Chapter 2

Background: Video Coding and Communication

2.1 Introduction

Video communication over wireless channels usually involves compression, i.e. source coding, of the video followed by channel coding and modulation, all at the transmitter end. On the receiver side, reverse process is performed to reconstruct the video which consist of demodulation, channel decoding and video decompression. Since, video signals are highly correlated, this feature is exploited by a video source coder, generally referred as video coder, to remove the redundancies for achieving compression. While the video coder removes the redundancies, the channel coder introduces it in a controlled manner to counter the effect of transmission errors. These added redundancies are used by the channel decoder to detect and correct the channel errors in the received video bitstream. The modulation converts the bitstream, generated by channel coder, into waveforms in such a way that they can be coupled and transmitted through the channel efficiently.

A simplified block diagram of video communication system, is shown in Fig. 2.1. Thus, transmission of video data over erroneous channels involves different complex processes, and exploring on this area requires good understanding of these processes.

In this chapter, firstly, the fundamentals of video and their compression techniques are described. Then, overview of channel coding, modulation and transmission channels are presented. Finally, at the end of this chapter, a detailed literature survey of various error resilient techniques, to minimize the effect of transmission errors, are provided.
2.2 Video Basics

Video is a sequence of images (also called frames or pictures). An image is a two dimensional spatial signal that generally represents a projected view of some natural scenes. Digital pictures and videos are digitized versions of analogue pictures and videos. The digitized pictures and videos can be stored and processed easily by the computer and other digital devices. In this thesis, images and videos mean digitized images and videos.

Fig. 2.2 shows a typical image and video structure. Fig. 2.2(a) shows the image structure. The analogue image is converted into digital image by sampling it into \( N \) rows and \( M \) columns yielding its resolution into \( N \times M \) sampling point
Table 2.1: Uncompressed (raw data) bit rates for some popular digital video applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Resolution</th>
<th>Frame Rate (frames/sec)</th>
<th>Bits/pixel</th>
<th>Bit Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Videophone</td>
<td>128x96 pixels/frame</td>
<td>7.5</td>
<td>24</td>
<td>2.1</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>352x288 pixels/frame</td>
<td>30</td>
<td>24</td>
<td>69.6</td>
</tr>
<tr>
<td>Standard TV</td>
<td>720x480 pixels/frame</td>
<td>30</td>
<td>24</td>
<td>237.3</td>
</tr>
<tr>
<td>High-definition TV</td>
<td>1920x1080 pixels/frame</td>
<td>30</td>
<td>24</td>
<td>1.4</td>
</tr>
<tr>
<td>Ultra high-definition TV</td>
<td>3840x2160 pixels/frame</td>
<td>30</td>
<td>24</td>
<td>5.56</td>
</tr>
</tbody>
</table>

(Also called pixels). Larger the resolution, the more clear the picture for fixed spatial dimension (height and width). Value of the pixel, \( s(x, y) \), is a combination of three primary colour signal called Red, Blue and Green, the so-called RGB signals. Each colour is generally represented in 256 levels requiring 8 bits (or byte) and thus, each pixel value is represented in 24 bits.

Fig. 2.2(b) shows the general video structure. A video is nothing but sequence of images captured at regular intervals of time. The rate at which the frames are captured or played back (called frames per second, fps) decides the resolution of the video in time domain. At least 24 fps of the video has to be played back so that eye does not perceive the discontinuity in watching it. The pixel values in video are recognized as \( s(x, y, t) \). Table 2.1 shows the typical resolution, frame rate and uncompressed bit rate for some standard video applications. Earlier decade videophone applications were popular and operated on low resolution, while in recent years, due to technological advancement in storage, processing, networking and display devices, demands for high resolution has increased which yields the latest ultra-high definition resolutions.

2.2.1 Color Spaces

A digital image or a video is represented in a colour space with three components used to convey the colour information [46, 47, 48]. There are two commonly used colour models: Red-Green-Blue (RGB) and Luminance-Chrominance (LC). The RGB colour space is commonly used in computer graphics as it is compatible with colour display devices, but three colour components are significantly correlated.

An LC space consist of a luminance (or intensity) component and two chrominance (colour difference) components. The LC colour space such as YIQ, YUV or YCbCr are mainly used in television applications. The LC spaces are popular because the luminance signal can be used to generate grey scale image, which is compatible with monochrome systems. The three colour components (LC colour space) have little correlation, which facilitates the encoding and modulation of the signals [49]. The YUV colour space is commonly used in colour image and
Chapter 2. Background: Video Coding and Communication

2.2.2 Chroma Formats

Since human eye is more sensitive to the luminance (Y) information, in YUV space, the chrominance components (U and V) can be sub-sampled while still preserving good viewable image quality [48]. An added benefit of sub-sampling is the reduction of bandwidth needed for the transmission. The chroma format refers to the relation of spatial sampling ratios between the luminance and chrominance components. The sub-sampling is denoted in the format X:X:X, where the first digit represent the number of the luminance sample, used as a reference and typically '4'. The second and third digits are the number of chrominance samples with respect to number of Y samples. For example, 4:1:1 format means that for every four Y samples, there are one U and one V samples. An exception of this nomenclature is 4:2:0 format, which has same number of chrominance samples as 4:1:1 format, but uses different method for down-sampling.

The commonly used chrominance sub-sample formats are 4:4:4, 4:2:2, 4:1:1 and 4:2:0 [46, 50, 51] which are shown in Fig. 2.3. A YUV space without any

Figure 2.3: YUV colour format with various chrominance sub-sampling (a) 4:4:4 format (no sub-sampling), (b) 4:2:2 format, (c) 4:1:1 format and (d) 4:2:0 format.

video compression. In a digital colour image, each pixel has three 8-bit values associated with it, one for each of the three colour components.
Table 2.2: Different Video formats: resolution for luminance component, frame rate and chrominance sub-sampling

<table>
<thead>
<tr>
<th>Video Format</th>
<th>Luminance Frame size</th>
<th>Frame Rate (Hz)</th>
<th>Chroma format</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of pixels/line</td>
<td>Number of lines</td>
<td></td>
</tr>
<tr>
<td>HDTV (European)</td>
<td>1920</td>
<td>1152</td>
<td>50</td>
</tr>
<tr>
<td>HDTV (American)</td>
<td>1920</td>
<td>1080</td>
<td>60</td>
</tr>
<tr>
<td>CCIR-601 (625 lines)</td>
<td>720 (704 active)</td>
<td>576</td>
<td>25</td>
</tr>
<tr>
<td>CCIR-601 (525 lines)</td>
<td>720 (704 active)</td>
<td>480</td>
<td>30 (29.97)</td>
</tr>
<tr>
<td>SIF-625</td>
<td>352 (360)</td>
<td>288</td>
<td>25</td>
</tr>
<tr>
<td>SIF-625</td>
<td>352 (360)</td>
<td>240</td>
<td>30</td>
</tr>
<tr>
<td>Sub-QCIF</td>
<td>128</td>
<td>96</td>
<td>30</td>
</tr>
<tr>
<td>QCIF</td>
<td>176</td>
<td>144</td>
<td>30</td>
</tr>
<tr>
<td>CIF</td>
<td>352</td>
<td>288</td>
<td>30</td>
</tr>
<tr>
<td>4CIF</td>
<td>704</td>
<td>576</td>
<td>30</td>
</tr>
<tr>
<td>16CIF</td>
<td>1408</td>
<td>1152</td>
<td>30</td>
</tr>
</tbody>
</table>

Chrominance sub-sampling is referred as 4:4:4 format as shown in Fig. 2.3(a), in which every pixel is associated with one Y sample, one U and one V samples. In 4:2:2 format as shown in Fig. 2.3(b), for every four luminance (Y) samples, there are two U and two V samples. The chrominance (U and V) components are down-sampled by a factor of two in horizontal direction only. Similarly, in 4:1:1 format, for every four Y samples, there is one U and one V samples. The chrominance components are sub-sampled by a factor of four in the horizontal direction, but no sub-sampling is performed in the vertical direction. The 4:2:0 format has the same number of samples for the chrominance components as 4:1:1, however the sub-sampling method and location of chrominance samples with respect to luminance samples is different as shown in Fig. 2.3(c) and (d).

