Vol. I, Issue II, September 2012 (ISSN: 2278-7720)

Digital Image Compression Comparisons using DPCM and DPCM with LMS Algorithm

Ranbeer Tyagi
Assistant Professor Department of Electronics &Telecommunication engg., PCRT (PU), Bhopal (M.P.) India

D. K. Sharma
Assistant Professor Department of Electronics & Communication engg., Ujjain Engineering College, Ujjain (M.P.) India

ABSTRACT

Image processing modifies pictures to improve, extract information and change their structure (composition, image editing, and image compression etc.). Images can be processed by optical, photographic and electronic means, but image processing using digital computers are the most common method because digital methods are fast, flexible and precise. Compression - reducing the redundancy in the image data to optimize transmission / storage. The DPCM and LMS may be used to remove the unused bit in the image for image compression. In this paper we compare the compressed image results for 1 and 3 bits DPCM Quantzation and DPCM with LMS Algorithm and also compare the histogram, prediction mean square using DPCM Quantzation and DPCM with LMS Algorithm for approximately same distortion levels. The LMS may provide almost 2 bits per pixel reduction in transmitted bit rate compared to DPCM when distortion levels are approximately the same distortion for both methods. The LMS Algorithm may be used to adapt the coefficients of an adaptive prediction filter for image source coding. In the method used in this paper we decrease the compressed Image size, distortion and also the estimation error.

Key words

DPCM, LMS Algorithm, Quantization, Histogram, Prediction mean square error,GUI.

1. INTRODUCTION

1.1 The basic idea of Image Compression process

In general the reduction of image data is achieved by the removal of redundant [1] data. In mathematics, compression may be defined as transforming the two-dimensional pixel array into a statistically uncorrelated data set. Usually image compression is applied prior to the storage or transmission of the image data. Later the compressed image is decompressed to get the original image or close to original image.

Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area. Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is the case when binary data such as executables, documents etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images (and music too) need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable.

1.2 What is an Image

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows.

Fig 1: An image — an array or a matrix of pixels arranged in columns and rows.

In a (8-bit) grey scale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey. A normal grayscale image has 8 bit colour depth = 256 grayscales.

1.3 The Principle of Image Compression

In a communication environment, the difference between adjacent time samples for image is small, coding techniques have envolved based on transmitting sample-to-sample differences rather than actual sample value. Successive differences are in fact a special case of a class of non-instantaneous converters called N-tap linear predictive coders. These coders, sometimes called predictor-corrector coders, predict the next input sample value based on the previous input sample values. This structure is shown in fig 2. In this type of converter, the encoder forms the prediction error (or the residue) as the difference between the next measured sample value and the predicted sample value. The equation for the prediction error [1] is

$$e(n) = x(n) - y(n) \tag{1}$$

In figure 1, Where Q=Quantizer, $x(n)$ is the nth input sample, $y(n)$ is the predicted value, and $e(n)$ is the associated prediction error. This is performed in the predict-and-compare loop, the loop shown in fig 2. It's prediction by forming the sum of its prediction and the prediction error

$$e_q(n) = quant[e(n)] \qquad (2)$$

Where quant (.) represents the quantization operation, $e_q(n)$ is the quantization [2] version of the prediction error, and $x_s(n)$ is the corrected and quantized version of the input sample. This is performed in the predict-and-correct loop.

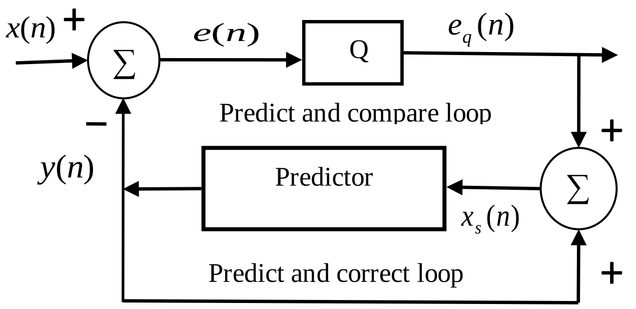


Fig. 2 Basic Block diagram of DPCM with LMS Algorithm image compression system

The communication task is that of transmitting the difference (the error signal) between the prediction and the actual data sample. For this reason, this class of coder is often called a differential pulse code modulator (DPCM) [3]. If the prediction model forms predictions that are close to the actual sample values, the residues variance (relative to the original signal).

