3 SCALAR QUANTIZATION
AND ARITHMETIC CODING

3.1 Overview
Quantization is an important step in lossy compression technique. Compression ratio and quality of reconstructed image is controlled by this stage. This chapter gives comparison study of different types of scalar quantizers such as scalar uniform & nonuniform quantizers, which give optimal mean square error.

Next step after quantization is entropy coding; in this chapter Scalar quantizer performance is evaluated along with entropy coders. Zero order and higher order entropy coders are considered for comparison. For higher order entropy coding, context based coding techniques are studied. Adaptive context based techniques are studied and compared.

Based on this study new context classification based entropy encoding is proposed. Also simple modification in MED predictor is suggested which improves result.

3.2 Motivation
From Literature survey it is found that for wavelet based compression system single uniform quantizer for all bands or separate uniform quantizers for each band is used [16] [18][19]. The performance comparison for these two approaches is not reported in literature. This provides motivation to evaluate performance of these two approaches.

For efficient entropy coding, arithmetic coding is used by most of the authors. Context based arithmetic coding is used to increase compression further. The predictors used in context based coding are linear as well as nonlinear. [27][28] [29] [30] Due to small correlation between wavelet coefficients, nonlinear predictors are expected to perform better than linear. Comparison of linear and nonlinear predictors is required to be done to select type of the predictor. While comparison of predictors' compression ability, actual probability distribution of the source i.e. image should be considered. Because if probability model is not exactly representing the source probability compression ratio due to context based coding will be less than maximum possible compression.
There is also scope for finding simple context classification techniques, which can exploit wavelet properties.

3.3 Introduction
Steps in of the transform based lossy compression technique are

1. Finding Transform coefficients
2. Quantization
3. Entropy coding

These steps are shown in following block diagram (fig 2.1)

![Block diagram of compression system](image)

Fig 3.1 Block diagram of compression system

The first step performs an invertible transformation on the image converting it to a basis in which coefficients are less correlated than the pixels in the original image. This step removes much of the redundancy in the representation. This step is lossless except the numerical precision errors.

The transformed image is then quantized. The real coefficients are replaced with lower precision, which can be represented by scaled integers values. Information loss occurs in quantization process. Hence this stage also decides quality and compression ratio. Therefore this block is extremely important in compression system. More the loss takes place quality of reconstructed image reduces while compression ratio increases. Quantized coefficients are entropy encoded to remove coding redundancy and get further compression.

Aim of quantizer design is to optimize quantizer, for best quality at minimum rate (bits/pixel). Following sections present theory behind quantizer design.
3.4 Quantization

Quantization is the process of transforming a given signal \( x(n) \) at a time \( n \), into an amplitude \( y(n) \) taken from finite state of amplitudes. It is defined as mapping \( Q(R \rightarrow C) \) where \( R \) is the real line and

\[
C = \{ y_1, y_2, \ldots, y_n \} \subset R
\]

(3-1)

The signal amplitude \( x \) is specified by index \( k \), if it falls into the interval \( I_k : \{ x_k \leq x < x_{k+1} \} \) where \( k = 1, 2, \ldots L \)

(3-2)

The \( L \) array no. \( k \) is transmitted to the receiver typically in binary format. Let \( L = 2^R \) then bit rate of

\[
R = \log_2 L
\]

(3-3)

Is needed to inform about the index. [12]

**Quantizer performance.**

Quantizer performance can be measured in-terms of distortion. Statistical average of the distortion is used to indicate performance

\[
D = E_d(X, Q(X)) = \int d(X, Q(X)) f(x)dx
\]

(3-4)

Where \( f(x) \) is the pdf of \( X \).

Statistical average of the mean square error distortion

\[
D = E(x - Q(x))^2 = \sum_{i=1}^{N} \int (x - y_i)^2 f(x)dx
\]

(3-5)

Quantizers are of basically two types – uniform quantizer and non-uniform quantizer

**3.4.1 Uniform Quantizer**

Quantizer is said to be uniform if

\[
X_{k+1} - X_k = \Delta
\]

(3-6)

\( k = 1, 2, 3, \ldots \ldots, L-1 \)
\[ Y_{k+1} - Y_k = \Delta \] (3-7)

Properties of uniform quantizer

- For this quantizer decision level intervals i.e. \( X_{k+1} - X_k \) & and reconstruction level intervals \( Y_{k+1} - Y_k \) are of equal length.
- For such quantizer reconstruction level is midway between decision levels, i.e. \( Y_k = \frac{X_{k+1} - X_k}{2} \)
- Uniform Quantizer is said to be symmetric if \( Q(x) = -Q(-x) \)
- Midtread quantizer— for uniform quantizer, if output level changes in between cell limit at mid point it is midtread quantizer.

3.4.2 Nonuniform Quantizer

Uniform quantization is the simplest form of quantization, however it may not result into smallest quantization error variance. This can be obtained by choosing smaller decision intervals where probability of occurrence of random variable \( X \) is high, i.e. where the pdf \( P_x(x) \) is comparably high and larger decision intervals otherwise.

The 'v' power difference distortion \( D(v) \) due to scalar quantization can be expressed as the sum of the distortions for each of the decision region.

3.4.3 Lloyd Max Quantizer

\[
D(v) = \int_{-\infty}^{\infty} |Q(x) - x|^v f_x(x) dx
\]

\[
= \int_{-\infty}^{\infty} |Y(k) - x|^v f_x(x) dx
\] (3-8)
\[ p_k = \int_{-\infty}^{\infty} p_k(x) dx \]  

(3-9)

Probability of \( x \) in interval \( d_{k-1} < x < d_k \)

Entropy \( H \) of \( Q(x) \), which measures minimum number of bits that must be used as average to represent \( Q(X) \) is given by

\[ H = -\sum p_k \log_2 p_k \]  

(3-10)

Because set of quantization regions is discrete the entropy can remain bounded even when number of quantization regions \( k \) allowed growing arbitrarily large. To represent \( Q(x) \) using a number of bits that approaches entropy, variable rate coding or entropy coding must be employed. If fixed rate encoding is used then number of bits required to represent \( Q(x) \) depends only on quantization levels \( M \).

The problem of quantizer design is to select decision levels and reconstruction levels that minimize \( D(v) \) subject to a constraint on either the number of levels \( K \) or on entropy \( H \). A quantizer which has the least value of distortion subject to a constraint on no. of levels is called Lloyd max quantizer. A quantizer, which has the least value of distortion subject to a constraint on entropy, is said to be optimal entropy constraint quantizer.

Widely used distortion measure is MSE i.e. \( v = 2 \). Quantizer, which minimizes \( D_2 \) subject to the appropriate constraint, is termed as MSE optimal or minimum mean square error.

