope overload that is typical of some DPCM schemes and guarantees that only a well known artifact, amplitude quantization, is introduced into the final image.

THE ALGORITHM

A block diagram of the DPCM compression algorithm is shown in Figure 1. The M bits per pixel (BPP) image data is scanned in conventional fashion; Y (K,J) is the pixel at row K column J. The predictor forms an estimate $\hat{Y}$ of Y and the resulting error E is quantized with a uniform quantizer of step size D. The quantized error, $\hat{E}$, is encoded with the code book shown in Table 1. The variable length code words are put into a buffer which transmits bits at a constant rate of N BPP. The number, T, of bits in the buffer and the rate, R, at which bits are entering the buffer are used to control the quantization interval, D.

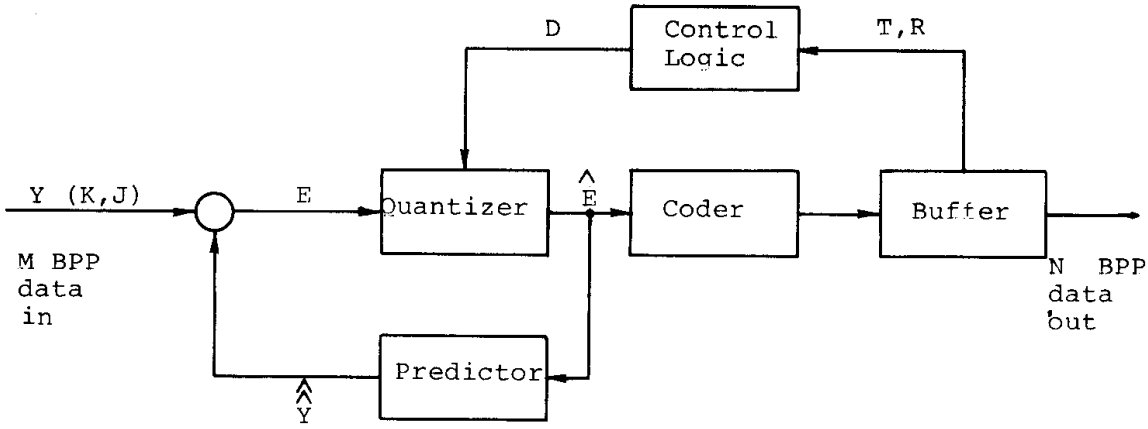

Figure 1 Compression Algorithm

Table 1. Code Book

quantized error, $\hat{E}$	code word
0	1
1	01
-1	001
2	0001
-2	00001
etc.	

PREDICTOR

The predictor uses six previous, reconstructed data points (A,B,C,D,E,F), for its estimate of Y. See Figure 2. We test these points for the likelihood of a vertical edge (V) or a horizontal edge (H) near Y. Then we set

$$\hat{\hat{Y}} = \begin{cases} B, \text{ if V is very likely} \\ A, \text{ if H is very likely} \\ 1/2\ (A+B), \text{ if neither V nor H is very likely.} \end{cases} \quad (1)$$

The formula for the last option is

$$\hat{\hat{Y}}\ (K,J) = .5\ \hat{Y}\ (K,J-1) + .5\ \hat{Y}\ (K-1,J), \quad (2)$$

where

$$\hat{Y}\ (K,J) = \hat{\hat{Y}}\ (K,J) + \hat{E}\ (K,J) \quad (3)$$

is the reconstructed data (Equation (3) is also the recipe for the decompression algorithm). For the first two options in (1) we use coefficients 0, 1 and 1,0, respectively, in (2).