2. IMAGE COMPRESSION USING DPCM

2.1 Basic idea of DPCM Method

Many different ideas have been proposed to improve the encoding efficiency of A/D conversion. In general, these ideas exploit the characteristics of the source signals. In analog messages we can make a good guess about a sample value from knowledge of past sample values. In other words, the sample values are not independent, and generally there is a great deal of redundancy in the Nyquist samples. Proper exploitation of this redundancy leads to encoding a signal with fewer bits. Considering a simple scheme; instead of transmitting the sample values, we transmit the difference between the successive sample values. Thus, if x(n) is the nth sample, instead of transmitting x(n), we transmit the difference e(n) = x(n) – x(n-1). At the receiver, knowing e(n) and several previous sample value x(n-1), we can reconstruct x(n). Thus, from knowledge of the difference e(n), we can reconstruct x(n) iteratively at the receiver. Now, the difference between successive samples is generally much smaller than the sample values Thus, we reconstruct the samples at the receiver iteratively. If our prediction is worth its salt, the predicted (estimated) value y(n) will be close to x(n), and their difference (prediction error) e(n) will be even smaller than the difference between the successive samples. Consequently, this scheme, known as the differential pulse code modulation (DPCM) [10][11], which is a special case of DPCM, where the estimate of a sample value is taken as the previous sample value, that is, y(n) =x(n-1).

2.2 Analysis of DPCM

In DPCM we transmit not the present sample $x(n)$, but $e(n)$ (the difference between $x(n)$ and its predicted value $y(n)$). At the receiver, we generate $y(n)$ from the past sample value to which the received $x(n)$ is added to generate $x(n)$. There is, however, one difficulty associated with this scheme. At the receiver, instead of the past samples $x(n-1), x(n-2),...$ as well as $e(n)$ we have their quantized version $x_s(n-1), x_s(n-2),..$ this will increase the error in reconstruction.

In such a case, a better strategy is to determine $y(n)$ the estimate of $x_s(n)$ (instead of $x(n)$), at the transmitter also from the quantized samples $x_s(n-1), x_s(n-2),...$ …………… difference $e(n) = x(n) - y(n)$ is now transmitted via PCM. At the receiver, we can generate $y(n)$, and from the received $e(n)$, we can reconstruct $x_s(n)$.Figure 1 shown a DPCM predictor. We shall soon show that the predictor input is $x_s(n)$. Naturally, its output is $y(n)$ the predicted value of $x_s(n)$.The difference of original image data, $x(n)$ and prediction image data, $y(n)$, is called estimation residual, $e(n)$ .So

$$e(n) = x(n) - y(n) \qquad (3)$$

is quantized to yield

$$e_q(n) = e(n) + q(n)$$

Where $q(n)$ is the quantization error, $e_q(n)$ quantized signal. And

$$q(n) = e_q(n) - e(n) \qquad (4)$$

The prediction output $y(n)$ is fed back to its input so that the predictor input $x_s(n)$ is

$$\begin{aligned} x_s(n) &= y(n) + e_q(n) \\ &= x(n) - e(n) + e_q(n) \\ &= x(n) + q(n) \qquad (5) \end{aligned}$$

This show $x_s(n)$ is quantized version of $x(n)$. The prediction input is indeed $x_s(n)$, as assumed. The quantized signal $e_q(n)$ is now transmitted over the channel. Flowchart of DPCM system is shown below fig 3.

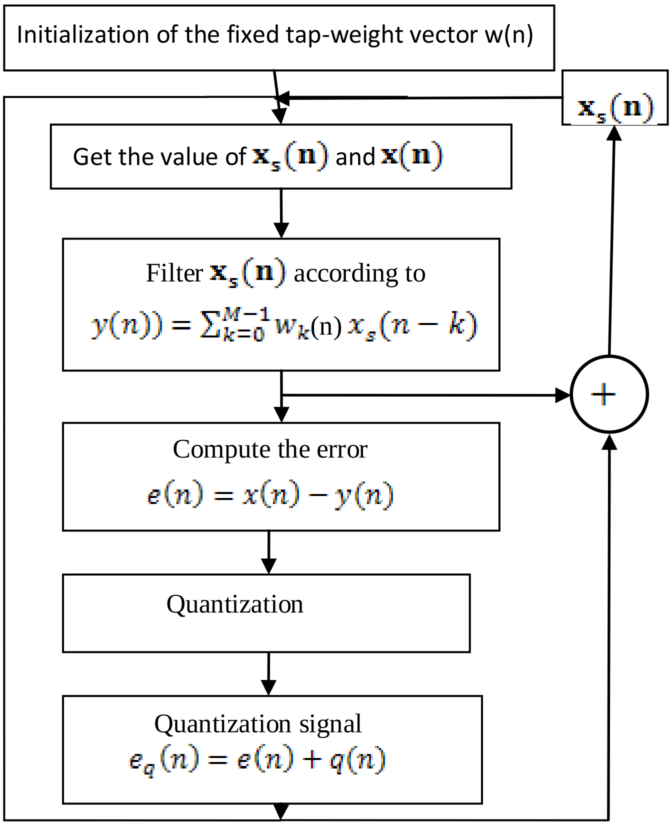

Initialization of the fixed tap-weight vector w(n)
$x_s(n)$
Get the value of $x_s(n)$ and $x(n)$
Filter $x_s(n)$ according to
$y(n)) = \sum_{k=0}^{M-1} w_k(n)\, x_s(n-k)$
+
Compute the error
$e(n) = x(n) - y(n)$
Quantization
Quantization signal
$e_q(n) = e(n) + q(n)$