### 2.2.3 Video Frame Formats

The size of the frames, that is their width and height, used in digital video are also standardised to facilitate interoperability. The Common Intermediate Format (CIF) was created to facilitate inter-operation between the NTSC and PAL television formats [48]. CIF is a non-interlaced format, with a luminance frame size of 352 pixels per line, 288 non-interlaced lines per frame at 30 frame per second (fps) [46, 52]. The chrominance components are at half the spatial resolution of luminance with 176 pixels per line and 144 lines per frame. The temporal resolutions of chrominance components are the same as for the luminance at 30 Hz (actually 29.97 Hz). It uses 4:2:0 chrominance sub-sampling format. CCIR-601 standard for European broadcasting is different from American standards.
in terms of the number of lines per frame and frame rate, but have the same number of pixels per line (720 pixels/line). To convert a frame from an NTSC format (525 line per field, 30 Hz) to a CIF format, only a line number conversion is needed, since frame rate is the same. To convert a frame from PAL format (576 active lines at 25 Hz) to a CIF format, only frame rate conversion needs to be performed, because CIF has half the active lines than PAL. Other picture formats are also commonly used, some of them are derived from CIF. The various frame formats are shown in Table 2.2.

The Source Input Format (SIF) is another popular video frame format, which is derived from CCIR-601 standard for NTSC. SIF is also a non-interlaced format, in which horizontal and vertical dimensions are half of the CCIR-601 standard. Accordingly, the horizontal resolution should be 360 pixels/line, but since in standard codecs the coding unit is based on macroblocks of $16 \times 16$ pixels, 360 is not divisible by 16. Therefore, four pixels each from leftmost and rightmost sides are removed, resulting into 352 pixels/line.

Throughout this thesis, only the sequences in GIF format are used. Moreover, the formats with frame rate of 29.97 Hz are usually referred to as 30 Hz or 30 frames/sec (fps). Lower frames rates than the indicated values for a given format can be obtained by simple frame-skipping on the original sequences.

### 2.2.4 Video Quality

Video processing (like conversion of digital pictures from one format to another or achieving compression for bit rate reduction) and communication over erroneous channels, leads to distortion. Therefore it is important to measure this distortion which will decide the effectiveness of processing and video quality to viewers. Quality assessment can be categorized into two methods: Subjective and Objective quality assessment [46]:

#### Subjective Quality Measurement

Many subjective quality measurement methodologies have been developed and validated over the years [46]. These procedures uses formal subjective tests in which users rate the quality using a 5-point scale, as shown in Fig. 2.4, with quality ratings ranging from bad to excellent. Usually, a reconstructed video is shown to the subjects (users) for certain time duration and asked to rate the impairments of the video ranging from bad to excellent. The average of the viewer's scores, defined as the mean opinion score (MOS), is a measure of video quality. At least 20-25 non-expert viewers are required to give a reliable MOS.
Objective Quality Measurement

The most common measure of objective quality assessment of frames is Peak-Signal-to-Noise-Ratio (PSNR\(_f\)), defined as [46]:

\[
\text{PSNR}_f = 10 \log_{10} \frac{255^2}{D} \quad (2.1)
\]

where \( D \) is the mean squared error (MSE), defined as:

\[
D = d(x, \hat{x}) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left( x(i,j) - \hat{x}(i,j) \right)^2 \quad (2.2)
\]

where \( x(i,j) \) and \( \hat{x}(i,j) \) is a pixel value of \( i^{th} \) row and \( j^{th} \) column of the reference and processed/reconstructed frames respectively, and \( M \) and \( N \) is the dimension of the frame. In Eqn. (2.2), the peak signal with an 8-bit resolution is 255, and the noise is the square of the pixel-to-pixel differences (error) between the reference frame and the frame under study. Although PSNR of each color component \( (Y,U,V) \) can be calculated but for comparison purposes only Luminance component is sufficient for consideration. The PSNR of a video sequence is calculated by taking the average of individual frames PSNR over all \( L \) frames as given by

\[
\text{PSNR} = \frac{1}{L} \sum_{(L \text{ frames})} \text{PSNR}_f \quad (2.3)
\]
Chapter 2. Background: Video Coding and Communication

2.3 Video Compression

As stated earlier, a raw or uncompressed video generates huge amount of data, therefore, it would require prohibitively large amount of bandwidth for communication. Fortunately however, video signals consist of lot of temporal, spatial and statistical correlations. Temporal correlation in a video is due to similarity of the adjacent frames as changes are mainly due to camera or object motion, whereas spatial correlation is due to the similarity of neighbouring pixels within a frame. The statistical redundancy is related with probability distribution function of pixel intensities/colours in a frame. Efficient reduction of these correlations is utilized to achieve the video compression as shown in Fig. 2.5.

As adjacent video frames are highly correlated, independent coding of each frame generally generates a lot of redundant bits. Changes in adjacent video frames are typically due to the motion of objects. By performing motion estimation (ME) between current and reference (previously decoded) frames, mechanism for efficient exploitation of temporal correlation can be devised. A practical and widely-used method for ME involves dividing the current frame into square blocks (usually of size $16 \times 16$) and finding a best match of each block within a predetermined search region in the reference frame. The matching criterion may be any error measure such as mean square error (MSE) or sum of absolute difference (SAD). The best match of the current block in the reference frame is a block which results in minimum error. The displacement between the two blocks is the estimated motion vector (MV). Often, ME accuracy is improved by using fraction-pixel accuracy.

The MVs are used to reduce the temporal redundancies either by performing motion-compensated prediction (MCP) [53] or motion compensated temporal filtering (MCTF) [54]. In MCP, a prediction of the current frame is generated from the reference frame by using motion-compensated prediction. By taking the difference between the current frame and its MCP, a prediction error or residual is generated. If prediction is good, then residual will be having small energy compared to the original frame. On the other hand, MCTF is a temporal decomposition technique that performs wavelet decomposition along the motion trajectory. Using MCTF, a group of frame (GOF) is temporally decomposed into a set of
temporal frequency bands. The first-level decomposition results in a temporal low-pass and high-pass subbands. The temporal high-pass subband represents the residual signal with most of the coefficients having small values. Since, most of the energy is concentrated in the temporal low-pass subband, therefore it is recursively decomposed to the desired temporal level to produce a temporal low-pass subband capturing most of the energy of the GOF and number of temporal high-pass subbands representing the residual.

