# 9

# Non-Predictive Multistage Lattice Vector Quantization Video Coding

M. F. M. Salleh[1] and J. Soraghan[2]
[1]*Universiti Sains Malaysia*
[2]*University of Strathclyde*
[1]*Malaysia*
[2]*United Kingdom*

## 1. Introduction

In recent years, the demand for mobile multimedia applications has increased tremendously. Since the volume of the application data such as video is high and the bandwidth of mobile channels is limited, efficient compression techniques are very much required (Ghanbari, 2003) (Sikora, 2005). This research area has attracted many researchers since the last 40 years, and many related works have been done as reviewed in (Sikora, 2005).

Generally, video compression technique aims to reduce both the spatial and temporal redundancy of a video sequence. The motion estimation and compensation is a very efficient technique to exploit the temporal redundancy of the video sequence (Sikora, 2005). Thus, it has been used in video coding standards for application in mobile communications such as in H.263 (H.263, 2000) and H.264 (Ostermann et al., 2004). Although this process offers significant gain in coding efficiency, the encoded bitstream suffers from channel errors during transmission in mobile channels which reduces the reconstructed frame quality at the receiver.

Motion JPEG2000 (ISO/IEC, 2002), uses Intra-frame video coding only which eliminates the prediction step uses in motion estimation process in the temporal domain. It offers less design complexity, reduces computational load and increases robustness in wireless environments (Dufaux & Ebrahimi, 2004). In another work done in (Akbari & Soraghan, 2003), a video coding scheme has been developed to omit the prediction step in temporal domain for robust video transmission in noisy mobile environment. In that work, the similar high frequency subbands from each frame within a Group of Frame (GOP) are joined to produce a number of group data. Each of the group data is processed using an Adaptive Joint Subband Vector Quantization (AJSVQ). The Adaptive Vector Quantization (AVQ) technique has been developed based on the work presented in (Voukelatos & Soraghan, 97).

In the past years, there have been considerable research efforts in Lattice Vector Quantization (LVQ) for image and video compression schemes (Conway & Sloane, 1988) (Barlaud et al., 94) (Kossentini & Smith, 99) (Sampson et al., 95) (Weiping et al., 97) (Kuo et Al., 2002) (Man et. al., 2002) (Feideropoulou et. al., 2007). The choice for LVQ has been for its property to reduce complexity of a vector quantizer. In video coding, the works have been inclined towards using LVQ with motion estimation and compensation process as explained in (Sampson et al., 95) (Weiping, et. al., 97) (Kuo et al., 2002) (Feideropoulou et al., 2007). However, only the work in (Man et. al., 2002) has been introduced to omit the motion estimation and compensation techniques and yet incorporates LVQ in the encoding process. The prediction step is omitted by the wavelet transform on the temporal domain, thus reducing the computational load in the video coding scheme. The LVQ is applied on the coefficients of the transformed data. The work is reported to achieve a good balance between coding efficiency and error resilience.

In our related work (Salleh & Soraghan, 2005) (Salleh & Soraghan, 2006) (Salleh & Soraghan, 2007), multistage lattice vector quantization (MLVQ) has been introduced. This technique has the capability to capture the quantization errors. For every pass of quantization process the errors are magnified by multiplication with the current scaling factor. The advantage of this process is that, it offers reduction in quantization errors and hence enhances reconstruction of frame quality as well it offers robustness for video transmission over mobile channels.

This chapter presents a video coding scheme that utilizes MLVQ algorithm to exploit the spatial-temporal video redundancy. Since LVQ reduces computational load of the codebook generation, this paves the way for the video coding scheme to have multistage processes (multistage lattice VQ). The Unequal Error Protection (UEP) and Equal Error Protection (EEP) schemes are also developed for robust video transmission. Results of the video coding scheme in erroneous Hilly Terrain (HT) and Bad Urban (BU) mobile environments are significantly better than H.263 codec using the TETRA channel simulator (ETSI, 1995). The performance under the same settings is comparable to H.264 codec for some test video sequences.

## 2. Background

The following subsections discuss briefly the basic concept of lattice vector quantization as well as a brief discussion about quad-tree coding. The use of vector quantization for lossy compression has been very common since the last decade. Lattice vector quantization technique offers great advantage in term of coding simplicity due to its regular structure (Conway & Sloane, 1988).

### 2.1 Lattice vector quantization

In lattice vector quantization (LVQ), the input data are mapped to the lattice points of a certain chosen lattice type. The lattice points or codeword may be selected from the coset points or the truncated lattice points (Gersho & Gray, 1992). The coset of a lattice is the set of points obtained after a specific vector is added to each lattice point. The input vectors surrounding these lattice points are group together as if there are in the same *voronoi* region. Some of the background of lattices, the quantizing algorithms for the chosen lattice type and the design of the lattice quantizer's codebook are now presented.

### 2.1.1 Lattice type

A lattice is a regular arrangement of points in k-space that includes the origin or the zero-vector. A lattice is defined as a set of linearly independent vectors [Conway and Sloane, 1988];

$$\Lambda = \left\{ X : X = a_1 u_1 + a_2 u_2 + ....... + a_n u_n \right\} \tag{1}$$

where $\Lambda \in \Re^k$, $n \le k$, $a_i$ and $u_i$ are integers for $\mathrm{i}=1, 2, \ldots, \mathrm{n}$. The vector set $\{u_i\}$ is called the basis vectors of lattice $\Lambda$ and it is convenient to express them as a generating matrix $U = [u_1, u_2, \ldots\ldots, u_n]$.
As an example consider a two-dimensional lattice $\Lambda_2$ with basis vectors as:

$$\Lambda_2 = \{X : X = a_{11}u_1 + a_{12}u_1 ;\, a_{21}u_2 + a_{22}u_2\}$$

Also let us assume that

$$a_{11} = 0, a_{12} = \sqrt{3}, a_{21} = 2, a_{22} = 1$$

The generating matrix is given by:

$$U = [u_1, u_2] = \begin{bmatrix} 0 & \sqrt{3} \\ 2 & 1 \end{bmatrix}$$

The vector $X$ represents a two dimensional coordinate system where each point can be represented $X = (x_1, x_2)$. Thus, it can be written that

$$x_1 = \sqrt{3}m_2$$

$$x_2 = 2m_1 + m_2$$

where $m_1$ and $m_2$ are any integer value. Figure 1 shows the lattice structure defined by this example.

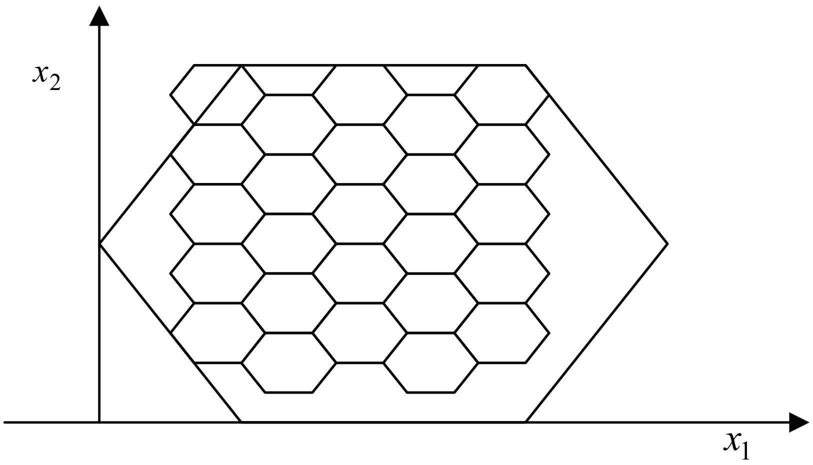


Fig. 1. Two-dimensional hexagonal lattice

The reciprocal or dual lattice $\Lambda^*$ consists of all points $Y$ in the subspace of $\Re^k$ spanned by $u_1, u_2, \ldots\ldots, u_n$ such that the inner product $X.Y = x_1y_1 + \ldots.. + x_ky_k$ is an integer for all $x \in \Lambda$ (Conway & Sloane, 1988). When the lattices are contained in their duals, there exist the cosets representatives $r_0, \ldots\ldots, r_{d-1}$ such that

$$\Lambda^* = \bigcup_{i=0}^{d-1} (r_i + \Lambda) \qquad (2)$$

where $d$ is the determinant of $\Lambda$. In a different approach, the dual lattice is obtained by taking the transpose of the inverse generating matrix given by $\left(U^{-1}\right)^t$ once the generating matrix is known (Gibson & Sayood, 1988).

The $Z^n$ or cubic lattice is the simplest form of a lattice structure. It consists of all the points in the coordinate system with a certain lattice dimension. Other lattices such as $D_n(n\geq 2)$, $A_n(n\geq 1)$, $E_n[n=6,7,8]$ and their dual are the densest known sphere packing and covering in dimension $n\leq 8$ (Conway & Sloane, 1988). Thus, they can be used for an efficient lattice vector quantizer. The $D_n$ lattice is defined by the following (Conway & Sloane, 1988):

$$D_n=(x_1,x_2,......,x_n)\in Z^n \tag{3}$$

$where \sum_{i=1}^{n} x_i = even$

Its dual i.e. $D_n^*$ is the union of four cosets of $D_n$:

$$D_n^*=\bigcup_{i=0}^{3}(r_i+D_n) \tag{4}$$

$where\ \ r_0=\left(0^n\right),\ r_1=\left(\frac{1}{2}^n\right),\ r_2=\left(0^{n-1},1\right),\ r_3=\left(\frac{1}{2}^{n-1},-\frac{1}{2}\right)$

The $A_n$ lattice for $n\geq 1$ consists the points of $(x_0,x_1,......,x_n)$ with the integer coordinates sum to zero (Conway & Sloane, 1982). The lattice quantization for $A_n$ is done in *n+1* dimensions and the final result is obtained after reverting the dimension back to *n* (Gibson & Sayood, 1988), (Conway & Sloane, 1982). Its lattice $A_n^*$ consists of the union of $n+1$ cosets of $A_n$ (Conway & Sloane, 1982):

$$A_n^*=\bigcup_{i=0}^{n}(r_i+A_n) \tag{5}$$

The expression for $E_n$ lattice with $n=6,7,8$ is explained in as the following:

$$E_8=\left(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)+D_8 \tag{6}$$

The dual lattice is given by the same definition i.e. $E_8^*=E_8$.