Fig 3: Flowchart of DPCM system

3. IMAGE COMPRESSION USING DPCM WITH LMS ALGORITHM

A block diagram of the LMS adaptive image compression system is shown in fig. 1 [4]. It is seen that the image prediction $y(n)$ is formed in a linear manner at the output of the LMS filter:

$$y(n) = \sum_{k=0}^{M-1} w_k(n) x_s(n-k) \qquad (6)$$

$$y(n) = w_0 x_s(n) + w_1 x_s(n-1) + \ldots w_{M-1} x_s(n-M+1)$$

$$y(n) = w^T(n) x_s(n) \qquad (7)$$

In equation (6), the $w_k(n)$ are N adaptive predictor coefficients, the $x_s(n)$ are the reconstructed image data, and k is 1, 2……….N integer values which select the previous image pixel on which base the current prediction. At each scanned pixel a prediction residual (error), $e(n)$ is computed

$$e(n) = x(n) - y(n) \qquad (8)$$

This quantized residual is send to the receiver. The quantization residual is determine

$$e_q(n) = e(n) + q(n) \qquad (9)$$

This residual is then quantized to form $e_q(n)$ and the quantized residual is also used to update the predictor coefficient for the next iteration by the well known least mean squares (LMS) algorithm [5].

$$w(n+1) = w(n) + \mu e_q(n) x_s(n) \qquad (10)$$

The parameter μ is known as the step size parameter and is a small positive constant, which control steady-state and convergent mean-square residual characteristics of the predictor. The LMS algorithm is an approximation to the gradient search method for iteratively computing the N optimal $w(n)$ coefficients which minimize the mean square prediction residual. It is known by [6][12] that the error between the original image and the reconstructed image at the receiver is simply the quantization error Thus, the distortion between the original discrete image $x(n)$ and the reconstructed value $y(n)$ at the receiver is given by

$$d(n) = y(n) - x(n) = e_q(n) - e(n) \qquad (11)$$

(Assuming the no channel-induced errors)

Therefore, if the goal of the system is an accurate reconstruction of the image, then an algorithm is desired which will form an accurate $y(n)$, so that $e(n)$ will have smaller variance and the quantizer levels may be adjusted to give a smaller quantization error.

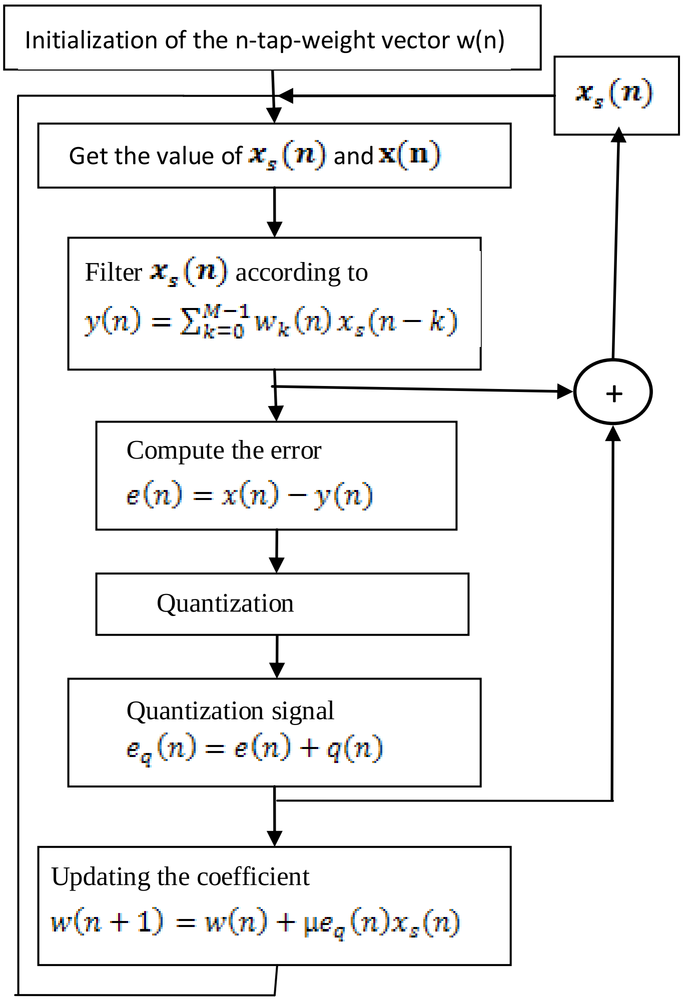

Initialization of the n-tap-weight vector w(n)
$x_s(n)$
Get the value of $x_s(n)$ and $x(n)$
Filter $x_s(n)$ according to
$y(n) = \sum_{k=0}^{M-1} w_k(n)\, x_s(n-k)$
+
Compute the error
$e(n) = x(n) - y(n)$
Quantization
Quantization signal
$e_q(n) = e(n) + q(n)$
Updating the coefficient
$w(n+1) = w(n) + \mu e_q(n) x_s(n)$