The residual frames obtained after temporal redundancy reduction has a lot of spatial redundancies as adjacent pixels are highly correlated. Often, this redundancy reduction is facilitated by 2-D signal transforms [54]. One of the beneficial properties of the transform is that they tend to compact most of the energy of the signal into a relatively small number of transform coefficients, with the majority of coefficients having little energy [55]. Efficient quantization of these low valued coefficients lead majority of them to the zero value, and with a small number of significant coefficients that provide a more compact representation of the residual frame. Over the years, a large number of signal transforms have been proposed. However, in video compression applications, only the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) are widely used [54]. DCT is a block-based transform which is applied after partitioning a frame into non-overlapping blocks of pixels and then processing each block individually. On the contrary, the DWT is applied on entire frame rather than to separate blocks of pixels.

The quantized symbols occurred with varying probabilities. It is a well known fact of information theory that symbols occurring with high probabilities have less information content compared to symbols having lower probabilities. This information is exploited to assign short codes to more probable symbols compared to less probable symbols, resulting in variable length code (VLC). A near-optimal method of designing VLCs is the Huffman coding. A more efficient method which can also be more easily designed to adapt to varying symbol statistics is the arithmetic coding [53].

2.4 DCT-based Standard Video Codecs

Almost all the standard video coders like MPEG-2, MPEG-4, H.261, and H.263 use a hybrid of motion compensated prediction (MCP) and DCT to exploit the temporal and spatial redundancies [46]. A generic hybrid video coder is shown in Fig. 2.6 which works as follows. Each frame is split into blocks of typically 16×16 pixels called macroblock (MB) and each MB is processed more or less independently. A MB may be coded in Inter- and/or Intra-mode. In Inter-mode,
a MB may use a unidirectional prediction (P-picture) or a bidirectional prediction (B-picture). In P-picture, for each MB in the current frame, a block matching ME process is used to find the best match in the previous frame. Only the luminance component is used in ME process. The estimated MV is used to generate a prediction of the current MB using MCP. In B-picture, block matching finds two best matching blocks, one in a previous frame and another in a future frame, and uses a weighted average of the two matches as the prediction for the current MB. In this case, two MVs are associated with each block. A prediction error or residual signal is generated by taking the difference between the current MB with its prediction. The residual is transformed using 8×8 DCT and coefficients are then scaled, entropy coded, and transmitted together with MV bits. Since MVs of adjacent MBs are usually similar, they are differentially coded. In the block matching process if no suitable match of the current MB is found, the block will be coded directly using transform coding. This is known as Intra-mode. In all the video coding standards, bit stream syntax follows hierarchical structure, in which a number of MBs form a group of blocks (GOB) or a slice, and several GOBs or slices form a picture. The size and shape of GOBs, slices, and picture sizes are different in various coding standards, which can often be tailored to the application’s needs.

A frame may be coded entirely in the intra-mode, and such a frame is called I-frame. The first frame of a sequence is always coded as an I-frame. A P-frame uses only a previous frame for prediction, and depending on the prediction
accuracy, an MB can be coded in either intra- or P-mode. Finally, a B-frame uses a bidirectional prediction, and an MB in a B-frame can be coded in intra-, P-, or B-mode. Further, a B-frame can only be coded after the surrounding I- or P-frames are already coded [56].

The video coding standard H.264/AVC also uses a similar architecture but with the following notable differences. Instead of 8x8 DCT, it uses a 4x4 integer transform. For more accurate MCP, quarter-pixel accuracy ME is used for luma and one eighth-pixel accuracy for chroma components. It also uses variable block size, bipredictive, multi-reference, and weighted MCP. In addition, it uses a very powerful intra prediction, which was not available in previous standards. It also uses an in-loop adaptive de-blocking filter to overcome the problem of blocking artifacts. When used all together, these new tools provide approximately a 50% bit rate savings for equivalent perceptual quality relative to the performance of prior standards [18].

2.5 DWT-based Non-standard Video Codecs

The outstanding performance of the wavelets for image compression has motivated researchers to extend them for video coding as well [57]. The straightforward approach is essentially an adaptation of the traditional hybrid architecture of Fig. 2.6 using DWT in Transform block [58, 59, 60, 61, 62]. A block-based ME with fractional-pixel accuracy is used to obtain the MVs. Overlapped block motion compensation (OBMC) [63] is then employed to remove temporal redundancy. Overlapping is necessary in order to remove the blocking artefacts in residual frames resulting from block matching. For block-based transform such as DCT, the blocking artefacts are not critical as long as the block boundaries of transformation and block matching are well aligned. However, wavelet-based coders that transform the entire residual frame will have to sacrifice much of their coding efficiency by coding the artificial high frequency information at the block boundaries of motion compensated frames.

A group of researchers have developed 3-D wavelet video coder as an alternative to hybrid video coder by extending wavelet transform in temporal direction. In contrast to the traditional hybrid video coding, 3-D video coding systems use 1-D temporal decomposition of a GOF using MCTF to exploit the temporal redundancy present in the video signal. The resulting temporal frames are then spatially decomposed using a 2-D wavelet transform followed by coefficients encoding as shown in Fig. 2.7 [54]. MCTF is a temporal decomposition technique that adaptively performs the wavelet decomposition and reconstruction along the motion trajectories. It is conveniently implemented using a lifting scheme. Theo-
Chapter 2. Background: Video Coding and Communication

2.5 Spatial and Temporal Analysis

Figure 2.7: Block diagram of a 3-D wavelet video codec

retically any filter could be used, however, research has shown that 5/3 filter has overall best performance for temporal decomposition [64]. For MCTF with 5/3 filter, the odd indexed frames are predicted from the adjacent even indexed frames to produce the highpass frames. The even indexed frames are then updated to generate lowpass frames using combination of the adjacent highpass frames. In both predict and update step, filtering is done along the motion direction. Temporal lowpass subband is recursively decomposed to give a lowpass subband and a number of highpass subbands. After temporal decomposition, each frame will undergo 2-D spatial wavelet decomposition, leading to 3-D spatio-temporal subbands as shown in Fig. 2.8. For coding, any 3-D wavelet coding algorithms such as 3D-SPIHT, JPEG2000 multi-component, or MC-EZBC is applied to generate the final bitstream [57]. MC-EZBC video coder [19] is currently considered to be the state-of-the-art in wavelet-based MCTF scalable coding. However, in this thesis WBTC [62, 65, 66] based video coder is considered, which will be described in section 2.7.

2.6 Scalable Video Coding

Video delivery to diversified clients through heterogeneous networks with time varying characteristics requires on-the-fly adaptation of the source bitstream. This is facilitated by scalable video coding (SVC), whereby the coded bitstream is organized such that it has number of truncation points. Decoding only the
Chapter 2. Background: Video Coding and Communication

Figure 2.8: 3-D wavelet decomposition of a GOF of sixteen frames (a) input GOF and (b) the resulting spatio-temporal subbands for 4-levels of spatial and temporal decomposition.

Partial streams will provide a range of scalability. The most common types of scalability are spatial, temporal, and quality (or SNR). In spatial scalability, complete decoding of the bitstream produces full resolution video, whereas partial decoding produces reduced spatial resolution. In temporal scalability, partial decoding leads to reduced decoded frame rates (temporal resolution), whereas in quality scalability, video quality (SNR) varies depending on how much of the bitstream is decoded [67].

