A Distributed Video Coding Framework for Higher Resolution Sequences

Asif Mahmood

PhD Thesis

School of Computing and Communications
Faculty of Science, Technology, Engineering and Mathematics
The Open University
Milton Keynes, UK

January 2020
Acknowledgements

All praise is due to Allah, Lord of the universe who created me and brought me where I am today.

I would like to express my sincerest and deepest gratitude to my supervisors Prof Laurence S. Dooley and Dr Patrick Wong for their excellent guidance and support throughout the journey of my PhD. Their suggestions, comments and criticisms have been invaluable. I am also grateful to them for patiently providing critical feedbacks despite my usual tardiness for completing drafts.

I would like to extend my thanks to my third-party monitor Dr Soraya Kouadri Mostefaaoui and Dr David Chapman for helping me through personal and administrative difficulties. I am also thankful to the Open University for funding my PhD and its staffs for their timely assistance on various occasions.

I want to thank all current and past members of the XGMT research group. Thanks to all my friends and colleagues who motivated me to complete the journey over the years. Special thanks to Shabuz, Riasat, Priota and Shagar for their support and care during the submission process.

I am indebted to my parents and other family members for their constant moral support and encouragements. I am particularly appreciative of my brother Faisal and his family, especially my lovely nieces Afrah and Atifah for making this arduous journey comfortable and vibrant.
Abstract

Video coding is the core enabling technology for compressing sequences where inherently very large bandwidth requirements must be reduced to enable efficient transmission and storage. Traditional video codecs (encoder-decoder) generally incur large encoder resources and so are not suitable for resource-scarce devices. Distributed video coding (DVC) has become an attractive alternative for such devices because of its simple encoder structure, which is achieved by shifting the major processing load to the decoder. Despite the established theory, current DVC architectures do not exhibit the same rate-distortion (RD) performance as traditional codecs, a performance gap which is compounded when coding higher resolution sequences. Available DVC codec rate-control options are also inflexible and difficult to guarantee a quality-of-service (QoS) level, while the aggregated encoder-decoder latency at higher spatial resolutions can significantly hinder practical DVC implementations.

This thesis presents a new DVC for Higher Resolution (DVC-HR) framework that addresses the aforementioned limitations arising in encoding higher spatial resolution sequences. It introduces a novel flexible content-aware quantisation (CAQ) mechanism which supports dynamic and robust rate-control to produce maximum output quality by lowering perceptible distortion as well as more efficiently utilising available bandwidth. Crucially, CAQ supports larger transform block sizes which are essential for effective higher resolution DVC. Finally, the latency issue is assuaged with the development of a new dynamic channel-coding algorithm to reduce decoding times thereby affording an important advance for practical DVC realisations.

Rigorous analysis of the DVC-HR framework confirms that it consistently outperforms the community-accepted benchmark DVC codec (DISCOVER) in RD performance, rate-control flexibility, bandwidth utilisation and lower latency at higher spatial resolutions. These contributions represent a notable step towards narrowing the DVC performance gap by enabling more practical, state-of-the-art implementations to better rival traditional codecs.
Declaration

The work presented in this thesis is an original contribution of the author. Parts of the thesis have appeared in the following peer-reviewed publications:


Table of Contents

Acknowledgements.................................................................................................................. II

Abstract.................................................................................................................................. III

Declaration................................................................................................................................. IV

Table of Contents ......................................................................................................................... V

List of Figures ............................................................................................................................... X

List of Tables ............................................................................................................................... XIV

Glossary ....................................................................................................................................... XVI

1 Introduction ............................................................................................................................... 19

1.1 Background ......................................................................................................................... 19

1.2 Video codecs ....................................................................................................................... 20

1.3 Low complexity encoding ................................................................................................. 22

1.4 Research question and objectives ...................................................................................... 28

1.5 Contributions ...................................................................................................................... 30

1.6 Thesis structure .................................................................................................................... 32

1.7 Summary ............................................................................................................................ 34

2 Survey of DVC literature ......................................................................................................... 35

2.1 Introduction ......................................................................................................................... 35

2.2 Theoretical foundation of DVC ......................................................................................... 35

2.2.1 Distributed source coding theories .................................................................................. 35
2.2.2 DSC realisation ..........................................................38
2.3 DVC architectures ..........................................................39
  2.3.1 The Berkeley architecture ...........................................40
  2.3.2 The Stanford architecture ...........................................42
2.4 The DISCOVER codec .......................................................45
  2.4.1 The encoder ..........................................................45
  2.4.2 The decoder ..........................................................47
  2.4.3 Evaluation ............................................................49
2.5 Innovations in DVC ..........................................................51
  2.5.1 SI generation and refinement ........................................51
  2.5.2 Transform and quantisation ..........................................53
  2.5.3 Correlation noise modelling ........................................58
  2.5.4 Channel coder and feedback channel ............................59
2.6 Gap identification and challenges .......................................62
3 Research Methodology .........................................................65
  3.1 Introduction ..................................................................65
  3.2 Development methods ....................................................65
  3.3 Overview of the framework development process .................67
  3.4 Implementation of the testbed ..........................................69
    3.4.1 Programming language and platform ..........................69
    3.4.2 Implementation of DVC modules ...............................72
3.4.3 Software testing and validation ........................................75