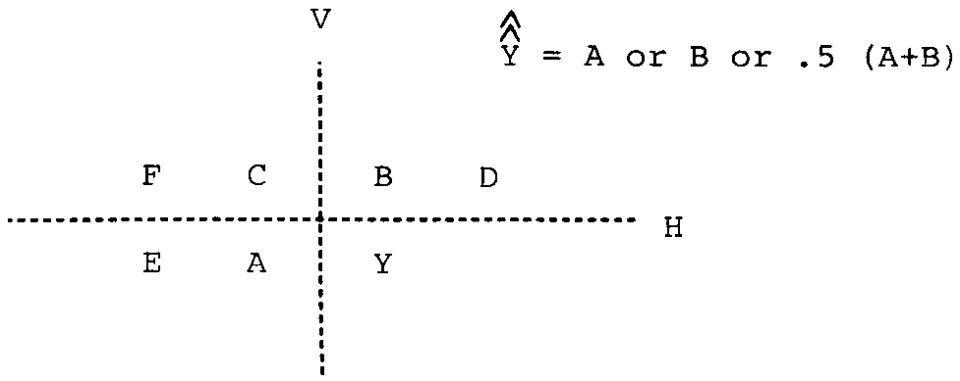

Figure 2 Details for the predictor

The tests in (1) are accomplished by estimating the two edge levels for each hypothesis H or V. For example, for a vertical edge (V) we estimate right and left edge levels

$$RE = 1/2\ (B+D) \quad (4)$$

and

$$LE = 1/4\ (A+C+E+F). \quad (5)$$

Then we calculate the mean square error between these estimates and the actual values: $(B-RE)^2$, etc. This gives a V metric,VM, which we compare to an H metric, HM, of similar construction.

Thus (1) becomes

$$\hat{\hat{Y}} = \begin{cases} B, \text{ if } 2\,VM < HM \\ A, \text{ if } 2\,HM < VM \\ 0.5\,(A+B), \text{ otherwise } . \end{cases} \tag{6}$$

The factor G=2 in (6) was chosen by examining the compression achieved for several pictures as we varied G. Figure 3 shows the effect on the average quantization interval D and indicates that the algorithm is not too sensitive to this factor.

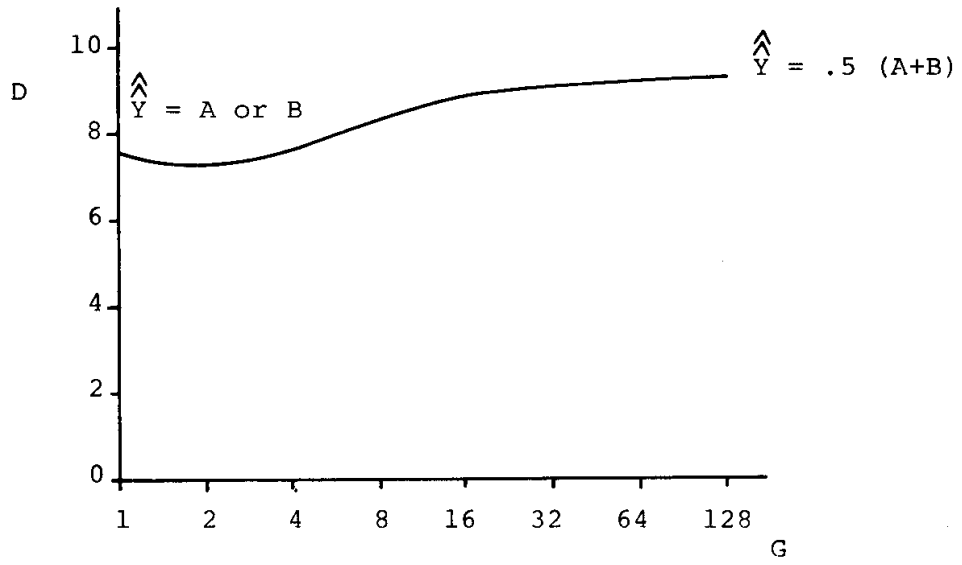

Figure 3 Average quantization interval, D, vs. edge test factor, G

ADAPTATION

The encoder puts bits into the buffer at a variable rate but we want to extract bits at a constant rate of, say, 2 bits per pixel. Moreover, we want the buffer to be of reasonable size, perhaps 16 times the row length. Thus, for an image of row length 256 the buffer will have a 4096 bits capacity.

To control the number of bits in the buffer we monitor the input bit rate, averaged over each line. This rate, R, for the Kth line and the total number, T, of bits for the Kth line are

used to adjust the quantization interval, D, for the next line. Clearly, if the input rate is high or the buffer is too full we want to generate fewer bits on the next line. This can be accomplished by increasing D, a distortion measure.