The lattice $E_7$ is defined as the following:

$$E_7=A_7\bigcup\left(\left(-\frac{1}{2}^4,\frac{1}{2}^4\right)+A_7\right) \tag{7}$$

The dual lattice is given by the following:

$$E_7^*=\bigcup\left(\left(-\frac{3}{4}^2,\frac{1}{4}^6\right)+E_7\right)=\bigcup_{i=0}^{3}(s_i+A_7) \tag{8}$$

$$where\ s_i = \left( \left(\frac{-j}{4}\right)^{2i}, \left(\frac{i}{4}\right)^{2j} \right),\ i + j = 4$$

Besides, other important lattices have also been considered for many applications such as the Coxeter-Todd $(K_{12})$ lattice, Barnes-Wall lattice $(\Lambda_{16})$ and Leech lattice $(\Lambda_{24})$. These lattices are the densest known sphere packing and coverings in their respective dimension (Conway & Sloane, 1988), (Gibson & Sayood, 1988).

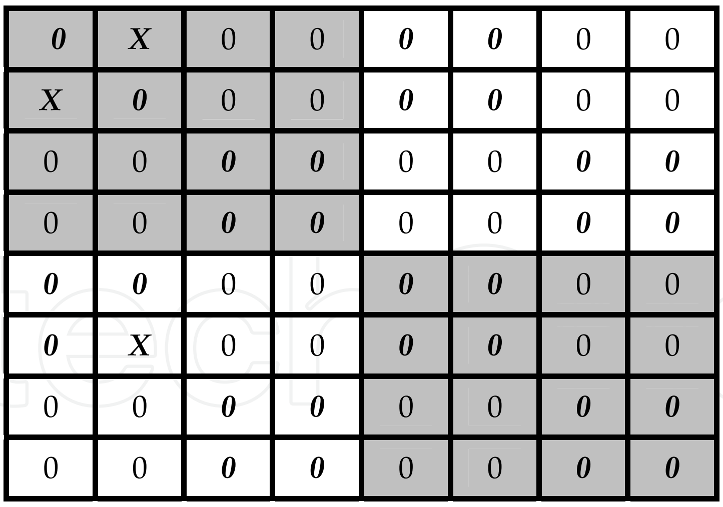
### 2.3 Quad-tree coding

Significant data often sifted from a set of data or subband using quad-tree coding technique. Often, a treshold is used for this purpose. If the data energy is higher than the threshold value, the block remains in the subband otherwise the block is replaced with zeros. This process continues until all the blocks in the subband are checked. At the end that particular subband has some zero coefficients as well as some preserved significant coefficients or the subband has been sifted. Then, the significant coefficients in the sifted subbands are searched and saved as a unit following a top down quadtree structure. Thus, there are two outcomes of this process namely; the significant units and the MAP sequence which tells the location of the significant subband coefficients. The pseudo code shown in Figure 2 illustrates the search of the preserved significant coefficients procedure on a particular sifted subband:

```
1.  Obtain maximum quad-tree level based on subband rectangular size
2.  FOR quad-tree level = 1 to maximum
          a.  call for TEST UNIT
          b.  save MAP sequence and significant unit
3.  END
TEST UNIT:
1.  Divide the subband into 4 descendent subregions
2.  FOR descendent subregion = 1 to 4
    a.  IF descendent  subregion has nonzero components
    b.  Attach "1" to MAP sequence, and further split the descendent  subregion into
        another 4 equal descendent subregions
          i.  IF size of subregion equal block size
         ii.  save block into significant unit
      ELSE
    a.  Attach "0" to MAP sequence and stop splitting the descendent subregion, and
        return to one level up of the quad-tree levels.
    b.  Return MAP sequence and significant unit
3.  END
```

Fig. 2. Search procedure of the significant coefficients

Figure 3 and Figure 4 show construction of a top down quad-tree structure of the MAP sequence out of a sifted subband. The symbol $X$ in Figure 3 shows the nonzero value of subband coefficients. The quad-tree structure produces a degree of compression to the MAP sequence as shown in Figure 4.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ***0*** | *X* | 0 | 0 | ***0*** | ***0*** | 0 | 0 |
| *X* | ***0*** | 0 | 0 | ***0*** | ***0*** | 0 | 0 |
| 0 | 0 | ***0*** | ***0*** | 0 | 0 | ***0*** | ***0*** |
| 0 | 0 | ***0*** | ***0*** | 0 | 0 | ***0*** | ***0*** |
| ***0*** | ***0*** | 0 | 0 | ***0*** | ***0*** | 0 | 0 |
| ***0*** | *X* | 0 | 0 | ***0*** | ***0*** | 0 | 0 |
| 0 | 0 | ***0*** | ***0*** | 0 | 0 | ***0*** | ***0*** |
| 0 | 0 | ***0*** | ***0*** | 0 | 0 | ***0*** | ***0*** |

Fig. 3. Part of a sifted subband

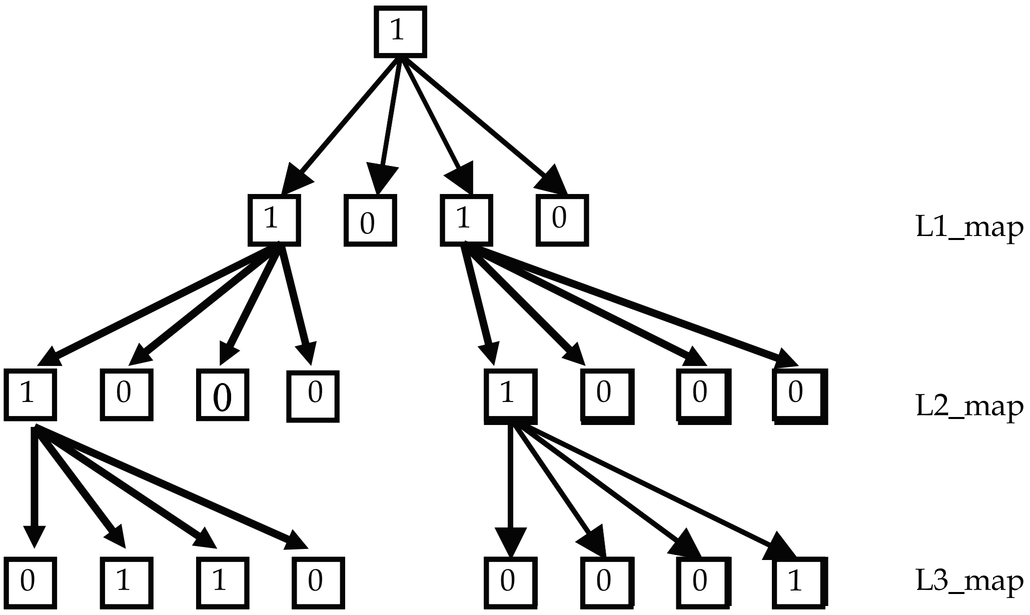


Fig. 4. Corresponding quad-tree representation of MAP sequence

## 3. Multistage lattice vector quantization video codec

In this section, the implementation of the proposed video codec based on the MLVQ technique is presented. The same high frequency subbands are grouped and their significant coefficients are processed by the MLVQ algorithm for lossy compression. The encoding procedure for MLVQ is also presented in the following subsections.

### 3.1 Overview MLVQ video codec

The block diagram of the MLVQ video codec is shown in Figure 5. The video codec takes a video sequence and passes it to a frame buffer. The buffer dispatches $n$ frames at a time to $m$ DWT blocks, thus effectively group the video sequence into a group of $n$ frames. Each of these $m$ DWT blocks performs 2-D discrete wavelet transform using JPEG2000 wavelet

(Lawson & Zhu, 2002). Then, the same high frequency subbands from each frame are joined together.

The coefficients of the high frequency subbands from each group are first subdivided into a predefine unit block size of $N \times N$, which ultimately defines the size of vector dimension. Then the significant vectors are searched or identified in each subband group by comparing the individual vector's energy with a threshold value. The preserved significant vectors from the subband group are then passed to the multistage lattice VQ (MLVQ) process for lossy compression. The MLVQ process produces two outputs i.e. the scale list and index sequence. The index sequence is then entropy coded using the Golomb coding (Golomb, 66). The location information of the significant units is defined as the MAP sequence (Man et al., 99) represented in the form of binary ones and zeros. If the coefficient block size is significant the MAP data is one otherwise it is zero. The lowest frequency subband group is compressed using the lossless coding. The non-predictive MLVQ video coder operates at a fixed frame rate and a fixed output bit rate by allocating a constant number of bits to all the frames of the input video sequence.