3.5 Simulation criteria ................................................................77
  3.5.1 Video format ..................................................................77
  3.5.2 Configuration parameters .............................................77

3.6 Performance benchmark .....................................................79

3.7 Key performance metrics ....................................................80

3.8 Test sequences ..................................................................83

4 Content-Aware Quantisation of Transform Coefficients ..............85
  4.1 Introduction ......................................................................85
  4.2 Limitations and design challenges .....................................86
  4.3 The CAQ mechanism .......................................................89
    4.3.1 CAQ algorithm ..........................................................91
    4.3.2 Enhancement-I: Exploiting human visual system in CAQ ....97
    4.3.3 Enhancement-II: Maximising the QM .........................99
  4.4 Critical evaluation of CAQ performance ...............................104
    4.4.1 Rate-control flexibility ..............................................104
    4.4.2 Bandwidth utilisation .................................................106
    4.4.3 Adaptation to scene changes ......................................109
    4.4.4 Qualitative evaluation ..............................................114
  4.5 Summary ........................................................................118

5 Investigation of larger transform block sizes for higher resolution DVC ......120
5.1 Introduction ......................................................................................... 120
5.2 Modifications of the DVC-HR framework ........................................ 121
  5.2.1 QM modification ............................................................................ 122
  5.2.2 Modification of the LDPCA coder ................................................ 124
5.3 Results discussion ............................................................................... 126
  5.3.1 4CIF sequences ............................................................................ 128
  5.3.2 CIF sequences ............................................................................... 131
  5.3.3 QCIF sequences ............................................................................ 133
  5.3.4 Performance evaluation across spatial resolutions ....................... 137
  5.3.5 Performance evaluation across scene characteristics ................. 137
  5.3.6 Qualitative performance evaluation ............................................. 138
  5.3.7 Transmission Overheads .............................................................. 142
  5.3.8 Parallel processing advantages .................................................... 142
  5.3.9 Latency considerations ................................................................. 143
5.4 Summary ............................................................................................ 144
6 Advanced Decoding Method for the Channel Coder ............................ 145
  6.1 Introduction ....................................................................................... 145
  6.2 Background ....................................................................................... 146
  6.3 Advanced BP algorithms ................................................................... 148
    6.3.1 The classic BP algorithm ............................................................ 148
    6.3.2 Sequential BP algorithms ........................................................... 151
6.3.3 Informed Dynamic Scheduling strategy for sequential BP ......152
6.3.4 The NWRBP algorithm .................................................154
6.3.5 The mNWRBP algorithm ...............................................156
6.3.6 mNWRBP algorithm complexity analysis..........................160
6.4 Simulation criteria ..........................................................163
6.5 Critical evaluation of mNWRBP performance .........................164
  6.5.1 Reduction of total iterations ........................................165
  6.5.2 RD performance ......................................................168
  6.5.3 Effect on parallel processing scenario ............................171
6.6 Summary ........................................................................172
7 Future Work .....................................................................174
8 Conclusion ........................................................................180
References ..........................................................................184
Appendices ..........................................................................192
A. Supplementary Results for the mNWRBP RD Performance ..........192
List of Figures

Figure 1.1: Block diagram of a DVC architecture: (a) separate encoder-joint decoder setup, (b) coding tools used in WZ encoder and decoder .................................................................24

Figure 1.2: Block diagram of the new DVC-HR framework with the novel contributions highlighted ..................................................................................................................31

Figure 2.1: Block diagram of (a) traditional SC and (b) DSC setups for coding of two correlated sources ..........................................................................................................................37

Figure 2.2: Block diagram of (a) traditional SC and (b) DSC setups for coding with SI ..................................................................................................................................................37

Figure 2.3: WZ coder conceptual block diagram comprising of a quantiser and a SW coder .................................................................................................................................................39

Figure 2.4: Simplified block diagrams of the PRSIM architecture (a) encoder and (b) decoder ..............................................................................................................................................41

Figure 2.5: Simplified block diagram of the Stanford DVC architecture ......................43

Figure 2.6: Block diagram of the DISCOVER codec ........................................................45

Figure 2.7: QM₁ – QM₈ as defined in (Brites et al., 2006; Aaron, Rane, Setton, et al., 2004) are shown in (a) – (h) respectively. Each entry in a QM denotes the number of quantisation steps for the quantiser ..........................................................54

Figure 2.8: Factor graph of a LDPC code of 8 variable nodes ($v_j$) and 4 check nodes ($c_i$) .................................................................................................................................................60

Figure 3.1: Block diagram of the DVC-HR framework development process ........68

Figure 3.2: Block diagram a generic DVC framework for testbed implementation .72
Figure 3.3: A sample frame from the test sequences used in DVC-HR evaluation: (a) Bus, (b) Coastguard, (c) Crew, (d) Football, (e) Foreman, (f) Hall, (g) Ice, (h) Mother & Daughter and (i) Soccer .................................................................84