The way R changes with D is a subject of rate distortion theory. In principle, this is calcuable if one makes certain assumptions (Gaussian statistics for Y, in particular) and can be estimated based on such simple parameters as the local mean, variance and X and Y decorrelation intervals. In practice, however, we adjust D only by integer values to simplify the implementation and we find that D needs to be changed only in steps of ±1 or 2 for an eight bit picture (integers 0 to 255). Thus we use the extremely simple logic shown in Table 2 to change D.

Table 2 Logic for the new D

IF (T.GT.(3/4)TMAX)	D = D+l
IF (T.LT.(1/4)TMAX)	D = D−1
IF (R.GT. DR (1.03))	D = D+l
IF (R.LT. DR (.97))	D = D−1

From Table 2 we see, for example, that D can be changed by +2 if the buffer is more than 3/4 full and R exceeds the desired rate, DR, by 3%; that D can be changed by −1 if R is 3% less than DR and the buffer is between 1/4 and 3/4 full.

This logic appears to be very robust and does not suffer from an instability that we observe when T (or R), alone, is used to select D.

EXAMPLES

In Figures 4,5, and 6 we show three scenes that have been used to illustrate the compression-reconstruction. (a) is the set of originals, (b) is the result, and (c) is a comparison picture which uses no line-by-line adaptation but achieves the same average distortion, D, as (b).

All images were 256 X 256 and were expanded by linear interpolation to 1024 X 1024 for display on a COMTOL screen. The bias and gain controls were set for best viewing of the original (a) and were untouched for (b) and (c).

In all sets there is not much to distinguish scenes (a), (b) and (c). Note, however, that scene (c) in the airport, Figure 4, contains noticeable degredations. In particular, quantization effects can be seen in the buildings on the lower left corner and the entire image seems to be softer. Both (b) and (c) have lost textural detail in the grassy regions of

Figure 5 and (c) appears to have lost details of a foot path in the upper center. The same loss of texture appears in Figure 6.

Some data from the compression algorithms,(b) and (c) are shown in Figures 7 and 8, respectively. Note how in Figure 7 the quantization interval D changes, in particular, gets larger as the scan moves into a busier (high R) area. Also note that, without adaptation, the buffer empties at row 230 in Figure 8.

DISCUSSION

The techniques presented here were selected to achieve a modest 4 to 1 compression with very simple hardware. The application is slanted towards an image processing system that has an 8 bit display so that images originating at the display will be compressed to 2 bits.

Our non-linear predictor is similar to that presented by Zschunke [2] and the encoder is the FS code of Rice and Plaunt. [3] The encoder uses a minimum code word length of unity so compression at or below one bit per pixel is impossible. However, if the entrophy H ($\hat{E}$) of the difference signal $\hat{E}$ is, itself, less than unity then a multi-mode encoding scheme such as that presented by Rice and Plaunt can reduce the bit rate to less than H ($\hat{E}$) + 0.3.[1]

The measure of fit between the original and decompressed pictures is the quantization interval D. That is, they differ by no more than ± D/2 on a line-by-line basis.

The line-by-line adaptive feature is used mainly to achieve real-time control over the number of bits per pixel in the final image. There is only a slight improvement over the nonadaptive DPCM. However the latter would require two passes to find a D that achieves a desired bit rate or, in one pass with a specified D, one would have to modify the hardware to accommodate an unspecified bit rate.

REFERENCES

1. A. Habibi, “Survey of Adaptive Image Coding Techniques”, IEEE Proc. on Comm., Vol. COM-25, No 11, November 1977, p. 1275.

2. W. Zschunke, “DPCM Picture Coding with Adaptive Prediction”, IEEE Proc. on Comm., Vol. COM-25, No 11, November 1977, p. 1295.

3. R.F. Rice and J.R. Plaunt, “Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data”, IEEE Proc. on Comm. Tech., Vol. COM-19, No 6, December 1971, p. 889.