### 3.4.4 Lloyd Max Quantizer Algorithm

To find optimum reconstruction and decision levels for fixed bit rate i.e. number of quantization levels iterative procedure can be used. This is known as Lloyd max quantizer. This quantizer is optimum quantizer in the sense of mean square error (MMSE) for fixed number of levels \( M \).

\[ E_{Q2} = E[ (x - Q(x))^2 ] = \int_{-\infty}^{\infty} (x - Q(x))^2 f(x) dx \]  

(3-11)
Quantizer is designed to minimize $E_{Q2}$

$$X_{i, opt} = \frac{1}{2}(Y_{i, opt} + Y_{i, _{opt}}) \quad k = 1, 2, \ldots L$$

(3-12)

$$Y_{i, opt} = \frac{\int x(p_i(x))dx}{\int p_i(x)dx} \quad k = 1, 2, \ldots L$$

(3-13)

For uniform distribution

$$\int p_i(x) = \frac{1}{x_{k+1} - x_k} \quad for \ k = 1, 2, \ldots L$$

(3-14)

$$\int px(x) = 1/ Xk+1 -xk \ for \ xk < x < Xk+1$$

$$Y_{i, opt} = \frac{x_{k+1} - x_k}{2}$$

(3-15)

From properties, an optimal MSE quantizer

$$E_{Q2} = \sigma^2 - \sum_j p_j r_j^2$$

(3-16)

$r_j^2$ = distortion

$$H = -\sum_j p_j \log_2 p_j$$

(3-17)

$H$=entropy of the quantized signal

$$p_j = \int f(x)dx$$

(3-18)
Has the entropy of quantized signal, i.e. of \{Y_k\} is the entropy H and the quantization error giving the rate-distortion function of the quantizer: as M increases, D decreases but R increases.

![Figure 3.2 Entropy Constrained Quantizer](image)

Here we select rate or quantization levels such that for given entropy constraint min distortion is achieved. This can be obtained by using langrage cost function

\[ J = D + \lambda R \]  

(3-19)

Where D is distortion, \( \lambda \) is langrage multiplier and R is rate we have to find R such that cost function gets minimized. [19] Has given systematic procedure for this.

**Compending**

Quantizer can be designed for input statistics to get better SNR. Nonuniform quantization can be achieved by using uniform quantizer and companding. Dynamic range can be increased for given number of bits of resolution.

Compander modeling of quantizers

![Figure 3.3 – Nonuniform quantization using compander](image)
Bit Allocation

For set of k random variables \( x_1, x_2, ..., x_k \) each with zero mean value and variance

\[
E_{x_i}^2 = \sigma_i^2 \quad \text{for} \quad i = 1, 2, ..., k
\]  

(3-20)

If the pdf of each variable is known and particular distortion measure is selected optimal quantizer \( Q_i \) can be selected for \( X_i \).

In transform coding signal is coded by two-step process

1. Linear transformation
2. Quantization of transformed coefficients

But all transformed coefficients are not of equal importance hence equal no. of bits or quantization levels need not be allocated for the quantized coefficients. Bit allocation for the coefficients can be done based on significance

**Bit allocation problem**

For set of k random variables \( X_1, X_2, ..., X_k \) each with zero mean value and with variance

\[
E_{x_i}^2 = \sigma_i^2
\]

(3-21)

For quantizer \( N_i \) let \( b_i \) are the number of bits allocated which results into mean square distortion \( W_i(b_i) \)

Let overall distortion \( D \)

\[
D(b) = \sum_{i=1}^{k} W_i(b)
\]

(3-22)

The bit allocation algorithm decides optimal values of \( b_1, b_2, b_k \) subject to total bits for the for the identically distributed random variables the bit allocation is given by

\[
b_i = \bar{b} + \frac{1}{2} \log_2 \left( \frac{\sigma_i^2}{p^2} \right)
\]

(3-23)

Where \( \bar{b} = \) no. Of bits per parameter, \( \sigma_i = \) variance

\[
p^2 = \left( \sum_{i=1}^{k} \sigma_i^2 \right)^{\frac{1}{k}}
\]

(3-24)

\( p^2 = \) geometric mean of the variances of the random variables
For non-identically distributed random variables

\[ b_i = b + \frac{1}{2} \log_2 \sigma_i^2 + \frac{1}{2} \log_2 \frac{p_i}{p^2} \]  

(3-25)

Where \( p_i \) is pdf function and \( p^2 \) is geometric mean of the coefficient.

Bit allocation problem can be also solved using lagrange cost function.

3.5 Entropy Coding

Statistical coding techniques use estimates of the probabilities of the events to assign the codewords. Short codewords are assigned to more probable events and longer codewords to less probable events. Data can be compressed whenever some events are more likely than others.

Given a set of mutually distinct events \( e_1, e_2, e_3, \ldots, e_n \) and an accurate assessment of the probability distribution \( P \) of the events, Shannon [15] proved that the smallest possible expected number of bits needed to encode an event is the entropy of \( P \), denoted by

\[ H(p) = \sum_{k=1}^{n} -p(k) \log_2 p(k) \]  

(3-26)

\( p(k) \) is the probability that event \( e_k \) occurs. An optimal code outputs \( -\log_2 p \) bits to encode an event whose probability of occurrence is \( p \). Huffman coding and arithmetic coding are efficient and popular entropy coding techniques.

Pure arithmetic codes supplied with accurate probabilities provides optimal compression.

3.5.1 Basic Arithmetic Coding

Arithmetic coding generates nonblock codes. Source symbols and codewords do not exist. Instead entire sequence of source symbols is assigned a single arithmetic code word. The codeword itself defines an interval of real numbers between 0 and 1.

As the number of symbols in the message increases the interval used to represent it becomes smaller and the number of information bits required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. This technique does not require source symbol to translate into an integral number of code symbols. This is different than
Huffman coding, where each symbol is transmitted with integral number of code symbols.

Figure 2.4 illustrates basic arithmetic coding process

![Arithmetic Coding Diagram](image)

Let symbol sequence is to be coded is a1, a3, a2, from four symbol source is coded.

Probabilities of the symbols are tabulated

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Probability</th>
<th>Initial subinterval</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>0.2</td>
<td>[0.0, 0.2)</td>
</tr>
<tr>
<td>a2</td>
<td>0.3</td>
<td>[0.2, 0.5)</td>
</tr>
<tr>
<td>a3</td>
<td>0.4</td>
<td>[0.5, 0.9)</td>
</tr>
<tr>
<td>a4</td>
<td>0.1</td>
<td>[0.9, 1.0)</td>
</tr>
</tbody>
</table>

To start with current interval is entire half open interval [0.0 1.0)

When first symbol in message string ‘a1’ is coded current interval shrinks to [0.0, 0.2)

Now current interval becomes [0.0, 0.2), it is now divided as per probability division as shown in figure, when next symbol a3 is coded current interval get modified to [0.1, 0.18)

Finally when third and last symbol a2 is coded current interval becomes [0.116, 0.140)

Thus arithmetic code becomes any number between 0.116 to 0.14 e.g. 0.120

65
Arithmetic decoding- decoding process is exactly opposite to encoder. When decoder receives 0.120 as arithmetic code, it compares this with the interval of the symbols as per table 1 as number is within range of interval of a1 it understands that first symbol is a1. Then it divides the interval 0.0 to 0.2 as per probabilities of the symbol. Again it compares arithmetic code no with the new symbol interval and find where it fits in the procedure is repeated till all symbols are decoded.