Fig 4: Flow chart of image compression using DPCM with LMS Algorithm

Hence, a lower reconstruction error, or distortion, will be present at the receiver. The quantizer levels themselves may be fixed or may vary as some function of the residual sequence $e_q(n)$.Although, in general, the position of the quantizer levels could be adaptive, for simplicity, and in this correspondence we only examine the case of a quantizer with fixed levels.

Alternatively, if the goal of the system is to reduce the bit rate over the channel subject to some distortion criteria, then we may reduce the number of quantizer levels which span the residual signal range and, hence, produce shorter code words per level. In this situation the LMS adaptive predictor reduces the average number of bits per image while maintaining an acceptable visual appearance at the receiver. The flowchart of the least mean square adaptive algorithm using differential pulse code modulation quantization system is shown in fig 4.

4. SIMULATION ENVIRONMENT

In this paper has used the fixed weight coefficient DPCM and adaptive tap weight coefficient LMS parameter. This parameter has been shown in bellow table. These parameters are the resulted parameter of our simulation for DPCM and LMS algorithm.

TABLE-1parametervalue/ Configuration of DPCM

Parameter	**value**
Image Matrix size	256×256
Original Image size	96.5 kB (98,915 bytes)
DPCM 1bit/pixel reconstructed Image size	83.0 kB (85,075 bytes)
DPCM 3bit/pixel reconstructed Image size	88.1 kB (90,243 bytes)
DPCM 1bit/pixel distortion level	-21.75
DPCM 3bit/pixel distortion level	-22.25
DPCM fixed Tap's	2
DPCM weight coefficient value	W=[.495, .456]
No of Bit's	1, 2, 3 bit's
Quantization level	2, 4, 8 level

TABLE-2 parameter value /Configuration of using DPCM with LMS Algorithm

Parameter	**value**
Image Matrix size	256×256
Original Image size	96.5 kB (98,915 bytes)
LMS 1bit/pixel reconstructed Image size	73.2 kB (74,960 bytes)
LMS 3bit/pixel reconstructed Image size	85.5 kB (87,618 bytes)
LMS 1bit/pixel distortion level	-22.4
LMS 3bit/pixel distortion level	-23.3
No of Filter Tap's	420
LMS adaptive weight coefficient	W=[ones(1,tap's)]
No of Bit's	1, 2, 3 bit's
Quantization level	2, 4, 8 level
LMS Parameter	µ=.0005

5. SIMULATION ANALYSIS PARAMETER

In this paper, the following parameters are used to analysis the image compression [14], [15], [16] using DPCM and using DPCM with LMS.

5.1 Histogram: A plot between the probability associated with each gray level versus gray level in the image is called histogram. Assume that the gray levels in this thesis image after normalization range from 0 to 1.

5.2 Prediction Error: The difference of original image data, $x(n)$, and prediction image data, $y(n)$, is called estimation residual or prediction error. At each scanned pixel a prediction residual, $e(n)$, is computed

$$e(n) = x(n) - y(n)$$

5.3 Distortion:

The distortion [12] between the original discrete image x(n) and the reconstructed value y(n) at the receiver is given by

$$d(n) = y(n) - x(n) = e_q(n) - e(n)$$

(Assuming the no channel-induced errors)

6. SIMULATION RESULT S AND DISCUSSION

6.1 Simulation Results and Discussion of Original image results

Simulations involving real image input signal consisted of 256×256 rows and columns' image matrix displayed in fig 5.

The histogram of original image is shown in fig 6. This histogram shows the gray levels are concentrated towards the dark end of the gray scale range. Assume that the gray levels in this image after normalization range from 0 to 1.

Fig 7 shows the histogram plot between gray level and sample. In this image samples range are from 0 to 255 but gray level changes according to sample value.