In most of the standard video coders using hybrid structure, scalability is achieved through a layered structure consisting of a base layer and a number of (usually two to three) enhancement layers as shown in Fig. 2.9. The base layer is generally coded such that it has very low quality, resolution and frame rate. The enhancement layers encodes the residual signal and when combined with base layer decoding, it provides a higher spatial resolution for spatial scalability, a higher frame rate for temporal scalability and a progressive quality improvement in case of SNR scalability.

Layered quality scalability is accomplished by using increasingly finer quantization step sizes. The base layer is obtained by applying a coarse quantizer to DCT coefficients giving low decoded quality. For any of the enhancement layer, the DCT coefficients of the original input frame is subtracted from the reconstructed DCT coefficients of the previous layer and the residual is quan-
Chapter 2. Background: Video Coding and Communication

Figure 2.9: Principles of layered scalability

tized using smaller step size than that of the previous layer. This type of quality scalability provides only coarse granularity scalability (CGS), in that the quality improvements are obtained with rate increases in large steps. On the other hand, in fine granularity scalability (FGS) a bitstream can provide continuously improving video quality with every additional bit. In FGS, a video is coded into a base layer and a fine granular enhancement layer. The base layer is coded with non scalable technique, whereas the enhancement layer bitstream is generated by coding the difference between the original DCT coefficients and the reconstructed base layer coefficients using a bitplane coding technique [68]. FGS is used in MPEG-4 [69] and SVC [70].

To achieve spatial scalability, the input video is decomposed into a spatial pyramid. Each spatial layer is encoded independently while the motion and temporal prediction are derived from the reference pictures at the same layer. To remove the redundancy among layers, the interlayer prediction comes from only the reconstructed frame of the most recent layer. Temporal scalability involves partitioning of the group of pictures (GOP) into temporal layers. A simple way to achieve temporal scalability is to use the well known IBBP prediction structure. Up to three frame rates are supported by decoding I-pictures only, both I- and P-pictures, or all of the I-, P-, and B-pictures, respectively. In the H.264/AVC and SVC, more levels of temporal scalability are possible with hierarchical B-pictures, which also improve the coding efficiency [70].

One of the problems with layered scalability is that every additional enhancement layer reduces the coding efficiency by about 15-20%, thereby restricting the number of enhancement layers to two-three only [46]. Whereas, in 3-D wavelet video coders, embedded coding achieves quality scalability that results in bit-level granularity. Temporal scalability of multiple levels is achieved using a combination of lowpass and highpass temporal subbands as shown in Fig. 2.8. Resolution scalability is achieved by coding the 3-D coefficients in resolution pyramid [68].
2.7 WBTC-based Video Coder

An efficient low bitrate wavelet video coder has been proposed in [62]. The video coder uses the same hybrid architecture as shown in Fig. 2.6. The key element of the video coder is the use of wavelet block-tree coding (WBTC) algorithm [65, 66] as an efficient means to quantize and code the motion compensated residual frames. WBTC combines the features of both SPIHT [71] and SPECK [72] in a single algorithm. The motivation for using WBTC is that after wavelet decomposition of the residual frames, the majority of the coefficients may be grouped and then quantized to zero values giving a good compression performance.

In temporal prediction, a combination of motion estimation and overlapped block motion compensation (OBMC) [63] is used to generate motion competed residual (predicted) frame. Overlapping is necessary in order to remove the blocking artifacts in the residual frames resulting from block matching. Conventional block-based motion estimation with half-pixel accuracy is used to obtain one motion vector for each macroblock of size 16 x 16 pixels. The differential motion vectors are lossless coded with adaptive arithmetic coding. The residual frame is wavelet transformed and the resulting coefficients are quantized and coded with color WBTC [65].

After wavelet decomposition, the wavelet coefficients in each colour plane (such as YUV) are divided into blocks of coefficients. Interdependency across the subbands and among the three color planes is exploited through the use of a composite spatial orientation block-tree. The proposed algorithm generates embedded bitstream, whereby bits are hierarchically organized from high to low distortion reduction capability. Such an embedded bitstream is generated by quantizing the wavelet coefficient using bitplane-based coding. In bitplane-based coding, first the most significant bit of all coefficients is coded, followed by the next most significant bit, and so forth. The coefficients are grouped into sets and magnitude ordered from the highest bitplane. The ordering information is encoded with a set partitioning algorithm which is facilitated by the use of three lists: a list of insignificant block (LIB), a list of insignificant block sets (LIBS), and a list of significant pixels (LSP). At the initialization step, only the blocks in the lowest band (the highest pyramid level) of luminance plane are added to LIB, and those with descendents also are added to LIBS. The LSP starts as an empty list.

The coding process starts with the most significant bitplane and proceeds towards the lowest bitplane. For each bitplane there are two passes; sorting and refinement. In the sorting pass, the encoder goes through the lists LIB followed by LIBS for locating and coding the significant coefficients. For each entry in LIB,
one bit is used to describe its significance. If the block is not significant, it remains
in LIB and no more bits will be generated. Otherwise, the block is recursively
partitioned to locate and code the significant coefficients. Similarly, each set in
LIBS requires one bit for significance information. Insignificant sets remain in
LIBS while significant sets will be partitioned into subsets to locate and code the
significant coefficients. The significant coefficients so found will be moved to LSP.
In the refinement pass, each pixel in LSP, except those just added at the current
bitplane, is refined with one bit to increase its precision. Such a progressive
refinement generates fully embedded bitstream. The algorithm then repeats the
above procedure for the next bitplane until the bit budget is exhausted.

2.8 Channel Coding

Channel encoding and decoding as a forward error correction (FEC) is performed
to detect and correct of any error which may have occurred due to noise and inter­
ference in the channel. Error control is almost must in video communication over
wireless channels which have high bit error rate than wired channels. Generally,
channel coding is performed using either convolutional or block coding. A convo­
lutional code is generated by passing the information sequence to be transmitted
through a linear finite-state shift register. It encodes \( k \) bit message sequence into
\( n \) bit code word. The encoded bits not only depend on the current \( k \) input bits
but also on past input bits. The main decoding strategy for convolutional codes
is based on the widely used Viterbi algorithm. On the other hand, in an \((n, k)\)
block coding, for a message block of \( k \) symbols, \( n \) symbols codeword is generated
by adding \( n-k \) parity symbols, resulting into coding rate of \( k/n \) [4].