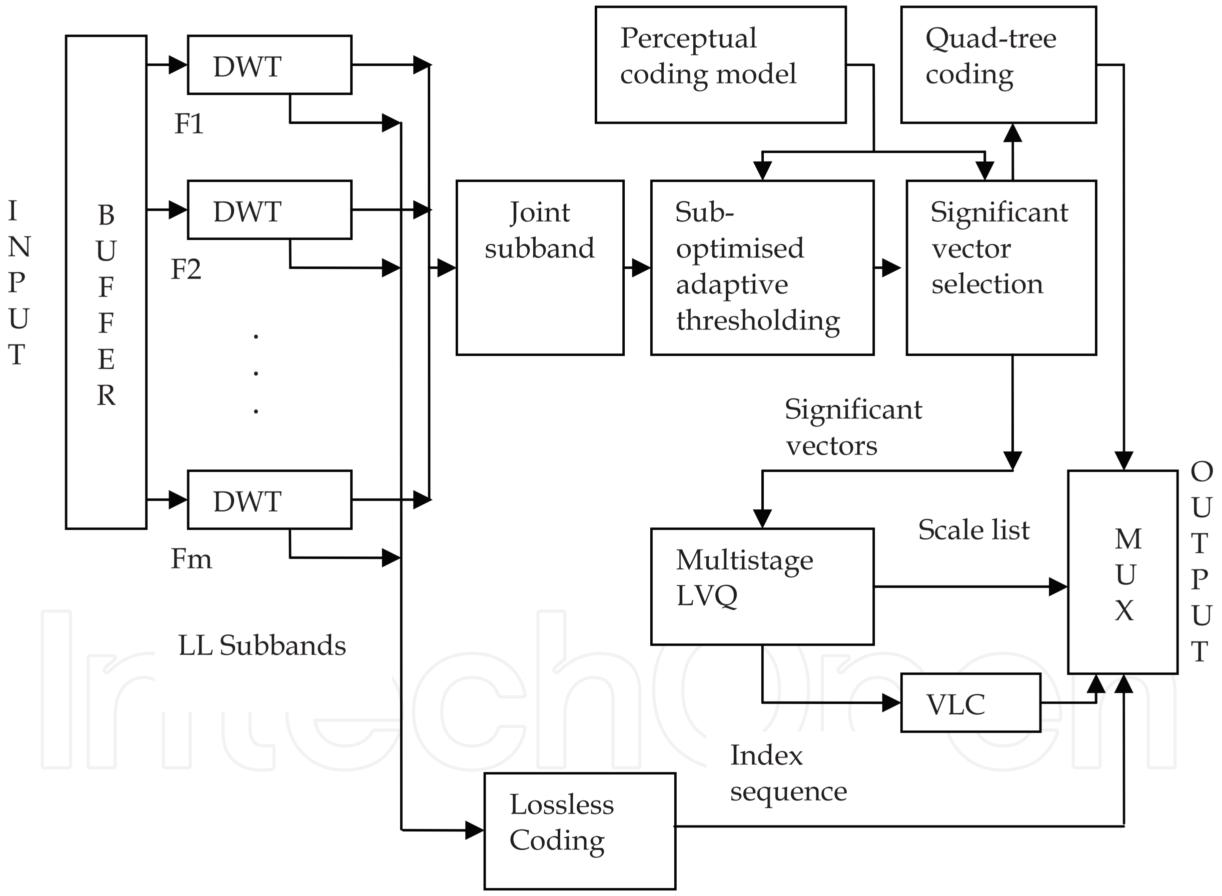


Fig. 5. Non-predictive MLVQ video encoder

### 3.2 Joining high frequency subbands

A video sequence contains high temporal redundancy due to the similarity of the successive frames content. One way to reduce the temporal redundancy is to get the difference frame between the successive frames in a group of picture (GOP). In many video coding standards that use motion estimation and compensation, this redundancy is exploited via the prediction process. However, this technique produces a video coding scheme that is not robust, particularly in mobile environments. Moreover, motion estimation technique involves high computational loads and design complexity. The advantage of joining the high frequency subbands within the GOP is that, if one or more of the code joint high frequency subbands bitstream are corrupted, the GOP can still be reconstructed using the other joint subbands as well as the low frequency subbands. Unlike video standards which employed motion estimation, the lost of motion vector data results in the lost of the prediction frames leaving only the intra frames for reconstruction.

The non-predictive MLVQ video codec joins the same subbands within the GOP results in $3L_{\max}$ groups of subbands as illustrated in Figure 6 below. In this case $L_{\max}$ denotes the number of DWT level. The significant coefficients of each subband group are then selected using the quad-tree coding. The preserved significant coefficients of each subband group are then coded using the multistage lattice vector quantization (MLVQ) encoding process. Applying the MLVQ encoding to the preserved significant coefficients of the subband groups exploits the spatial and temporal redundancy of the video sequence.

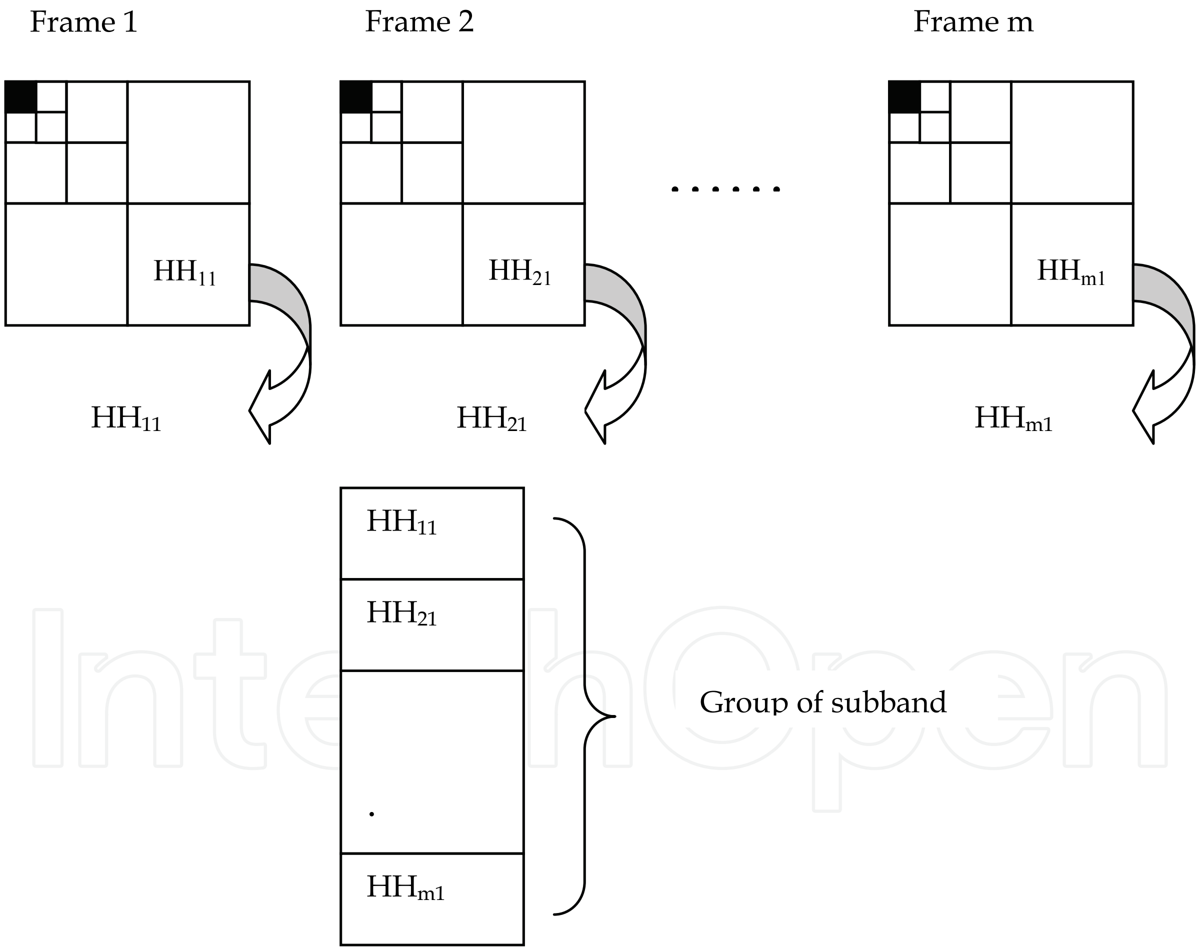


Fig. 6. Joining subband significant coefficients in a GOP

### 3.3 Single pass LVQ encoder

The significant coefficients or vectors of every joint subband are quantized and output as the scale list and index list. In this work, the $Z_n$ lattice quantizer has been chosen to quantize the significant vectors. The $Z_n$ spherical codebook is enumerated by theta series. The spherical codebook is chosen since the data to be quantized are the preserved coefficients relative to a threshold, rather than the entire wavelet coefficients of the high frequency subbands. They do not exhibit the Laplacian distribution which requires a pyramidal codebook. In this work, a four-dimensional $Z_4$ lattice codebook has been chosen due its simple codebook construction. The codebook is derived with the first energy level $(m=1)$ has 8 lattice points or vectors, second level $(m=2)$ has 24 vectors, and third level $(m=3)$ has 32 vectors. Therefore, a total of 64 codewords are in the codebook which can be represented by 6-bit index.