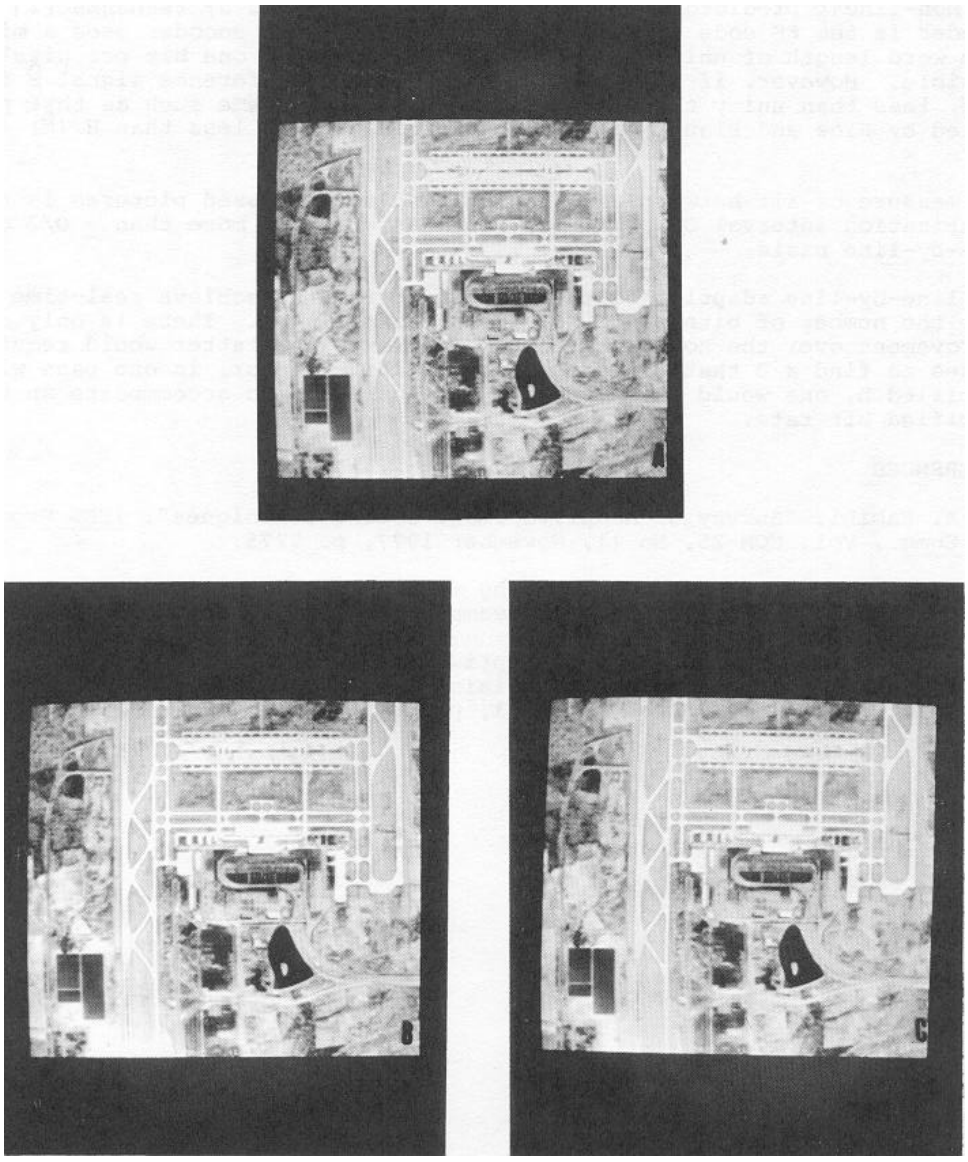

Figure 4 Airport: a) 8 bit original, b) 2 bit adaptive DPCM, c) 2 bit non-adaptive DPCM

Figure 5 Missile Site: a) 8 bit original, b) 2 bit adaptive DPCM, c) 2 bit non-adaptive DPCM