Reed-Solomon (RS) and low density parity check (LDPC) codes are the most
widely used block codes in video communication applications. RS code [73], a
special class of linear non-binary block codes, is very popular and mostly used
due to its high error correcting capability against channel errors and erasures. It
achieves ideal error protection against packet loss since it is a maximum distance
separable (MDS) code. Though, LDPC codes have low encoding/decoding com­
plexity and capability to operate over a large block size, these codes are not MDS
code and suffer with poor coding efficiency. Recent studies reports that LDPC
codes are more suitable for large block transmission over unidirectional channels,
whereas RS codes are more appropriate for small block size and real time video
streams [20] and this is the motivation for using RS codes in this thesis as FEC
for video communication over noisy channel. The following section describes the
fundamentals of RS code briefly [74, 75, 4, 76].
Chapter 2. Background: Video Coding and Communication

Reed-Solomon (RS) code

Reed-Solomon (RS) codes are linear block codes and subset of Bose, Chaudhuri, and Hocquenghem (BCH) codes [75]. A codeword structure of RS Code is shown in Fig. 2.10. The RS code is specified as RS(n, k) with m-bit symbols, where n is the total symbols in a codeword, k is the data symbols and 2t is the amount of parity symbols added to data symbols to form a codeword. Each symbol in a codeword comprises of m-bit. The relation between n and m is given as:

\[ n = 2^m - 1 \]  

(2.4)

And the total parity symbols, 2t, is given as:

\[ 2t = n - k \]  

(2.5)

A RS decoder can correct up to t symbol errors or up to 2t symbol erasures in a codeword. A symbol error occurs when atleast 1 bit in a symbol is wrong. An erasure occurs when the position of an error symbol is known. When a codeword is decoded, the original transmitted data symbols will always be recovered if

\[ 2s + r < 2t \]  

(2.6)

where s is the no of symbol errors and r is the no of symbol erasures. Otherwise original data symbol cannot be recovered and sometime it will mis-decode and recover an incorrect codeword.

The computational complexity of Reed-Solomon codes is related to the number of parity symbols per codeword. A large value of t means that a large number of errors can be corrected but requires more computational power than a small
value of $t$.

The channel code rate, $r_c$, of RS code is given as:

$$r_c = \frac{k}{n} \quad (2.7)$$

The RS decoded symbol-error probability, $P_E$, in terms of the RS input symbol-error probability, $p_s$, can be written as follows [74, 75]:

$$P_E \approx \frac{1}{2^m - 1} \sum_{j=t+1}^{2^m-1} \binom{2^m - 1}{j} p_s^j (1 - p_s)^{2^m-1-j} \quad (2.8)$$

Where $p_s$ can be expressed in term of bit-error probability, $p_b$, as

$$p_s = 1 - P(\text{all } m \text{ digits received correctly})$$
$$p_s = 1 - (1 - p_b)^m \quad (2.9)$$

### 2.9 Modulation

Modulation is one of the key functions performed at the physical layer in transporting information over wireless networks. The modulator is the interface device that maps the digital information into analog waveforms that match the characteristics of the channel. The mapping is generally performed by taking blocks of $k = \log_2 M$ binary digits at a time from the information sequence $\{a_n\}$ and selecting one of $M = 2^k$ deterministic, finite energy waveforms $\{s_m(t), m = 1, 2, \cdots, M\}$ for transmission over the channel. The waveforms may differ in either amplitude (Amplitude Modulation) or in phase (Phase Modulation) or in frequency (Frequency Modulation), or some combination of two or more signal parameters (Quadrature Amplitude Modulation). In this section only Quadrature Amplitude Modulation (QAM) is described briefly. For further details and reading please refer to [4].

#### 2.9.1 Quadrature Amplitude Modulation (QAM)

Almost in all current wireless communication standards like WiMAX, Long Term Evolution (LTE), Digital Video Broadcast-Terrestrial (DVB-T), QAM is used at physical layer to map the bits into signal waveforms. QAM is obtained by simultaneously impressing two separate $k$-bit symbols from the information sequence $\{a_n\}$ on two quadrature carriers $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ The signal waveform
for QAM, $s_m(t)$, may be expressed as [4]:

$$s_m(t) = A_{mi}g(t)\cos(2\pi f_c t) - A_{mq}g(t)\sin(2\pi f_c t) \quad (2.10)$$

where $A_{mi}$ and $A_{mq}$ are the information-bearing signal amplitudes of the quadrature carriers and $g(t)$ is the signal pulse. For rectangular signal space constellation with $M = 2^k$, the signal amplitudes, $A_{mi}$ and $A_{mq}$, take the discrete values (levels) from

$$(2m - 1 - \sqrt{M})\frac{d}{2}, \quad m = 1, 2, \ldots, \sqrt{M} \quad (2.11)$$

where $d$ is the distance between adjacent signal amplitudes.

Alternatively, the QAM signal waveforms may be expressed as

$$s_m(t) = V_mg(t)\cos(2\pi f_c t + \theta_m) \quad (2.12)$$

where $V_m = \sqrt{A_{mi}^2 + A_{mq}^2}$ and $\theta_m = \tan^{-1}(A_{mq}/A_{mi})$. From this expression, it is apparent that QAM signal waveforms may be viewed as combined amplitude and phase modulation.

The QAM signal waveforms may be represented as a linear combination of two orthonormal signal waveforms, $f_1(t)$ and $f_2(t)$, as:

$$s_m(t) = s_{m1}f_1(t) + s_{m2}f_2(t) \quad (2.13)$$

where

$$f_1(t) = \sqrt{\frac{2}{E_g}}g(t)\cos(2\pi f_c t) \quad (2.14)$$

and

$$f_2(t) = -\sqrt{\frac{2}{E_g}}g(t)\sin(2\pi f_c t) \quad (2.15)$$

and

$$s_m = \begin{bmatrix} s_{m1} & s_{m2} \end{bmatrix} = \begin{bmatrix} A_{mi}\sqrt{\frac{1}{2}E_g} & A_{mq}\sqrt{\frac{1}{2}E_g} \end{bmatrix} \quad (2.16)$$

where $E_g$ is the energy of the signal pulse $g(t)$. 

The Euclidean distance between any pair of signal vector is
\[
d^{(c)}_{mn} = \|s_m - s_n\| = \sqrt{\frac{1}{2}E_s[(A_{m1} - A_{n1})^2 + (A_{m2} - A_{n2})^2]}
\]  
(2.17)

Figure 2.11: Signal space constellation diagram for \(M = 4\) and \(M = 16\) QAM

The signal space constellation diagram for \(M = 4\) and \(M = 16\) QAM is shown in Fig. 2.11. The \(d_1\) and \(d_2\) are the minimum intra-quadrant and inter-quadrant Euclidean distances. In case of rectangular QAM these distances are \(d_1 = d_2 = d\).

If the distance \(d_1\) is greater than distance \(d_2\), it will result into the signal constellation diagram of Hierarchical QAM, which is used as the basis of providing unequal error protection and will be described in chapter 3.

### 2.9.2 Bit Error Rate of QAM

Probability of a symbol error or bit error rate (BER), \(P_M\), for the \(M\)-ary QAM with rectangular constellations over AWGN channel with power spectral density of \(N_0/2\) [4] is given by

\[
P_M = 1 - \left[1 - 2 \left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3E_s}{(M - 1)N_0}}\right)\right]^2
\]  
(2.18)

where \(E_s\) is the average energy of the QAM symbol.

The bit error rate performance of QAM with \(M = 4\), 16 and 64 over wide range
Average signal-to-noise ratio, $E/N_0$ (dB)

Figure 2.12: Bit error rate performance of QAM

of average signal-to-noise ratio is shown in Fig. 2.12. It is observed from the figure that for fixed value of $M$, the BER decreases as SNR increases. This is due to the fact that as average power increases, for same number of constellation points, the distance between these points will increases and may be able to withstand more noise, which will result in reduced BER. It is also observed from the figure that as the value of $M$ increases, the BER also increases at fixed SNR. This is due to the fact that for fixed average power, as $M$ is increased the signal constellation points come closer to each other, and for same noisy conditions, the demodulator may detect the symbols more erroneously. Thus BER performance will drop as $M$ increases.

