ISTP
Research Publication

INTERNATIONAL SOCIETY OF THESIS PUBLICATION
A Society of Research Publication

IMAGE COMPRESSION USING DPCM WITH LMS ALGORITHM

Ranbeer Tyagi
B.E.(EC), M.E.(ET)

www.istp.org.in

ABSTRACT

The image compression in telecommunication is very demanding application to control the distortion in image in channel after transmitting the image.

The Differential pulse code modulation (DPCM) may be used to remove the unused bit in the image for image compression. In this thesis we compare the compressed image for 1, 2, 3, bits (2, 4 and 8 quantization levels, respectively), estimation error and average square distortion using DPCM with fixed coefficient and using DPCM with LMS algorithm. The LMS algorithm may be used to adapt the coefficients of an adaptive prediction filter for image source encoding. Results are presented which show LMS may provide almost 2 bit per pixel reduction in transmitted bit rate compare to DPCM when distortion levels are approximately the same for both methods. Alternatively, LMS can be used in fixed bit-rate environments to decrease the reconstructed image distortion and prediction mean square error. When compared with fixed coefficient DPCM and adaptive coefficient LMS, reconstructed image distortion is reduced and the prediction mean square error is reduced using DPCM with LMS.

This thesis is presenting the performance of using DPCM, using DPCM with LMS algorithm for image compression. Real data for leena.PNG image signal. The performance of these methods for image compression has been verified via computer simulations using MATLAB.

INDEX

CHAPTER - 4

DATA COLLECTION /OBJECTIVE

CHAPTER - 5

ANALYSIS OF DATA DEPLOYMENT OF MODEL

CHAPTER - 6

RESULTS & DISCUSSION/INTERPRETATION OF RESULT

CHAPTER - 7

CONCLUSION OF THE STUDY & SCOPE FOR FUTHER STUDY

LIST OF FIGURES

ABBREVIATIONS AND ACRONYMS

LMS : Least mean square

DPCM : Differential pulse code modulation

RD : Rate distortion

FIR : Finite impulse response

IIR : Infinite impulse response

MSE : Mean square error

PMSE : Prediction mean square error

A/D : Analog to digital Converter

PCM : Pulse code modulation

PE : Prediction error

ASD : Average square distortion

DL : Distortion level

CHAPTER - 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] (Redundancy: If m1 and m2 are the number of words to represent the same information then the redundancy is given as a function of compression ratio.) 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.

A two-layer image compression device is disclosed with a halftone circuit, an inverse halftone circuit and a quantization circuit. In this circuit, the halftone circuit converts the input gray-scale image into a binary image and rearranges the binary image output sequence to serve as a base layer of the input gray-scale image. The inverse halftone circuit recovers a predicted image from the binary image using the LMS algorithm. The quantization circuit then compares the input gray-scale image with the predicted image and encodes the difference between them to obtain an enhancement layer of the input gray-scale image. In DPCM, a prediction of the next sample value is formed from past values. This prediction can be thought of as instruction for the quantizer to conduct its search for the next sample value in a particular interval. By using the redundancy in the signal to form a prediction, the region of uncertainty is reduced and the quantization can be performed with a reduced number of decisions (or bits) for a given quantization level or with reduced quantization levels for a given number of decisions(or bits). The reduction in redundancy is realized by subtracting the prediction from the next sample value. This difference is called the prediction error.

1.2 WHAT IS AN IMAGE?

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

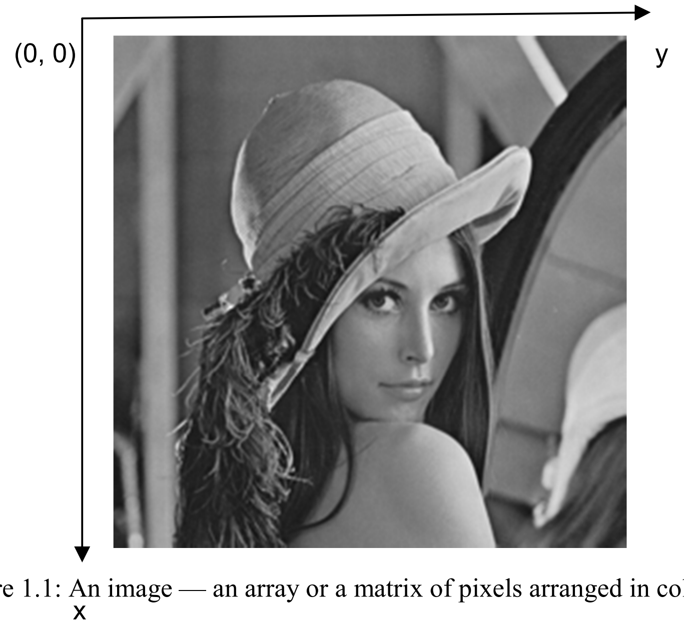


Figure 1.1: An image — an array or a matrix of pixels arranged in columns and rows.

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.

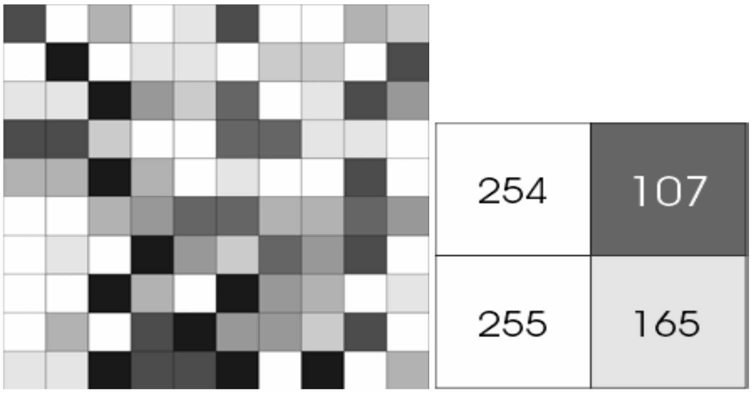


Figure 1.2: Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 tones or grayscales.

A normal grayscale image has 8 bit colour depth = 256 grayscales. A "true colour" image has 24 bit colour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours.

1.3 THE USUAL STEP OF IMAGE COMPRESSION

1. Specifying the Rate (bits available) [2] parameter for the target image.
2. Dividing the image data into various classes, based on their importance.
3. Dividing the available bit budget among these classes, such that the distortion is a minimum.
4. Quantize each class separately using the bit allocation information derived in step 3.
5. Encode each class separately using an entropy coder and write to the file.

1.4 BIT ALLOCATION

The first step in compressing an image is to segregate the image data into different classes. Depending on the importance of the data it contains, each class is allocated a portion of the total bit budget, such that the compressed image has the minimum possible distortion. This procedure is called Bit Allocation.

The Rate-Distortion theory is often used for solving the problem of allocating bits to a set of classes, or for bit rate control in general. The theory aims at reducing the distortion for a given target bit rate [2], by optimally allocating bits to the various classes of data. Initially, all classes are allocated a predefined maximum number of bits.

1. For each class, one bit is reduced from its quota of allocated bits, and the distortion due to the reduction of that 1 bit is calculated.
2. Of all the classes, the class with minimum distortion for a reduction of 1 bit is noted, and 1 bit is reduced from its quota of bits.
3. The total distortion for all classes D is calculated.
4. Compare the target rate and distortion specifications with the values obtained above. If not optimal, go to step 2.

In the approach explained above, we keep on reducing one bit at a time till we achieve optimality either in distortion or target rate, or both [2]. An alternate approach is to initially start with zero bits allocated for all classes, and to find the class which is most 'benefitted' by getting an additional bit. The 'benefit' of a class is defined as the decrease in distortion for that class.

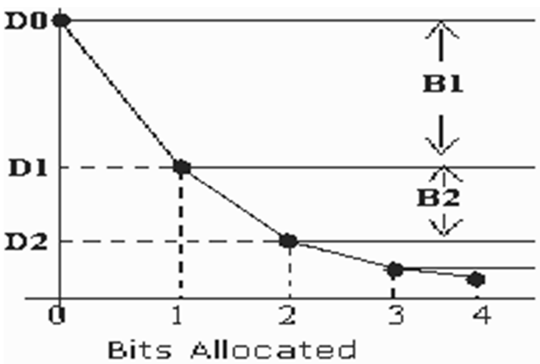


Figure 1.3: 'Benefit' of a bit is the decrease in distortion due to bit.

As shown above, the benefit of a bit is a decreasing function of the number of bits allocated previously to the same class. Both approaches mentioned above can be used to the Bit Allocation problem.

1.5 IMAGE REPRESENTATION

An image defined in the "real world" is considered to be a function of two real variables, for example, $f(x, y)$ with f as the amplitude (e.g. brightness) of the image at the *real* coordinate position (x, y). The effect of digitization [3] is shown in Figure 1.4.

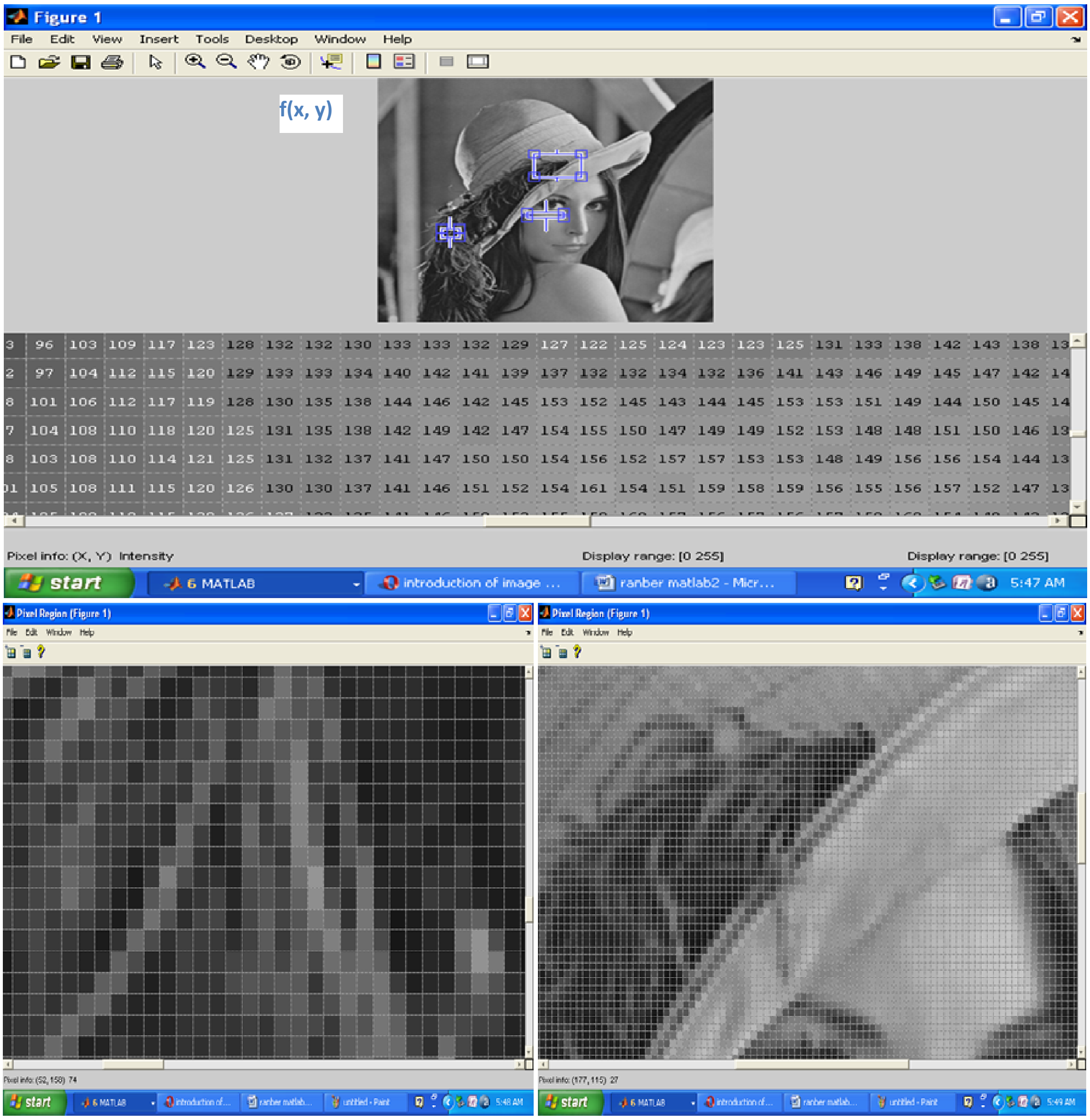


Figure 1.4: Image representation

The 2D continuous image $f(x, y)$ is divided into N *rows* and M *columns*. The intersection of a row and a column is called as *pixel*. The value assigned to the integer coordinates $[m, n]$ with $\{m=0,1, 2,...,M-1\}$ and $\{n=0,1,2,...,N-1\}$ is $f[m, n]$.In fact, in most cases $f(x, y)$ which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF, PNG etc., has some header and picture information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

1.6 THE PRINCIPLE OF IMAGE COMPRESSION

In a communication environment, the difference between adjacent time samples for image is small, coding techniques have involved 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 figure 1.5. 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 loop is

$$e(n) = x(n) - y(n) \quad 1.1$$

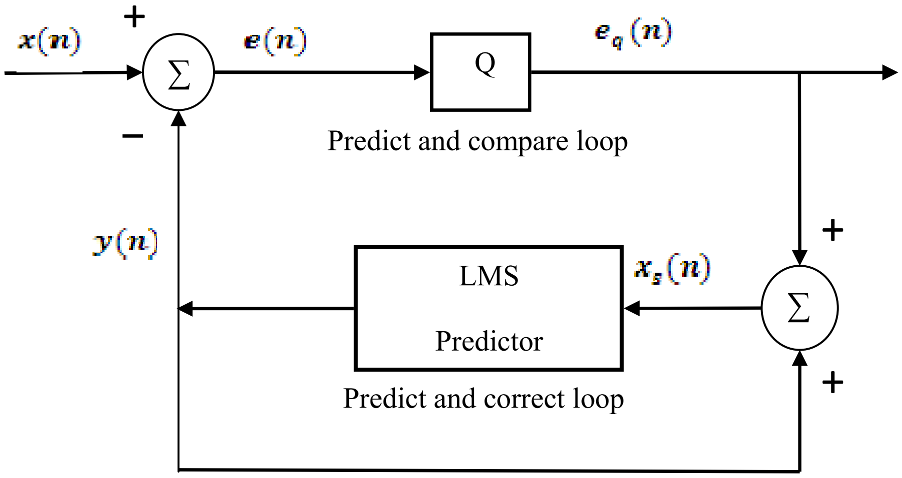


Figure1.5: N-tap predictive differential pulse code modulator (DPCM) for image compression system

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 figure 1.5. The encoder corrects its prediction by forming the sum of its prediction and the prediction error.

$$e_q(n) = quant\,[e(n)] \quad 1.2$$

$$x_s(n) = y(n) + e_q(n) \quad 1.3$$

Where quant (.) represents the quantization operation, $e_q(n)$ is the quantization 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.

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) [4]. If the prediction model forms predictions that are close to the actual sample values, the residues will exhibit reduced variance relative to the original signal. Thus the reduced variance sequence of residues can be moved through the channel with a reduced data rate.

The predictive converters must have a short –term memory that supports the real-time operations required for prediction algorithm. In addition, they will often have a long-term memory that supports the slow time, often data-dependent operations, such as automatic gain control and filter adjustments. Predictors that incorporate the slower, data-dependent adjustment algorithms are called adaptive predictors. Adaptive predictors use the theory of adaptive filters.

In an image compression, a model of predict and correct loop may vary continuously hence the model to be updated continuously. This is done by adaptive filtering algorithms.

CHAPTER – 2

LITRETURE REVIEW & CRITIQUE

In order to understand the content presented in this thesis it is first necessary to provide some background information regarding digital image signal theory. It will start out rather elementary and then progress to more complicated matters. Later chapters will build upon this theoretical basis in the derivation and implementation of the differential pulse code modulation and the adaptive filtering technique used in image compression.

2.1 DIGITAL IMAGE REPRESENTATION

The important aspect in digital image processing is image representation [1]. Any monochrome image can be represented by means of a two-dimensional light intensity function f(x, y), where x and y denotes spatial coordinates and the value of x at any point (x, y) is the gray level or the brightness of the image at that point. The axis convention used to represent the image is shown in figure 2.1. The origin is taken at the top left corner and the horizontal line and the vertical line through the origin are taken as y and x axes, respectively.

Figure 2.1: Coordinates, conventions and image

The monochrome image f(x, y) is discretized both in spatial coordinates and gray level values to obtain the digital image. A digital image can be represented as a matrix whose rows and columns are used to locate a point in the image and the corresponding element values give the gray level at that point. Each element in this matrix/digital array is called as image elements or pixels. A typical digital image of size N × M is represented as given in equation 2.1.

$$f(x, y) = \begin{pmatrix} f(0,0) & f(0,1) & \cdots & f(0, M-1) \\ \vdots & \vdots & \ldots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1, M-1) \end{pmatrix} \qquad 2.1$$

The images we normally perceive in our daily visual activities consist of light reflected from objects. Hence the function f (x, y) may consist of two components

1. The amount of light incident on the scene being viewed.
2. The amount of light reflected by the object in the scene.

The light incident and reflected can be denoted as i(x, y) and r(x, y), respectively. Then the image function f(x, y) is nothing but the product of i(x, y) and r(x, y) and the same is given in equation 2.2.

$$f(x, y) = i(x, y) \times r(x, y) \qquad 2.2$$

Where $0 < i(x, y) < \infty$ and $0 < r(x, y) < 1$.