The algorithm for encoding message stream using arithmetic coding works conceptually as follows:

1. We begin with a current interval \([L, H]\) initialized to \([0, 1)\).
2. For each event in the data stream, we perform two steps.
   (a) We subdivide the current interval into subintervals. The size of an event's subinterval is proportional to the estimated probability that the event will be the next event, according to the model of the input.
   (b) We select the subinterval corresponding to the event that actually occurs next, and make it the new current interval.
3. We output enough bits to distinguish the final current interval from all other possible final intervals. The length of the final subinterval is clearly equal to the product of the probabilities of the individual events. We need some mechanism to indicate the end of the message, either a special end-of-message event coded just once, or some external indication of the message length. Either method adds only a small amount to the code length.

**Difficulties while coding long message string**

As number of symbols being coded increase current interval shrinks to very small range. Difference between lower bound and higher bound becomes so small such that with the fixed precision arithmetic it becomes impossible to distinguish the difference as if they coincide. Similar problem also occurs when some of the symbols have very less probability. To avoid this interval scaling is to be used.

**Probability model**

To obtain maximum compression of a data, we need both a good probability model and an efficient way of representing (or learning) the probability model. To ensure decodability, the encoder is limited to the use of model information that is available to
the decoder. There are no other restrictions on the model; in particular, it can change as the data is being encoded. The models can be adaptive (dynamically estimating the probability of each event based on all events that precede it), semi-adaptive (using a preliminary pass of the input file to gather statistics), or non-adaptive (using fixed probabilities for all files).

Adaptive codes allow one-pass coding but require a more complicated data structure. Semi-adaptive codes require two passes and transmission of model data as side information; if the model data is transmitted efficiently they can provide slightly better compression than adaptive codes, but in general the cost of transmitting the model is about the same as the learning" cost in the adaptive case.

The basic implementation of arithmetic coding described above has two major difficulties: the shrinking current interval requires the use of high precision arithmetic, and no output is produced until the entire file has been read. The most straightforward solution to both of these problems is to output each leading bit as soon as it is known, and then to double the length of the current interval so that it reflects only the unknown part of the final interval. If interval lies in first half ‘0’ bit is sent while if it is in upper half ‘1’ bit will be sent [35].

3.5.2 Context Based Coding

Arithmetic coding efficiency depends upon how exactly we can get probability distribution or statistical data model. State of art coding techniques used in JPEG 2000 or H.263 works in two steps. First step is data modeling and second coding as per data model.

A model is representation of the source that generates data being compressed. Modeling is the process of constructing this representation. Coding maps the representation of the source into compressed representation. Arithmetic coder codes input symbols provided by the statistical modeler. It gives optimal compression with respect to the model used to generate the statistics. Thus assuming existence of optimal encoder, modeling becomes the key feature for effective data compression. The selection of a modeling paradigm and its implementation decides compression performance. Context modeling is a new approach to statistical modeling or Markov modeling.

Shannon [15] showed that the lossless compression scheme could encode a memoryless source with an average number of bits equal to the entropy of the source. Such
memoryless source coder is known as zero order entropy coder, which uses unconditional probabilities. When previous symbols are considered while coding current symbol i.e. source with memory, dependencies between the symbols will be exploited, by adapting the coder to the current “context” higher compression ratio will be achieved through the context-based entropy coder. In context based coding reference pixel is predicted based on context in which it exists and then prediction error is coded using arithmetic coding [18]. Compression efficiency depends upon how well context is able to predict current pixel. Thus conditional probability of the source is considered for such type of coding simplest context can be previous neighbor pixel. But if context template include more pixels prediction will be more accurate.

**Context-based Entropy Coding**
Given a finite source $X_1, X_2, X_3, \ldots, X_n$, compressing this sequence losslessly requires to process the symbols in some order and try to estimate the conditional probability distribution for the current symbol based on the previously processed symbols [41]. If we use conditional probabilities, we can do better than the zeroth-order entropy. Therefore, the optimal code length of the sequence in bits is $- \log \prod_{i=0}^{n-1} p(x_{i+1} | x_1, x_2, \ldots)$

To achieve maximum compression it is necessary to know probability distribution at decoder. To save the cost of this overhead we estimate model of distribution and use same while decoding. Let

$$P^*(x_n = a_i | x_{i-1}^{n-1})$$

As actual model

$$Q(a_i | x_{i-1}^{n-1})$$

And as estimated model

Average length $L_{\text{avg}}$ in case we use true model

$$L_{\text{av}}^* = - \sum_{a} P^*(x_n = a_i | x_{i-1}^{n-1}) \log(P^*(x_n = a_i | x_{i-1}^{n-1}))$$

Average length $L_{\text{avg}}$ in case estimated model
The challenge is to balance the benefit of using extra conditioning information against the performance cost of poor probability estimates. There are many ways to find good models of conditional distribution for a given data set. One of them is to impose a finite-state structure on the source by assuming the conditional distributions are dependent on the current state of the process. The current state in turn could be defined by a particular neighborhood of nearby pixel values, or in other words the context. Conditional probabilities can then be estimated by keeping track of symbol occurrences in each state. The methods of selecting the order of the model and the specific contexts on which the current symbol is to be conditioned result in two general kinds of context models: causal and non-causal. Figure 2.6 illustrates these two models in raster-scan images, scanning in row order, from top to bottom, and from left to right in each row.

In raster-scan images, because the non-causal models include not only causal neighboring pixels but also non-causal ones as its context, it provides more precise context information; the causal model avoids the use of header files. Causal context is a simple scheme in which all the context information relies upon the previous pixels. The less accurate causal model may contain less context information and lead to poor compression results. Ideally, if the context model reflects the conditional distributions of the source data exactly, the conditional entropy coder based on the probability distributions \( P(x_{i+1} | x_1, x_2, \ldots ) \) will obtain asymptotically good compression performance. The context size is also an issue for efficient compression. On one hand, the larger the context size the more context information it contains; on the other hand, when the context size is large while the image size is not accordingly large enough, there won’t be enough symbols in the image to form an accurate context model, leading to the “data dilution” problem.

Therefore, in practice the context size is a tradeoff between compression performance and the image size.

Context based coding method has four components
1. Context selection
2. Prediction
3. Context modeling of prediction error
4. Entropy coding of prediction errors

Context selection - For causal context model previous pixels are chosen as context. No. of pixels in context decides order. Simplest context is order 1.