2.10 Communication Channels

Channel is a medium through which the information signal is conveyed from transmitter to the receiver. The channel may be wired such as wire, coaxial cable, a waveguide, an optical fiber or it can be a wireless like a radio link. In current scenarios, the backbone channels are almost wired while end users are connected with wireless links. These wireless link have high bit error rate (BER) as compared to wired one. Upon this high BER, the wireless link faces varying
channel conditions, due to mobility in the surrounding or the relative movement between transmitter and receivers, which leads to fading of the signal. Therefore, transmitting compressed video without or weak error resilient technique over these types of channel leads to vary poor end-to-end quality. Therefore, simulation of these types of channel is necessary to see the behaviour of any proposed error resilient technique for a encoded video bitstream over these erroneous channels.

Most widely used channel model is Additive White Gaussian Noise (AWGN) in the communication systems. The reason behind using the AWGN comes from the central limit theorem [4], which states that the sum of a large number of statistically independent and identically distributed random variable (that is, noise) with finite mean and variance approaches a Gaussian probability density function (pdf). In wireless channels, a large number of signals from the other sources interfere the transmitted signals. As these interferences are usually independent to each other, the sum of these interferences can be model as Gaussian noise. Further, in most of the communication these are added in the transmitted signal, that's why the whole channel model can be simulated using AWGN. The Additive white Gaussian Noise (AWGN) channel is described briefly as follows.

Additive White Gaussian Noise Channel (AWGN)

The AWGN channel model is shown in the Fig. 2.13. In this channel model, the transmitted signal \( s(t) \) is corrupted by an additive random noise process \( n(t) \) (usually in communication these are the sum of all interferences arriving from other sources). Generally in a broad class of physical communication channel this random noise is taken to be Gaussian process. The power spectral density of this process is taken to be flat (i.e. white) and equals to \( N_0/2 \) W/Hz, which means that the samples of this noise at different time instances are uncorrelated. Hence, the resulting mathematical model for the channel is usually called the additive white Gaussian noise (AWGN) channel. The received signal, \( r(t) \), at the
Chapter 2. Background: Video Coding and Communication

Figure 2.14: Probability density function of Gaussian distribution for various parameter $\mu$ and $\sigma^2$

output of this channel is

$$r(t) = s(t) + n(t) \quad (2.19)$$

The probability density function (pdf), $f_{\mu,\sigma^2}(x)$, of Gaussian distribution is given as:

$$f_{\mu,\sigma^2}(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x - \mu)^2}{2\sigma^2}} \quad (2.20)$$

where $\mu$ is the mean and $\sigma^2 = N_0/2$ is the variance of the distribution. The pdf of guassian distribution is shown in Fig. 2.14 for various value of parameter $\mu$ and $\sigma^2$.

### 2.11 Error Resilient Techniques: A Review

As discussed in previous chapter, that all over the world, there is a huge demand of video based applications over wireless and IP networks. These applications are now frequently accessed on portable devices like cell-phone, smart-phone, tablets, etc. But, transmission of compressed video bitstream over these channels
Chapter 2. Background: Video Coding and Communication

is a very challenging task. This is due to the fact that these channels are error prone and dynamically varying in nature either because of fading and interference or high congestion at the routers. Usually, fading results in loss of packets at the receiver, whereas, router drops the packets to control the congestion. In addition, the compressed video generates highly dependent bitstream, and even single bit error may result catastrophic failure in reconstruction of the received video. Furthermore, the portable devices on which the video services are accessed, have limited computational power, which in turn requires low complexity coders. Above it, video coding and decoding process is also constraint to delay. That is, in playback of video sequences, delay between two frames should be at most $1/10^{th}$ of second. Therefore, an error resilient technique with low computational complexity is required to combat the effect of errors to the coded bitstream.

Over the years, a variety of solutions have been proposed to cope with these challenges [2, 77, 3, 78]. These solutions can be classified into two broad categories:

1) Feedback-based error resilient techniques
2) Redundancy-based error resilient techniques

Feedback-based solution uses feedback channel to get the information of the distortion occurred in the reconstructed video. Accordingly, the video encoder adjusts it encoding strategy such that the error propagation stops in the reconstructed video at the receiver. Whereas, redundancy-based solution uses some sort of redundancies, added at the transmitter side, either at single layer or multiple layers (e.g. application layer source coding or transport layer FEC, etc.), to detect and correct the channel distortions at the receiver. Further, some redundancy based error resilient solutions uses feedback channel to sense the channel condition for adapting the added redundancies in the transmission. Note that the difference in using the feedback channel in both the techniques is totally different. In former technique, video decoding information is transmitted back from receiver, while in later, the feedback channel does not provide the information regarding decoded video but only about the channel condition occurring at the receiver. These two types are discussed in detail in following subsections:

### 2.11.1 Feedback-based Error Resilient Techniques

Error resilience of video transmission can be improved by using the feedback channel, if available [2, 77, 78]. The decoder at the receiver, transmit back the error pattern of the reconstructed video sequence to the video coder using feedback channel. That is, the feedback channel gives the information to video coder that which part of the video sequence has been recovered or corrupted at the
receiver, using ACK (acknowledgement) or NACK (negative acknowledgement), respectively. On the basis of this feedback, video coder analyses the error pattern and accordingly, adjust its coding or transmission strategy such that the quality of the reconstructed video recovered as early as possible. For example, one way of achieving the resiliency of video transmission is to use source adaptive coding based on feedback, and other may be to use retransmission strategies, if permitted. Thus feedback-based error resilient techniques can further be categorized into two major groups, which are as follows

**Adaptive Source Channel Coding**

In this type of techniques, based on the information of reconstructed video sequence received on the feedback channel, the video coder adjusts its coding modes such that the error propagation in the reconstructed video sequence be eliminated or reduced as early as possible. This is based on the fact that if errors in the video persist only for few frames, the human eye can hardly perceive the effect. For example, in one technique [2], when errors are detected in decoder, it sent back the information on NACK packet to the video coder. On the other side, the video encoder, after receiving the NACK packet, encodes the next frame (to be transmitted) in intra-mode. This intra-mode frame (I-frame) when decoded at the receiver will stops the error propagation. This approach will reduce the compression efficiency but stops the error propagation.

In error tracking approach proposed in [79, 80], decoder feedbacks the identity message of damaged blocks on NACK packet to video coder, if errors are detected in the reconstructed frame. Meantime, the decoder continues its normal decoding using error concealment method on the erroneous blocks. At the encoder, using the information received of the corrupted frame, its erroneous blocks, and the motion vectors, stored in buffers, the affected blocks are estimated in the current frame which is going to be encoded. These affected blocks are then may be encoded in the intra mode. This method efficiently stops the error propagation in the reconstructed video sequence. In another method of error tracking approach [80], the same above procedure is adopted except that the encoder uses the error concealment procedure as that performed by the decoder to the erroneous blocks of the corrupted frame. Then local decoding is redone from the corrupted frame up to the frame going to be encoded. This will enable the encoders prediction frame buffer matches that at the decoder and thus further frame will be reconstructed almost without errors.