2.2 SAMPLING AND QUANTIZATION

Sampling and quantization are the two important processes used to convert continuous analog image into digital image. Image sampling refers to discretization(A process in which signals and data samples are considered at regular intervals) of spatial coordinates where as quantization refers to discretization of gray level values. Normally, the sampling and quantization [1] deal with integer values. After the image is sampled, with respect to x and y coordinates the number of samples used along the x and y directions are denoted as N and M, respectively. The N and M are usually the integer powers of 2.Hence N and M can be represented by the mathematical equation as follows:

$$M = 2^n \quad \text{and} \quad N = 2^k \qquad 2.3$$

Similarly, when we discretize the gray levels, we use the integer values and the number of integer values that can be used is denoted as G. The numbers of integer gray level values used to represent an image usually are integer powers of 2.

$$G = 2^m \qquad 2.4$$

Where m represents the number of bits used to represent a gray level value in the image. An image of size N × M consisting of NM pixels and the number of bits required to store a digital image can be given by the following equation:

$$b = M \times N \times m \qquad 2.5$$

If

$$M = N \text{ then } b = N^2 m \qquad 2.6$$

The number of pixels that can be accommodated in a unit area is called the resolution of an image.

2.3 TRANSVERSAL FIR FILTERS

A filter can be defined as a piece of software or hardware that takes an input signal and processes it so as to extract and output certain desired elements of that signal. There are numerous filtering methods, both analog and digital which are widely used. However, this thesis shall be constrained to adaptive filtering using a particular method known as transversal finite response (FIR) [5] filters.

The characteristics of a transversal FIR filter can be expressed as a vector consisting of values known as tap weights. It is these tap weights which determine the performance of the filter. These values are expressed in column vector form as,

$$w = [w_0, w_1, w_2, \ldots\ldots w_{M-1}]^T$$

This vector represents the impulse response of the FIR filter. The number of elements on this impulse response vector corresponds to the order of the filter, denoted in this thesis by the character M. The utilization of an FIR filter is simple; the output of the FIR filter at time n is determined by the sum of the products between the tap weight vectors, w_k and M time-delayed input values.

$$y(n) = \sum_{k=0}^{M-1} w_k \, x_s(n-k) \qquad 2.7$$

If these time delayed input values are expressed in vector form by the column vector

$$x_s(n) = [x_s(n), x_s(n-1), \ldots\ldots x_s(n-M+1)]^T \qquad 2.8$$

Figure 2.2 shows a block schematic of real transversal FIR filter, here the input values is denoted by $x_s(n)$, the filter order is denoted by M, and z^{-1} denoted a delay of one sample period.

Adaptive filters utilize algorithm to iteratively alter the values of the impulse response vector in order to minimize a value known as the cost function. The cost function, e(n) is a function of the difference between a desired output and the actual output of the FIR filter. This difference is known as the estimation error of the adaptive filter.

$$e(n) = x(n) - y(n) \qquad 2.9$$

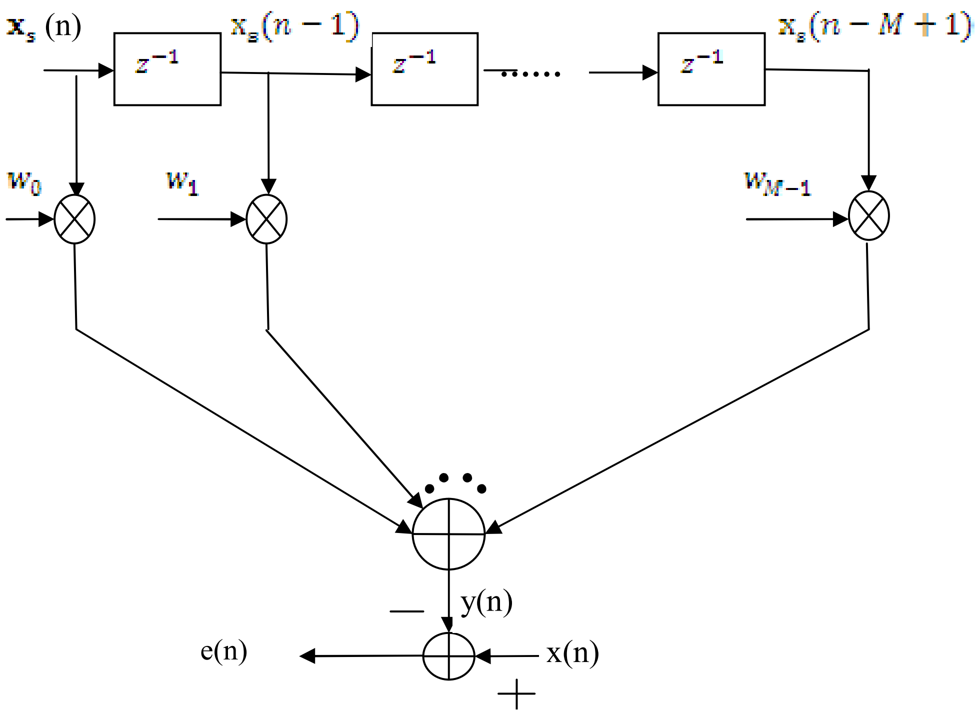


Figure 2.2: Transversal FIR filter

2.4 DPCM QUANTIZATION ERROR

A quantizer is defined by the number, size and location of its quantizing levels or step boundaries, and the corresponding step sizes. In a uniform quantizer, the step sizes are equal and are equally spaced. The number of levels N is typically a power of 2 and N $=2^b$, where b is the number of bits used in the conversion process. This number of levels is equally distributed over the dynamic range of possible input levels. Figure 2.3 shows the quantization process. Normally, this range is defined as $\pm I_{max}$. Such as $\pm 1.0V$ or $\pm 5.0V$. Thus according for the full range of I_{max} the size of quantization step is

$$q(n) = \frac{I_{max}}{2^b} = \frac{(simg)_{max}}{2^b} \qquad 2.10$$

Here $(simg)_{max}$ is maximum value of an image signal.

is quantized to yield $\quad e_q(n) = e(n) + q(n) \qquad 2.11$

Where $e_q(n)$ is the quantization signal.

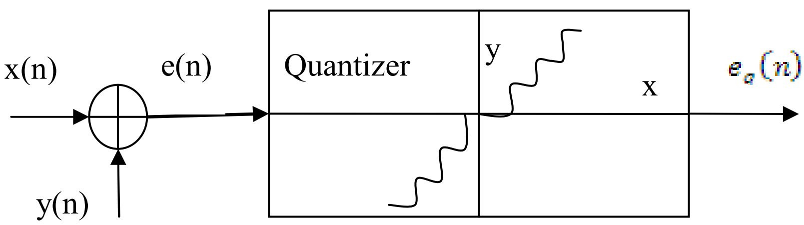


Figure 2.3: Process and model of quantization error

It is well known and easily shown [12] that the error between the original image and the reconstructed image at the receiver is simply the quantization error incurred from quantizing the prediction residual.Thus, the distortion between the original discrete image x(n) and the reconstructed value y(n) at the receiver is given by

$$q(n) = d(n) = y(n) - x(n) = e_q(n) - e(n) \qquad 2.12$$

(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. 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, in this thesis 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 [6] adaptive predictor reduces the average number of bits per image while maintaining an acceptable visual appearance at the receiver.

2.5 ADAPTIVE FILTER STRUCTURE AND ALGORITHM

Essentially the performance of an image compression depends on the selection of the adaptive filter structure and algorithm for the adaptation. In other words, the adaptive filter structure and the algorithm used determine the accuracy in estimating the reconstructed image path and the speed to adapt to its variation. In this investigation, a finite impulse response (FIR) transversal filter structure is used. The structure is chosen in favor of its counterpart infinite impulse response (IIR) [5] filters because of its convergence superiority and stability.

The adaptive algorithm on the other hand, adjusts the weight coefficients in the filter to minimize the error e(n). Common adaptive algorithm is the Least Mean Square (LMS) Algorithm.

2.6 WIENER FILTERS

Wiener filters [5] are a special class of transversal FIR filters which build upon the mean square error cost function to arrive at an optimal filter tap weight vector which reduces the MSE signal to a minimum. Consider the output of the transversal FIR filter as given below, for a filter tap weight vector, w and input vector, x_s(n).

$$y(n) = \sum_{k=0}^{M-1} w_k(n)\, x_s(n-k) \qquad 2.13$$

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

$$y(n) = \boldsymbol{w^T(n) x_s(n)} \qquad 2.14$$

Where the subscript T denotes the transpose of the w(n) . w(n) is the weight vector

$$\text{Where } \boldsymbol{w(n)} = [w_0(n) w_1(n) w_2(n) \ldots \ldots w_{M-1}(n)]^T \qquad 2.15$$

and x_s(n) is the tap input vector And

$$\boldsymbol{x_s(n)} = [x_s(n), x_s(n-1), \ldots \ldots, x_s(n-M+1)]^T \qquad 2.16$$

The mean square error cost function can be expressed in terms of the Cross-correlation vector between the desired and input signals,

$$\boldsymbol{P} = E[x_s(n) x(n)] \qquad 2.17$$

and the autocorrelation matrix of the input signal is given as

$$\boldsymbol{R} = E[x_s(n) {x_s}^T(n)] \qquad 2.18$$

Prediction mean squared error is

$$J = E[e^2(n)] \qquad 2.19$$

$$= E[\{x(n) - y(n)\}^2]$$

$$= E[\{x(n) - \boldsymbol{w^T(n) x_s(n)}\}^2]$$

$$= E[x^2(n) - 2x(n) w^T(n) x_s(n) + w^T(n) x_s(n) {x_s}^T(n) w(n)]$$

$$= E[x^2(n)] - 2w^T(n) E[x_s(n) x(n)] + w^T(n) E[x_s(n) {x_s}^T(n)] w(n)$$

$$= E[x^2(n)] - 2w^T(n) \boldsymbol{P} + w^T(n) \boldsymbol{R} w(n) \qquad 2.20$$

The minimum value of J can be found by calculating its gradient vector related to the filter tap weights and equating it to 0.

$$\nabla J = 0 \qquad 2.21$$

$$\text{Where} \nabla = \left[\frac{\partial}{\partial w_0(n)} \; \frac{\partial}{\partial w_1(n)} \cdots \frac{\partial}{\partial w_{M-1}(n)}\right]^T \qquad 2.22$$

After differentiating equation 2.20 we obtain

$$2\boldsymbol{R} w_{opt} - 2\boldsymbol{P} = 0$$

$$w_{opt} = \boldsymbol{R}^{-1} \boldsymbol{P} \qquad 2.23$$

The optimal wiener solution is the set of filter tap weights which reduce the cost function to zero. This vector can be found as the product of the inverse of the input vector autocorrelation matrix and the cross correlation vector between the desired signal and the input vector which gives the filter coefficients of a Wiener Filter, optimal in the sense of the prediction mean square error (PMSE). The Wiener filter is a linear optimum filter. It depends on the known statistics **R** and **P**. In practice, we do not know **R** and **P** exactly, and in an adaptive context they may be slowly varying with time. The adaptive filter should be able to track the changes in the statistics hence a changing w_{opt}, so some approximations are necessary. One idea is to approximate the **R** and **P** values, which leads to the Recursive Least Squares (RLS) algorithm.

CHAPTER - 3

METHODOLOGY/ EXPERIMENTAL SETUP

3.1 METHODS OF IMAGE PROCESSING

There are two basic methods available in Image Processing.

3.1.1 ANALOG IMAGE PROCESSING

Analog Image Processing refers to the alteration of image through electrical means. The most common example is the television image.

The television signal is a voltage level which varies in amplitude to represent brightness through the image. By electrically varying the signal, the displayed image appearance is altered. The brightness and contrast controls on a TV set serve to adjust the amplitude and reference of the video signal, resulting in the brightening, darkening and alteration of the brightness range of the displayed image.

3.1.2 DIGITAL IMAGE PROCESSING

Digital image processing methods were introduced in 1920[1], when people were interested in transmitting picture information across the Atlantic Ocean. The time taken to transmit one image of size 256×256 was about a week. The pictures were encoded using specialized printing equipment and were transmitted through the submarine cable. At the receiving end, the coded pictures were reconstructed.

In this case, digital computers are used to process the image. The image will be converted to digital form using a scanner – digitizer [7] and then process it. It is defined as the subjecting numerical representations of objects to a series of operations in order to obtain a desired result. It starts with one image and produces a modified version of the same. It is therefore a process that takes an image into another.

The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer [8, 9]. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits.

The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision.

There are so many numbers of methods for image compression, however, in this thesis only two methods for image compression are described and compared. First one is fixed two coefficients DPCM system and second is an image compression using DPCM with Adaptive coefficients using LMS Algorithm. These methods are described below.

3.2 DIFFERENTIAL PULSE CODE MODULATION

3.2.1 BASIS IDEA OF DPCM

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. DPCM [10] is one such scheme.

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 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

We can improve upon this scheme by estimating (predicting) the value of nth sample x(n) from a knowledge of several previous sample values. If this estimate is y(n), then we transmit the difference (prediction error) e(n)=x(n)-y(n). At the receiver also, we determine the estimate y(n) from the previous sample values, and then generate x(n) by adding the received e(n) to the estimate y(n). 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],** 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).

3.2.2 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 $x_s(n-1), x_s(n-2), \ldots$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 $x_s(n-1), x_s(n-2), \ldots \ldots \ldots$The difference e(n)=x(n)-y(n) is now transmitted via PCM [10]. At the receiver, we can generate y(n), and from the received e(n), we can reconstruct $x_s(n)$.

Figure 3.1 is 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 $x_s(n)$.

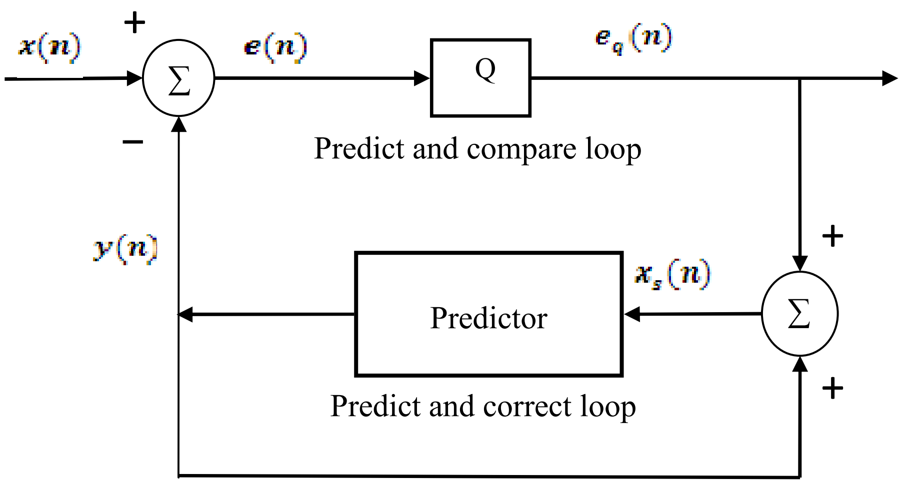


Figure 3.1: N-tap predictive differential pulse code modulator (DPCM)

The difference of the original image data, $x(n)$, and prediction image data, $y(n)$, is called estimation residual, $e(n)$. So

$$e(n) = x(n) - y(n) \quad 3.1$$

is quantized to yield

$$e_q(n) = e(n) + q(n) \quad 3.2$$

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

And

$$q(n) = e_q(n) - e(n) \quad 3.3$$

$$q(n) = \frac{\mathrm{I_{max}}}{2^{\mathrm{b}}} = \frac{(\mathrm{simg})_{\mathrm{max}}}{2^{\mathrm{b}}} \quad 3.4$$

Here b is number of bit. $\mathrm{I_{max}}$, $(\mathrm{simg})_{\mathrm{max}}$ is maximum value of an image signal.

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

$$x_s(n) = y(n) + e_q(n) \quad 3.5$$

$$= x(n) - e(n) + e_q(n)$$

$$= x(n) + q(n) \quad 3.6$$

This shows $x_s(n)$ 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 diagram.