Pixel in same row pixel in previous row Order 2 model
and previous column and same column

Figure 3.6 context models of order 1 and 2

Prediction – for $n^{th}$ order context model we predict current pixel i.e. from previous $n$ pixels as

$$\hat{Y} = \sum \alpha_i y_i$$

(3-31)

Value of $\hat{Y}$ becomes context for current pixel ‘x’. Different values of $\hat{Y}$ gives different contexts. Even if $y$, have finite values, every context will have different probability model. This may result in large number of contexts. With finite number of input samples, we will not be able to train the data or inaccurate data model. This problem is known as context data dilution. Context dilution results into inefficient compression. To overcome this problem number of contexts to be reduced or context quantization is required. Context quantization groups several $Y$ values into a single group.

Variety of prediction schemes are used for prediction. They can be classified as Fixed predictor and adaptive predictor.
3.5.3 Predictors

Fixed linear predictor
It predicts current pixel from previous pixel using fixed values of weights i.e. \( a_i \), which can be found using Yule-Walker equation as

\[
\begin{pmatrix}
R_{01} \\
R_{02} \\
R_{03}
\end{pmatrix} =
\begin{pmatrix}
R_{00} & R_{01} & R_{02} \\
R_{01} & R_{11} & R_{12} \\
R_{21} & R_{21} & R_{22}
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
a_3
\end{pmatrix}
\]

Where \( R_{xy} \) specifies autocorrelation function

GAP Predictor
Linear predictors have some disadvantages because their predictors are fixed for all symbols. The quality of the prediction can be improved by considering image gradients in the neighborhood of the pixel being predicted. The GAP technique proposed by Wu in CALIC [31], is a simple and efficient predictor which adapts prediction according to local gradients. It introduces two parameters \( h_d \) and \( v_d \) to represent horizontal and vertical gradients for each image sample during encoding.

The GAP is a simple, adaptive, nonlinear predictor that can adapt itself to the intensity gradients near the predicted pixel. For each pixel \( x_{[i,j]} \) Wu uses \( n, w, ne, nw, nn, ww, nne \) to denote north, west, northeast, northwest, north-north, west-west and north-northeast pixels respectively that form a context model as shown in Figure 2.7

---

**Figure 3.7 Context template for GAP predictor**
In the prediction algorithm, the predictor becomes either pixel \( n \) or \( w \) when there is a reasonable likelihood that a vertical or horizontal edge exists; otherwise, it is selected from several choices according to the local Smoothness, which needs to be estimated. When the samples of \( n, w, ne, nw, nn, ww \) and \( nne \) are known to both the encoder and decoder, the parameters \( d_h \) and \( d_v \) can be estimated by the following equations:

\[
d_h = |w - ww| + |n - nw| + |ne - n|
\]

\[
d_v = |w - mw| + |n - nn| + |ne - nne|
\]

In the algorithm, \( d_h \) and \( d_v \) are introduced to represent the horizontal and vertical gradients for each image sample during encoding. Detected by several thresholds for \( d_h \) and \( d_v \). Wu [30] uses GAP for his context modeling to predict the continuous-tone values for the pixels to be coded. An energy estimator of the predictor error is then calculated and quantized to classify the errors according to their variances. Finally, a conditioned adaptive arithmetic coder is run on the prediction errors. Thus entropy coding of errors using estimated conditional probability improves coding efficiency.

**MED Predictor**

The MED predictor is also known as MAP (Median Adaptive Predictor) Hewlet Packard's proposal; LOCO-I (low-complexity lossless coder) uses the MED predictor, which adapts in the presence of local edges.

Producing the predictor is described in Figure 2.8

![Figure 3.8 Context template for MED predictor](image)

The north pixel is used as the predictor in case of vertical edge.
Detection. The west pixel is applied in case of horizontal edge. Finally, if Neither a vertical edge nor a horizontal edge is detected, an interpolation

\[ x[I,j] = \begin{cases} \min(n, w) & \text{if } nw \geq \max(n, w) \\ \max(n, w) & \text{if } nw < \leq \min(n, w) \\ n+w-nw & \text{otherwise} \end{cases} \]

The context is obtained from uniformly quantizing the predictor. Similar to GAP, MED also detects horizontal or vertical edges by examining the values of north(n), west(w) and north-west(nw) neighbors of the current sample \( x[I,j] \).

3.5.4 Context Modeling And Context Quantization

Arithmetic coding or any entropy coding technique achieves compression by coding source symbols on their source statistics i.e. probability distribution. A mechanism for estimating this probability distribution is known as statistical model of the source.

For context modeling conditional probability distribution is required i.e. \( P(x_i | x_{i-1}, x_{i-2}, \ldots, x_1) \). The set of past observations on which the probability of the current symbol is conditioned is called the context modeling. Context modeling of gray scale images in transform domain leads to large number of possible model states or contexts if the number of model states is too large with respect to size of the image one may not have enough samples to reach good estimates of conditional probabilities within each model state leading to poor coding efficiency. This is known as sparse context or context dilution.

Efficient modeling should make minimum model cost. Model cost is the side information necessary to describe the source model to the decoder in the form of side information or loss in coding efficiency due to adaptive model. (Model on fly)

We can reduce number of contexts by context quantization

Consider a random variable \( C \). the minimum bit rate required to code \( C \) under the condition of a random variable \( E \) is the conditional entropy \( H(C/E) \).

If \( E \) is quantized to \( M = Q(E) \) then the conditional entropy becomes \( H(C/M) \)

\[ H(C/E) = H(C/E, M) \]
\[ H(C/M) = H(C/E, M) + I(C; E/M) \]
\[ = H(C/E) + I(C; E/M) \]
This means context quantization increases entropy. However, context quantization reduces the model cost at the same time. By applying a properly designed context quantizer, better compression performance can be achieved by an arithmetic coder with the quantized context. Thus, the objective of the optimal context quantization is to find a context quantizer, which can minimize the difference for a given number of quantization levels.

3.5.5 Context Classification

Another approach for context reduction uses context based classification. When used in wavelet transform domain every coefficient is classified based on context neighborhood. Number of classes and classification criterion is selected such that it results into minimum entropy.

Separate quantizer is used for every class. Quantized coefficients are modeled using parametric model. Parametric model assumes probability distribution according to some standard distribution such as Gaussian distribution. Parameters of this model are found from classified quantized coefficient using either minimum like hood function or it obtained on fly.

3.6 Experimentation

3.6.1 Experiment1-single quantizer for all bands

The purpose of the experiment is to evaluate performance of single uniform quantizer (used for all bands). For performance evaluation quality and bit rate are the criteria. Quality is measured in terms of PSNR (peak signal to noise ratio). Bit rate is equal to quantization levels if no entropy encoding is used. While if arithmetic coding is used as entropy encoder, bit rate approaches zero order entropy of the source.

Hence comparison is done on the basis of both. Performance evaluation is done on the test set used in chapter 1.