Figure 4.1: QM₁ – QM₈ used in the DISCOVER codec are shown in (a) – (h) respectively in terms of number of bit-planes .................................................................87

Figure 4.2: Maximum magnitudes of coefficient bands for Frame #42 of Foreman CIF sequence ..............................................................................................................95

Figure 4.3: (a) QM₆ and (b) QM generated by the CAQ algorithm for frame #42 of Foreman CIF sequence .........................................................................................96

Figure 4.4: HVS weight matrix for 4x4 block size .................................................98

Figure 4.5: (a) original maximum magnitudes, (b) HVS-weighted maximum magnitudes and (c) resulting QM generated by the CAQ algorithm for frame #42 of Foreman CIF sequence after Enhancement-I ......................................................99

Figure 4.6: QM derived at different stages of the CAQ algorithm (a) original, (b) Enhancement-I only and (c) both Enhancements I & II for Foreman CIF sequence Frame #42 ..................................................................................................................100

Figure 4.7: The CAQ module and the relationship among its components.........102

Figure 4.8: Block diagram of the DVC-HR framework with the new CAQ module highlighted ..................................................................................................................103

Figure 4.9: RD points produced by CAQ and DISCOVER for CIF test sequences: (a) Coastguard, (b) Football, (c) Foreman, (d) Hall, (e) Mother & daughter and (f) Soccer ..................................................................................................................105

Figure 4.10: Bandwidth utilisation comparison of CAQ and pre-determined QM quantisation mechanism of DISCOVER .................................................................107

Figure 4.11: Simulation of adaptive ABP for Foreman CIF sequence .................111
Figure 4.12: Bandwidth usage of adaptive ABP and fixed ABP simulations across all frames of CIF sequences (a) Foreman (b) Soccer

Figure 4.13: Perceptual comparison of frame #104 for Football decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM

Figure 4.14: Perceptual comparison of frame #184 for Foreman decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM

Figure 4.15: Perceptual comparison of frame #90 for Foreman decoded using different quantisation mechanisms at 800kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM

Figure 4.16: Perceptual comparison of frame #78 for Coastguard decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM

Figure 5.1: HVS weight matrices for (a) 8×8 and (b) 16×16 DCT block sizes

Figure 5.2: RD curves for different DCT block size configurations for (a) Crew, (b) Ice and (c) Soccer 4CIF test sequences

Figure 5.3: RD curves for different DCT block size configurations for (a) Coastguard, (b) Crew, (c) Football, (d) Foreman, (e) Mother & Daughter and (f) Soccer CIF test sequences

Figure 5.4: RD curves for different DCT block size configurations for (a) Coastguard, (b) Crew, (c) Foreman, (d) Hall, (e) Mother & daughter and (f) Soccer QCIF test sequences

Figure 5.5: RD curves and PSNR vs log(bit-rate) curves for Mother & Daughter QCIF sequence for calculation of BD-rate metric: (a) the original curve when plotting PSNR
vs log(bit-rate), (b) RD curve for DISCOVER when QM₅ is used instead of QM₁, and (c) resulting ‘normal’ PSNR vs log(bit-rate) curve. ......................................................... 136

Figure 5.6: A sample frame #150 of Crew 4CIF sequence decoded by different configurations of the DVC-HR framework .................................................................................. 138

Figure 5.7: A sample frame #164 of Ice 4CIF sequence decoded by different configurations of the DVC-HR framework .................................................................................. 139

Figure 5.8: A sample frame #148 of Soccer 4CIF sequence decoded by different configurations of the DVC-HR framework .................................................................................. 140

Figure 5.9: A sample frame #76 of Coastguard CIF sequence decoded by different configurations of the DVC-HR framework .................................................................................. 141

Figure 6.1: Factor graph of an LDPC code of 8 variable nodes (vᵢ) and 4 check nodes (cᵢ) ........................................................................................................................................ 149

Figure 6.2: Example demonstrating message propagations in the LDPC code during an iteration of the NWRBP algorithm .................................................................................. 156

Figure 6.3: RD performance curves generated by mNWRBP and Flooding for (a) Crew, (b) Ice and (c) Soccer 4CIF resolution sequence at different DCT block sizes ........................................................................................................... 169

Figure A.1: RD performance curves generated by mNWRBP and Flooding for (a) Coastguard, (b) Foreman, (c) Mother & Daughter and (d) Soccer QCIF resolution sequence at different DCT block sizes ........................................................................................................... 192

Figure A.2: RD performance curves generated by mNWRBP and flooding for (a) Bus, (b) Coastguard, (c) Crew, (d) Football, (e) Foreman, (f) Hall, (g) Mother & Daughter and (h) Soccer CIF resolution sequence at different DCT block sizes ... 193
List of Tables

Table 1.1: Comparison of coding tools to reduce statistical redundancies employed by traditional codecs and DVC codecs ................................................................. 26

Table 2.1: Bit-rate and PSNR comparison of Hall and Soccer QCIF sequences coded by the DISCOVER codec using the same QM .................................................. 50

Table 3.1: Platform and tools used for developing the testbed software .................. 70