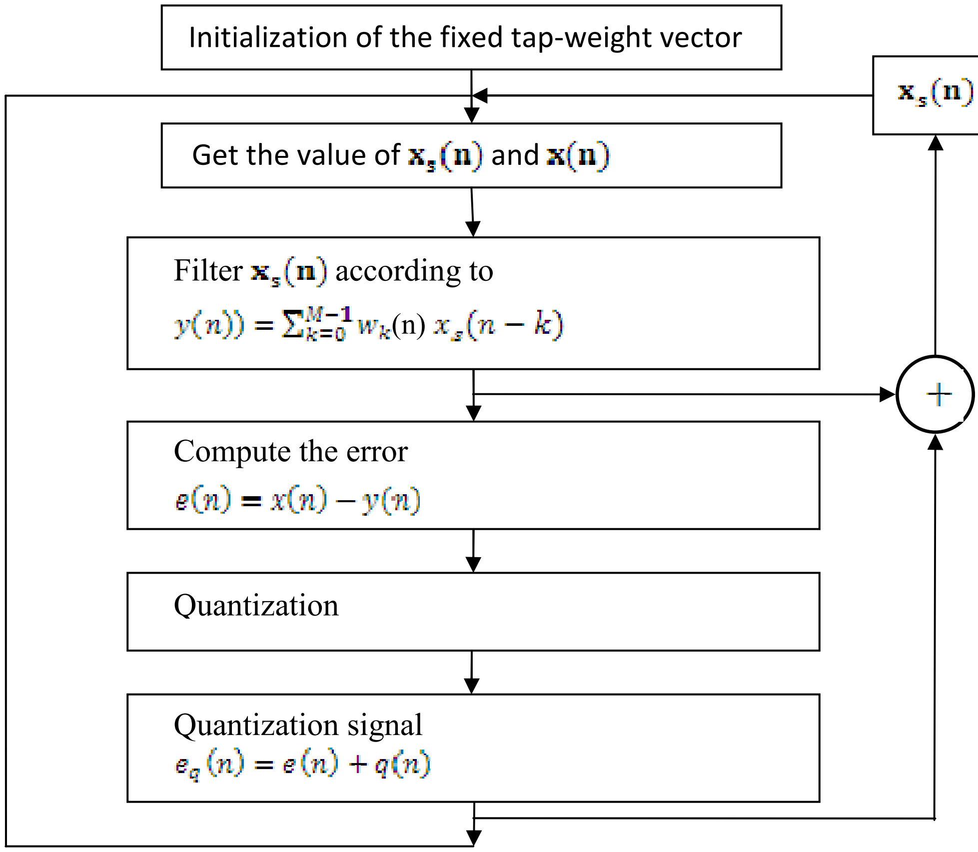


Figure 3.2: Flowchart of DPCM system

3.3 THE LEAST MEAN SQUARE ALGORITHM

3.3.1 INTRODUCTION OF LMS

The Least Mean Square (LMS) [2] algorithm was first developed by Widrow and Hoff in 1960 through their studies of pattern recognition (Haykin 1991, p. 67). From there it has become one of the most widely used algorithms in adaptive filtering. The LMS algorithm is a type of stochastic gradient-based algorithms as it utilizes the gradient vector of the filter tap weights to converge on the optimal Wiener solution. It is well known and widely used due to its computational simplicity. It is this simplicity that has made it the standard against which all other adaptive filtering algorithms are benchmarked. If the image information is to be sent over a channel with a fixed number of bits per transmitted sample then the reconstruction made using LMS has significantly less distortion than that obtained using a fixed coefficient DPCM predictor.LMS algorithm consists of two basis processes

1. A filtering process- which involves computing and outputting the output y(n) of the transversal filter in response to an input signal and generating an estimation error by comparing this output with a desired input image signal x(n).

2. An adaptive process- which involves the automatic adjustment of the parameters of the filter in accordance with the estimation error.

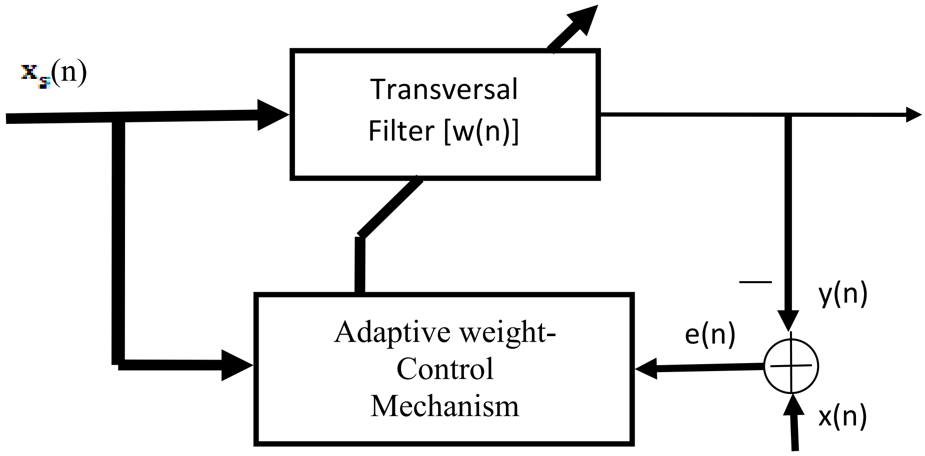


Figure 3.3: Block diagram of Adaptive Transversal Filter

With each iteration of the LMS algorithm [11], the filter tap weights of the adaptive filter are updated according to

$$w(n+1) = w(n) + \mu e(n) x_s(n) \qquad 3.7$$

Where $w(n)$ is the tap weight vector at time n.

The parameter μ is known as the step size parameter and is a small positive constant. This step size parameter controls the influence of the updating factor. Selection of a suitable value for μ is imperative to the performance of the LMS algorithm, if the value is too small the time the adaptive filter takes to converge on the optimal solution will be too long; if μ is too large the adaptive filter becomes unstable and its output diverges.

It is noted that the existence of feedback e(n) in the LMS Algorithm may cause the algorithm to be unstable. Fortunately, the stability of the algorithm can be determined by the step-size parameter. The step size parameter should satisfy the following

$$0 < \mu < \frac{2}{S_{max}} \qquad 3.8$$

Where S_{max} is maximum value of the input signal power.

3.3.2 DERIVATION OF THE STANDERD LMS ALGORITHM

The derivation of the LMS algorithm builds upon the theory of the wiener solution for the optimal filter tap weightsw_{opt}. The LMS algorithm minimizes the expected value of the squared error. Thus the criterion function, mean squared error is

$$J = E[e^2(n)] \qquad 3.9$$

$$\nabla J = \nabla E[e^2(n)]$$

$$\nabla J = 2e(n)\nabla E[e(n)]$$

$$\nabla J = 2e(n)\nabla E[d(n) - w^T(n)x_s(n)]$$

$$\nabla J = -2e(n)x_s(n) \qquad 3.10$$

For simplicity, the tap input vector x(n) and the desired response x(n) are assumed to be jointly wide-sense stationary. With this assumption, the method of steepest descent can be used to compute a tap weight vector.

$$w(n+1) = w(n) - \mu\nabla J \qquad 3.11$$

$$w(n+1) = w(n) + 2\mu e(n)x_s(n) \qquad 3.12$$

For convenience, the factor two in equation 3.12 is absorbed into the constant µ yielding

$$w(n+1) = w(n) + \mu e(n)x_s(n) \qquad 3.13$$

The LSM algorithm has a correction factor of µe(n) x_s(n) to the tap weight vector w(n). One notable fact is that the correction factor is directly proportional to the tap input vector x_s(n) and hence when x_s(n) is large, the LMS algorithm faces a **gradient noise amplification** problem. This means the error in the gradient estimate gets magnified.The main reason for the LMS algorithms popularity in adaptive filtering is its computational simplicity, making it easier to implement than all other commonly used adaptive algorithms. For each iteration the LMS algorithm requires 2N additions and 2N+1 multiplications (N for calculating the output, y(n), one for 2µe(n) and an additional N for the scalar by vector multiplication).

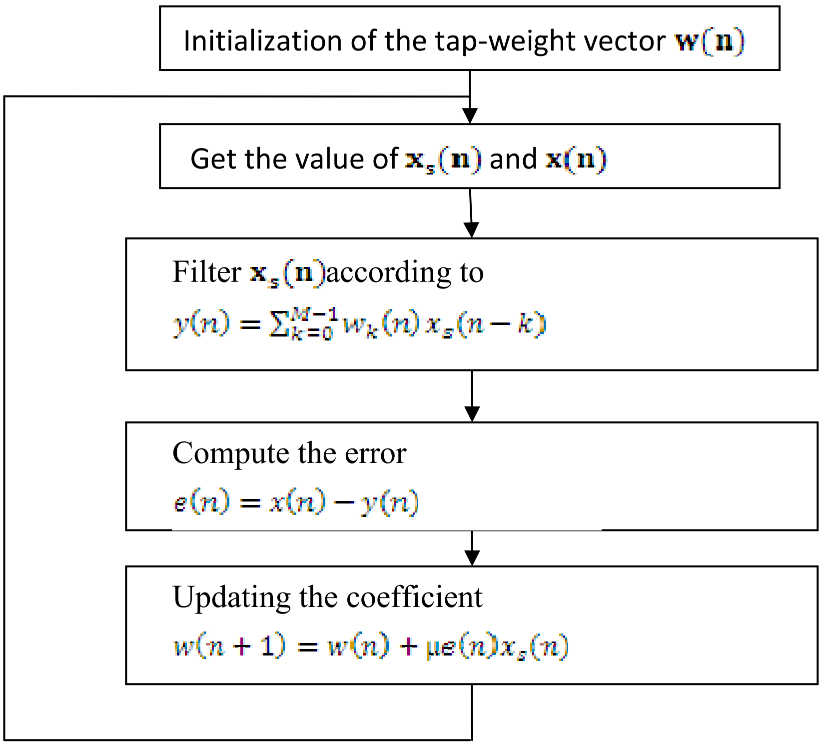


Figure 3.4: Flow chart of the Basic LMS Algorithm

3.3.3 IMAGE COMPRESSION USING DPCM WITH LMS ALGORITHM

A block diagram of the LMS adaptive image compression system is shown in figure 3.5. It is seen that the image prediction y(n) is formed in a linear manner at the output of the LMS filter:

$$y(n) = \sum_{k=0}^{M-1} w_k(n)\, x_s(n-k) \qquad 3.14$$

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

$$y(n) = w^T(n) x_s(n) \qquad 3.15$$

In equation 3.14, 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) [12], is computed

$$e(n) = x(n) - y(n) \qquad 3.16$$

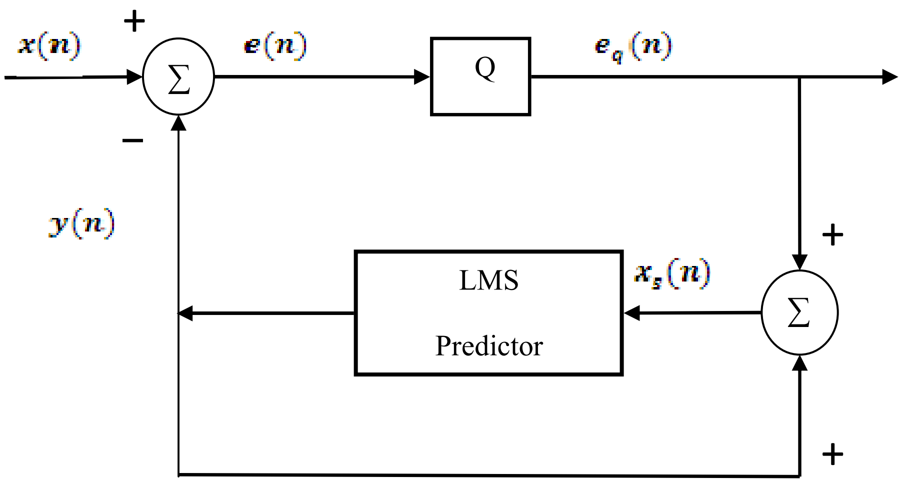


Figure 3.5: Block Diagram of image compression using DPCM with LMS Algorithm

This residual is then quantized to form $e_q(n)$ and this quantized residual is send to the receiver. The quantization residual is determine

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

The quantized residual is also used to update the predictor coefficient for the next iteration by the well known least mean squares (LMS) [11] algorithm.

$$w(n+1) = w(n) + \mu e_q(n) x_s(n) \qquad 3.17$$

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.

The flowchart of the least mean square adaptive algorithm using differential pulse code modulation quantization system as shown in figure 3.6

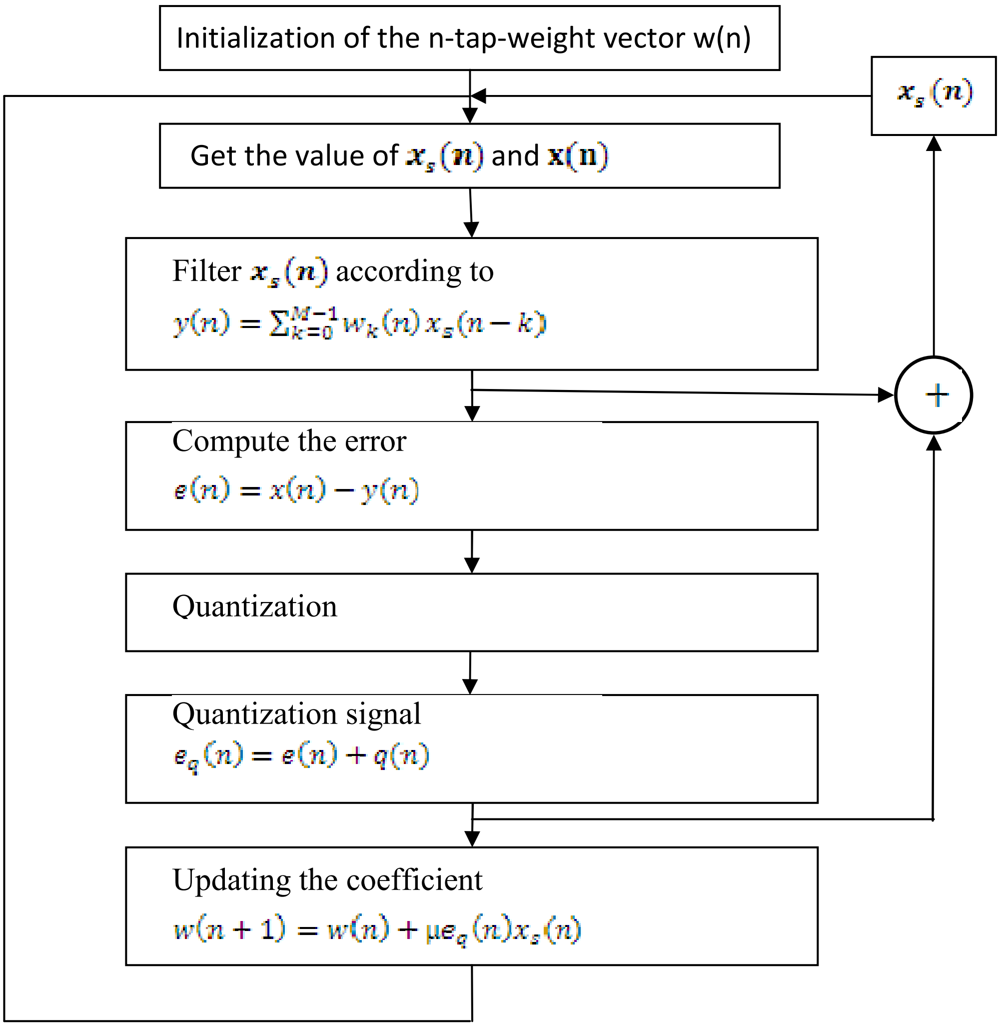


Figure 3.6: Flow chart of image compression using DPCM with LMS Algorithm

CHAPTER - 4

DATA COLLECTION /OBJECTIVE

4.1 SIMULATION ENVIRONMENT

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

4.1.1 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

4.1.2 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

CHAPTER - 5

ANALYSIS OF DATA DEPLOYMENT OF MODEL

5.1 SIMULATION ANALYSIS PARAMETER

In this thesis, the following parameters are used to analysis the image compression using DPCM and using DPCM with LMS.

5.1.1 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.

5.1.2 HISTOGRAM EQUALIZATION

Let r be the variable representing the gray levels in the image to be enhanced. Assume that the gray levels in this thesis image after normalization range from 0 to 1. For any value of r in the original image in the interval (0, 1) the transformation in the form

$$I=T(r)$$

Here I is produces a gray level. It is assumed that above equation satisfies the following two conditions:

1. T(r) is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$.
2. $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

The first condition preserves the order from black to white in the gray scale, whereas the second condition guarantees a mapping that is consistent with the allowed range of pixel values.

5.1.3 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

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

5.1.4 PREDICTION MEAN SQUARE ERROR

$$PMSE = e^2/samples$$

$$MSPEdB = 10log10(MSPE)$$

5.1.5 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).

5.1.6 AVERAGE SQUARE DISTORTION

$$ASD = mean(e_q^2)$$
$$ASDdB = 10log10(ASD)$$

CHAPTER - 6

RESULTS & DISCUSSION/INTERPRETATION OF RESULT

6.1 SIMULATION AND ANALYSIS

The DPCM image quantization [13,14] and the DPCM with LMS was simulated using Matlab 7.5 with respect to the application of image compression depicted in respectively figure 3.1, 3.5 chapter 3.