**Spherical $Z_n$ LVQ Encoding Procedure**

The significant vectors are quantized using the $Z_n$ spherical LVQ. The encoding procedure of a single stage or pass LVQ process is summarized below:

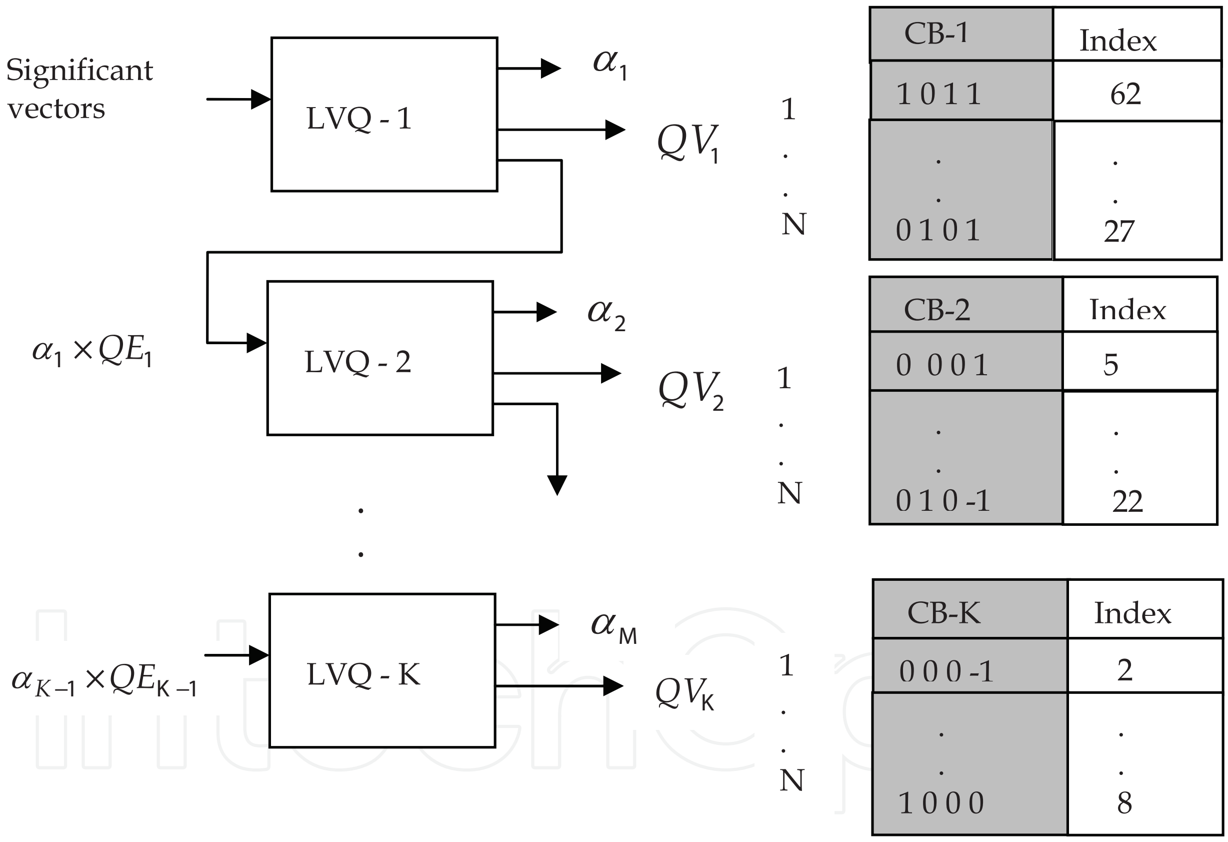
1. *Scale the input vectors.*
   a. *Obtained an energy list from the input vector list. Let $E_i$ be the individual element in the energy list set. $E_i = \sum_{j}^{N} X_j^2$ where $j$ is the column and $i$ is the row of a matrix respectively while $N$ is the dimension of the vector.*
   b. *Find the maximum energy from the list $(E_{\max})$.*
   c. *Define the energy list normalized factor $\beta$ $((0 < \beta \leq 1)$ ), where 1 indicates the maximum energy.*
   d. *Define selected energy: $E_s = \lfloor E_{\max} \times \beta \rfloor$ where $\lfloor . \rfloor$ is a floor function.*
   e. *Scaling factor: $\alpha = \sqrt{\frac{E_s}{m}}$*
   f. *Scale vectors are obtained by dividing each input vectors by the scaling factor $\alpha$ .*
2. *The scaled vectors are quantized using the $Z_n$ lattice quantizing algorithm.*
3. *The output vectors of this algorithm are checked to make sure that they are confined in the chosen spherical codebook radius $m$ .*
4. *If the output vectors exceed the codebook radius $m$ then they are rescaled and remapped to the nearest valid codeword to produce the final quantized vectors $(QV)$.*

In lattice VQ, the vectors can be scaled in such a way that the resulting scaled vector will reside in one of the three combinations of regions i.e. the granular region, overlap regions, or both. The normalized factor $\beta$ serves as the indicator as where the scaled vectors would reside in one of these three regions. If the value of $\beta$ is 1 all the scaled vectors are in the granular region. For example, if the value is 0.8 the majority of the scaled vectors are in the granular region, and few are in the overlap region. Therefore, the optimum value of $\beta$ is obtained from experiment. In the experiment the value of $\beta$ starting from 0.5 up to 0.9 with 0.01 increments are used. Each time the image is reconstructed, and the value of PSNR is the calculated. Therefore, the optimum $\beta$ is found from the best value of PSNR. This value will be used in the first stage or pass of multistage LVQ and the subsequent LVQ stages the scaled vectors are forced to reside only in granular regions $(\beta = 1)$. This is because the quantization errors data have small magnitude variation.

### 3.4 MLVQ encoder

The multistage LVQ (MLVQ) process is illustrated in Figure 6 using an example of a particular group subband. In each LVQ pass three outputs i.e. the scale factor $(\alpha)$, quantized vectors $(QV)$, and the quantization error vectors $(QE)$ are produced. At every pass, the quantized vectors $(QV)$ are obtained using the spherical $Z_n$ LVQ. If the four-dimensional vectors are quantized, and if the origin is included, the outer lattice point will be removed to accommodate the origin.

The first LVQ stage processes the significant vectors and produces a scale factor $(\alpha_1)$, the quantized vectors $(QV_1)$ or codewords, and the quantization error vectors $(QE_1)$ etc. Then the quantization error vectors $(QE_1)$ are "blown out" by multiplying them with the current stage scale factor $(\alpha_1)$. They are then used as the input vectors for the subsequent LVQ stage, and this process repeats up to stage K. Figure 7 illustrates the resulting K-stage codebook generation and the corresponding indexes of a particular subband. Next, the index sequences of the K-stage codebook are variable length coded.



| CB-1 | Index |
|---|---|
| 1 0 1 1 | 62 |
| . | . |
| . | . |
| 0 1 0 1 | 27 |

| CB-2 | Index |
|---|---|
| 0 0 0 1 | 5 |
| . | . |
| . | . |
| 0 1 0 -1 | 22 |

| CB-K | Index |
|---|---|
| 0 0 0 -1 | 2 |
| . | . |
| . | . |
| 1 0 0 0 | 8 |

*K*-stage codebook and the corresponding indexes

Fig. 7. MLVQ process of a particular subband

The flow diagram of MLVQ algorithm that process all the high frequency subbands as well as a simple bit allocation procedure is described in Figure 8. In this work, a three-level DWT system $(L_{\max} = 3)$ results in nine subbands. The encoding process starts from the lower frequency subbands towards the higher frequency subbands. This is because the lower