Table 3.2: Configuration parameters to control the testbed software behaviour ...... 77

Table 3.3: Summary of test sequences selected for evaluation .............................. 84

Table 4.1: Number of bit-planes produced by QM1 – QM8 ................................. 87

Table 4.2: The length of a coefficient band and subsequent bit-planes corresponding to different spatial resolutions for 4×4 DCT block size .............................. 88

Table 4.3: Step by step procedure of Algorithm 4.1 for frame #42 of Foreman CIF sequence ........................................................................................................... 96

Table 4.4: Bandwidth utilisation statistics for CAQ and DISCOVER quantisation methods ........................................................................................................... 108

Table 4.5: Performance comparison of fixed ABP and adaptive ABP simulation for Foreman CIF sequence .................................................................................. 111

Table 4.6: Performance comparison of fixed ABP and adaptive ABP simulation for Soccer CIF sequence ..................................................................................... 112

Table 5.1: Dynamic range of DC and AC coefficients for various DCT block sizes .............................................................................................................................. 124

Table 5.2: Bit-plane length for various spatial resolutions and DCT block sizes combinations ........................................................................................................... 125

Table 5.3: Simulation parameters for different DCT block size configuration ...... 127
Table 5.4: Average performance gain by DCT-16 and DCT-8 with respect to DISCOVER for 4CIF sequences ................................................................. 130
Table 5.5: Average performance gain by DCT-16 and DCT-8 with respect to DISCOVER for CIF sequences ................................................................. 132
Table 5.6: Average performance gain by DCT-8 with respect to DISCOVER for QCIF sequences .................................................................................. 134
Table 5.7: Updated BD-PSNR and BD-rate metric for Mother & daughter test sequence when DISCOVER RD curves include QM6 instead of QM1 ............... 136
Table 6.1: Total iterations and rate performance of the mNWRBP algorithm for various values of IC for Soccer QCIF sequence ........................................... 159
Table 6.2: Simulation criteria to evaluate the mNWRBP performance .................. 163
Table 6.3: ABP value configurations for different DCT block sizes ................. 164
Table 6.4: Decoding performance of mNWRBP with respect to Flooding for various 4CIF sequences using DCT-16 configuration ........................................ 165
Table 6.5: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several 4CIF sequences ...................................................................... 167
Table 6.6: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several CIF sequences ........................................................ 167
Table 6.7: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several QCIF sequences ......................................................... 167
Table 6.8: BD-rate metrics of mNWRBP with respect to Flooding for Soccer sequence at different resolutions and DCT block sizes ................................. 170
Table 6.9: % Reduction of iterations achieved by mNWRBP with respect to Flooding for bit-planes corresponding to DC coefficients of Soccer 4CIF sequences ......... 171
### Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABP</td>
<td>Available Bit-Planes</td>
</tr>
<tr>
<td>AVC</td>
<td>H.264/Advanced Video Coding</td>
</tr>
<tr>
<td>BP</td>
<td>Belief Propagation</td>
</tr>
<tr>
<td>bps</td>
<td>bits per second</td>
</tr>
<tr>
<td>BSC</td>
<td>Binary Symmetric Channel</td>
</tr>
<tr>
<td>CAQ</td>
<td>Content-Aware Quantisation</td>
</tr>
<tr>
<td>CIF</td>
<td>Common Interchange Format</td>
</tr>
<tr>
<td>CNM</td>
<td>Correlation Noise Modelling</td>
</tr>
<tr>
<td>CODEC</td>
<td>enCOder and DECoder</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DISCOVER</td>
<td>DIStributed COding for Video sERvices</td>
</tr>
<tr>
<td>DSC</td>
<td>Distributed Source Coding</td>
</tr>
<tr>
<td>DVC</td>
<td>Distributed Video Coding</td>
</tr>
<tr>
<td>DVC-HR</td>
<td>DVC for Higher Resolutions</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>fps</td>
<td>frames per second</td>
</tr>
<tr>
<td>Gbps</td>
<td>Giga-bits per second</td>
</tr>
<tr>
<td>GOP</td>
<td>Group Of Pictures</td>
</tr>
<tr>
<td>HD</td>
<td>High Definition</td>
</tr>
<tr>
<td>HEVC</td>
<td>H.265/High Efficiency Video Coding</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IDS</td>
<td>Informed Dynamic Scheduling</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>kbps</td>
<td>kilo-bits per second</td>
</tr>
<tr>
<td>KF</td>
<td>Key Frames</td>
</tr>
<tr>
<td>LDPC</td>
<td>Low-Density Parity Check</td>
</tr>
<tr>
<td>LDPCA</td>
<td>Low-Density Parity Check Accumulate</td>
</tr>
<tr>
<td>LLR</td>
<td>Log Likelihood Ratio</td>
</tr>
<tr>
<td>LR</td>
<td>Likelihood Ratio</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>MCTI</td>
<td>Motion Compensated Temporal Interpolation</td>
</tr>
<tr>
<td>mNWRBP</td>
<td>modified NWRBP</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>MV</td>
<td>Motion Vector</td>
</tr>
<tr>
<td>NWRBP</td>
<td>Node-Wise Residual Belief Propagation</td>
</tr>
<tr>
<td>OpenCV</td>
<td>Open Computer Vision</td>
</tr>
<tr>
<td>PRISM</td>
<td>Power-efficient, Robust, high-compression, Syndrome-based Multimedia coding</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>QCIF</td>
<td>Quarter Common Interchange Format</td>
</tr>
<tr>
<td>QM</td>
<td>Quantisation Matrix</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RBP</td>
<td>Required Bit-Planes</td>
</tr>
<tr>
<td>RD</td>
<td>Rate-Distortion</td>
</tr>
<tr>
<td>SC</td>
<td>Source Coding</td>
</tr>
<tr>
<td>SD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>SF</td>
<td>Scaling Factor</td>
</tr>
<tr>
<td>SI</td>
<td>Side Information</td>
</tr>
<tr>
<td>SW</td>
<td>Slepian-Wolf</td>
</tr>
<tr>
<td>VCC</td>
<td>Virtual Correlation Channel</td>
</tr>
<tr>
<td>WZ</td>
<td>Wyner-Ziv</td>
</tr>
<tr>
<td>WZF</td>
<td>Wyner-Ziv Frames</td>
</tr>
<tr>
<td>WZVC</td>
<td>Wyner-Ziv Video Coding</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