Simulations involving real image input signal consisted of 256×256 rows and columns image matrix displayed in figure 6.1.This image has been saved by the name of leena.png in MATLAB. So this image is .PNG format image.

The histogram of original image is shown in figure 6.2. 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. And histogram equalization is shown in figure 6.3 of this original image after normalization range.

Figure 6.4 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. The full MATLAB source code of the original image is include in Appendix B.1.

Figure 6.1: Original image with 256×256 matrix dimension

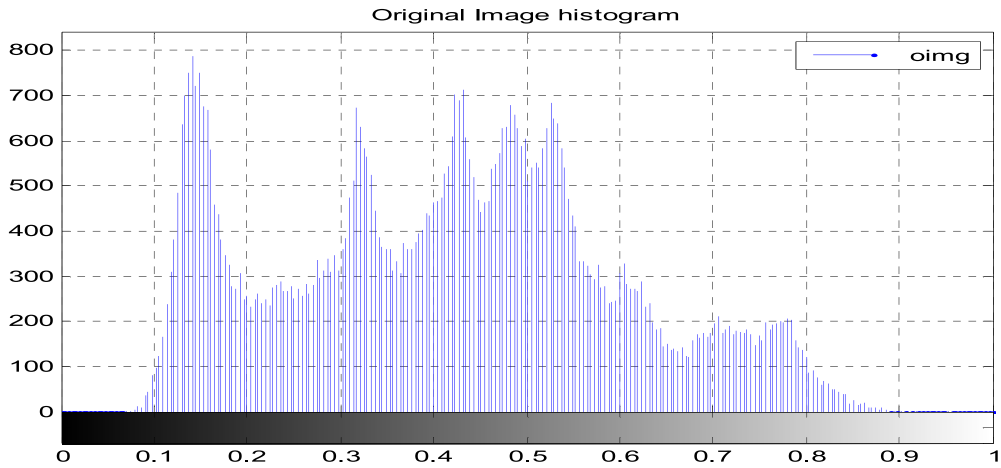


Figure 6.2: Original Lena image Or Dark image histogram

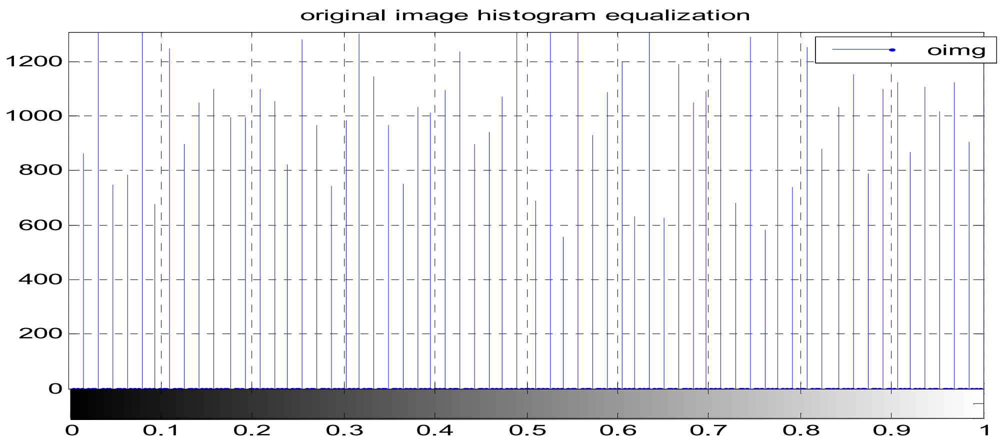


Figure 6.3: Histogram equalization original Lena image

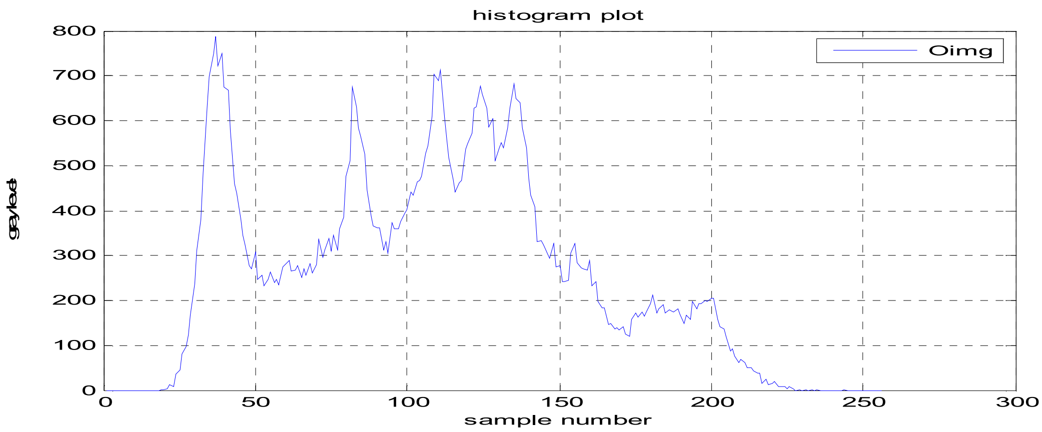


Figure 6.4: Histogram plot Original Lena image

6.2 SIMULATION RESULTS FOR IMAGE COMPRESSION USING DPCM

6.2.1 SIMULATION RESULTS OF IMAGE COMPRESSION FOR 1BIT DPCM

The DPCM was simulated using Matlab 7.5 with respect to the application of 1bit/pixel image compression. The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image pass with 1bit/pixel DPCM with fixed coefficient w [.495 .456] fixed 2taps. The full MATLAB source code of the DPCM is include in Appendix B.1.The simulation result shown in bellow figures.

Figure 6.2.1 shows the reconstructed 1bit/pixel image. This is a bright image. It is reduce approximately 13.5kb (13,840bytes) for these parameters and the reconstructed image size 83.0kb (85,075bytes). Figure 6.2.2 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. And histogram equalization is shown in figure 6.2.3 of this compressed image after normalization range.Figure 6.2.4 shows the histogram plot 1bit/pixel DPCM compressed Lena image this plot between gray level and samples. In this image samples range are from 0 to 255 but gray level changes according to sample value. Figure 6.2.5 shows the PMSE 1bit/pixel DPCM compressed Lena image. This shows the PMSE [dB] versus sample from 0 to 255.

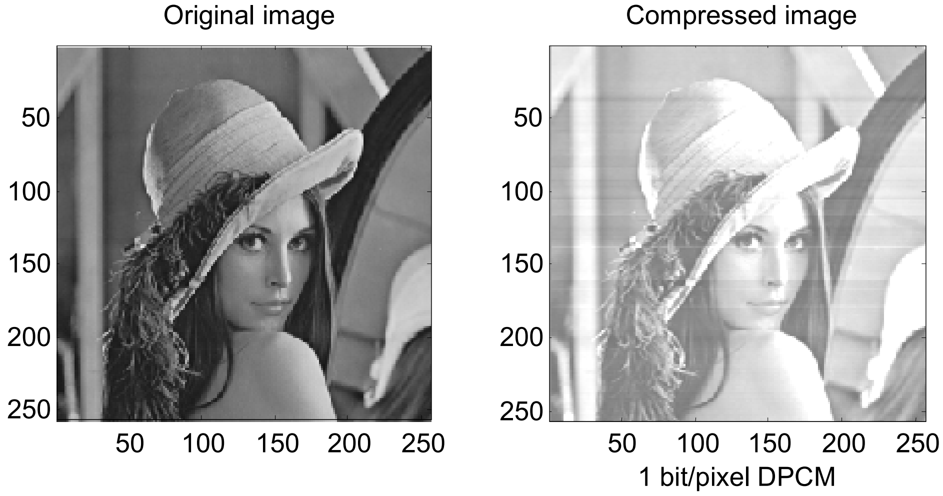


Figure 6.2.1: 1bit/pixel DPCM compressed Lena image

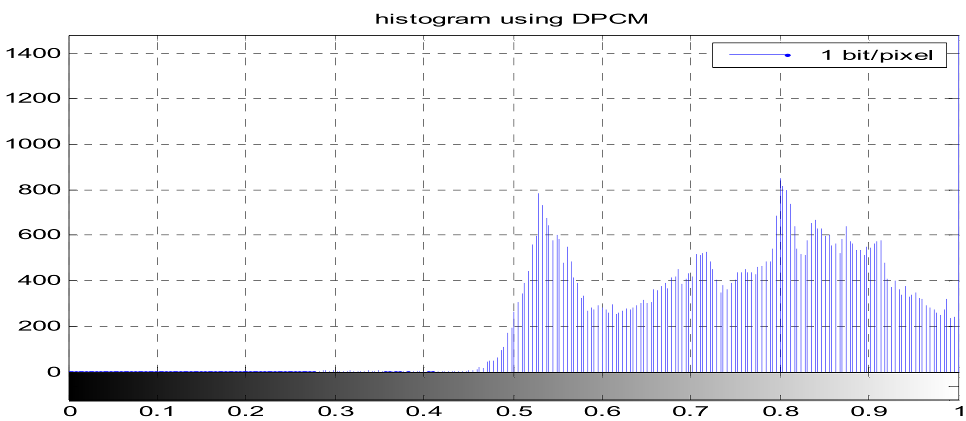


Figure 6.2.2: Histogram1bit/pixel DPCM compressed or bright Lena image

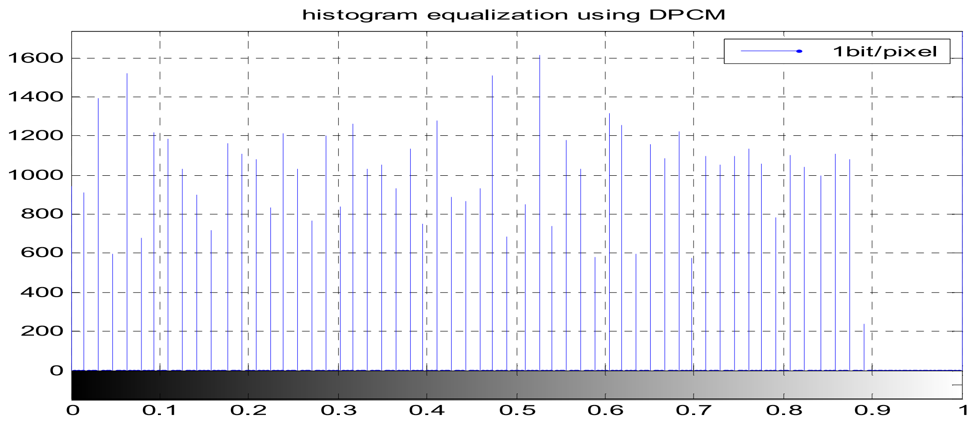


Figure 6.2.3: Histogram equalization 1bit/pixel DPCM compressed Lena image

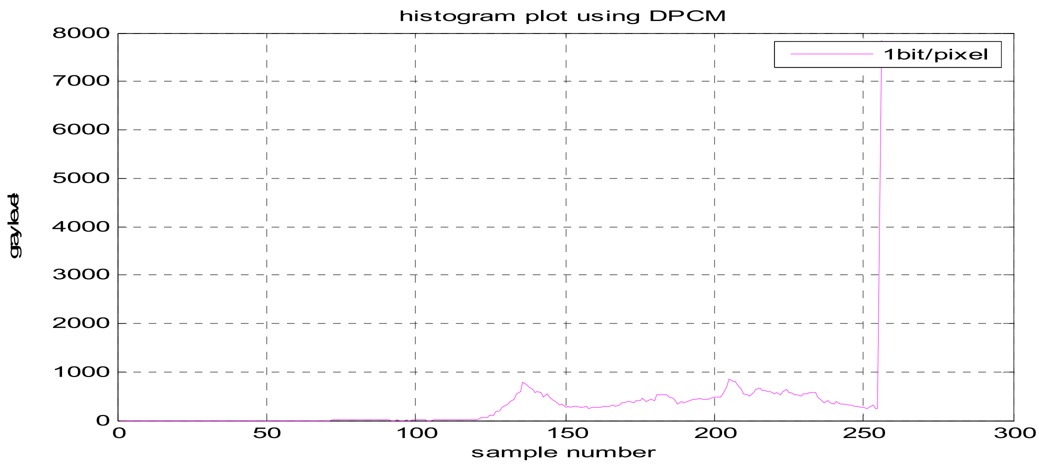


Figure 6.2.4: Histogram plot1bit/pixel DPCM compressed Lena image

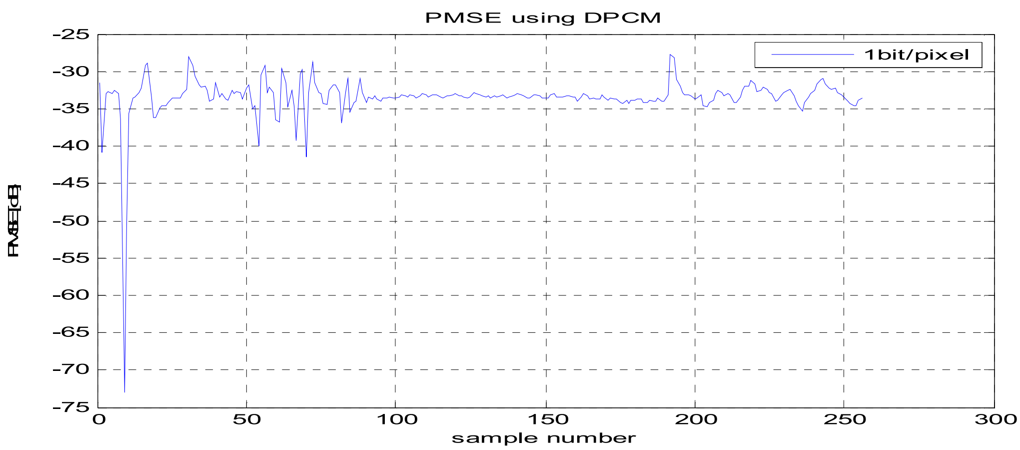


Figure 6.2.5: PMSE1bit/pixel DPCM compressed Lena image

6.2.2 SIMULATION RESULTS OF IMAGE COMPRESSION FOR 3BIT DPCM

The DPCM was simulated using Matlab 7.5 with respect to the application of 3bit/pixel image compression. The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image pass with 3bit/pixel DPCM with fixed coefficient w [.495 .456] fixed 2taps. The full MATLAB source code of the DPCM is include in Appendix B.2.The simulation result shown in bellow figures.

Figure 6.2.6 shows the reconstructed 3bit/pixel image. This is a dark image. This image was approximately same the original image. It is reduce approximately 6.4kb (8,672bytes) for these parameters and the reconstructed image size 88.1kb (90,243bytes). Figure 6.2.7 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. And histogram equalization is shown in figure 6.2.8 of this compressed image after normalization range.

Figure 6.2.9 shows the histogram plot 3bit/pixel DPCM compressed Lena image this plot between gray level and samples. In this image samples range are from 0 to 255 but gray level changes according to sample value. Figure 6.2.10 shows the PMSE 3bit/pixel DPCM compressed Lena image. This shows the PMSE [dB] versus sample from 0 to 255.