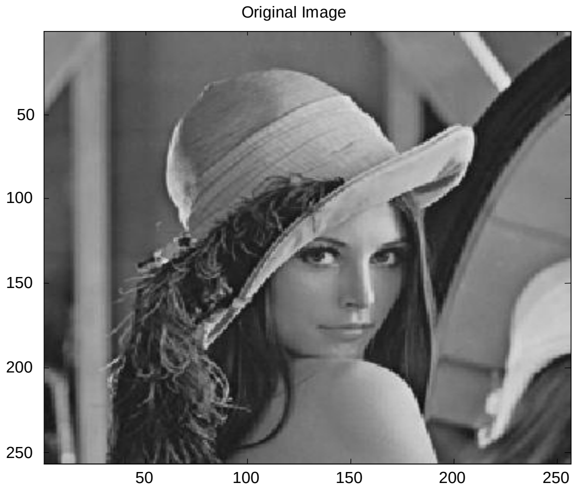


Fig 5: Original image with 256×256 matrix dimension

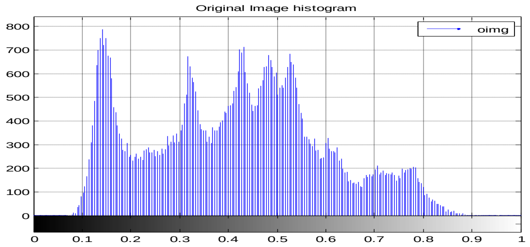


Fig 6: Original Lena image Or Dark image histogram

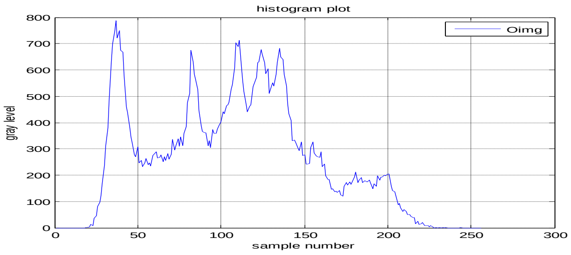


Fig 7: Histogram plot Original Lena image

6.2 Simulation results Discussion of image Compression for 1, 3bit DPCM and DPCM with LMS Algorithm

The LMS algorithm was simulated using Matlab 7.5 with respected to the application of image compression comparison using DPCM with LMS algorithm [7] [12] depicted in figure 1. LMS algorithm is easy to implement and computationally inexpensive. This feature makes the LMS algorithm attractive for image compression.

Simulation involving real image input signal consisted of 256 sample points. Filter length was taken to be 420 taps. The parameter of LMS algorithm µ was set to be 0.0005.

The 256×256 original image with matrix dimension is shown in fig 5. This image size is 96.5 kB (98,915bytes). This original image passed with the residual quantizer (Q in fig 2) consisting of b=1, 2 and 3 bits (2, 4 and 8 quantization levels, respectively) using DPCM with LMS algorithm adaptive coefficient w. The characteristics of the quantizer follow the Laplacian density model [8, 9]. The coefficients of the fixed DPCM predictor were chosen in accordance with the globally optimum model [10] and fixed coefficient value taken by w=[0.495 0.456]. The dynamic range of the data was eight bits from grey level 0 to 255. The simulation result shown in bellow figures.

Figure 8 shows the histogram 3bit/pixel using DPCM with LMS algorithm compressed or dark Lena image. Assume that the gray levels in this image after normalization range from 0 to 1. Fig 9 shows the histogram 3bit/pixel DPCM compressed or dark Lena image. Assume that the gray levels in this image after normalization range from 0 to 1. Fig 10 shows the histogram 1bit/pixel using DPCM with LMS algorithm compressed or bright Lena image. Assume that the gray levels in this image after normalization range from 0 to 1. Fig 11 shows the histogram 1bit/pixel DPCM compressed or bright Lena image. Assume that the gray levels in this image after normalization range from 0 to 1. Fig 12 shows the PMSE 3bit/pixel using DPCM with LMS algorithm compressed Lena image [13]. This shows the PMSE [dB] versus sample points from 0 to 255. Fig 13 shows the PMSE 3bit/pixel DPCM compressed Lena image. This shows the PMSE [dB] versus sample from 0 to 255.Fig 14 shows the PMSE 1bit/pixel using DPCM with LMS algorithm compressed Lena image. This shows the PMSE [dB] versus sample points from 0 to 255. Fig 15 shows the PMSE 1bit/pixel DPCM compressed Lena image. This shows the PMSE [dB] versus sample from 0 to 255.

Fig 16 plots the average square distortion versus transmitted bit rate for the Lenna image. All values of average squared error in dB referenced to the performanced of the 1bit/pixel fixed coefficient predictor. The bit rate is in bits/ pixel and is controlled by the number of levels in the quantizer. The top graph is for the fixed DPCM predictor and the lower is for LMS with µ=0.0005-value. The LMS filter was initialized at the beginning of the picture reception.