Video is one of the most important forms of modern communications, involving not only traditional usages such as news, documentaries and motion pictures, but increasingly also new forms of communications are being invented and popularised. These include informal communications such as video calls to friends and families, along with more formal communications like video conferencing and video surveillance for security purposes. With the surge of multimedia-enabled mobile devices and widespread use of social media, casual video recording has reached unprecedented levels. YouTube, a website dedicated for sharing videos, sees over 300 hours of video being uploaded every minute (Statistic Brain, 2016) while leading social media website Facebook users watch more than 100 million hours of video every day (Facebook Newsroom, 2018).

The transmission of video data is constrained by its enormous bandwidth requirement. For instance, an uncompressed full High Definition (HD) video with 60 frames per second (fps) requires 1.5 Giga-bits per second (Gbps) bandwidth assuming 12 bits per pixel. Such bandwidth requirement is impractical for many applications so it is essential to compress the data before transmission and storage. A compression mechanism, i.e. a codec (enCODer and DECoder) is thus generally employed in communication systems to compress the video by exploiting statistical redundancies (known as encoding) in the data prior to transmission and then to decompress it back to the original (known as decoding) at the receiver. However, since video comprises an ordered sequence of frames and thus is multidimensional, complex computations
are necessary for effective compression, so a general-purpose codec does not necessarily provide the best compression performance.

### 1.2 Video codecs

A general-purpose codec exploits various statistical redundancies implicit in uncompressed bitstream to achieve compression. However, video data also contains both spatial redundancy, i.e. redundancy within a frame, and temporal redundancy, i.e. redundancy between frames apart from this implicit redundancy. A video codec is specifically designed to effectively exploit these redundancies. Traditionally, this exploitation is performed at the encoder, but since this involves complex computations, a traditional encoder demands significantly more computing resources than its decoder counterpart. For instance, the most popular traditional codec H.264/Advanced Video Codec (AVC) (Wiegand et al., 2003) encoder is typically 5–10 times more complex from a processing viewpoint, than the AVC decoder. This gap is further widened in the state-of-the-art H.265/High Efficiency Video Codec (HEVC) (Sullivan et al., 2012) whose encoder is several times more complex than the AVC encoder (Bossen et al., 2012).

Different codecs employ different methods for successfully exploiting statistical redundancies. For instance, AVC and its successor HEVC employ the Discrete Cosine Transform (DCT) of pixels in a frame, followed by quantisation of DCT coefficients to reduce spatial redundancy. The quantisation of DCT coefficients facilitates better compression however, some information is permanently lost potentially leading to distortion in the decoded frames. Therefore, the choice of quantiser is critical since the amount of distortion depends on the degree of quantisation. Interestingly, the subjective quality of a video is not directly proportional to its distortion quantity and some distortion is not perceived by the human eye. Besides different applications have
varying Quality of Service (QoS) requirements for a video determined by its frame rate, spatial resolution, bandwidth and fidelity (Wiegand et al., 2003). This enables the codec to control distortion by adjusting the degree of quantisation of DCT coefficients to match the QoS requirement. If high quality video is required, the encoder uses a fine quantiser to retain the fidelity, while conversely, in scenarios where bandwidth is critical, the encoder can select a coarser quantiser to discard less significant information and achieve superior compression. For this reason, performance comparison of video codecs is conducted based on their Rate-Distortion (RD) performance which considers not only the compression performance but also the fidelity of a decoded video characterised by how similar it is to the original video. Respective distortions are measured over a range of bit-rate configurations of a codec and the resulting RD curve is compared with the benchmark to assess its performance. The relevant metrics for comparing video codec performance and their measurement process are discussed in detail in Section 3.7.