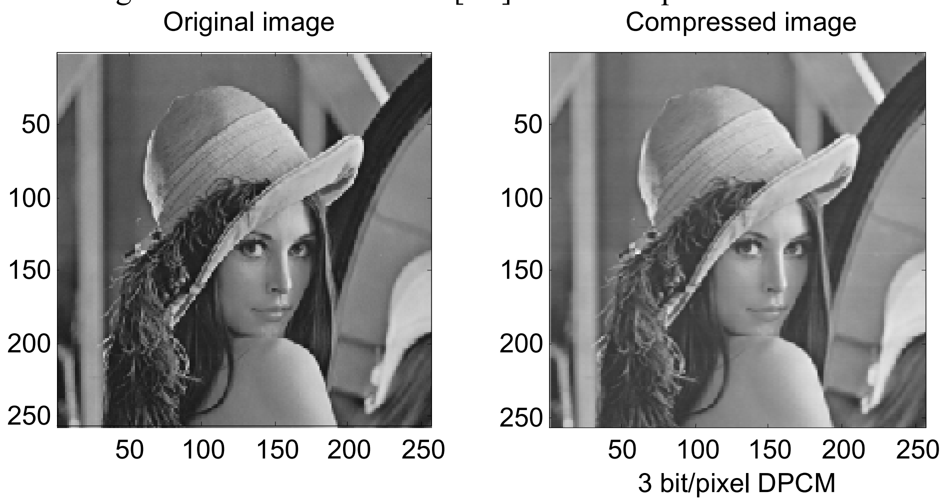


Figure 6.2.6: 3bit/pixel DPCM compressed Lena image

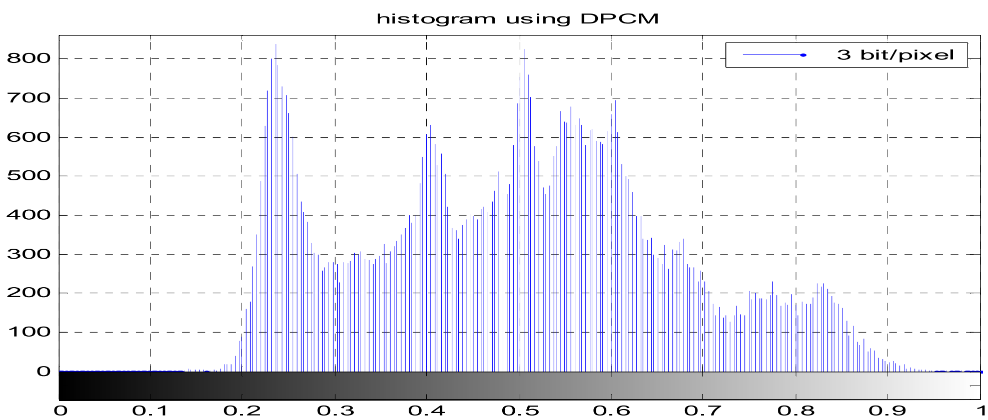


Figure 6.2.7: Histogram3bit/pixel DPCM compressed or Dark Lena image

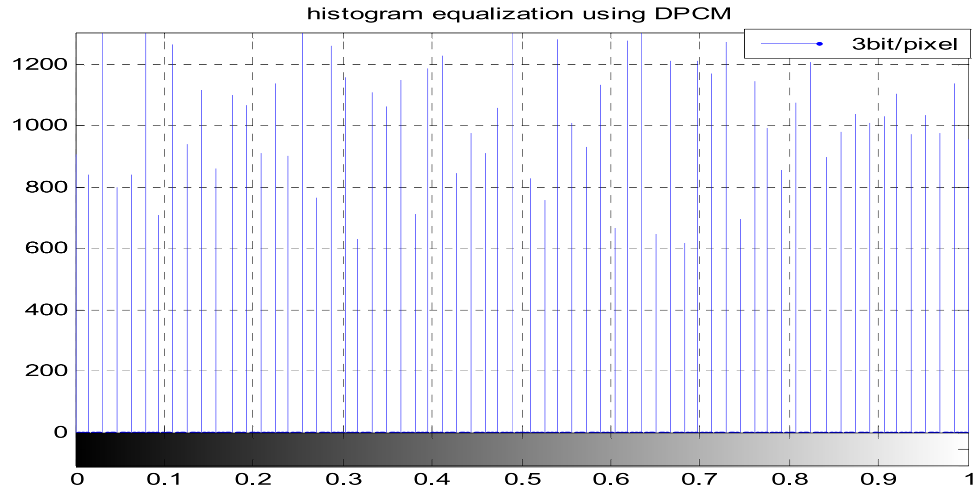


Figure 6.2.8: Histogram equalization 3bit/pixel DPCM compressed Lena image

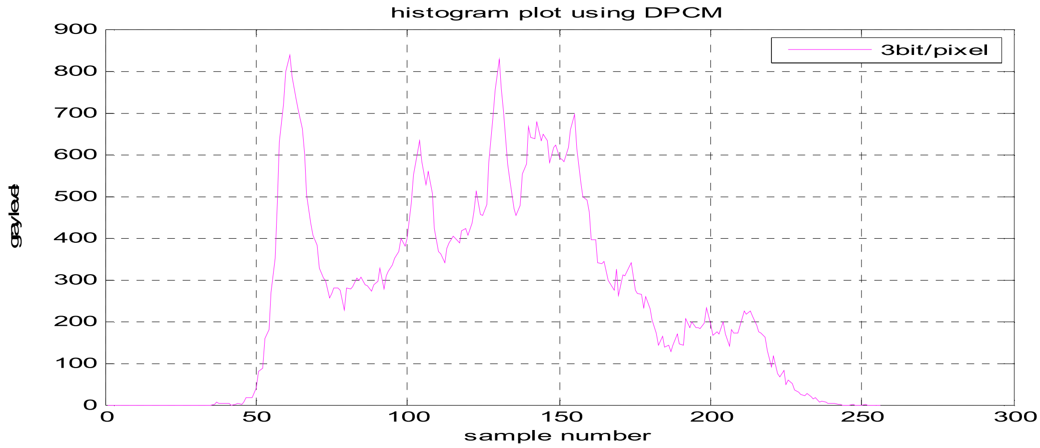


Figure 6.2.9: Histogram plot 3bit/pixel DPCM compressed Lena image

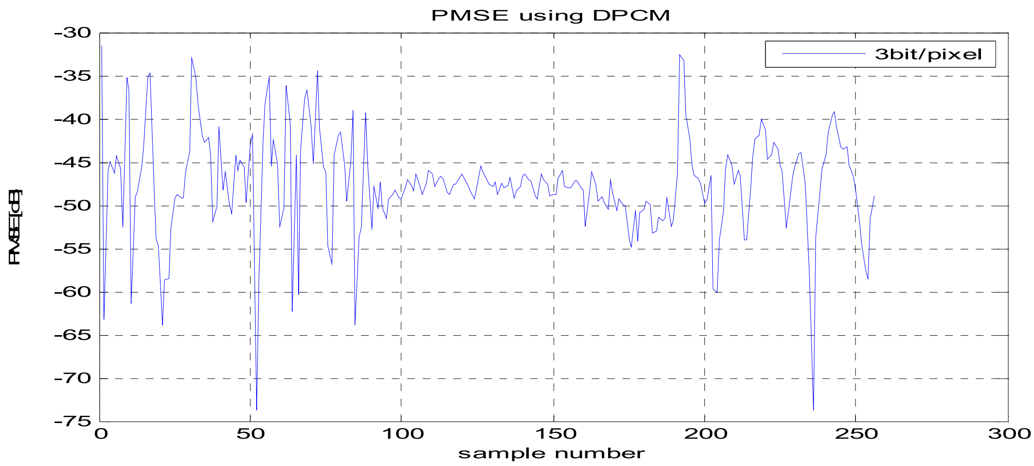


Figure 6.2.10: PMSE 3bit/pixel DPCM compressed Lena image

6.2.3 SIMULATION RESULTS OF IMAGE COMPRESSION FOR 1 AND 3 BIT'S DPCM COMPARISON

The DPCM was simulated using Matlab 7.5 with respect to the application of 1 and 3bit/pixel image compression comparison. The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image pass with 1 and 3bit/pixel DPCM with fixed coefficient w [.495 .456] fixed 2taps. The full MATLAB source code of the DPCM is include in Appendix B.3.The simulation result shown in bellow figures.

Figure 6.2.11(a) shows the reconstructed 1bit/pixel image. This is a bright image. It is reducing approximately 13.5kb (13,840bytes) respect to original image for these parameters and the reconstructed image size 83.0kb (85,075bytes). Figure 6.2.11(b) shows the reconstructed 3bit/pixel image. This is a dark image. This image was approximately same the original image. It is reduce approximately 6.4kb (8,672bytes) for these parameters and the reconstructed image size 88.0kb (90,243bytes). So we can say 1bit/pixel DPCM image reduction was more compare to 3bit/pixel DPCM. But the distortion of 1 bit/pixel DPCM is compare to 3bit/pixel DPCM. This has been showed in figure 6.2.12 and the figure 6.2.13 shows histogram plot comparison between gray label and sample number. The PMSE in 3bit/pixel DPCM less 9-10 dB compare to 1bit/pixel DPCM. This has been showed in figure 6.2.14. This figure shows the PMSE [dB] versus sample from 0 to 255.

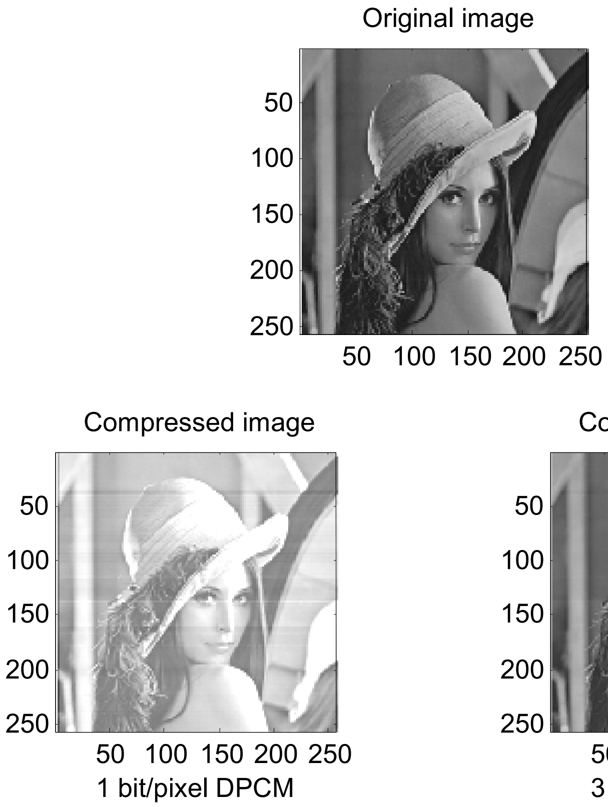


Figure 6.2.11: Visual results for processing Lena image using DPCM

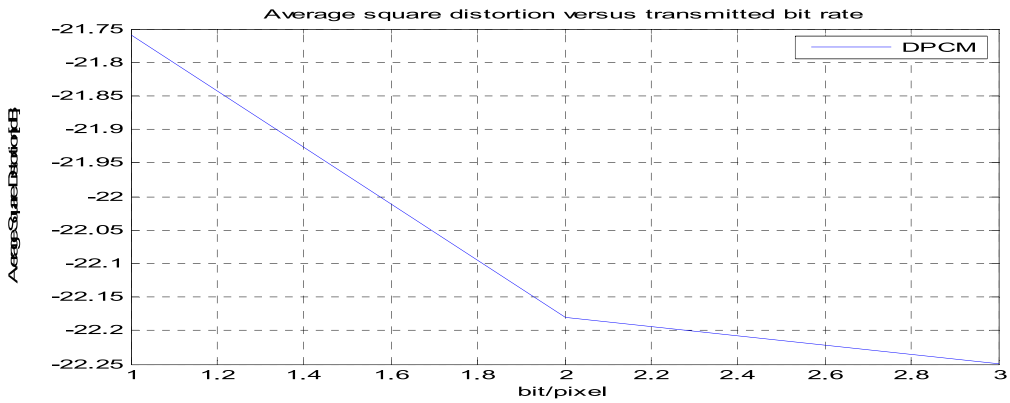


Figure 6.2.12: Average square distortion versus transmission bit rate using DPCM.

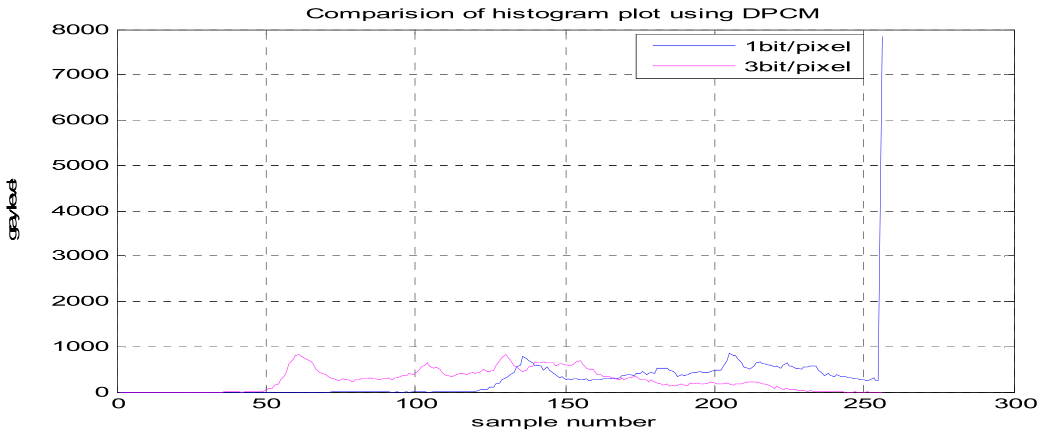


Figure6.2.13: 1, 3 bit/pixel comparison of histogram using DPCM

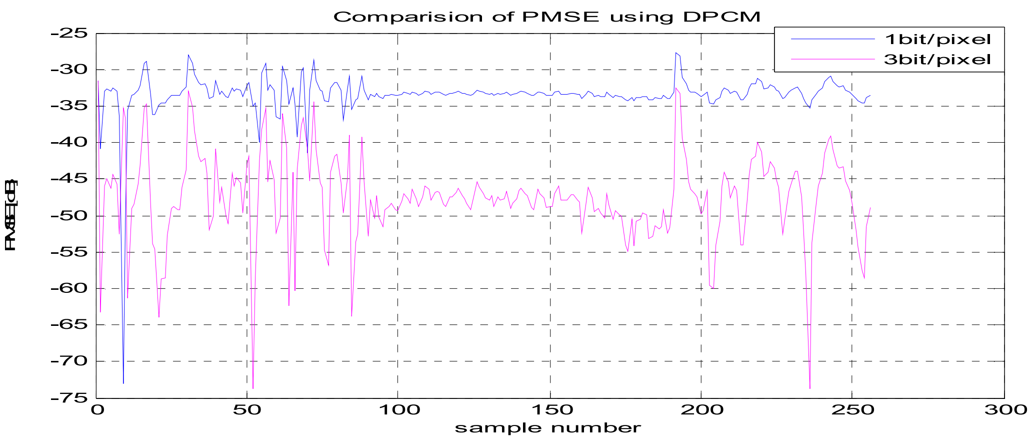


Figure6.2.14: 1, 3 bit/pixel comparison of PMSE using DPCM

6.3 SIMULATION RESULTS FOR IMAGE COMPRESSION USING DPCM WITH LMS ADAPTIVE ALGORITHM

6.3.1 SIMULATION RESULTS OF IMAGE COMPRESSION FOR 1 BIT LMS

The LMS algorithm was simulated using Matlab with respected to the application of 1bit/pixel image compression using DPCM with LMS algorithm depicted in figure 3.5 in chapter3. 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 .0005 and bit set to be 1bit/pixel for LMS. The full MATLAB source code of the LMS Algorithm is included in Appendix C.1.

The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image passed with 1bit/pixel using DPCM with LMS algorithm adaptive coefficient w .The simulation result shown in bellow figures.

Figure 6.3.1 shows the reconstructed 1bit/pixel image. This is a bright image. It is reduce approximately 23.3kb (23,955bytes) for these parameters and the reconstructed image size 73.2kb (74,960bytes). Figure 6.3.2 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. And histogram equalization is shown in figure 6.2.3 of this compressed image after normalization range. Figure 6.3.4 shows the histogram plot 1bit/pixel using DPCM with LMS algorithm compressed Lena image this plot between gray level and samples points. In this image samples range are from 0 to 255 but gray level changes according to sample value. Figure 6.3.5 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.