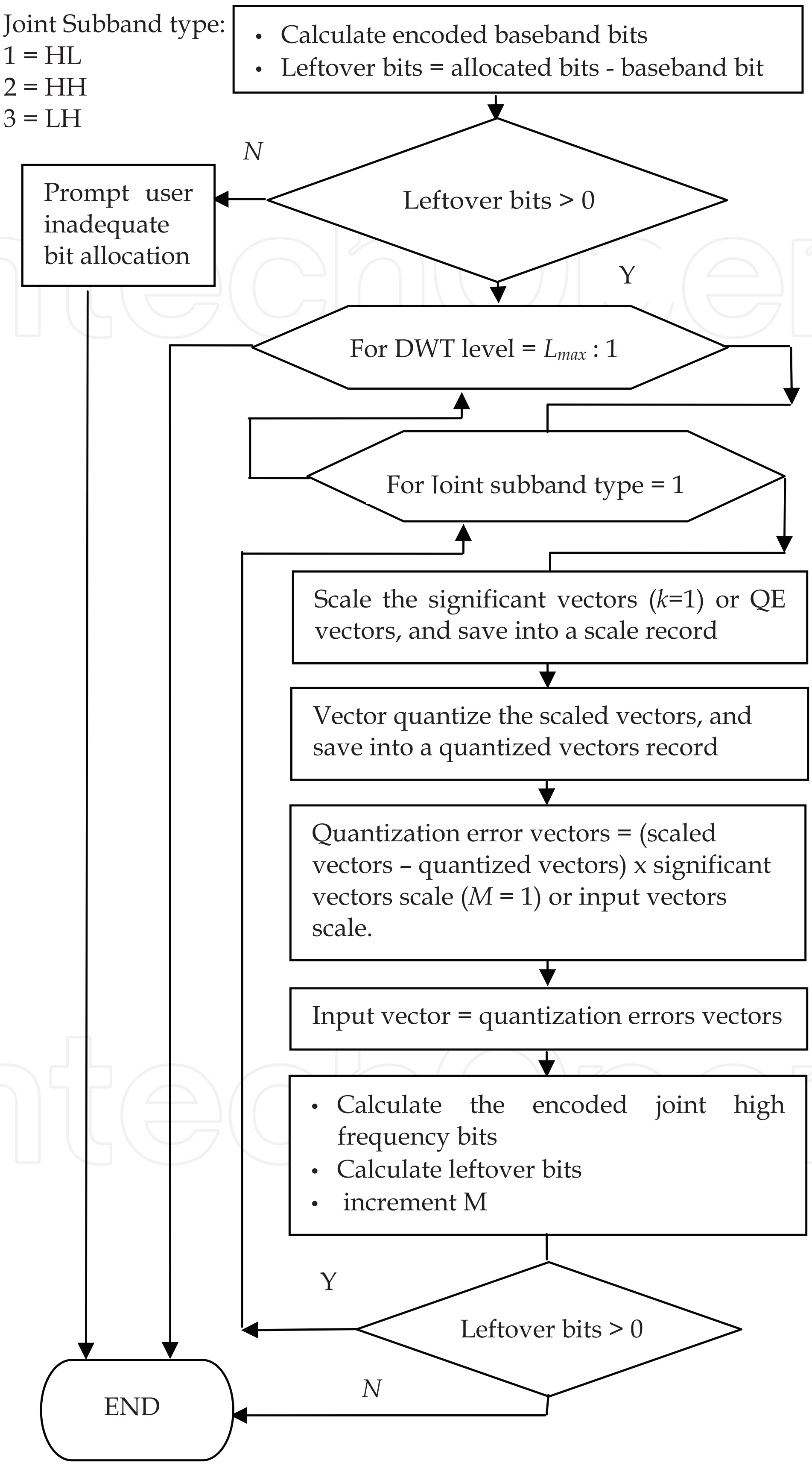

Joint Subband type:
1 = HL
2 = HH
3 = LH
Calculate encoded baseband bits
Leftover bits = allocated bits - baseband bit
Leftover bits > 0
N
Y
Prompt user inadequate bit allocation
For DWT level = $L_{max}$ : 1
For Ioint subband type = 1
Scale the significant vectors ($k$=1) or QE vectors, and save into a scale record
Vector quantize the scaled vectors, and save into a quantized vectors record
Quantization error vectors = (scaled vectors – quantized vectors) x significant vectors scale ($M$ = 1) or input vectors scale.
Input vector = quantization errors vectors
Calculate the encoded joint high frequency bits
Calculate leftover bits
increment M
Y
Leftover bits > 0
N
END

Fig. 8. Flow diagram of MLVQ algorithm

subbands contain more important information and therefore have more impact on the reconstructed image.

In this work, a simple bit allocation procedure is employed. The encoder calculates first the corresponding amount of bit usage for the lower subband. Then, the left over bits for high frequency subbands is obtained by subtracting this amount from the total bit allocated. Subsequently, the high frequency subbands are encoded starting from the joint HL, HH, and LH subbands. In each joint subband the amount of encoded bits used is calculated, and the leftover bit is obtained. The encoding process continues for the subsequence quantization stage if all the three high frequency subbands have been encoded. The process ends when the left over bit is exhausted. In this work, the experimental data has prevailed that the optimum performance of the codec occurs when there are three multistage processes to encode the video sequence.

In this algorithm, the residual data or quantization error vectors are captured and sent to the decoder to increase the reconstruction of frames quality. The quantization errors vectors are produced and *magnified* as the extra set of input vectors to be quantized. The advantage of magnifying the quantization errors vectors is that many vectors which have components near zero can be quantized to many more lattice points during the subsequent LVQ stages. Thus, more quantization errors can be captured and the MLVQ can produce better frame quality.

## 4. MLVQ video transmission system

The block diagram of the MLVQ video codec for transmission in mobile channel is shown in Figure 9. The *Lossy Video Compression* is the same process as the MLVQ encoder which encodes the video sequence with some compression gain. The compressed bitstream is then classified according to a predefine syntax structure in the *Bitstream Constructor* process. In this stage the bitstream is enciphered into two parts i.e. the header and texture. In the *RS Encoder* process the forward error correction (FEC) codes are added to the bitstream using Reed Solomon codes (Reed & Solomon, 60). In the next stage, the coded header and texture

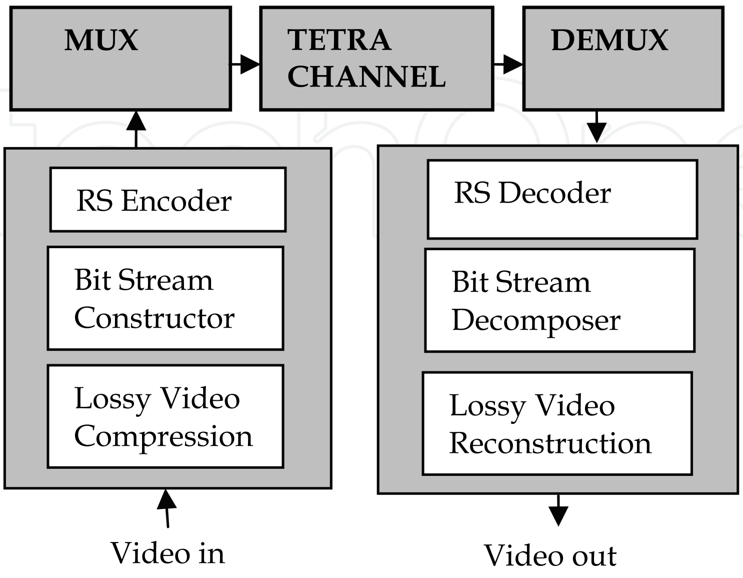


Fig. 9. Block diagram of MLVQ video transmission

are combined together in and alternating structure before they are passed through the TETRA Channel simulator. The received data are first de-multiplexed to the header and texture bitstreams. Then the *RS Decoder* process eliminates the added bit redundancy in the coded bitstreams. Then, the *Bitstream Decomposer* process deciphers the received bitstreams to the meaningful data for video decoding. The final stage is the *Lossy Video Reconstruction* process, where the compressed video sequence is reconstructed. The bitstream syntaxes are protected using the forward error correction codes (FEC) using the RS codes before they are transmitted in mobile channels. In this work, two error resilient schemes are developed i.e. the Equal Error Protection (UEP) and Unequal Error Protection (EEP) schemes.

### 4.1 Unequal error protection

In this work the shortened Reed Solomon codes are selected for forward error correction due to their burst error and erasure correcting capabilities, which makes them suitable to be used for error correction in mobile applications. Table 1 shows the properties of the shortened RS codes for FEC (Akbari, 2004).

| Original Code | Shortened Code | Code rate | Errors/Erasures Corrected |
|---|---|---|---|
| RS (255 , 237) | RS (54 , 36) | 2/3 | 9 errors/18 erasures |
| RS (255 , 219) | RS (54 , 18) | 1/3 | 18 errors/36 erasures |
| RS (255 , 215) | RS (54 , 14) | 1/4 | 20 errors/40 erasures |

Table 1. RS codes properties (Akbari, 2004).

The UEP scheme applied to the MLVQ codec is shown in Figure 10. The labels $S_1$ and $S_2$ represent the streams of different levels of priority produced by the MLVQ encoder. The $C_1$ and $C_2$ denote the channel codes being used with rates of $r_1$ and $r_2$ respectively, where $r_1 < r_2$.

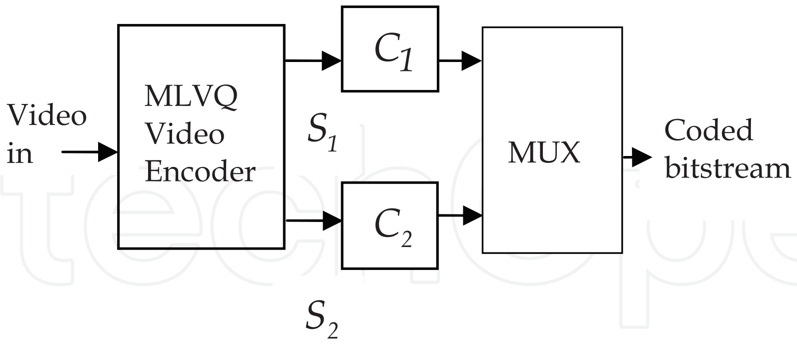


Fig. 10. UEP scheme for MLVQ video codec

The MLVQ video codec bitstream is partitioned into a header and texture syntaxes. The header field contains parameters that can potentially cause decoding failure or loss of synchronisation, while the texture data defines the quality of the decoded frames without having any influence on the decoding process or codec synchronization. The purpose of partitioning the video bitstream is to have the UEP scheme where the header data are protected with more bit redundancy (code rate $r_1$) and the texture data is protected with less bit redundancy (code rate $r_2$) where $r_1 < r_2$.

The header data for each group of pictures (GOPs) contains important data for decoding such as picture start code, quad-tree parameters (MAP sequence), index sequence and scale list, which are transmitted through the channel with code rate $r_1$. The texture data contain the low frequency subband data are passed through the channel with code rate $r_2$. In this work the code rate of $r_1 = 1/4$ and $r_2 = 2/3$ are used UEP scheme. The bit allocation for each group of pictures takes into account the additional bits consumed by the UEP scheme in the following way:

- Suppose $F$ is the allocated bits for a GOP
- First, calculate the LL subands bits used after lossless coding
- Then, computes the bits used to encode the texture data: $bit_{TXT} = bit_{LL} \times \frac{1}{r_2}$
- Then, calculate the left over bits: $bit_{LFOVR} = F - bit_{TXT}$
- Thus, bits requires to encode header is given: $bit_{HDR} = bit_{LFOVR} \times r_1$

### 4.2 Equal Error Protection (EEP)

The equal error protection (EEP) uses the same amount of bit redundancy to protect both header and texture data. In this work, this scheme is developed for the purpose of comparing the performance of the MLVQ video codec with EEP with the H.263 video standard with EEP scheme. In addition, for further comparison with the current video coding standard, the the EEP scheme for H.264 has also been developed. The bitstream of H.264 format obtained from the JM reference software version 10.1 (JM H.264, http://iphome.hhi.de/suehring/tml/) is protected globally with the RS codes before sending the protected bitstream through TETRA channel. In this work the code rate of 1/3 is used for the protection of the source data for both $C_1$ and $C_2$.