More complex computations are necessary to address temporal redundancy between frames. Consecutive frames of a sequence generally have a strong correlation, with their difference being characterised by either objects (local) or camera (global) motion. Some frames are encoded independently and referred to as a reference or Key Frames (KF), while others are prediction frames, so only the residual with respect to the KF are encoded instead of all the frames. This is achieved by using motion estimation and compensation techniques. The encoder estimates the motion between predicted frames and KF by computing motion vectors (MV), while the decoder applies these vectors to the transmitted KF to compensate for the motion, thus reconstructing predicted frames. Both AVC and HEVC transmit only the residual frame and the MV for predicted frames, thus achieving significantly superior
compression performance compared to the KF. The decoder performs motion compensation by copying pixels from the KF along the MV, thus reconstructing the predicted frame.

Operational overheads of traditional codecs, particularly finding the motion vectors, require significant computational resources making the encoder much more complex than its decoder counterpart. This processing imbalance between the transmitter and receiver sides is suitable for the traditional broadcasting regime with adequate processing capability, as the video is usually encoded once and transmitted to many receivers before being decoded by devices with diverse computational resources. This means decoder complexity is a key codec design consideration with traditional approaches seeking to reduce it. This however, becomes impractical in emerging processing and memory constrained mobile or handheld devices, which are often employed in remote sensing scenarios (Alvi et al., 2015). These devices do not possess sufficient processing power necessary for traditional complex encoders. Furthermore, with the emergence of resource-scarce wireless mobile devices and the Internet of Things (IoT), the demand for low complexity encoder designs has become more prominent.

1.3 Low complexity encoding

To reduce encoder complexity, traditional codecs like AVC and HEVC provide intra-frame configurations which encode each frame independently (Sullivan et al., 2012; Wiegand et al., 2003). Since they do not employ complex motion estimation methods, the complexity of such configurations is much lower however, intra-frame codecs do not exploit temporal redundancy, so they exhibit poorer compression performance than inter-frame codecs which do exploit this redundancy. So lower complexity versions of traditional codecs are achieved at the cost of compression performance.
Distributed Video Coding (DVC) is an alternate coding paradigm which exploits statistical data redundancies at the decoder (Girod et al., 2005; Aaron et al., 2002; Puri and Ramchandran, 2002). This makes the DVC encoder much simpler than a traditional encoder, while its decoder counterpart becomes more complex. This complexity redistribution facilitates some new and emerging application scenarios such as wireless video cameras, low-power surveillance networks and drones (Alvi et al., 2015; Pereira et al., 2008). Since IoT devices are usually battery powered, the complex computational requirement of a traditional encoder leads to the quick drain of energy. On the other hand, since the resources required of a DVC encoder is lower, it consumes less energy giving a prolonged battery life.

The foundations of DVC are built upon the work of Slepian and Wolf (1973), and Wyner and Ziv (1976). These establish the information theoretic bounds for coding of multiple correlated sources in a separate encoding-joint decoding scenario. Specifically, each source is encoded independently and the corresponding encoder has no access to other correlated sources. On the other hand, the joint decoder has access to all encoded sources and exploit their correlation while decoding. To assist decoding each source, Side-Information (SI) is used, which is knowledge of the other correlated sources available only at the decoder. The Slepian-Wolf (SW) theorem proves that the lossless compression performance of such a setup is equal to a traditional coding scenario where each encoder is aware of the other sources and their correlation is exploited at the encoder. This is important because the encoder does not need access to the other sources and does not exploit their correlation, so requiring lower resources than a traditional encoder. The Wyner-Ziv (WZ) theorem extends this conclusion for lossy coding scenarios with further compression achieved at the cost of distortion by discarding unimportant information.
As video coding is generally lossy, DVC based upon the WZ theorem is often referred to as Wyner-Ziv Video Coding (WZVC). To recreate the scenario of separate encoding, joint decoding as in the WZ theorem, a video sequence is firstly partitioned into KF and WZ Frames (WZF). The KF are encoded independently, possibly using a low complexity intra-frame configuration of AVC or HEVC, while the WZF are encoded using DVC techniques. At the joint decoder, the decoded KF are used to generate the SI. This process is conceptualised in the generic block diagram in Figure 1.1 (a), where every other frame is classified as a WZF. The simplified block diagram in Figure 1.1 (b) shows the various DVC constituent blocks used in a WZ encoder and decoder.

Figure 1.1: Block diagram of a DVC architecture: (a) separate encoder-joint decoder setup, (b) coding tools used in WZ encoder and decoder
At the WZ encoder, WZF are firstly transformed and quantised before being channel encoded to produce parity bits. Interestingly, only these parity bits are transmitted to the decoder, with the actual WZF discarded. At the WZ decoder, the received KF is used to generate the SI by creating an estimate of the WZF. This SI is both transformed and quantised before being processed by the channel decoder, which essentially views the SI as a noisy version of the original WZF. Channel noise is the difference between the original WZF and corresponding SI and is normally represented by a noise model in the channel decoding process. Channel decoding uses the parity bits from the encoder to remove the noise by error correction. As the number of parity bits required for noise removal varies with the SI, their transmission is controlled via a feedback channel to the WZ encoder. After successful noise removal, the inverse transform is applied to reconstruct the decoded WZF.