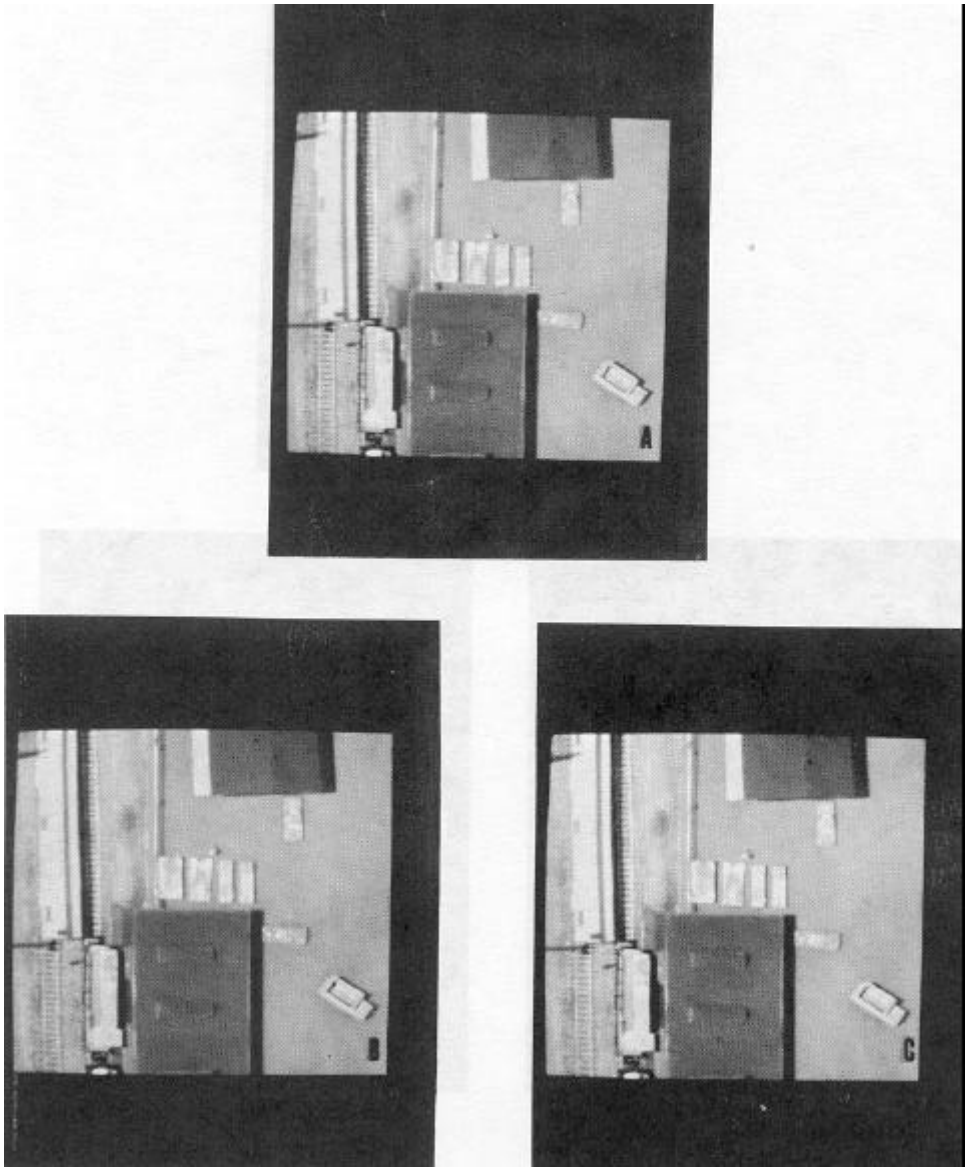

Figure 6 Model: a) 8 bit original, b) 2 bit adaptive DPCM, c) 2 bit non-adaptive DPCM.