Step in experimentation

1. Image is decomposed upto three levels using bior6.8 wavelets basis
2. Uniform quantization of wavelet coefficients is done with number of levels 4, 16 & 64
3. Entropy of quantized coefficient is calculated
4. PSNR is calculated

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bit rate 2</td>
</tr>
<tr>
<td>Goldhill</td>
<td>Bior6.8</td>
<td>31.15 .23</td>
</tr>
<tr>
<td>Peppers</td>
<td>Bior 6.8</td>
<td>27.61 .24</td>
</tr>
<tr>
<td>Boat</td>
<td>Bior 6.8</td>
<td>26.55 .27</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior 6.8</td>
<td>25.08 .33</td>
</tr>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>25.43 .30</td>
</tr>
<tr>
<td>Mandrill</td>
<td>Bior6.8</td>
<td>18.49 .138</td>
</tr>
<tr>
<td>Test pat1</td>
<td>Bior6.8</td>
<td>16.09 .147</td>
</tr>
<tr>
<td>Testpat2</td>
<td>Bior6.8</td>
<td>10.55 .098</td>
</tr>
<tr>
<td>Parrot</td>
<td>Bior6.8</td>
<td>20.98 .097</td>
</tr>
<tr>
<td>Peppers</td>
<td>Bior6.8</td>
<td>19.74 .104</td>
</tr>
<tr>
<td>House</td>
<td>Bior6.8</td>
<td>19.65 .142</td>
</tr>
</tbody>
</table>

Table 3.1-Single quantizer for all bands

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bit rate 6</td>
</tr>
<tr>
<td>Goldhill</td>
<td>Bior6.8</td>
<td>32.15 .76</td>
</tr>
<tr>
<td>Boat</td>
<td>Bior6.8</td>
<td>33.55 .767</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior6.8</td>
<td>33.10 1</td>
</tr>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>34.31 .58</td>
</tr>
<tr>
<td>Mandrill</td>
<td>Bior6.8</td>
<td>30.05 1.6</td>
</tr>
<tr>
<td>Test pat1</td>
<td>Bior6.8</td>
<td>31.57 1</td>
</tr>
<tr>
<td>Testpat2</td>
<td>Bior6.8</td>
<td>26.59 1.46</td>
</tr>
<tr>
<td>Parrot</td>
<td>Bior6.8</td>
<td>34.60 1.8</td>
</tr>
<tr>
<td>Peppers</td>
<td>Bior6.8</td>
<td>32.21 1.71</td>
</tr>
<tr>
<td>House</td>
<td>Bior6.8</td>
<td>32.62 2</td>
</tr>
</tbody>
</table>

Table 3.2- Single quantizer for all bands

3.6.2 Experiment 2- separate quantizer for each band
Uniform quantization using bit allocation scheme.
Wavelet transform coefficients in different band and different orientation carry unequal amount of energy. Standard deviation of these bands show wide range of values. Bit allocation scheme allocates more bits to the band having larger standard deviation as explained in section [14]. Performance of this scheme is evaluated in this experiment for Performance evaluation PSNR and bit rate are considered. Same test set is used as experiment1.

1. Input image is decomposed using different bior6.8 wavelet functions up to level three
2. Bit allocation procedure is used for wavelet decomposed coefficients of test images.
3. Bit allocation is done based on variance of the subbands. Using equation 2.23
4. More number of bits are allocated for bands of higher standard deviation
5. ‘Zero bin’ is kept equal to step size of quantization. Which is more for higher standard deviation band.
6. Entropy and PSNR is calculated

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>Bit rate 2 entropy</th>
<th>Bit rate 4 entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>26.38 .23</td>
<td>34.42 .67</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior6.8</td>
<td>25.55 .256</td>
<td>34.31 1.25</td>
</tr>
<tr>
<td>Boat</td>
<td>Bior6.8</td>
<td>25.55 .26</td>
<td>33.51 .82</td>
</tr>
<tr>
<td>Mandrill</td>
<td>Bior6.8</td>
<td>21.74 .38</td>
<td>29.74 1.61</td>
</tr>
<tr>
<td>Goldhill</td>
<td>Bior6.8</td>
<td>25.90 .28</td>
<td>33.70 1.08</td>
</tr>
<tr>
<td>Teatpat1</td>
<td>Bior6.8</td>
<td>21.86 .35</td>
<td>30.15 .99</td>
</tr>
<tr>
<td>Teatpat2</td>
<td>Bior6.8</td>
<td>14.06 .37</td>
<td>24.27 1.25</td>
</tr>
<tr>
<td>Hurricane</td>
<td>Bior6.8</td>
<td>37.97 .35</td>
<td>35.41 .88</td>
</tr>
</tbody>
</table>

Table 3.3- Separate quantizer for each band

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>Bit rate 6 PSNR</th>
<th>Bit rate 8 PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>42.85 2.12</td>
<td>54.55 4.02</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior6.8</td>
<td>43.92 2.85</td>
<td>55.58 4.79</td>
</tr>
<tr>
<td>Boat</td>
<td>Bior6.8</td>
<td>42.64 2.18</td>
<td>54.03 4.09</td>
</tr>
</tbody>
</table>
Table 3.4 - Separate quantizer for each band

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>Bit rate 2</th>
<th>Bit rate 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSNR</td>
<td>entropy</td>
</tr>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>17.29</td>
<td>.70</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior6.8</td>
<td>14.55</td>
<td>1.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Filter</th>
<th>Bit rate 6</th>
<th>Bit rate 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Bior6.8</td>
<td>17.63</td>
<td>3.87</td>
</tr>
<tr>
<td>Barb</td>
<td>Bior6.8</td>
<td>18.43</td>
<td>3.90</td>
</tr>
</tbody>
</table>

Table 3.5 - Lloyds non uniform codebook
Figure 3.9 – Comparison between uniform quantizer and non uniform quantizer (varying quantization levels)
3.6.4 Discussions and Conclusions (From Experiment 3.10.1, 3.10.2 and 3.10.3)

1. To achieve compression, quantization of wavelet coefficients is done. Number of quantization levels are dependant on bit rate or compression ratio. Two quantization schemes are implemented.
   a. Uniform single quantizer for all wavelet bands
   b. Uniform separate quantizers for every band of wavelet decomposition.

   Here number of quantization levels for each band are decided by bit allocation scheme.

2. For the given number of quantization levels bit allocation scheme works better on the basis of MSE distortion. For all images separate quantization scheme gives higher PSNR by 5 to 6 dbs compared to single quantizer, for equal
quantization levels. This is as per expectation because bit allocation scheme reduces quantization error.