Reference picture selection method [81, 82] is used where multiple prediction frames are considered in video coding and decoding. In this method, as in error tracking, the decoder uses the feedback channel to inform the video coder about
which frames are correctly received and which are damaged. Accordingly, the encoder uses the undamaged frame to predict the subsequent frames. Information of the selected frame for prediction is also transmitted along the encoded frame so that the same frame can be used for prediction at the decoder.

Retransmission

For general data communication, retransmission of corrupted data has been used effectively using automatic repeat request (ARQ) schemes [83] while video communications are considered as highly sensitive to delay, and retransmissions are not considered in most of the video applications. However, retransmission of erroneous blocks can be considered in video services where the delay incurred is small [84, 2]. More specifically, the real time video communication can tolerate an end-to-end delay of less than 200 ms [17]. The end-to-end quality of the video can be improved by controlling the retransmission of corrupted blocks appropriately. For example [85] instead of using indefinite retransmission trials to recover the damaged block, the retransmission trials can be restricted by determining the allowed delay.

In usual method of retransmission of damaged blocks, the decoder had to wait for the arrival of requested retransmission before decoding the subsequent received frames. This will result in freezing of the displayed video temporarily. Another approach of retransmission in [86, 87] solved the above problem effectively. In this scheme, after sending the retransmission request to video encoder, the decoder uses some error-concealment method to the damaged blocks and continues its normal decoding. Also, at the same time, the trace of affected blocks in subsequent decoded frames and their associated coding information is recorded. When retransmission blocks arrives, decoder uses these retransmitted blocks and motion vector information, stored in the decoder buffer, to reproduced the affected blocks in subsequent frames (which is to be displayed next) as if these blocks are received undamaged. Thus error propagation stops here without freezing the video display, but this is quite computationally complex method which is not very much suitable for portable devices having low processing power and small battery backup.

2.11.2 Redundancy-based Error Resilient Techniques

These techniques are used where feedback channel is unavailable like broadcasting, multicasting, etc. or feedback channel is only used to inform channel condition. In these techniques some sort of redundancy is added in network layers. For example, small amount of redundancy is kept purposely in source coding (in
application layer), which can be exploited to detect and correct the damaged video blocks. Similarly redundant information can be added in transport, MAC and Physical layers. In addition, instead of applying at single layer, cross-layer redundancies can be added to multiple network layers. In the following, some of these approaches are briefly reviewed.

Robust Source Coding

The video communication can be made robust by keeping redundancy in the source encoding of the video. This redundant information can be utilized at the decoder to confine the channel error to very small region and control its propagation to other block/frames. Sometime it can provide better error concealment at the decoder. In one approach [2], the auxiliary information is added to counter the effect of transmission errors. For example, motion vectors for intra-coded macroblocks may also be transmitted which would usually not required at the decoder to decode the intra block. The corrupted intra coded macroblock can be recovered with the help of surrounded error free macroblock and motion vectors.

In another approach [2], to minimize the effect of error propagation due to prediction, the prediction can be restricted within non-overlapping spatial and temporal regions. To reduce spatial prediction error, whole frame is divided into number of slices with each slice include group of macroblock. So prediction from inter-slice macroblock is avoided. For reducing temporal error propagation group of frames called thread is formed. Each thread is coded without using prediction from other threads.

Redundancies can also be added in entropy coding. One of the techniques is self-synchronized entropy coding [88, 89, 90, 91, 92], in which a code word called synchronization code is added in the entropy encoded streams. This codeword has the capability to resynchronize the entropy decoder if erroneous bits have corrupted the data. Thus error propagation is kept confined between two synchronization code words. To avoid high bit rate overhead, shorter synchronization code words are used but it increases the probability that a bit error may make a false synchronization code words. On the other hand long codes increase bit overhead but are strong in synchronization and robustness.

In error resilient entropy coding (EREC) [93] scheme, equal size of slots are formed which contains the variable length bitstreams from individual blocks. Bits from each image block, is placed into the designated slot. If the slot becomes full, the remaining bits from the image block are placed in another slot, which is searched using predefined offset sequence. With EREC, synchronization can be obtained at the start of the each block in the decoder operation. The redundancy introduced by using EREC is negligible.
Reversible variable-length code (RVLC) is also employed as error resilient entropy coding [46]. In RVLC the coded bitstream can be decoded backward from any synchronization code word. If error occurred in the bitstream, the subsequent coded bitstream can be recovered by decoding backward from the next synchronization code word detected. The RVLC codes give improved robustness to errors but at the cost of reduced coding efficiency.

Multiple Descriptive Coding

Multiple description coding (MDC) [94, 95, 96, 97] is another approach to provide error resilient transmission of video sequences. This technique exploits the availability of several parallel paths between the transmitter and the receiver, and the channel conditions of each path are independent to each other such that the probability that all channels are simultaneously down (i.e. suffering from long error burst) is very small. These path may be physically or virtually (e.g. using frequency division) different from each other. For example, this scheme can be used over wireless multihop networks.

In MDC, several coded bit stream, called as descriptions, of the same source signals are generated and transmitted over different paths. The video coder and decoder are designed such that, if any one description is correctly received, the quality of the reconstructed video is acceptable, and that when more descriptions are correctly received, the quality improves in incremental steps. As each description carries sufficient information about the video, this will reduce coding efficiency which usually happens in redundancy based error resilient techniques. However, MDS methods are highly effective in very poor channel conditions [2].

Various MDC schemes have been proposed over the years for providing the error resiliency to coded bitstreams. These includes multiple-description scalar quantization [98, 99, 100, 101], rate-distortion optimal selection of temporal splitting, spatial splitting and repetition modes [102] using pairwise correlating transforms [103, 104], matching pursuits for generating multiple descriptions [105, 106], using different prediction paths while performing motion compensation in individual descriptions [107], use of B frames for making multiple description, etc.

Forward Error Correction

Forward Error Correction (FEC) is a very powerful and commonly used technique for error detection as well as error correction for video communication. FEC can be applied using either the block or convolutional coding. Usually, FEC codes are capable of correcting both random errors and erasures in a block of symbols. Reed-Solomon (RS) and low density parity check (LDPC) codes are the most
widely used block codes in video communication applications. RS code [73], a special class of linear non-binary block codes, is very popular and mostly used due to its high error correcting capability against channel errors and erasures. It achieves ideal error protection against packet loss since it is a maximum distance separable (MDS) code. RS codes suffer from block size limitation and computational complexity. In recent years, LDPC codes have attracted the researchers due to its low encoding/decoding complexity and capability to operate over a large block size. However, LDPC codes are not MDS code and suffer with poor coding efficiency. Recent studies have pointed out that LDPC codes are more suitable for large block transmission over unidirectional channels, while RS codes are more appropriate for small block size and real time video streams [20].

FEC has been applied at application layer, transport layer, link layer, and physical layer. For slow fading channel, a deep fade will cause burst errors of long duration. To improve the performance under such conditions, interleaving is often used. Interleaving randomizes the bit errors which can be then corrected by FEC. However, use of interleaving introduces delay and therefore, care must be taken while designing an interleaving scheme. Various types of FEC coding has been studied and applied for video communications [20, 21, 22, 23, 24, 25, 26]. Moreover, recently some new codes such as raptor codes [108] have emerged as a powerful alternative to classical channel codes.