The generic block diagrams in Figure 1.1 and supporting narrative on the DVC encoding and decoding processes are based on the DISTRibuted COding for Video sERvices (DISCOVER) codec (Artigas et al., 2007). The underlying feedback channel-based architecture has several crucial advantages over feedback-free alternatives, but constrains the coding to be done ‘online’ i.e. encoding and decoding must be performed simultaneously. Thus, this codec is not suitable for either storage purposes or where a feedback channel is unavailable. Its extensions, as well as other alternative DVC solutions, will be examined in greater detail in the literature review in Chapter 2. The different coding tools employed by DVC architectures to exploit various statistical data redundancies compared to a traditional codec are summarised in Table 1.1.
Table 1.1: Comparison of coding tools to reduce statistical redundancies employed by traditional codecs and DVC codecs

<table>
<thead>
<tr>
<th>Statistical redundancy</th>
<th>Traditional codec</th>
<th>DVC codec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal</td>
<td>Motion estimation</td>
<td>SI generation</td>
</tr>
<tr>
<td>Spatial</td>
<td>Transform coding</td>
<td>Transform coding</td>
</tr>
<tr>
<td>Implicit</td>
<td>General purpose compressor</td>
<td>Channel coder</td>
</tr>
</tbody>
</table>

The most significant difference between the two codec types is in the way temporal redundancy between frames is exploited. The traditional codec like HEVC employs motion estimation methods at the encoder to reduce temporal redundancy which is primarily responsible for the increased encoder complexity. DVC in contrast exploits temporal redundancy by generating SI at the decoder so enabling a simple encoder design by transferring the responsibility for exploiting temporal redundancy to the decoder. Another unique aspect of DVC is the employment of channel coder to compress the WZ bitstream. Usually, channel coding increases the payload redundancy to compensate for possible channel errors (distortion) during transmission. In DVC however, the payload is discarded and only the added redundancies, i.e. the parity bits are transmitted. A simple request-transmit protocol is employed to further lower the transmission overhead incurred by the parity bits however, this crucially introduces latency between the encoder and decoder which can be detrimental for practical applications (Qiu et al., 2013). For example, real-time coding has been reliably achieved for low-motion sequences by employing complex parallel processing architectures (Shen et al., 2017). For sequences with higher motion activity however, the DVC framework fails to achieve a sufficient frame rate even in relatively straightforward surveillance applications. Furthermore, in higher spatial resolution sequences, the latency aggregates due to the larger bandwidth requirements and more information must be transmitted, which further compounds this problem.
Although both traditional and DVC codecs employ transform coding, i.e. DCT and quantisation to exploit spatial redundancy, the process is significantly more constrained in DVC. HEVC and AVC can both use hierarchical transform blocks with varying block sizes to improve compression in specific scenarios. This is difficult in DVC since the resulting bitstream is processed by a channel coder which has contradictory requirements. Generally, channel coders work on a fixed length bitstream and show better performance on longer bitstreams. Changing transform block sizes leads to variable length bitstreams, which impacts upon channel coding performance. Moreover, the total length of bitstream depends on the DCT quantised coefficients which add to the requirements of any suitable quantiser thereby making it significantly more challenging to design for DVC and achieve a consistent QoS provision.

The reduced complexity of a DVC encoder is promising for low-power IoT applications. However, the state-of-the-art DVC solutions have yet to realise the theoretical benefit of equal compression performance (Jun, 2019; Taheri et al., 2019; Zhou et al., 2019; Shen et al., 2017). They have established that the RD performance of DVC is consistently superior to the traditional intra-frame configuration but inferior to more complex inter-frame alternatives. Moreover, its performance is highly dependent on the motion characteristics of the sequence being encoded and shows degraded performance for high motion sequences. Crucially, DVC performance enhancements have only been reliably verified on low resolution Quarter Common Interchange Format (QCIF) sequences which is now unrealistic for modern video applications. The corresponding coding performance however, noticeably deteriorates at the higher spatial resolution Common Interchange Format (CIF) sequences (Taheri et al., 2018; Shen et al., 2017). This provided the key motivation to investigate the
development of coding tools to improve DVC performance in higher resolution sequences, without crucially compromising encoder complexity which underpins the main research question addressed by this thesis.

1.4 Research question and objectives

The above discussion reveals that despite being an attractive low-complexity solution, it is not practical to currently apply DVC at higher spatial resolutions, and this has become a major performance bottleneck. To overcome this limitation, the following overarching research question has been framed:

*How can distributed video coding at higher spatial resolution be effectively achieved?*

A DVC codec will be considered effective at higher spatial resolutions if it retains the RD advantages established at lower resolutions. Specifically, the codec exhibits superior RD performance to the traditional intra-frame codec, while encoding CIF and higher resolution sequences. The RD performance must also be significantly better for low-motion sequences and comparable for sequences characterised by high motion activity. It must also retain the inherently attractive core characteristics of DVC, namely that the encoder should have a low overhead design and its complexity does not exceed that of a typical intra-frame codec.