3. When quantization index is encoded using arithmetic coding, it is observed that entropy with single uniform quantization is less, i.e. we get more compression. Wavelet transform coefficients have large values for coarser resolution and smaller values for finer resolution. When we use single quantizer, most of the finer coefficients get quantized to zero bin. Size of finer band is higher than coarser band. This makes probability of zero high and reduces average entropy. Finer coefficients are approximated to zero but they carry less signal energy. Therefore when image is reconstructed it does not cause much error. Thus single uniform quantizer gives better performance i.e. more compression for same PSNR when entropy coder is used, compared to separate quantizer for each band.

4. This simple experimentation shows that uniform quantization using single quantizer is better choice than using separate quantizers. However literature shows that separate quantization scheme is used for most of the cases.

5. Nonuniform quantizer using Lloyds algorithm is implemented. Quantization thresholds are decided from training set of images. This is iterative method. Quantization decision threshold are changed to minimize quantization error. This quantizer is expected to give optimum (minimum) MSE. However result found to be inferior. We get less PSNR for same number of quantization levels that of uniform quantizer. This may be because though we have performed optimization for entire training set, it may not result into optimum decision threshold for specific input image.

3.6.5 Experiment 4- arithmetic coding using actual probability distribution
Arithmetic coding of quantized image using actual probability distribution

1. Using bit allocation equation image is quantized to get average quantization levels equal to given number of levels.

2. Binary arithmetic coding is performed assuming that actual probability distribution is available to the encoder and decoder
### Table 3.6- Binary arithmetic coding using actual distribution

<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
<th>Average b/p including model cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>4.0288</td>
<td>4.0316</td>
</tr>
<tr>
<td>Barb</td>
<td>4.7942</td>
<td>4.7970</td>
</tr>
<tr>
<td>Boat</td>
<td>4.0924</td>
<td>4.0952</td>
</tr>
<tr>
<td>Mandrill</td>
<td>5.4657</td>
<td>5.4655</td>
</tr>
<tr>
<td>Goldhill</td>
<td>4.8053</td>
<td>4.8081</td>
</tr>
<tr>
<td>Testpat1</td>
<td>4.2534</td>
<td>4.31</td>
</tr>
<tr>
<td>Testpat2</td>
<td>3.8307</td>
<td>3.8335</td>
</tr>
<tr>
<td>Hurricane</td>
<td>4.5736</td>
<td>4.61</td>
</tr>
<tr>
<td>Parrots</td>
<td>9.1448</td>
<td>9.1548</td>
</tr>
<tr>
<td>Peppers</td>
<td>11.8469</td>
<td>11.9372</td>
</tr>
<tr>
<td>House</td>
<td>12.3524</td>
<td>12.4624</td>
</tr>
</tbody>
</table>

#### 3.6.6 Experiment 5 - Arithmetic coding assuming Gaussian probability distribution

1. Using bit allocation equation image is quantized to get average quantization levels equal to given number of levels.
2. Binary arithmetic coding is performed assuming Gaussian probability distribution model.
3. Model parameters are estimated using MATLAB function

<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
<th>Rate in b/pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>4.0288</td>
<td>5.87</td>
</tr>
<tr>
<td>Barb</td>
<td>4.7942</td>
<td>5.78</td>
</tr>
<tr>
<td>Boat</td>
<td>4.0924</td>
<td>5.65</td>
</tr>
<tr>
<td>Mandrill</td>
<td>5.4657</td>
<td>6.07</td>
</tr>
<tr>
<td>Goldhill</td>
<td>4.8053</td>
<td>6.12</td>
</tr>
</tbody>
</table>
Table 3.7- Binary arithmetic coding using Gaussian distribution

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testpat1</td>
<td>4.2534</td>
<td>5.71</td>
</tr>
<tr>
<td>Testpat2</td>
<td>3.8307</td>
<td>6.98</td>
</tr>
</tbody>
</table>

3.6.7 Discussions and Conclusions

1. Arithmetic coding give coding rate very close to zero order entropy of the source. For arithmetic coding encoder and decoder is expected to know probability distribution of the source. To get best results exact distribution should be obtained and should be also available to the encoder as well as decoder. Number of bits to send the probability table to the decoder is model cost. It is also overhead of the coding.

2. Model cost i.e. overhead to send actual probability distribution is smaller for big images. For Lena or barb (512 x 512) overhead cost is less than 256 x 256 testpat1 and testpat2 image
3. To eliminate or reduce model cost we can estimate probability distribution of the source. In that case we have to send only model parameters. However mismatch in actual probability distribution and modeled distribution reduces efficiency of coding, i.e. it reduces compression ratio.

4. Assuming simple model as Gaussian model, we find coding bit rate. Which is quiet higher than entropy. This emphasize on the need of accurate model estimation.

3.6.8 Experiment 6 - Context based arithmetic coding using linear and non-linear predictors

In context based arithmetic coding two parameters are important, i.e. context selection and predictor.

1. Using bit allocation equation image was quantized to get average quantization levels equal to given number of levels.

2. Two context template are used
   - Context of three samples such as a(i-1,j).a(i,j-1) and a(i-1,j-1)

   ![Ref pixel](Ref pixel)

   a1,a2,a3 pixel form context template

   - Context of 6 samples

   ![Context of 6 samples](Context of 6 samples)

   a1,a2,a3,a4,a5,a6 pixel form context template
3. Context based arithmetic coding is performed using linear predictor mean
predictors.

4. Fixed Linear predictor is implemented for context1. From the magnitudes of
correlation tabulated in chapter two, it is observed that there exists very less
correlation between reference pixel and ‘a1’ pixel of context1. Therefore fixed
weights are taken as (0.1, 0.45 , 0.45) for a1, a2,a3 respectively. We can also
use Yule-Walker equation to calculate weights. But we need to calculate weights
for every band and of every image.

5. Non-linear predictors such as min, max and median, GAP and MED are used for
comparison. For gap prediction contexts are used without classification to keep
comparison platform similar.

6. Arithmetic encoder and decoder used actual distribution (non parametric).
Model cost is included for bit/pixel calculation

<table>
<thead>
<tr>
<th>Context1 results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
</tr>
<tr>
<td>Lena</td>
</tr>
<tr>
<td>Barb</td>
</tr>
<tr>
<td>Boat</td>
</tr>
<tr>
<td>Goldhill</td>
</tr>
<tr>
<td>Mandrill</td>
</tr>
<tr>
<td>Testpat1</td>
</tr>
<tr>
<td>Testpat2</td>
</tr>
<tr>
<td>Hurrican1</td>
</tr>
</tbody>
</table>

Table 3.8- arithmetic coding using context1
Context2 results

<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.670</td>
<td>.577</td>
<td>.579</td>
<td>.649</td>
<td>.643</td>
</tr>
<tr>
<td>Barb</td>
<td>1.254</td>
<td>.993</td>
<td>.997</td>
<td>1.172</td>
<td>1.022</td>
</tr>
<tr>
<td>Boat</td>
<td>.829</td>
<td>.568</td>
<td>.711</td>
<td>.799</td>
<td>.798</td>
</tr>
<tr>
<td>Goldhill</td>
<td>1.085</td>
<td>.982</td>
<td>.978</td>
<td>1.046</td>
<td>1.047</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1.619</td>
<td>1.492</td>
<td>1.490</td>
<td>1.567</td>
<td>1.862</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.990</td>
<td>.777</td>
<td>.766</td>
<td>.830</td>
<td>.881</td>
</tr>
<tr>
<td>Testpat2</td>
<td>1.259</td>
<td>1.169</td>
<td>1.010</td>
<td>1.225</td>
<td>1.192</td>
</tr>
<tr>
<td>Hurrican1</td>
<td>.885</td>
<td>.714</td>
<td>.709</td>
<td>.856</td>
<td>.855</td>
</tr>
</tbody>
</table>

Table 3.9- arithmetic coding using context2

3.6.9 Experiment 7 -GAP and MED predictor

1 Non-linear predictors MED and gap predictors are implemented.

2 For GAP prediction contexts are used without classification to keep comparison platform similar context based arithmetic coding is performed using GAP and MED predictor.