**Joint Source-Channel Coding**

In Joint Source-Channel Coding (JSCC) [27] method of providing error resiliency to video bitstream, an interaction between source and channel takes place at lower level. This method is contradicted to Shannons separation principle [109], which state that the design of a source coding and channel coding scheme could be optimized separately, and when cascaded in a communication system could also achieves optimal performance. However, it assumes two assumptions, namely a) the use of infinite block length for both source and channel codes and b) the availability of arbitrarily high computational resources. But, both of these assumptions are not valid in practice for the transmission of images and videos because of practical limits on delay and computational resources. This motivates a design of joint source-channel coding. The JSCC scheme adapts the source coding parameters (e.g. quantizer step size, entropy-coder design) and channel coding or modulation parameters according to the channel condition such that the channel distortions are minimized. In [110], optimal quantizers are designed to minimize the total distortion due to quantization and transmission errors for the given input data probability distribution and the channel error matrix. In [111], in addition to above approach, code-word assignment to match the channel
Chapter 2. Background: Video Coding and Communication

error characteristics is also proposed. Similar schemes are studied in [112, 113]. The JSCC approach to image signals is firstly proposed in [114] using Differential Pulse Coded Modulation (DPCM) and convolution codes. This scheme is further extended for Discrete Cosine Transform (DCT) based image coders in [115] [116]. The JSCC for video transmission has also been widely studied [28, 117, 29, 118, 119].

Layered Coding and Unequal Error Protection

The most popular and effective scheme for providing error resiliency to a video transmission system has been layered coding combined with prioritized error protection [35, 34]. In layered coding, video is partitioned into a base layer and one or more enhancement layers on the basis of non-uniform importance of the coded video in reconstruction. The base layer contains the basic and significant information of the video, and alone, it can yield acceptable quality on decoding. Whereas, the enhancement layers contain details of the video, which on decoding, further improves the quality of the reconstructed video. As more enhancement layers are received higher quality can be obtained. Thus the video data is partitioned into different priority layers, in which base layer has high priority while enhancement layers have low priorities. In the literature, layered coding is some time also referred as scalable coding. Several different types of layered coding can be implemented based upon how the video data is partitioned. Four types of layered coding or scalability are most popular, namely, a) Temporal Scalability b) Spatial scalability, c) Signal-to-Noise ratio (SNR) scalability, and d) Data partitioning [2, 120].

To achieve robust video transmission, the layered coding is combined with some transport prioritization in such a manner that the base layer gets high protection against channel errors while enhancement layers gets relatively lower error protection [36, 37]. The graceful degradation of video is achieved with increased channel error probability, as enhancement layer can be dropped in decoding while base layer can withstand this increased channel distortion and thus decoded to achieve acceptable quality. This type of error protection of layered coding is usually designated as unequal error protection (UEP) or unequal loss protection (ULP) or non-uniform error protection and has been widely reported [35, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130].

UEP at Individual Network Layers The UEP for layered coded video has been achieved at different network protocol layers, namely, physical, data link, network and application layers using network prioritization, forward error correction [131, 114, 132, 133, 134], modulation [135, 136, 30], power allocation
Chapter 2. Background: Video Coding and Communication

In network prioritization based UEP, the transmitted packets are assigned certain bits in its header to inform about its priority. Based upon priority information, network node discards the low priority packets to control the traffic congestion. Thus instead of catastrophic failure of video, an acceptable quality of it is achieved.

At application layer, UEP for layered video coded bitstream can also be achieved based on channel codes such as rate-compatible punctured convolutional code (RCPC), Turbo codes, Low Density Parity codes (LDPC), BCH code, Reed Solomon (RS), etc. [138, 139, 140, 141, 142, 143, 144]. In this type of UEP, the base layer is protected with a strong channel code and enhancement layers are protected relatively with weak channel codes. Under poor channel condition, the enhancement layer may be damaged, whereas the base layer can be recovered successfully to reconstruct the video with certain acceptable quality. Furthermore, JSCC has also been used to provide adaptive UEP for varying channel conditions [145] [146].

At physical layer, hierarchical modulation can also be used to provide UEP [147] [148] [149] [39] [150] [151] [152] [153]. This type of UEP uses non-uniform signal space constellations of hierarchical modulation to give different degrees of error protection. More specifically, in hierarchical modulation, certain blocks of constellation points have larger Euclidean distance between them than other block of points. The message points are mapped over the constellation diagram such that the specific bits assigned to the widely separated points have lower error probabilities than the others. This feature could be exploited to achieve UEP to layered coded video. Similarly, UEP has been also achieved using different power level (unequal power control) to transmit the coded bits. The advantage of this type of UEP is that it is achieved without any additional bandwidth requirement, which is usually required in FEC based UEP.

**UEP using Cross-layer Approach** Usually, most of the error resilient schemes discussed above are employed at various network layers such as application, transport, link and physical layers, separately and independently. Although, the decoupled layer protocol enables interoperability and reduces network design complexity with hidden implementation details and internal parameters to remainder layers, but the independently optimized protocol paradigm is not very much suitable for wireless networks. The limited performance of overall architecture for wireless network is due to the lack of coordination among layers to tackle the user mobility, limited bandwidth and time-varying nature of the channels. Therefore, a cross-layer design has been proposed to improve the overall system performance.
Chapter 2. Background: Video Coding and Communication

The idea behind the cross-layer design is to maintain the functionality associated to the original layers, while allowing coordination, interaction and joint optimization of protocols over multiple layers.

Cross-layer strategies in communication was firstly studied using adaptive modulation and coding (AMC) [158]. After its success, the cross-layer optimizations have been pursued from the perspective of cross-layer networking architecture [159], video distortion driven routing [160], scheduling [161], link adaptation [162], energy efficiency [163], video analysis and delivery [164, 165], end-to-end quality-of-service (QoS) requirements [166, 167] and so on.

For implementing error control strategies, a number of researchers have used application layer FEC and Media Access Control (MAC) layer ARQ to provide cross-layer error protection for wireless LANs [168, 169, 155]. The work in [170] combines cross-layer error protection techniques composed of FEC in the MAC and application layers, and ARQ across the MAC and application layer. However, for large networks such as cellular mobile, the delay introduced by MAC layer ARQ may not be suitable for delay sensitive video applications. Also, for applications such as multicast, there is no provision of feedback channel to support ARQ protocol. Therefore, for such scenarios, cross-layer error protection employing different combinations of application layer FEC and adaptive modulation is a viable alternative [39, 40, 41]. For example, in [39], a combination of turbo code and hierarchical QAM is used to provide unequal error protection (UEP) for two layer scalable H.264 bitstream. Similarly, a combination of rate compatible punctured convolution code (RCPC) and non-uniform phase shift keying (PSK) modulation is suggested in [40] and [41] to achieve UEP in H.263+ layered video coded bitstream.

2.12 Summary

This chapter begins with some background material which are related to the thesis work. It provided the fundamentals of video coding and review of various video coding standards. The wavelet-block tree coder (WBTC) has been discussed in details, which is used in this thesis for research study. Then, overview of different channel coding, modulation schemes and communication channel models were given. Finally, a detailed literature survey of current state-of-the art error resilient techniques were presented. In the next chapter, unequal error protection (UEP) at physical layer will be investigated.