In fact, the DPCM at 3bit/pixel has approximately the same distortion than LMS 1 bit/pixel. The lenna image more compress 1bit/pixel LMS compare to 3bit/pixel DPCM with approximately same distortion level. The difference of 1bit/pixel LMS to 3bit/pixel DPCM is 14.9kB (15,283bytes) more in 1bit/pixel LMS. Lastly, the visual characteristics of LMS distortion are presented in fig 17, displaying the results for 3 bits/pixel and 1 bit/pixel transmission. At 3 bit/pixel, comparing the LMS predictor figure 17(a) and the DPCM prediction figure 17(b) shows there is no significant visual different between either method or original leena image.

How ever, as the bit rate is decreased to 1 bit/pixel, there is significance difference between the LMS processing and that using DPCM. Figure 17(c) displays the reconstruction using LMS, which provides a significantly sharper image than that shown in figure 17(d) which results of using 1 bit/pixel DPCM.

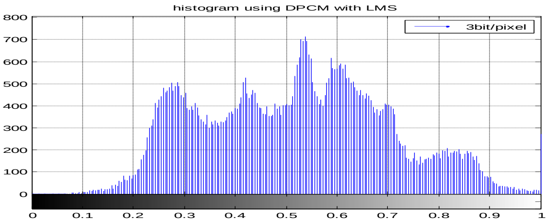


Fig 8: Histogram 3 bit/pixel using DPCM with LMS compressed or Dark Lena image

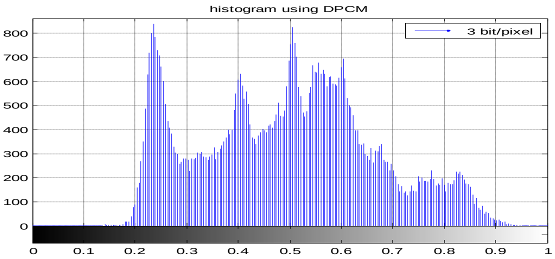


Fig 9: Histogram3bit/pixel DPCM compressed or Dark Lena image

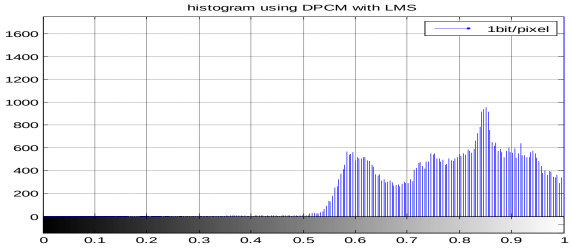


Fig 10: Histogram 1 bit/pixel using DPCM with LMS compressed or Bright Lena image