3 As minimum predictor performs best in above set so MED predictor is implemented using modified predictor.

Conventional predictor is

\[ x[I,j] = \min(n, w) \text{ if } nw \geq \max(n, w) \]
\[ \max(n, w) \text{ if } nw \leq \min(n, w) \]
\[ n + w - nw \text{ otherwise} \]

We modify it to

\[ x[I,j] = \min(n, w) \text{ if } nw \geq \max(n, w) \]
\[ \max(n, w) \text{ if } nw \leq \min(n, w) \]
\[ \min(n + w - nw) \text{ otherwise} \]
<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
<th>Gap</th>
<th>Med</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.670</td>
<td>.656</td>
<td>.563</td>
</tr>
<tr>
<td>Barb</td>
<td>1.254</td>
<td>1.142</td>
<td>1.00</td>
</tr>
<tr>
<td>Boat</td>
<td>.829</td>
<td>.799</td>
<td>.699</td>
</tr>
<tr>
<td>Goldhill</td>
<td>1.085</td>
<td>1.042</td>
<td>.964</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1.619</td>
<td>1.577</td>
<td>1.510</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.990</td>
<td>.896</td>
<td>.781</td>
</tr>
<tr>
<td>Testpat2</td>
<td>1.259</td>
<td>1.231</td>
<td>.8958</td>
</tr>
<tr>
<td>Hurricane</td>
<td>.885</td>
<td>.8625</td>
<td>.8055</td>
</tr>
</tbody>
</table>

Table 3.10- GAP and MED Predictor

<table>
<thead>
<tr>
<th>Image</th>
<th>MED</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.563</td>
<td>.552</td>
</tr>
<tr>
<td>Barb</td>
<td>1.00</td>
<td>.975</td>
</tr>
<tr>
<td>Boat</td>
<td>.699</td>
<td>.687</td>
</tr>
<tr>
<td>Mandrill</td>
<td>.964</td>
<td>1.498</td>
</tr>
<tr>
<td>Goldhill</td>
<td>1.510</td>
<td>.954</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.781</td>
<td>.764</td>
</tr>
<tr>
<td>Testpat2</td>
<td>.8958</td>
<td>.882</td>
</tr>
<tr>
<td>Hurricane</td>
<td>.8055</td>
<td>.799</td>
</tr>
</tbody>
</table>

Table 3.11- Modified MED Predictor
3.6.10 Conclusions and discussion

1. Seven predictors are implemented. Out of which two are linear, three are nonlinear predictors (min, max, median) and two are adaptive predictors (Gap, MED).

2. Using these predictors we can achieve better compression ratio, i.e. average bit/pixel ratio than simple arithmetic coding.

3. Performance of the mean predictor is most inferior. The image in wavelet domain has two types of regions, smooth uniform region and activity region of edges. Mean predictor finds mean of the context pixels. This operation is equivalent to low pass filter. Therefore any variation in the context gets smoothened. This may give higher prediction errors in edge region and therefore less compression. Results are further degraded for bigger contexts because more smoothening occurs.
4. Median filter is also type of low pass filter and therefore does not give good compression. Min and max predictors perform better than mean and median. Average bits/pixel using min and max predictor is less than mean or median. In case of min or max operator, when it is used for edge region, it tries to catch discontinuity or variation in coefficient values. This makes prediction error small and therefore entropy reduces.

5. Images with more activity such as Mandrill, Testpat1 gives good results with max predictor than min predictor. For such images combinations of min & max predictor also improves results, i.e. if for level 3 min predictor and level 2 and 1 max predictor is used results get improved.

6. Bigger context gives more compression in case of min and max predictors but for mean and median performance reduces (more number of pixels). Bigger template also increases computation complexity.

7. Performance of GAP predictor is not so good. It is more complex than all other predictors discussed. Performance is also dependent on gradient threshold used in algorithm, which needs to be adjusted for optimum results for every image.

8. MED predictor uses smaller template and gives best results

9. From above observations modification is suggested in MED predictor which improves results further.

3.6.11 Experiment 8- Context quantization

Arithmetic coding using Context quantization. Predictor discussed in above experimentation give rise to larger number of context values. Therefore to transmit large number of these contexts to the decoder side, big overhead will be created. To reduce number of contexts context quantization is used.
In this experiment we use two methods of context quantization
1. Uniform quantization
2. Codebook based quantization.
For uniform quantization total number of contexts are divided into 8 quantization bins. Results are taken only with min predictor as it has found best predictor. Context2 i.e.
template of six pixels are used. Decoder and encoder are using actual probability distribution. Model cost is included while calculating bits/pixel.

<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
<th>Context quantization (8 level quantization)</th>
<th>Context quantization (6 level quantization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.670</td>
<td>.488</td>
<td>.533</td>
</tr>
<tr>
<td>Barb</td>
<td>1.254</td>
<td>1.138</td>
<td>1.143</td>
</tr>
<tr>
<td>Boat</td>
<td>.829</td>
<td>.72</td>
<td>.742</td>
</tr>
<tr>
<td>Goldhill</td>
<td>1.085</td>
<td>.90</td>
<td>.98</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1.619</td>
<td>1.43</td>
<td>1.461</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.990</td>
<td>.814</td>
<td>.928</td>
</tr>
<tr>
<td>Testpat2</td>
<td>1.259</td>
<td>1.18</td>
<td>1.198</td>
</tr>
<tr>
<td>Hurricane</td>
<td>.885</td>
<td>.716</td>
<td>.725</td>
</tr>
</tbody>
</table>

Table 3.12- Uniform context quantization

**Codebook based quantization**

Purpose of the quantization is to reduce contexts at the same time to get min bit/pixel ratio for given number of contexts. In uniform quantization we group contexts based on their magnitude, e.g. if context values vary from 0 to 100 and we want to reduce number of contexts to only 25. We can group contexts 0-3 in context no.1, 4 to 7 values in context no.2 and so on. In other words we are using equally apart thresholds for quantization. This scheme may not result into optimum partition. Lloyd’s algorithm is used to partition context values in required number of bins.