## 5. Results

This section presents the results of video transmission on error free channel. In this experiment the performance of the proposed MLVQ video codec is compared with the AJSVQ (Akbari & Soraghan, 2003) over noiseless channel in order to show the incremental results. The performance comparison with other video standards is also conducted for various bit rates. Then, the performance comparison between the codecs in mobile channels using the TETRA channel simulator is conducted.

### 5.1 Transmission in error free channel

The error free channel assumes an ideal channel condition with no lost to video data. All of the test video sequences used are in the form of Quarter Common Intermediate Format (QCIF) format. In this format, the luminance components consist of $144 \times 176$ pixels, while the chrominance components have $72 \times 88$ pixels. The optimized value found from experiment for the normalized energy factor $(\beta = 0.76)$. This value is used in spherical LVQ encoding scheme of the first stage or pass of the multistage quantization process found after using "Foreman", "Carphone", "Miss America" and "Salesman" video sequences. The value is used throughout the high frequency subbands encoding process. In the subsequent passes of the MLVQ scheme the value is set to one $(\beta = 1.0)$.

A preliminary experiment is conducted as to show the incremental performance of the new MLVQ video codec, where the test video sequence "Miss America" is first encoded using 64kbps at 8 fps and compared to the non-predictive video codecs. In this case, the performance results are compared to the motion JPEG2000 standard and AJSVQ video codec (Akbari & Soraghan, 2003) (Akbari 2003). In order to emulate the motion JPEG2000 video scheme, the individual gray frame of "Miss America" of size 176x144 pixels per frame is coded using the JPEG2000 image still coding standard. Fig. 11 below shows the relative performance of the first frame of "Miss America" sequence between MLVQ and AJSVQ. Other test sequences like "Salesman" and "Carphone" are also used in simulation. Fig. 12 and Fig. 13 show the results taken at various bit rates for frame rate of 12 fps. The results show that in noiseless environment H.264 always has the best performance as compared to the rest of the codecs.

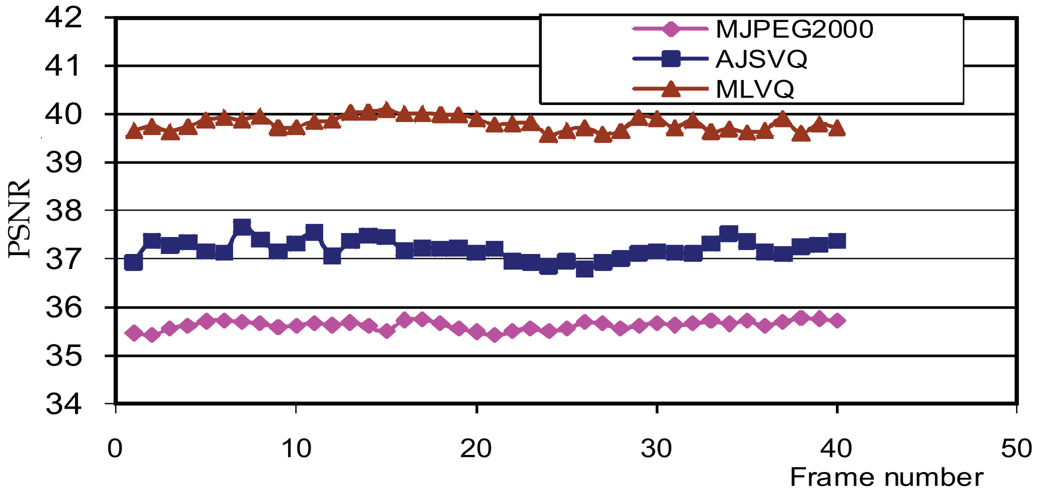


Fig. 11. Relative performance of "Miss America" at 8 fps

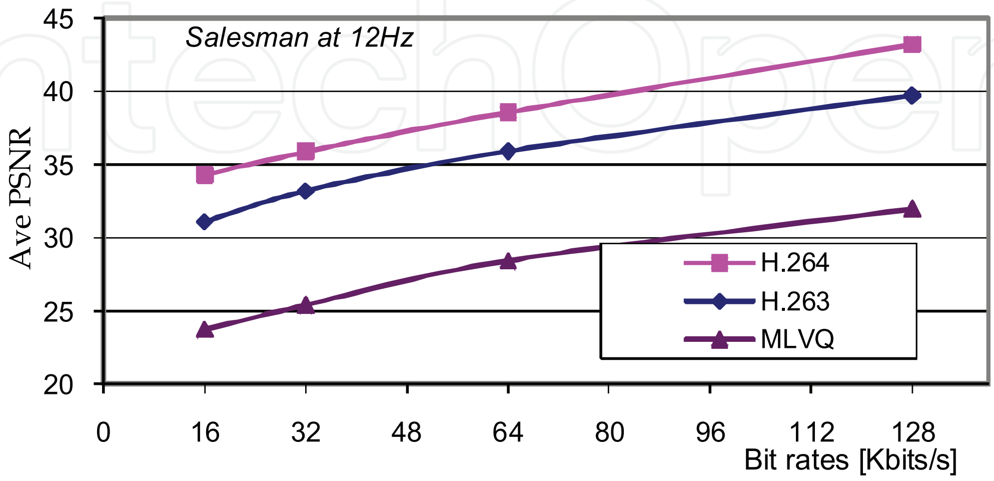


Fig. 12. "Salesman" over noiseless channel

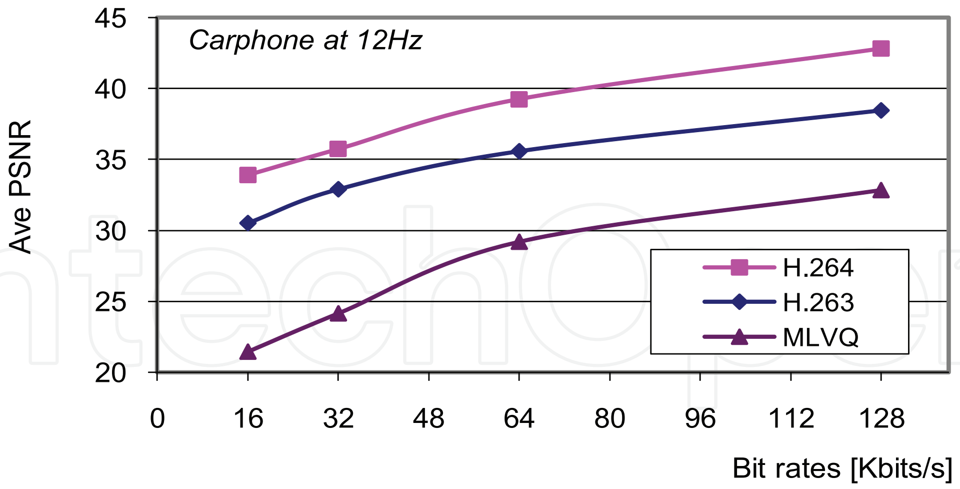


Fig. 13. “Carphone” over noiseless channel

### 5.2 Transmission in mobile channels

In this subsection, two experimental results will be presented. First, the performance comparison results between the proposed MLVQ video codec with H.263 are presented. Secondly, the comparison performance results of the new codec against H.264 video standard are also presented.

### 5.3 Comparison with H.263

In this experiment the H.263+ encoder that employs Annexes D (Advanced Prediction Mode), F (Unrestricted Motion Vectors) and G (PB-frames) is used for simulation. The shortened RS codes with code rate of 1/3 and block interleaving at a depth of four with each ECC are utilized to equally protect the compressed H.263 bitstream and mitigate the effects of burst errors. The Bad Urban (BU) and Hilly Terrain (HT) environments with channel SNR equal to 18 dBs of the TETRA channel simulator (ETSI, 95) are used in the experiment. The video is encoded using bit rate 64 kb/s and frame rate of 15 fps throughout this experiment.
Fig. 14 and Fig. 15 show the performance of the ‘Foreman’ and ‘Carphone’ test sequences particularly on frame rate 15 fps, at 18 dB channel SNR in bad urban (BU) and hilly terrain (HT) mobile environments respectively. The results show the performance of MLVQ with UEP is always better than the MLVQ with EEP. The performance of the MLVQ codec with EEP scheme is also compared to the H.263 with EEP schemes. This gives a fair comparison since both codecs use the same technique for error protection. The performance of MLVQ with UEP scheme is then compared to the MLVQ with EEP scheme to show the improvement gain due to the UEP scheme.