		R	D	T
ROW 200	AVG BITS =	1.949	7.	2852.0
ROW 201	AVG BITS =	2.004	7.	2853.0
ROW 202	AVG BITS =	1.898	7.	2827.0
ROW 203	AVG BITS =	2.195	6.	2877.0
ROW 204	AVG BITS =	2.137	7.	2912.0
ROW 205	AVG BITS =	2.055	8.	2926.0
ROW 206	AVG BITS =	2.270	8.	2995.0
ROW 207	AVG BITS =	1.957	9.	2984.0
ROW 208	AVG BITS =	1.906	9.	2960.0
ROW 209	AVG BITS =	2.098	8.	2985.0
ROW 210	AVG BITS =	1.797	9.	2933.0
ROW 211	AVG BITS =	1.949	8.	2920.0
ROW 212	AVG BITS =	1.934	8.	2903.0
ROW 213	AVG BITS =	2.180	7.	2949.0
ROW 214	AVG BITS =	1.980	8.	2944.0
ROW 215	AVG BITS =	1.902	8.	2919.0
ROW 216	AVG BITS =	2.066	7.	2936.0
ROW 217	AVG BITS =	1.891	8.	2908.0
ROW 218	AVG BITS =	2.273	7.	2978.0
ROW 219	AVG BITS =	1.953	8.	2966.0
ROW 220	AVG BITS =	1.789	8.	2912.0
ROW 221	AVG BITS =	1.863	7.	2877.0
ROW 222	AVG BITS =	2.082	6.	2898.0
ROW 223	AVG BITS =	1.777	7.	2841.0
ROW 224	AVG BITS =	2.219	6.	2897.0
ROW 225	AVG BITS =	1.961	7.	2887.0
ROW 226	AVG BITS =	1.965	7.	2878.0
ROW 227	AVG BITS =	2.090	7.	2901.0
ROW 228	AVG BITS =	1.848	8.	2862.0
ROW 229	AVG BITS =	1.914	7.	2840.0
ROW 230	AVG BITS =	2.008	6.	2842.0
ROW 231	AVG BITS =	2.070	6.	2860.0
ROW 232	AVG BITS =	1.891	7.	2832.0
ROW 233	AVG BITS =	1.875	6.	2800.0
ROW 234	AVG BITS =	2.242	5.	2862.0
ROW 235	AVG BITS =	2.063	6.	2878.0
ROW 236	AVG BITS =	2.219	6.	2934.0
ROW 237	AVG BITS =	1.836	7.	2892.0
ROW 238	AVG BITS =	2.535	6.	3029.0
ROW 239	AVG BITS =	1.887	8.	3000.0
ROW 240	AVG BITS =	1.844	7.	2960.0

Figure 7 Line-by-line bit rate (R), quantization interval D and bits in the buffer T for Figure 4 (b)

		R	D	T
ROW 200	AVG BITS =	1.785	8.	846.0
ROW 201	AVG BITS =	1.801	8.	795.0
ROW 202	AVG BITS =	1.750	8.	731.0
ROW 203	AVG BITS =	1.727	8.	661.0
ROW 204	AVG BITS =	1.938	8.	645.0
ROW 205	AVG BITS =	2.035	8.	654.0
ROW 206	AVG BITS =	2.258	8.	720.0
ROW 207	AVG BITS =	2.176	8.	765.0
ROW 208	AVG BITS =	2.145	8.	802.0
ROW 209	AVG BITS =	2.078	8.	822.0
ROW 210	AVG BITS =	2.012	8.	825.0
ROW 211	AVG BITS =	1.949	8.	812.0
ROW 212	AVG BITS =	1.953	8.	800.0
ROW 213	AVG BITS =	1.941	8.	785.0
ROW 214	AVG BITS =	1.992	8.	783.0
ROW 215	AVG BITS =	1.898	8.	757.0
ROW 216	AVG BITS =	1.891	8.	729.0
ROW 217	AVG BITS =	1.914	8.	707.0
ROW 218	AVG BITS =	1.996	8.	706.0
ROW 219	AVG BITS =	1.980	8.	701.0
ROW 220	AVG BITS =	1.801	8.	650.0
ROW 221	AVG BITS =	1.664	8.	564.0
ROW 222	AVG BITS =	1.711	8.	490.0
ROW 223	AVG BITS =	1.617	8.	392.0
ROW 224	AVG BITS =	1.711	8.	318.0
ROW 225	AVG BITS =	1.852	8.	280.0
ROW 226	AVG BITS =	1.789	8.	226.0
ROW 227	AVG BITS =	1.863	8.	191.0
ROW 228	AVG BITS =	1.836	8.	149.0
ROW 229	AVG BITS =	1.691	8.	70.0
ROW 230	AVG BITS =	1.629	8.	-25.0
ROW 231	AVG BITS =	1.691	8.	-104.0
ROW 232	AVG BITS =	1.734	8.	-172.0
ROW 233	AVG BITS =	1.512	8.	-297.0
ROW 234	AVG BITS =	1.652	8.	-386.0
ROW 235	AVG BITS =	1.754	8.	-449.0
ROW 236	AVG BITS =	1.773	8.	-507.0
ROW 237	AVG BITS =	1.699	8.	-584.0
ROW 238	AVG BITS =	1.953	8.	-596.0
ROW 239	AVG BITS =	1.941	8.	-611.0
ROW 240	AVG BITS =	1.680	8.	-693.0

Figure 8 Repeat of Figure 7 for the non-adaptive DPCM of Figure 4 (c)