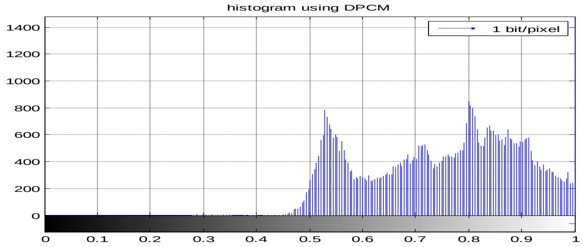


Fig 11: Histogram1bit/pixel DPCM compressed or bright Lena image

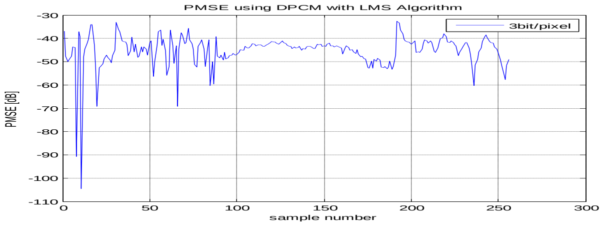


Fig 12: PMSE 3 bit/pixel using DPCM with LMS compressed Lena image

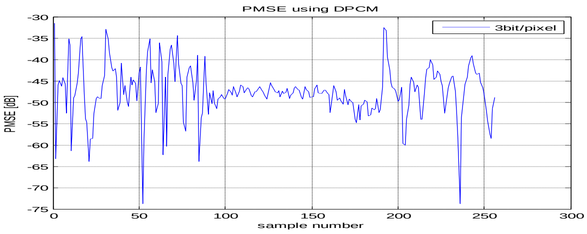


Fig 13: PMSE 3bit/pixel DPCM compressed Lena image

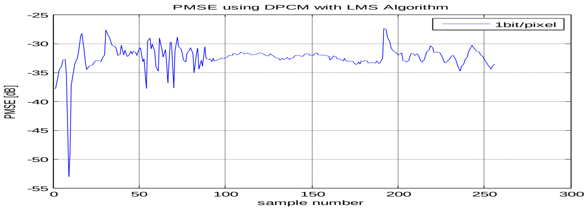


Fig 14: PMSE 1 bit/pixel using DPCM with LMS compressed Lena image

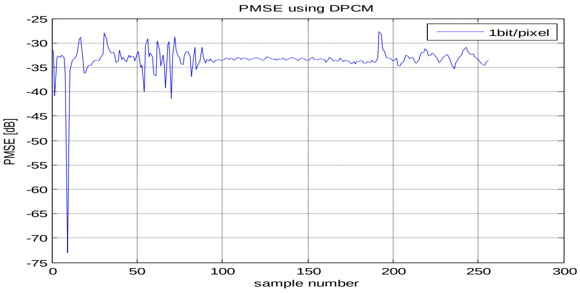


Fig 15: PMSE1bit/pixel DPCM compressed Lena image

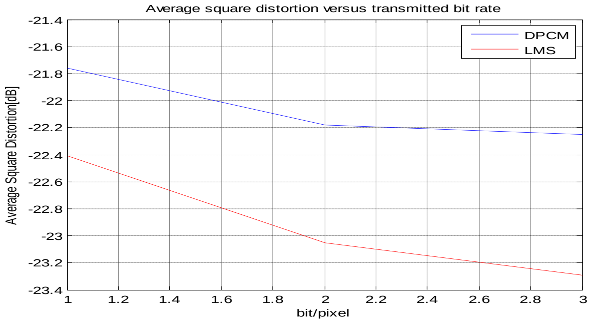
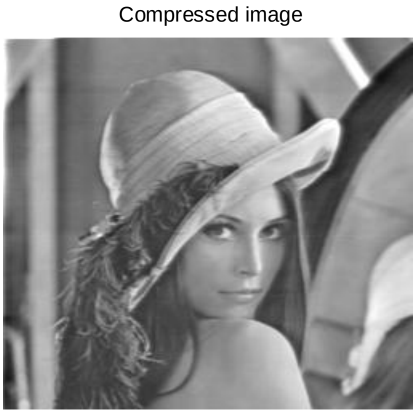
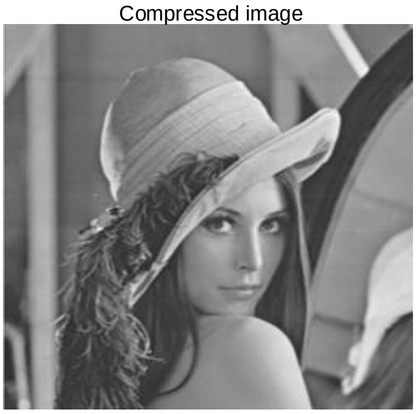


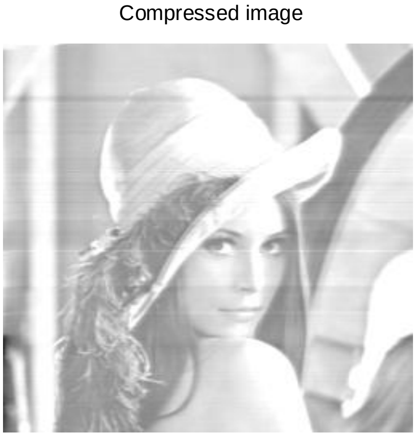
Fig 16: Average Square distortion versus transmission bit rate.



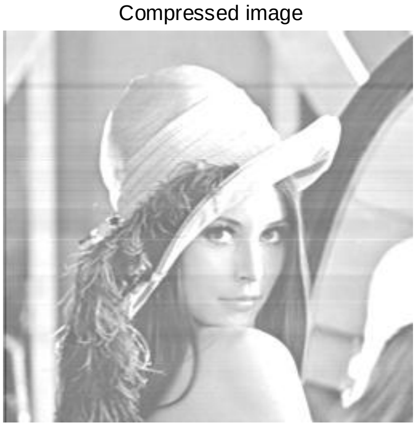
(a) 3 bit/pixel LMS



(b) 3 bit/pixel DPCM



(c) 1 bit/pixel LMS



(d) 1 bit/pixel DPCM

Fig 17: Visual results for processing Leena image with LMS and DPCM

6.3 GUI block diagrams for image compression using DPCM and LMS Algorithm

The basic block diagram of guide is shown below figures. It has been made by the help of buttons. Each button represents the relative output of the bits. When we click the button and we receive the output of related bit.