### 5.4 Comparison with H.264

In this experiment, MLVQ codec with UEP scheme has been chosen since it offers the best performance of forward error correction scheme. The baseline profile of the H.264 standard (JM H.264, http://iphome.hhi.de/suehring/tml/) is used since it is used for application in mobile environment. The baseline profile H.264 is equipped with the error resilient tools

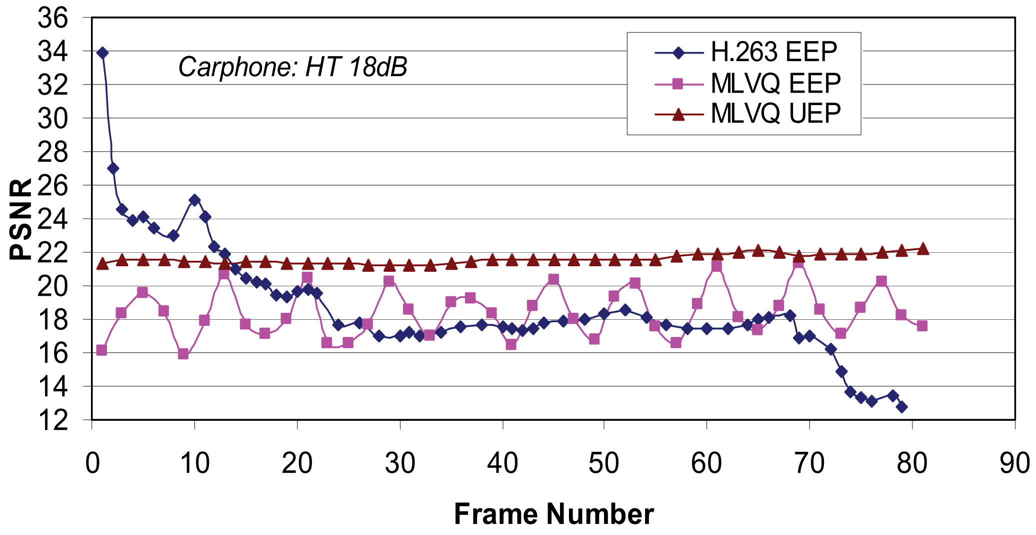


Fig. 14. “Foreman” sequence over BU18 channel, 15 fps.

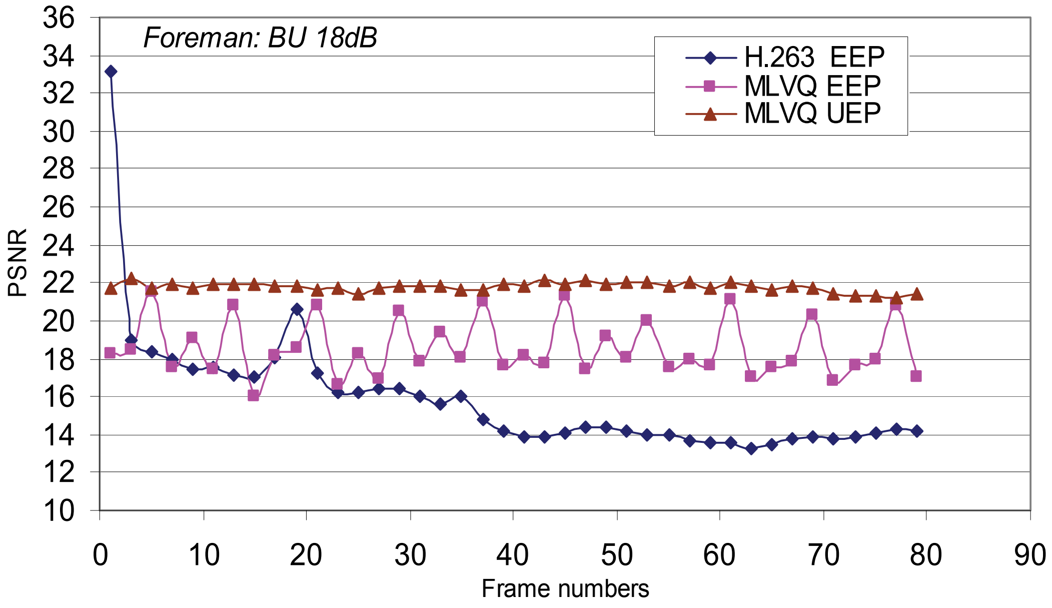


Fig. 15. “Carphone” sequence over HT18 channel, 15 fps.

such as redundant slices, flexible macroblock ordering, macroblock line intra refreshing and feedback channel. However, the baseline profile does not support data partition. Hence, the equal error protection (EEP) scheme with RS code rate ¼ is used for bits error protection before being transmitted in the mobile channels. In this experiment the TETRA Hilly Terrain (HT) channel is used as the mobile environment. The test sequences "Foreman", "Carphone" and "Miss America" with bit rate of 64kbits/s and frame rate at 15 fps are used. In this experiment, the MLVQ codec are compared to the H.264 with error resilient tools enabled. Table 2 below summarizes the error resilient tool used in the experiment.

| Error resilient tool features | H.264 bitstream |
|---|---|
| Slice mode | 50 MB per slice |
| Error concealment | No |
| Redundant slices | No |
| FMO | Interleave mode |
| MB line intra refreshing | Yes |
| Feedback channel | No |

Table 2. Error resilient tools used in H.264 video standard

Table 3 shows the average PSNR obtained for the two codecs, where the average PSNR of H.264 is higher than MLVQ codec. This is due to the fact that the PSNR values of the earlier decoded H.264 frames are always higher. As the number of frame gets higher the PSNR values of H.264 are comparable to MLVQ as shown in Figure 16.

| Mobile Environment | Sequence | H.264 | MLVQ |
|---|---|---|---|
| | | PSNR | PSNR |
| HT | Carphone | 25.36 | 21.70 |
| | Foreman | 24.43 | 21.80 |
| | Miss America | 31.20 | 29.86 |

Table 3. Average PSNR of test sequences as compared to H.264 at SNR18dB

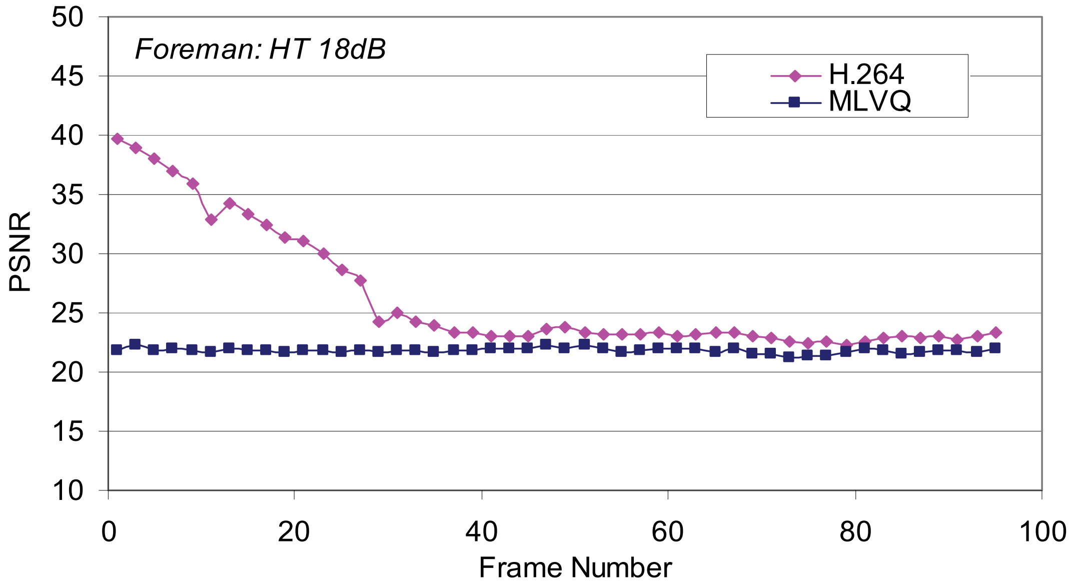


Fig. 16. “Foreman” sequence over HT18 channel

## 6. Conclusion

In this paper the multistage lattice vector quantization (MLVQ) coding scheme for mobile applications has been presented. The video codec groups the video sequence into a group of m-frames by grouping the high frequency subbands from each frame. The significant coefficients or vectors are then quantized using the MLVQ coding. The lattice vector quantization offers less amount of computation in generating the codebook due to its regular structure. Since the codebook generation does not require any computation, this facilitates the use of multistage quantization process to capture the quantization residual errors. This enhances the reconstructed frame quality. In noiseless environment, MLVQ is always inferior from the H.263 and H.264 standard codecs. However, the performance of the non-predictive MLVQ scheme is superior to the H.263 in mobile environment since the new video codec does not contain any motion vectors data. The non-predictive MLVQ codec performs comparably near to the performance of the latest H.264 video codec in erroneous mobile channels.

## 7. Future research

The forward error correction adds redundant bits to the coded bitstream, which allows the decoder to correct errors up to certain level. However, this reduces compression ratio. Moreover, the FEC must be designed with the assumption of the worst case scenario. If for example, the coded video is transmitted through an error-free channel, the additional bits are unnecessary. In another situation, where the channel might have highly variable quality the worst case situation also vary. Therefore, this suggests the need to employ the very powerful codes. In other words, these scenarios address a problem of efficient bits allocation for forward error correction technique, while minimizing the reduction of compression ratio. One area of future direction could be to investigate the use of multiple descriptions coding (MDC). In this way, the joint source i.e. the MLVQ video data and the channel coding method could provide an effective way for error resilience with relatively small reduction in compression ratio.

## 8. References

Adoul, J. P.; and Barth, M. (1988). "Nearest neighbor algorithm for spherical codes from the Leech lattice," *IEEE Trans. Information Theory*, Vol. 34, No. 5, Sept. 1988, pp. 1188-1202, ISSN: 0018-9448.