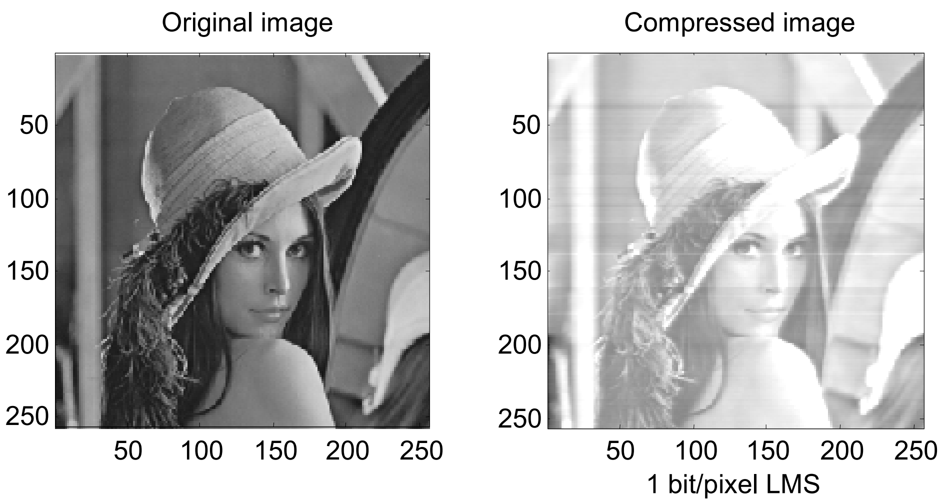


Figure 6.3.1: 1 bit/pixel using DPCM with LMS compressed Lena image

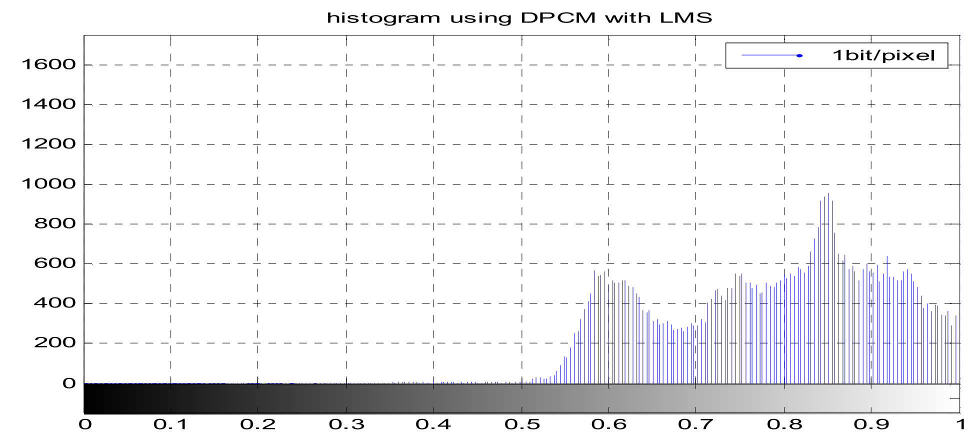


Figure 6.3.2: Histogram 1 bit/pixel using DPCM with LMS compressed or Bright Lena image

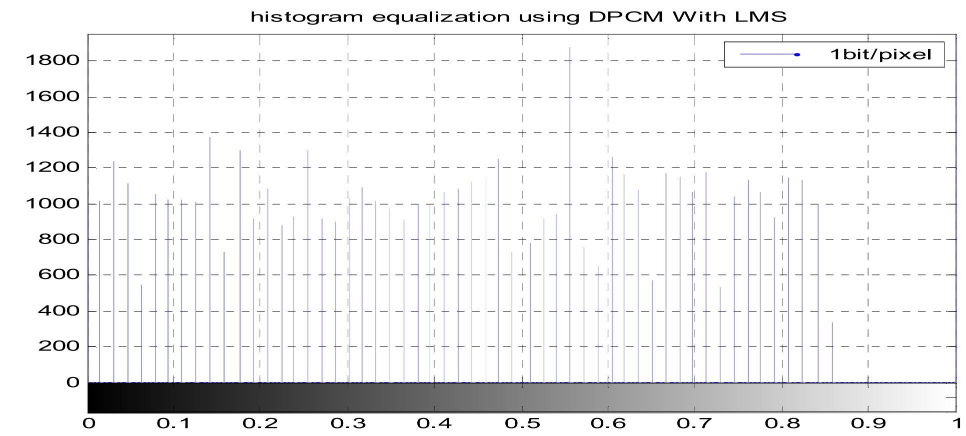


Figure 6.3.3: Histogram equalization1 bit/pixel using DPCM with LMS compressed Lena image

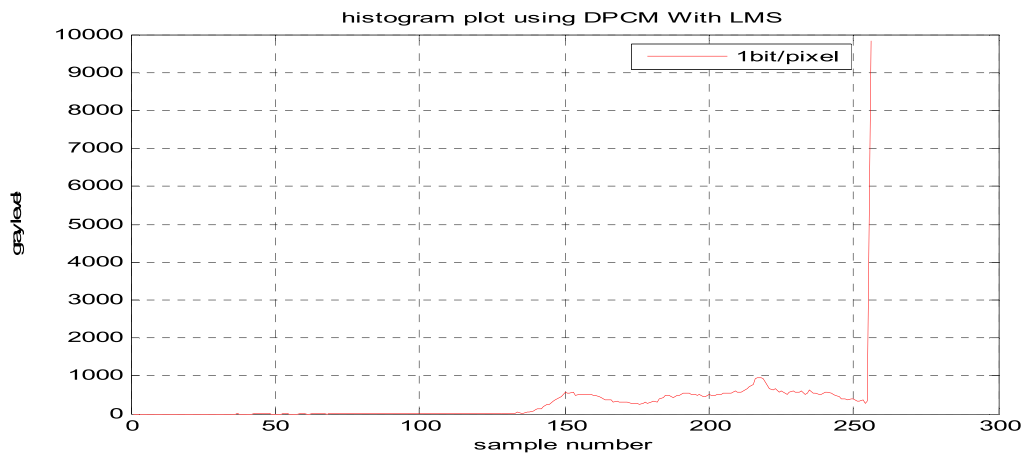


Figure 6.3.4: Histogram plot 1 bit/pixel using DPCM with LMS compressed Lena image

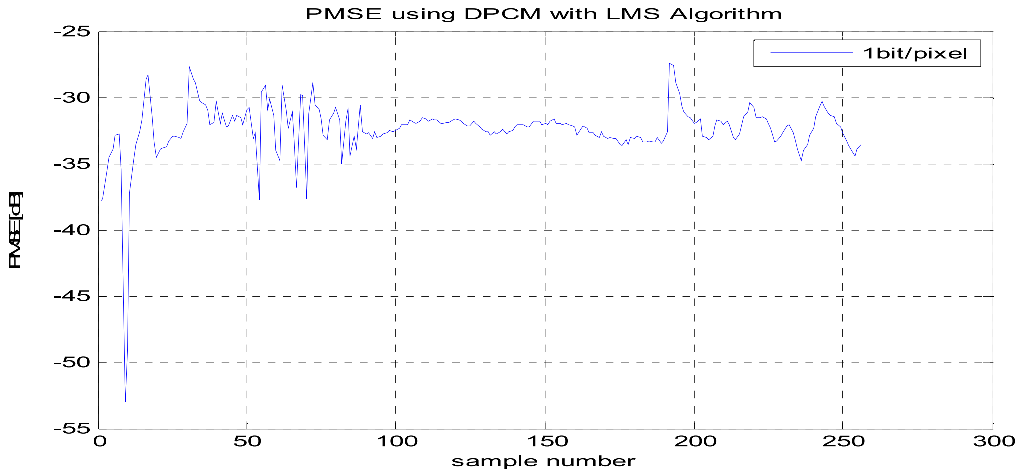


Figure 6.3.5: PMSE 1 bit/pixel using DPCM with LMS compressed Lena image

6.3.2 SIMULATION RESULTS OF IMAGE COMPRESON FOR 3 BIT LMS

The LMS algorithm was simulated using Matlab with respected to the application of 3bit/pixel image compression using DPCM with LMS algorithm depicted in figure 3.5 in chapter3. 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 .0005 and bit set to be 3bit/pixel for LMS. The full MATLAB source code of the LMS Algorithm is included in Appendix C.2.

The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image passed with 3bit/pixel using DPCM with LMS algorithm adaptive coefficient w .The simulation result shown in bellow figures.

Figure 6.3.6 shows the reconstructed 3bit/pixel image. This is a dark image. This image was approximately same the original image. It is reduce approximately 11.0kb (11,297bytes) for these parameters and the reconstructed image size 85.5kb (87,618bytes). Figure 6.3.7 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. And histogram equalization is shown in figure 6.3.8 of this compressed image after normalization range. Figure 6.3.9 shows the histogram plot 3bit/pixel using DPCM with LMS algorithm compressed Lena image this plot between gray level and samples points. In this image samples range are from 0 to 255 but gray level changes according to sample value. Figure 6.3.10 shows the PMSE 3bit/pixel using DPCM with LMS algorithm compressed Lena image. This shows the PMSE [dB] versus sample points from 0 to 255.

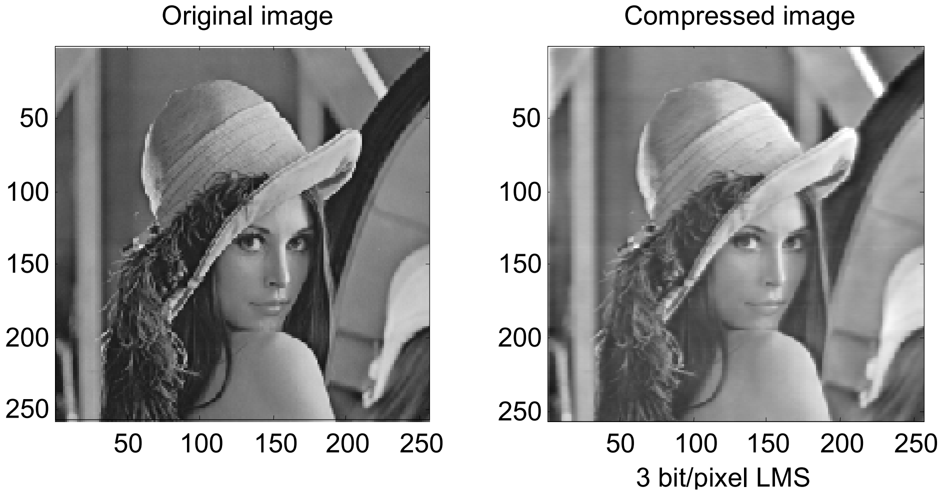


Figure 6.3.6: 3 bit/pixel using DPCM with LMS compressed Lena image

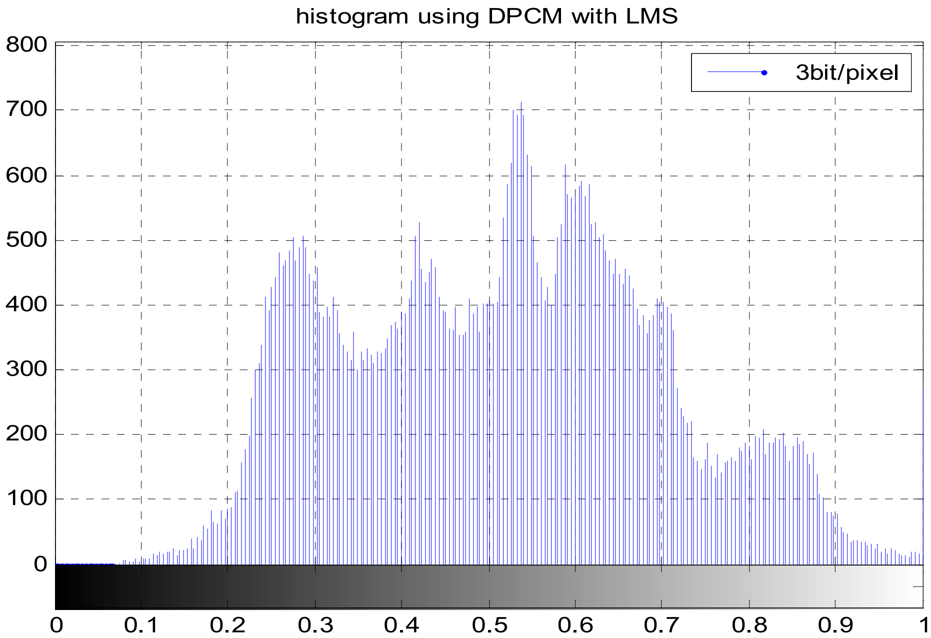


Figure 6.3.7: Histogram 3 bit/pixel using DPCM with LMS compressed or Dark Lena image

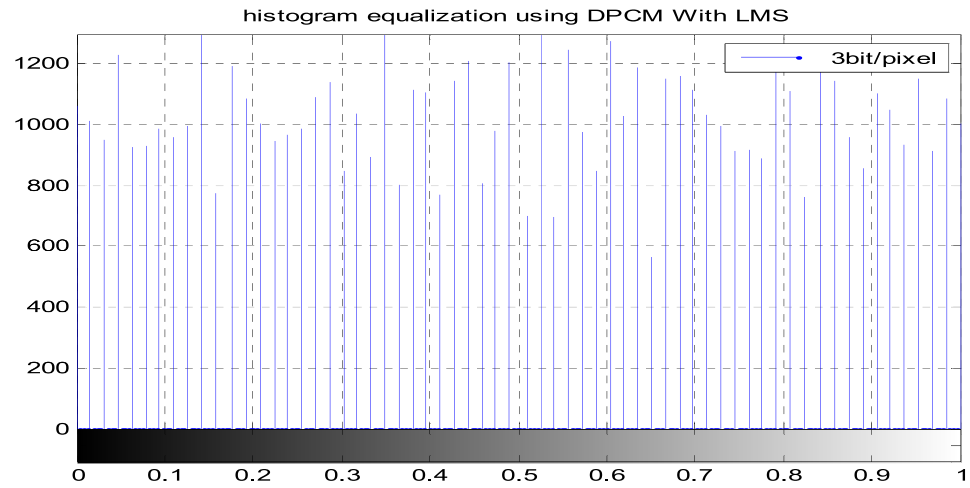


Figure 6.3.8: Histogram equalization 3 bit/pixel using DPCM with LMS compressed Lena image

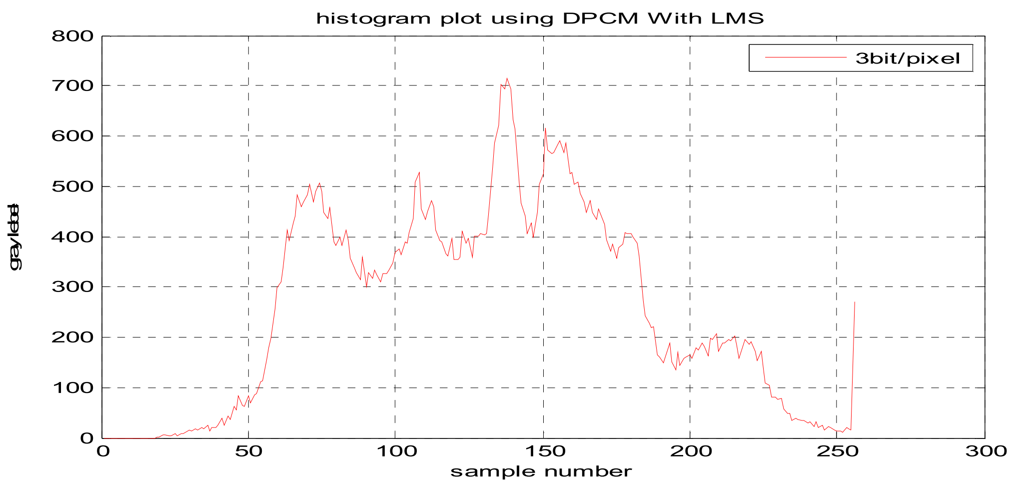


Figure 6.3.9: Histogram plot 3 bit/pixel using DPCM with LMS compressed Lena image

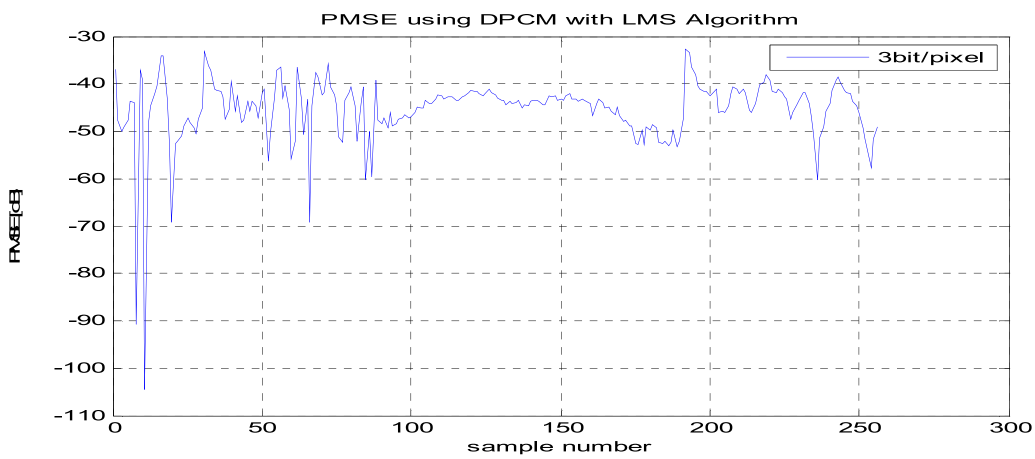


Figure 6.3.10: PMSE 3 bit/pixel using DPCM with LMS compressed Lena image

6.3.3 SIMULATION RESULTS OF IMAGE COMPRESSION FOR 1-3 BIT'S LMS COMPARISON

The LMS algorithm was simulated using Matlab 7.5 with respected to the application of 1 and 3bit/pixel image compression comparison using DPCM with LMS algorithm depicted in figure 3.5 in chapter3. 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 .001 and bit set to be 1 and 3bit/pixel for LMS. The full MATLAB 7.5 source code of the LMS Algorithm is included in Appendix C.3.

The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes). This original image passed with 1and 3bit/pixel using DPCM with LMS algorithm adaptive coefficient w .The simulation result shown in bellow figures. Figure 6.3.11(a) shows the reconstructed 1bit/pixel image. This is a bright image. It is reducing approximately 23.3kb (23,955bytes) respect to original image for these parameters and the reconstructed image size 73.2kb (74,960bytes). Figure 6.3.11(b) shows the reconstructed 3bit/pixel image. This is a dark image. This image was approximately same the original image. It is reduce approximately 11.0kb (11,297bytes) for these parameters and the reconstructed image size 85.5kb (87,618bytes). So we can say 1bit/pixel using DPCM with LMS algorithm image reduction was more compare to 3bit/pixel LMS. But the distortion of 1 bit/pixel LMS is more compare to 3bit/pixel DPCM. This has been showed in figure 6.3.12 and the figure 6.3.13 shows histogram plot comparison between gray label and sample number. The PMSE in 3bit/pixel LMS less 9-12 dB compare to 1bit/pixel LMS. This has been showed in figure 6.3.14. This figure shows the PMSE [dB] versus sample from 0 to 255.