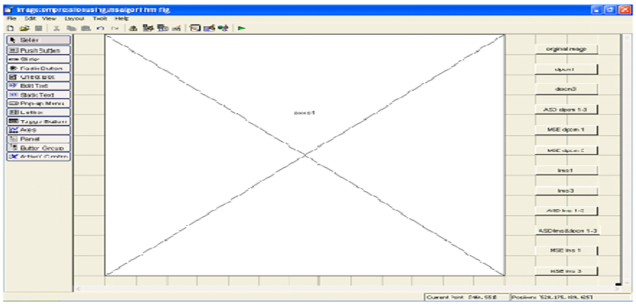


Fig 18: Guide basic block diagram

Fig 19: original image GUI diagram click original button

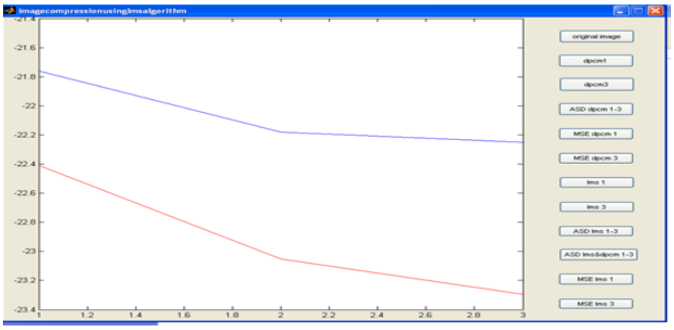


Fig 20: ASD LMS & DPCM 1-3 bit GUI diagram click on ASD LMS & DPCM button

7. Conclusion

This paper has used fixed weight coefficient DPCM and the LMS uses the adaptive coefficient for image compression with same distortion level. A comparison on using DPCM and using DPCM with LMS algorithm with respect to image compression has been carried out based on their coefficient and the number of bits. Results are presented which show LMS may provide more reduction in transmitted bit rate compared to DPCM when distortion levels are approximately the same for both methods. The results show that the LMS algorithm has the least computational complexity but more reduction in compressed image compare to DPCM with same distortion. The LMS can be used in fixed bit rate environments to decrease the reconstructed Image size and distortion.

8. REFERENCES

[1] B. P. Lathi and Zhi ding "Modern Digital and Analog Communication Systems" International Fourth Edition. New York Oxford University Press-2010, pp.292.

[2] J. E. Modestino, and D. G. Daut, "Source-channel coding of images," IEEE Trans. Commun., vol. COM-27, pp. 1644-1659, Nov. 1979.

[3] A. Habbi, "Comparison of Nth-order DPCM encoder with linear transformation and block quantization techniques," IEEE Trans. Commun., vol. COM-19, pp. 948-956, Dec. 1971.

[4] S.Haykin and T.Kailath "Adaptive Filter Theory" Fourth Edition. Prentice Hall, Pearson Eduaction 2002.

[5] J. R. Zeidler et al., "Adaptive enhancement of mulyiple sinusoids in uncorrelated noise," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 240-254, June 1978.

[6] J. G. Prokis, Digital Communications. New York: McGraw-Hill, 1983.

[7] S. T. Alexander and S. A. Rajala, "Analysis and simulation of an adaptive image coding system using the LMS algorithm," in Proc. 1982 IEEE Int. Conf. Acoust., Speech Signal Processing, Paris, France, May 1982.

[8] J. E. Modestino, and D. G. Daut, "Source-channel coding of image," IEEE Trans. Commun., vol. COM-27, pp. 1644-1659, Nov1979.

[9] M.D. Paez, and T. H. Glisson, "Minimum mean-square error quantization in speech PCM and DPCM systems," IEEE Trans. Commun. Vol. COM-20, pp.225-230, Apr. 1972.

[10] W. K. Pratt, Digital Image Processing. New York: Wiley, 1978.

[11] Prema Karthikeyan, Narayanan Sreekumar, "A Study on Image Compression with Neural Networks Using Modified Levenberg Maruar d t Method" Global Journal of Computer Science and TechnologyVolume 11 Issue Version 1.0 March 2011. Online ISSN: 0975-4172 & Print ISSN: 0975-4350

[12] Chun-Chieh Tseng, and Jer-Guang Hsieh, "Study on Huber Fractal Image Compression" De-partment of Computer Science, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 18, NO. 5, MAY 2009.

[13] R. Distasi, M. Nappi, and D. Riccio, "A range/domain approximation error-based approach for fractal image compression," IEEE Trans. Image Process., vol. 15, no. 1, pp. 89–97, Jan. 2006.

[14] Xiao-Yan xu, Philip Chen, Juan dai, "Hybrid encoding analysis of fractal image compression method based on wavelet transform" Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008.

[15] Pusit Borwonwatanadelok, Boonchana Purahong, Somkait Udomhunsakul, "Selection of Wavelet Filters for Panoramic Dental X-Ray Image Compression" Authorized licensed use limited to: University of Allahabad on November 15, 2009 IEEE Xplore. Restrictions apply.

[16] Mohammad Kabir Hossain1, Shams MImam, Khondker Shajadul Hasan , William Perrizo, "A Lossless Image Compression Technique Using Generic Peano Pattern Mask Tree" Proceedings of th International Conference on Computer and Information Technology (ICCIT 2008) 25-27 December, 2008, Khulna, Bangladesh.