Akbari, A. S.; and Soraghan, J. J. (2003). "Adaptive joint subband vector quantization codec for handheld videophone applications," *Electronic Letters*, vol. 39, no. 14, July 2003, pp.1044-1046, ISSN: 0013-5194.

Akbari, A. S. (2004). "Fuzzy Multiscale based Image and Video Processing with Applications," PhD Thesis, Institute for Communication and Signal Processing, Department of Electrical and Electronic Engineering, University of Strathclyde, Glasgow, UK, December 2004.

Barlaud, M.; et al., (1994). "Pyramidal Lattice Vector Quantization for Multiscale Image Coding," *IEEE Trans. Image Processing*, Vol. 3, No. 4, July 1994, pp. 367-381, ISSN: 1057-7149.

Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the Leech lattice", *IEEE J. Selected Areas in Comm.,* Vol. 7, No. 6, Aug. 1989, pp. 959 – 967, ISSN: 0733-8716.

Conway, J. H.; and Sloane, N. J. A; (1988). *Sphere Packings, Lattices, and Groups*, Springer-Verlag, ISBN-10: 0387985859, New York.

Conway, J. H.; and Sloane, N. J. A. (1982). "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes," *IEEE Trans. Information Theory*, vol. IT-28, March 1982, pp. 227-232, ISSN: 0018-9448.

Dufaux F.; and Ebrahimi, T. (2004). "Error-Resilient Video Coding Performance Analysis of Motion Jpeg-2000 and MPEG-4," *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, January 2004.

ETSI: TETRA Voice and Data, Part 2: (1995). "Air Interface, ETS 300 392-2," November 1995.

Feideropoulou, G.; et al., (2007). "Rotated Constellations for Video Transmission Over Rayleigh Fading Channels," *IEEE Signal Processing Letters*, Vol. 14, No. 9, Sept. 2007, pp. 629 – 632, ISSN: 1070-9908.

Forney Jr., G. D. (1988). "Coset codes. II. binary lattices and related codes," *IEEE Trans. Information Theory*, Vol. 34, No. 5, Sept 1988, pp. 1152 -1187, ISSN: 0018-9448.

Gersho, A.; and Gray, R.M. (1992). *Vector Quantization and Signal Compression*, Kluwer Academic, ISBN-10: 0792391810, New York.

Ghanbari, M. (2003). *Standard Codecs: Image Compression to Advanced Video Coding*, The Institution of Engineering and Technology, ISBN-10: 0852967101, London, UK.

Gibson, J. D.; and Sayood, K. (1988). "Lattice Quantization," *Advances in Electronics and Electron Physics*, Vol. 72, pp. 259-330, P. Hawkes, ed., Academic Press.

Golomb, S. W. (1966) "Run-length Encodings," *IEEE Trans. Inf. Theory*, Vol. IT-12, No. 3, July 1966, pp. 399-401, ISSN: 0018-9448.

ISO/IEC 15444-3:2002, (2002). "Information technology – JPEG2000 image coding system – Part 3: Motion JPEG2000," 2002.

ITU-T Recommendation H.263, (2000). “Video codec for Low bit rate communication,” Version 3, November 2000.

JM H.264/AVC Reference Software available at homepage http://iphome.hhi.de/suehring/tml/.

Kuo, C. M. et al., (2002). “Multiresolution Video Coding Based on Kalman Filtering Motion Estimation,” *Journal of Visual Communication and Image Representation*, Vol. 13, No. 3, September 2002, pp. 348-362, ISSN: 1047-3203.

Lawson, S.; and Zhu, J. (2002). “Image compression using wavelets and JPEG2000: a tutorial,” *Electronics & Communication Engineering Journal*, Vol. 14, No. 3, June 2002, pp. 112-121, *ISSN:* 0954-0695.

Man, H. et al., (1999). “A Family of Efficient and Channel Error Resilient Wavelet/Suband Image Coders,” *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.9, No.1, Feb. 1999, pp. 95–108, ISSN: 1051-8215.

Man, H. et al., (2002). “Three-dimensional subband coding techniques for wireless video communications,“ *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 12, No. 6, June 2002, pp. 386-397, ISSN: 1051-8215.

Ostermann, et al, (2004). “Video coding with H.264/AVC: Tools, Performance, and Complexity,” *IEEE Circuits and System Magazine*, Vol. 4, No. 1, First Quarter 2004, pp. 7-28, ISSN: 0163-6812.

Ran, M.; and Snyders, J. (1998). “Efficient decoding of the Gosset, Coxeter-Todd and the Barnes-Wall Lattices” *Proc. IEEE Int. Symp. Information Theory*, ISBN: 0-7803-5000-6, Cambridge, MA, August 1998, pp. 92, IEEE, New York.

Reed, I. S.; and Solomon, G. (1960). "Polynomial codes over certain finite fields," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 8, June 1960,pp. 300--304, *ISSN*: 03684245.

Salleh, M. F. M.; and Soraghan, J. (2005). “A new multistage lattice VQ (MLVQ) technique for image compression,” *CD-ROM Proc. European Signal Processing Conference 2005*, Antalya, Turkey.

Salleh, M. F. M.; and Soraghan, J. (2006). “A new adaptive subband thresholding algorithm for multistage lattice VQ image coding,” *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP 2006)*, Vol. 2, pp. 457-460, ISBN: 1-4244-0469-X, Toulouse France, IEEE, New York.

Salleh, M. F. M.; and Soraghan, J. (2007). “Non-predictive multi-scaled based MLVQ video coding for robust transmission over mobile channels,” *Electronics Letters*, Vol. 43, No. 13, June 2007, pp. 711-712, ISSN: 0013-5194.

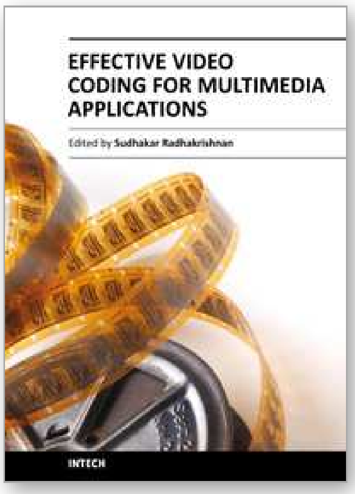
Sampson, D. G. et al., (1995). “Low bit-rate video coding using wavelet vector quantisation,” *IEE Proceedings Vision, Image and Signal Processing*, Vol. 142, No. 3, June 1995, pp. 141 – 148, ISSN: 1350-245X.

Sikora, T. (2005). “Trends and perspectives in image and video coding,” *Proceedings of IEEE*, Vol. 93, No. 1, Jan 2005, pp. 6-17, ISSN: 0018-9219.

Sloane, N. J. A. (1981). “Tables of Sphere Packings and Spherical Codes”, *IEEE Transactions on Information Theory*, Vol. IT-27, No. 3, May 1981, pp. 327-338, ISSN: 0018-9448.

Voukelatos, S. P.; and Soraghan, J. J. (1997) “Very Low Bit Rate Colour Video Coding Using Adaptive Subband Vector Quantization with Dynamic Bit Allocation,” *IEEE Transactions. on Circuits and Systems for Video Technology,* Vol. 7, no. 2, April 1997, pp. 424-428, ISSN: 1051-8215.

Weiping, L. et al., (1997). “A video coding algorithm using vector-based techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, Feb. 1997, pp.146 – 157, ISSN: 1051-8215.



**Effective Video Coding for Multimedia Applications**
Edited by Dr Sudhakar Radhakrishnan

ISBN 978-953-307-177-0
Hard cover, 292 pages
**Publisher** InTech
**Published online** 26, April, 2011
**Published in print edition** April, 2011

Information has become one of the most valuable assets in the modern era. Within the last 5-10 years, the demand for multimedia applications has increased enormously. Like many other recent developments, the materialization of image and video encoding is due to the contribution from major areas like good network access, good amount of fast processors e.t.c. Many standardization procedures were carrried out for the development of image and video coding. The advancement of computer storage technology continues at a rapid pace as a means of reducing storage requirements of an image and video as most situation warrants. Thus, the science of digital video compression/coding has emerged. This storage capacity seems to be more impressive when it is realized that the intent is to deliver very high quality video to the end user with as few visible artifacts as possible. Current methods of video compression such as Moving Pictures Experts Group (MPEG) standard provide good performance in terms of retaining video quality while reducing the storage requirements. Many books are available for video coding fundamentals.This book is the research outcome of various Researchers and Professors who have contributed a might in this field. This book suits researchers doing their research in the area of video coding.The understanding of fundamentals of video coding is essential for the reader before reading this book. The book revolves around three different challenges namely (i) Coding strategies (coding efficiency and computational complexity), (ii) Video compression and (iii) Error resilience. The complete efficient video system depends upon source coding, proper inter and intra frame coding, emerging newer transform, quantization techniques and proper error concealment.The book gives the solution of all the challenges and is available in different sections.

**How to reference**
In order to correctly reference this scholarly work, feel free to copy and paste the following:

M. F. M. Salleh and J. Soraghan (2011). Non-Predictive Multistage Lattice Vector Quantization Video Coding, Effective Video Coding for Multimedia Applications, Dr Sudhakar Radhakrishnan (Ed.), ISBN: 978-953-307-177-0, InTech, Available from: http://www.intechopen.com/books/effective-video-coding-for-multimedia-applications/non-predictive-multistage-lattice-vector-quantization-video-coding

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China

www.intechopen.com

51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821