1. Centroids of the quantization bins in above experiments are used to get starting codebook
2. For every context value in training image (which is outside this set) distance between context value and bin centroid is found. Context is assigned bin for which distance is minimum. Centroid of all those context values, which are
assigned same bin, are grouped together. Centroid of these grouped values become new codebook entries. Procedure is repeated no substantial change found.

<table>
<thead>
<tr>
<th>Image</th>
<th>Context quantization (8 level quantization)</th>
<th>6 level</th>
<th>4 level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.584</td>
<td>.574</td>
<td>.60</td>
</tr>
<tr>
<td>Barb</td>
<td>1.048</td>
<td>1.01</td>
<td>1.21</td>
</tr>
<tr>
<td>Boat</td>
<td>.71</td>
<td>.725</td>
<td>.76</td>
</tr>
<tr>
<td>Goldhill</td>
<td>.97</td>
<td>.97</td>
<td>1.05</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1.49</td>
<td>1.47</td>
<td>1.53</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.765</td>
<td>.845</td>
<td>.908</td>
</tr>
<tr>
<td>Testpat2</td>
<td>1.060</td>
<td>1.069</td>
<td>1.09</td>
</tr>
<tr>
<td>Hurrican1</td>
<td>.746</td>
<td>.751</td>
<td>.783</td>
</tr>
</tbody>
</table>

Table 3.13- Context quantization using Lloyds codebook

3.6.12 Conclusions

1. Context quantization is used to reduce number of contexts

2. This reduces model cost i.e. number of bits to send probability table to the decoder. However at the same time this reduces exactness in prediction and therefore compression ratio decreases.

3. It is found that with eight levels of context quantization we got less compression compared to MED predictor. MED predictor is found to be most effective than mean max. or min. When quantization levels are decreased to six, model cost reduces. However prediction error also increases, therefore six level quantizer is found to give less compression compared to eight levels. Hence quantization levels should not be very less.

4. Effectiveness of context quantization also depends on the method of quantization. Lloyds algorithm gives quantization decision levels as per distribution of contexts. Resultant quantization is non-uniform. This is expected
to improve compression results. Nonuniform context quantization using Lloyds algorithm is implemented for 8, 6 and 4 quantization levels. When quantization levels are changed from 8 to 6, compression ratio is increased. But when quantization levels are decreased to 4 i.e. only four contexts, compression ratio decreases. When we go from 8 levels to 6 levels model cost decreases, which may have given more compression. For 4 levels of quantization model cost further reduces, however exactness in prediction decreases or prediction error increases. Therefore compression ratio decreases.

3.6.13 Experiment 10- New Context classification based arithmetic coding.
To reduce number of contexts, context classification can be also used. Most of the authors have used classification criterion, which depends on magnitude of coefficients in context template. For classification thresholds of context magnitude are defined. For optimum compression these thresholds need to be properly chosen. Therefore thresholds may vary from image to image.
Hence classification criterion defined for new context classification techniques is not dependent on magnitude. Context classification is done in following manner.

Context template is small of only three pixels as shown in figure. All three coefficients are scanned in the order 1, 2, 3 and converted into binary value. If transform coefficients value is greater than '0' it is assumed as '1' otherwise '0'. Resultant binary number with combination of 1,2,3 coefficients is the class of reference pixel. e.g. if all coefficients are positive reference context value is binary 111 i.e. '7' hence reference pixel is classified in the class 7. We get such 8 total classes.

![Context template for context classification](image)

Classification distribution for Lena is shown in the figure 2.13, while fig 2.14 shows distribution of class ‘0’ and class ‘4’.
Figure 3.14 - Distribution of the classification

Figure 3.15 - Distribution of the class '0' and '4'

Class '0' is most frequent or highly probable class. Variance of this class is very small compared to class 4, which is less probable class. Performance is tested using nonparametric probability distribution.
<table>
<thead>
<tr>
<th>Image</th>
<th>Min</th>
<th>MED</th>
<th>Context classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.58</td>
<td>.563</td>
<td>.498</td>
</tr>
<tr>
<td>Barb</td>
<td>1.448</td>
<td>1.00</td>
<td>.91</td>
</tr>
<tr>
<td>Boat</td>
<td>.72</td>
<td>.699</td>
<td>.625</td>
</tr>
<tr>
<td>Goldhilll</td>
<td>.98</td>
<td>.964</td>
<td>.894</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1.51</td>
<td>1.510</td>
<td>1.41</td>
</tr>
<tr>
<td>Testpat1</td>
<td>.794</td>
<td>.781</td>
<td>.691</td>
</tr>
<tr>
<td>Testpat2</td>
<td>1.108</td>
<td>.8958</td>
<td>.974</td>
</tr>
<tr>
<td>Hurrican1</td>
<td>.816</td>
<td>.8055</td>
<td>.75</td>
</tr>
</tbody>
</table>

**Table 3.14- Context classification**

**Figure 3.16- Comparison of Context Quantization and Classification**
3.6.14 Discussions and Conclusions

1. New context classification techniques gives entropy less than min., max, mean GAP and even MED predictor. Thus it gives best compression results compared to min., max, GAP and MED predictors implemented. The given classification technique is not dependent on context magnitude and therefore not image specific.

2. It is observed while studying different classification techniques that, classification technique should distribute the coefficients (after classification) unevenly so as to get more compression. Also variance of the more probable class should be less to get more compression. Because less variance means smaller dynamic range of the class or less symbols in the class. This increase probability of every symbol and reduces entropy. From the graph (fig 3.14) it is found class ‘0’ is more probable than remaining classes, while its distribution is very narrow. This makes given classification technique more effective i.e. gives best compression.

3.7 Summary

In this chapter compression performance using scalar quantization and arithmetic coding is evaluated. Two types of scalar quantizers are implemented. They are Uniform and nonuniform scalar quantizers. Quantization levels are changed and PSNR is calculated. Performance of uniform quantization i.e. compression ratio is more than non-uniform quantizers. In case of uniform quantizer two schemes are implemented ‘Single quantizer for all bands’ and ‘Separate quantizers for each band’. Single quantizer scheme gives more compression than separate quantizer scheme when used along with arithmetic coding.

Quantized coefficients are encoded using context based arithmetic coding. Seven predictors which include linear, nonlinear and adaptive type and used in context based coding are evaluated. Maximum compression is achieved using MED predictor. Modification in conventional MED is done. This improves results further.
Context quantization schemes and context classification are used to reduced number of contexts. Uniform context quantization and nonuniform context quantization is implemented.

New context classification Scheme is implemented. Classification criterion is order of coefficients. Based on the criterion coefficients are classified in seven classes. This method gives best compression compared to the encoding using above seven predictors.