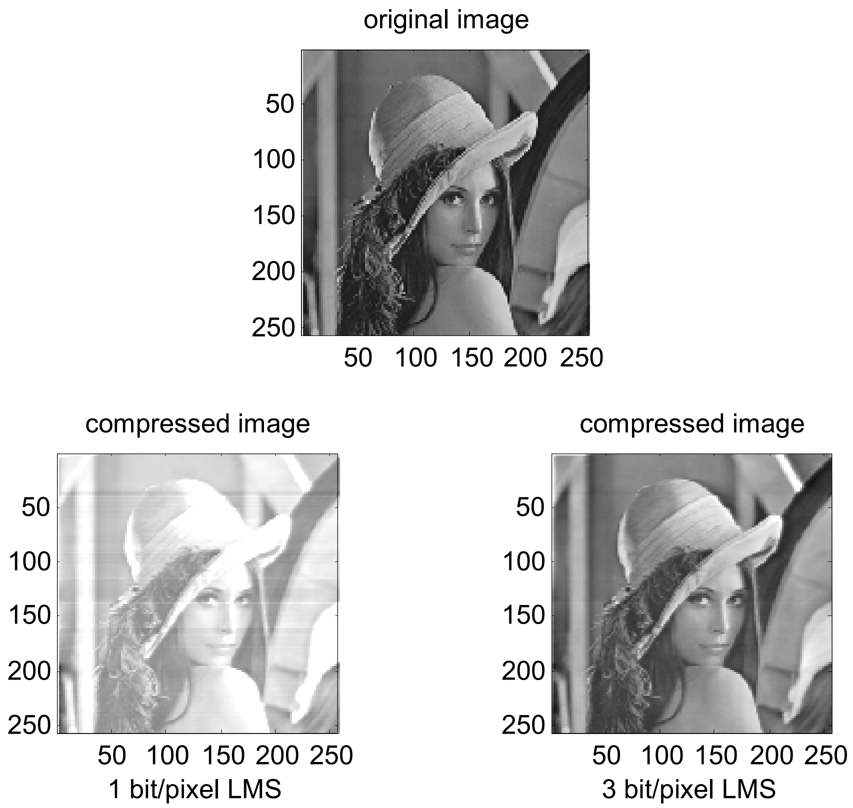


Figure 6.3.11: Visual results for processing Lena image using DPCM with LMS

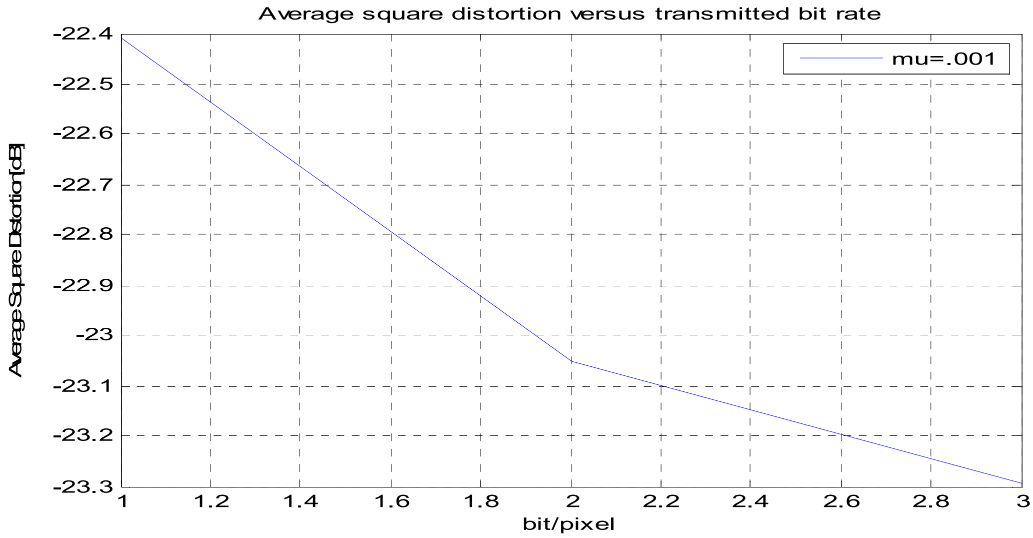


Figure 6.3.12: Average square distortion versus transmission bit rate using DPCM with LMS.

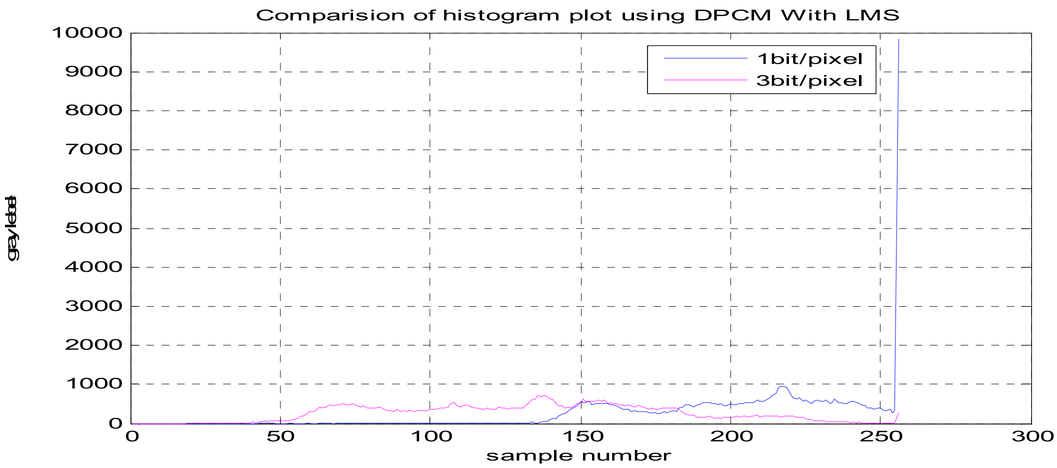


Figure 6.3.13: 1, 3 bit/pixel comparison of histogram using DPCM with LMS

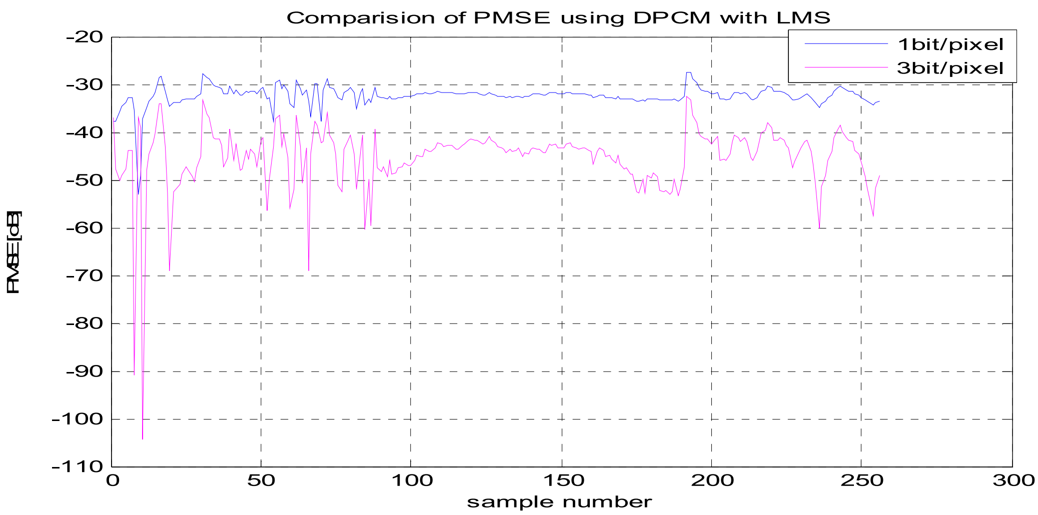


Figure 6.1.14: 1, 3 bit/pixel comparison of PMSE using DPCM with LMS.

6.4 SIMULATION RESULTS OF IMAGE COMPRESSION USING DPCM AND DPCM WITH LMS ADAPTIVE ALGORITHM FOR 1-3 BIT'S COMPARISON

The LMS algorithm was simulated using Matlab 7.5 with respected to the application of image compression comparison using DPCM with LMS algorithm depicted in figure 3.5 in chapter3. 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 .0005. The full MATLAB 7.5 source code of the LMS Algorithm is included in Appendix B3.

The 256×256 original image is shown in figure 6.1. This image size is 96.5 kB (98,915bytes. This original image passed with the residual quantizer (Q in figure 4.2.3 in chapter 4)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 [15]. The coefficients of the fixed DPCM predictor were chosen in accordance with the globally optimum model [14] and fixed coefficient value taken by w=[.495 .456]. The dynamic range of the data was eight bits from grey level 0 to 255. The simulation result shown in bellow figures.

Figure 6.4.1 plots the average square distortion versus transmitted bit rate for the Leena 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=.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 leena 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 figure 6.3.2, displaying the results for 3 bits/pixel and 1 bit/pixel transmission. At 3 bit/pixel, comparing the LMS predictor figure 6.4.2(a) and the DPCM prediction figure 6.4.2(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 6.4.2(c) displays the reconstruction using LMS, which provides a significantly sharper image than that shown in figure 6.4.2(d) which results of using 1 bit/pixel DPCM. Figure 6.4.3 and figure 6.4.4 shows the same bit level PMSE for LMS and DPCM. These figure shows less PMSE of LMS compare to DPCM.

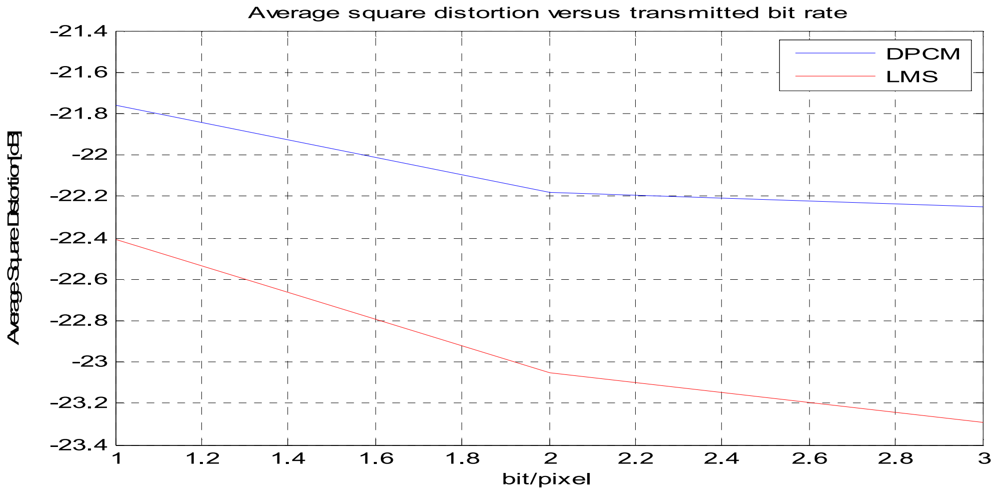
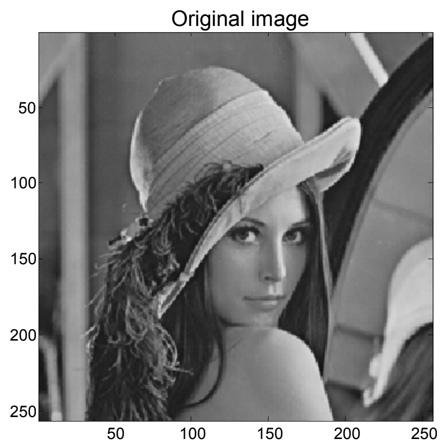
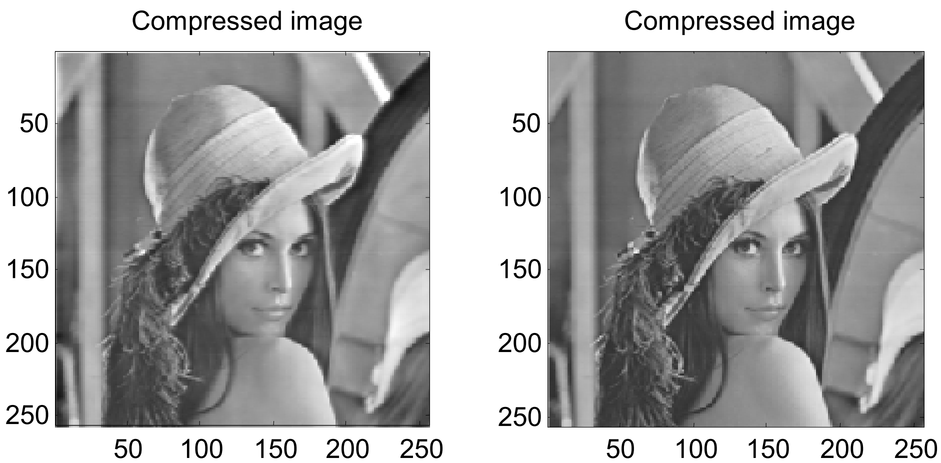
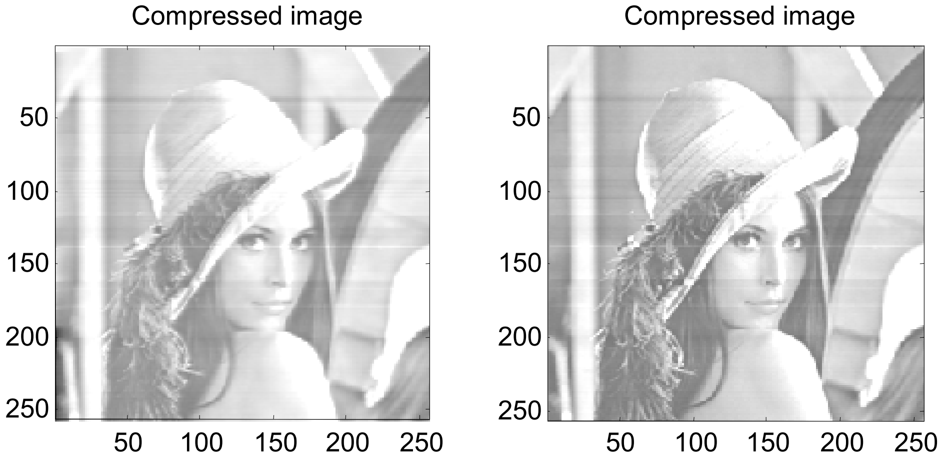


Figure 6.4.1: Average Square distortion versus transmission bit rate.





(a) 3 bit/pixel LMS (DL=-23.3) (b) 3 bit/pixel DPCM (DL=-22.25)



(c) 1 bit/pixel LMS (DL=-22.4) (d) 1 bit/pixel DPCM (DL=-21.75)

Figure 6.4.2: Visual results for processing Lena image with LMS and DPCM

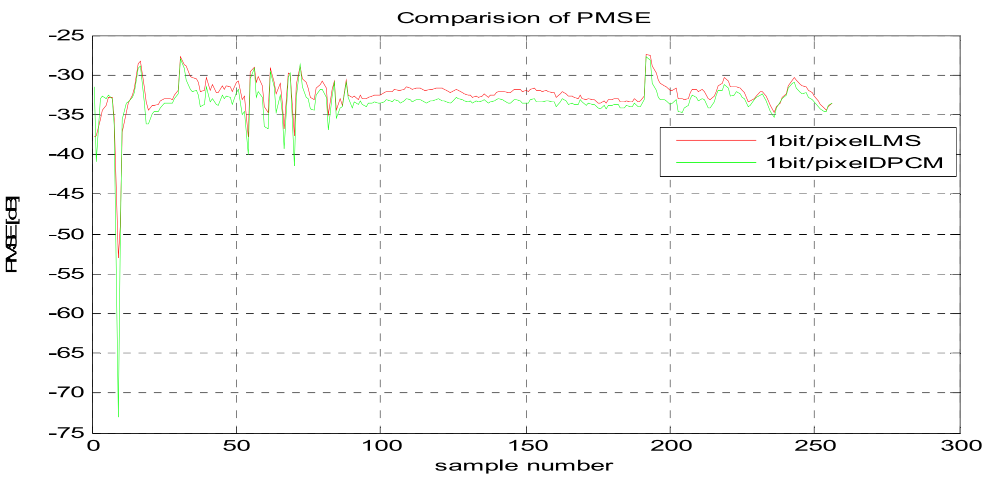


Figure 6.4.3: Comparison of PMSE using 1bit/pixel DPCM and LMS.

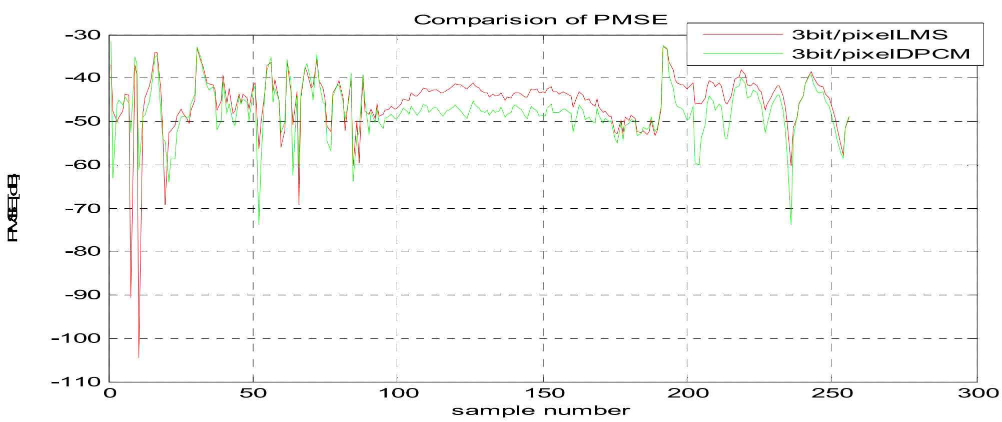


Figure 6.4.4: Comparison of PMSE using 3bit/pixel DPCM and LMS.

6.5 GUIDE 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.

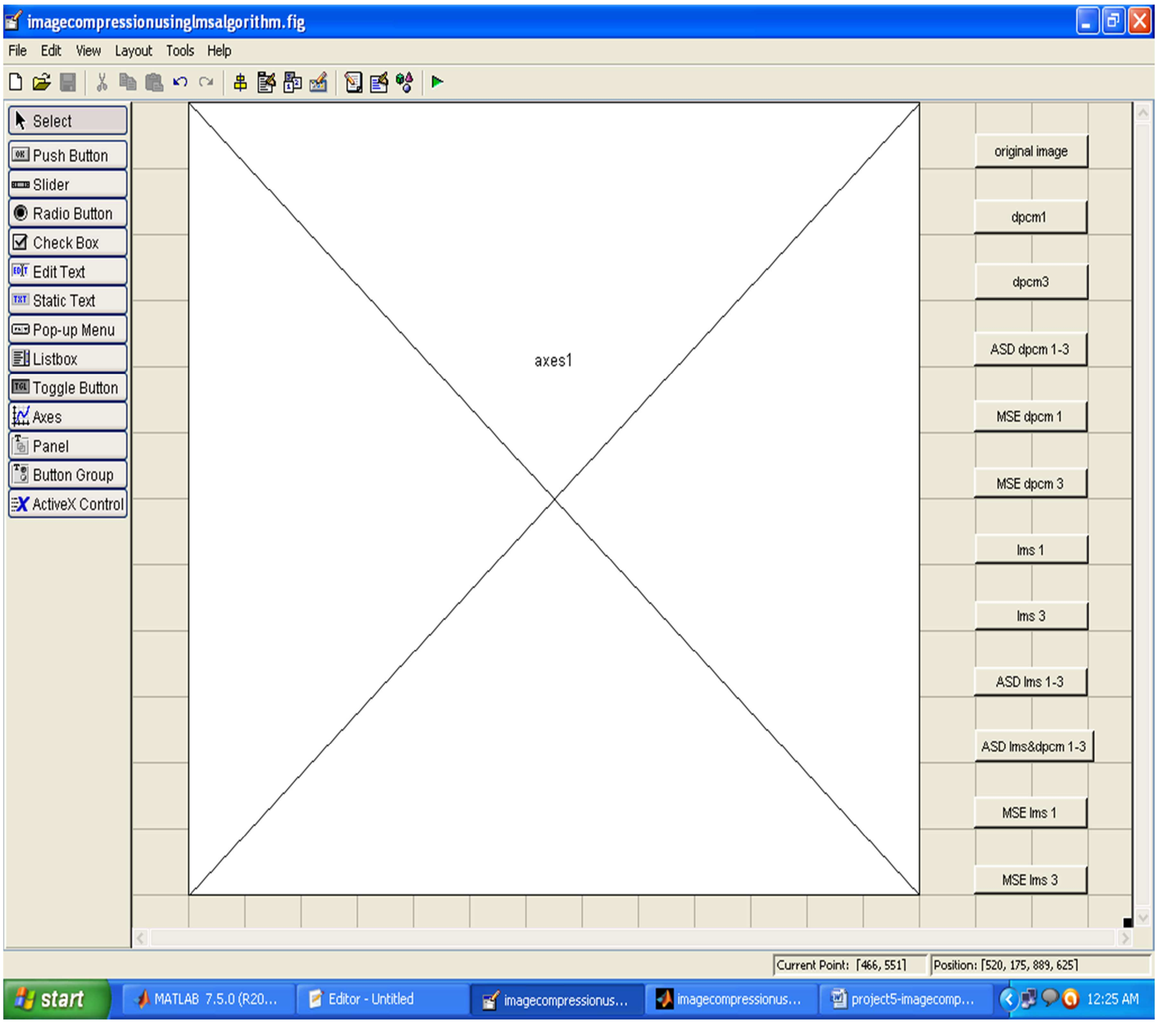


Figure 6.5.1 Guide basic block diagram

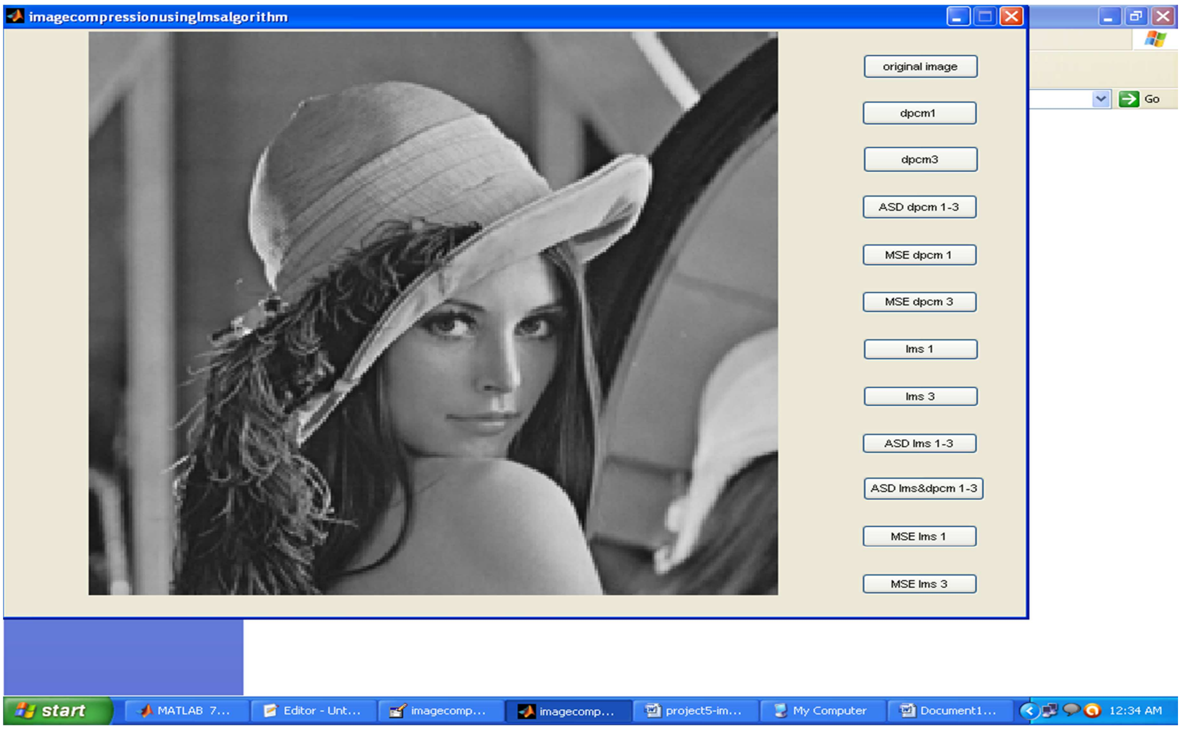


Figure 6.5.2 original image button block diagram

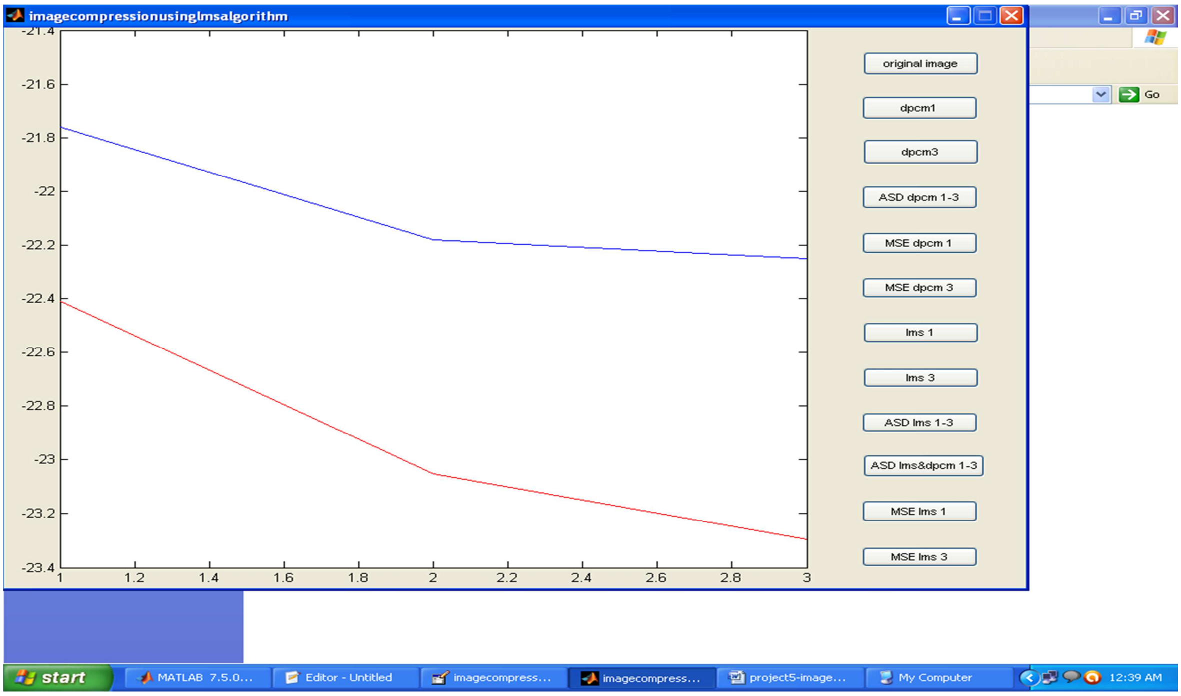


Figure 6.5.3 AVD LMS & DPCM 1-3 bit comparison block diagram

CHAPTER - 7

CONCLUSION OF THE STUDY & SCOPE FOR FUTHER STUDY

7.1 CONCLUSION OF THE STUDY

This thesis has presented the image compression using DPCM and using DPCM with LMS adaptive algorithm. The LMS is a simple and robust adaptive algorithm and DPCM use the LMS for prediction. The DPCM is use for A/D conversion in transmission side. This thesis has used fixed weight coefficient DPCM and the LMS uses the adaptive coefficient for image compression.

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 almost 2 bits/pixel 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.

The LMS can be used in fixed bit rate environments to decrease the reconstructed image distortion. In this thesis if the same distortion level for 3bit/pixel DPCM and 1bit/pixel LMS then LMS provide 14.9kB (16,283bytes) more reduction compare to DPCM.

7.2 POSSIBLE FUTURE WORK

The test of the algorithm was performed totally 'off-line'. The testing image was saved before as input to the algorithm and the output was looked over after simulation. Therefore, the real-time application for testing purpose could be the most interesting future work.

Besides that, the image compression can be done by help of other adapting filtering algorithm such as NLMS and RLS. This work carried out in future using these algorithms.

REFERENCES

[1]. S. ANNADURAI and R. SHANMUGALAKSHMI “Fundamentals of digital image processing” Published by Dorling Kindersley (India) 2007.
[2]. E. A. Riskin, "Optimum bit allocation via the generalized BFOS algorithm," IEEE Transactions on Information Theory, vol. 37, pp. 400-402, Mar. 1991
[3]. Digital Image Processing - A Remote Sensing Perspective, Jhon R. Jenson, 3rd Edition, Prentice – Hall, 2003.
[4]. 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.
[5]. S.Haykin and T.Kailath “Adaptive Filter Theory” Fourth Edition. Prentice Hall, Pearson Eduaction 2002.
[6]. 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.
[7]. KMM et al., Design and Fabrication of Color Scanner, Indian Journal of Technology, Vol 15, Apr 1997.
[8]. Fundamentals Of Digital Image Processing - Anil K. Jain, Prentice-Hall, 1989.
[9]. Digital Image Processing - R.C. Gonzalez Woods, Addison Wesley, 1992
[10]. B. P. Lathi and Zhi ding “Modern Digital and Analog Communication Systems” International Fourth Edition. New York Oxford University Press-2010, pp.292.
[11]. 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.
[12]. J. G. Prokis, Digital Communications. New York: McGraw-Hill, 1983.
[13]. J. E. Modestino, and D. G. Daut, “Source-channel coding of images,” IEEE Trans. Commun., vol. COM-27, pp. 1644-1659, Nov. 1979.
[14]. W. K. Pratt, Digital Image Processing. New York: Wiley, 1978.
[15]. M. D. Paez, and T. H. Glission, “Minimum mean square error quantization in speech PCM and DPCM system,” IEEE Trans. Commun. Vol. COM-20, pp. 225-230, Apr. 1972.
[16] www.wikipedia.com
[17]. Liang-jin Lin, Antonio Ortega, "Bit-Rate Control Using Piecewise Approximated Rate-Distortion Characteristics", IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, No. 4, pp. 446-459 , Aug. 1998
[18]. A. Ortega, "Optimal rate allocation under multiple rate constraints", Data Compression Conference'96, Snowbird, UT, Mar. 1996
[19]. Digital Image Processing - Kenneth R. Castleman, Prentice-Hall, 1996.
[20]. Digital Image Processing - Chellappa, 2nd Edition, IEEE Computer Society Press, 1992.

LIST OF RESEARCH PUBLICATIONS

PUBLICATIONS OF RESEARCH PAPERS IN INTERNATIONAL JOURNALS

1. **Ranbeer Tyagi**, Rohit Yadav, Shekhar Sharma, **"Image Compression using DPCM with LMS Algorithm",** "Journal of Communication and Computer, USA" Volume-8, Number-6, pp. 490-493, June-2011 (Serial Number -79).**ISSN 1548-7709(Print), ISSN 1930-1533 (Online)**.
2. Rohit Yadav, **Ranbeer Tyagi**, L.D. Malviya, **"Low Magnitude Edge Detection Algorithm"** , "International Journal of Computer Applications" (IJCA) (0975 – 8887) ,Vol.-23, No.-2,pp. 16-19, **June 2011. ISBN: 978-93-80752-76-0**
   Digital Library URI:http://www.ijcaonline.org/achieves/volume23/number2/2863-3691

3. **Ranbeer Tyagi,** D. K. Sharma, **"Digital Image Compression Comparisons using DPCM and DPCM with LMS Algorithm,"** International Journal of Computer Applications and Information Technology (IJCAIT), Vol.-I, Issue-II, pp.-65-71, **Sept.-2012** (ISSN: 2278-7720).

PUBLICATIONS OF RESEARCH PAPERS IN NATIONAL JOURNALS

1. **Ranbeer Tyagi**, Rohit Yadav, Akhilesh R. Upadhyay, **"Digital image compression comparison results for fixed weight coefficient using DPCM Quantization", "**People's Journal of Science & Technology (PJST)" Bhopal (M.P.), India. **ISSN: 2249-5487,** vol.-1, no.-2, pp. 29-31, Dec.-2011.

PUBLICATIONS OF RESEARCH PAPERS IN INTERNATIONAL CONFERENCE PROCEEDINGS AND CONFERENCE ATTENDED

1. Devendra Sharma, Lekhraj Sharma, **Ranbeer Tyagi**, **"Design and analysis of Auto-Tuned based PID controller",** International Conference on Advances in Communication, Embedded System and Computing, ICACEC, Bhopal, India, pp. 478-482, Vol.-1, Jan.- 2011.

PUBLICATIONS OF RESEARCH PAPERS IN NATIONAL CONFERENCE PROCEEDINGS AND CONFERENCE ATTENDED

1. **Ranbeer Tyagi**, Shekhar Sharma, **"Image compression results for 1, 2, 3 bits using LMS Algorithm",** A National Conference on "Advances in Electrical and Electronics Engineering (AEEE-2011)" at S.V.C.E. Indore, India, pp. 17, Paper ID-EC/S/025, Feb 2011.
2. **Ranbeer Tyagi,** Shekhar Sharma**, "Acoustic Echo Cancellation Using NLMS Adaptive Algorithm",** "III National Conference Emerging Trends in Engineering and Technology (ETET – 10)" at MIT, Ujjain, India, pp. 26, April 2010.

ABOUT THE AUTHOR

Mr. Ranbeer Tyagi obtained Master of Engineering degree from SGSITS Indore, Madhya Pradesh, India in year 2010, and B.E. in Electronics and Communication Engineering from RGPV University, Bhopal, and Madhya Pradesh, India in year 2008. He is currently working as an Assistant Professor in Electronics and Communication Department at People's College of Research and Technology (People's University) Bhopal, Madhya Pradesh, India. His areas of interest are Image Processing, Speech Processing, Communication, Acoustic Echo Cancellation, and Signal Processing. He has more than 2 years experience in teaching and research. He is Editor Member of International Journal of Advanced Technology & Engineering Research (IJATER).