A comparative study of traditional codecs and DVC reveals that exploiting temporal redundancy is the key driver behind compression performance. In DVC, this is performed at the decoder through the SI generation process, though this is very challenging because the decoder, does not have direct access to the original WZF. Hence much DVC research has focused on improving SI generation to better exploit temporal redundancies (Taheri et al., 2019; Zhou et al., 2019; Shen et al., 2017; Abou-
Elailah et al., 2015; Brites and Pereira, 2015; Luong et al., 2014; Salmistraro et al., 2014; Brites et al., 2013; Ascenso et al., 2005). In contrast, little work has been conducted to improve the performance of the transform and channel coding blocks in Figure 1.1 (b) which exploit spatial and implicit redundancies. Existing DVC literature has been critically reviewed and several promising research avenues identified which afford the opportunity to successfully address the above overarching research question. In particular, the following three research objectives have been defined:

1. **To critically analyse the performance of larger transform block sizes at higher spatial resolutions**
   
   **Justification:** Since existing DVC codec performance deteriorates at higher spatial resolutions, it is intuitive to examine in more depth how spatial redundancy is exploited in DVC. This is generally achieved by applying transform coding such as the DCT, on fixed sized pixel blocks, with 4×4 pixels being the usual size applied (Brites et al., 2006; Aaron, Rane, Setton, et al., 2004). Since the DVC performance limitation arises at higher spatial resolutions, employing larger transform block-sizes could be beneficial and worthy of critical investigation.

2. **To develop an efficient and flexible rate control mechanism**
   
   **Justification:** As discussed in Section 1.2, different applications have different QoS requirements. A practical video codec should be flexible enough to accommodate these different application scenarios. Existing DVC codecs provide very few options to flexibly control quality and bandwidth. Moreover, at higher spatial resolutions, these limited options intensify the design problem making the achievement of desired QoS and bandwidth extremely difficult. A
more flexible rate control functionality to provide a range of QoS options is an essential requirement for a practical DVC framework.

3. To synthesise and critically evaluate advanced decoding mechanisms to assuage aggregated latency constraints.

Justification: While the basic request-transmit protocol used in DVC ensures a minimum bit-rate, it often leads to greater latency between the encoder and decoder because the channel decoder frequently makes multiple failed attempts before succeeding. This situation worsens at both higher spatial resolutions and in higher bandwidth scenarios since more information must be processed by an inherently slow protocol. New advanced channel decoding algorithms (Aslam et al., 2017; Liu et al., 2017; Lee et al., 2013; Casado et al., 2007) promise faster and more accurate decoding and these merit further investigation into how they can be modified and embedded into the proposed DVC framework to reduce the processing overheads sustained in each attempt and thereby the overall latency.

In summarising, the overall aim is to investigate and develop a DVC framework that effectively exploits spatial redundancy by employing larger transform blocks, provides a more flexible rate control mechanism for different QoS application scenarios, and employs advanced channel coding techniques to improve the latency issue at higher spatial resolution.

1.5 Contributions

This thesis presents three original contributions to knowledge to fulfil the research objectives and these are integrated into a new DVC framework, known as, DVC for Higher Resolutions (DVC-HR) with the different constituent blocks highlighted in the block diagram in Figure 1.2. The three contributions are specifically:
1. A novel *Content Aware Quantisation* (CAQ) mechanism has been developed for DVC-HR to overcome the limitations of existing quantisation methods. CAQ relates to the Quantisation block in Figure 1.2, though in reality, CAQ is introduced as a separate block in the WZ decoder and does not incur any encoder costs, while being able to dynamically control the degree of quantisation applied to produce desired QoS outputs. CAQ also considers the perceptual impact of DCT coefficients to reduce visible distortion, thus produces the best quality output for a given bandwidth constraint, so fulfilling Objective #2. Crucially CAQ is scalable and can be applied in the quantisation of larger DCT block sizes, so is an integral component to successfully resolving Objective #1.

2. To fulfil Objective #1, larger DCT block sizes have been implemented within the DVC-HR framework with quantisation achieved using CAQ. Two larger DCT block sizes of 8×8 and 16×16 pixels are considered in addition to the normal 4×4 size. The effects of larger DCT block sizes on the overall RD performance have been critically analysed. Several key performance indicators relating to different application scenarios have been identified and
appropriate block sizes for these scenarios proposed. This particular contribution relates to the Transform block in Figure 1.2.

3. The third contribution focuses on the channel coding block of Figure 1.2 where a modified version of the *Node-Wise Residual Belief Propagation* (NWRBP) algorithm (Casado et al., 2007) has been seamlessly integrated into the DVC-HR framework to give superior decoding performance by ensuring each failed attempt is detected more quickly in each processing iteration. This leads to lower decoder overheads in each attempt and an improvement in the overall latency. The potential benefit of lower latency is critically analysed for a range of test sequences comprising different spatial resolutions and DCT block sizes and its corresponding impact on RD performance is evaluated, together with a theoretical complexity analysis of the *modified NWRBP* (mNWRBP) algorithm.

1.6 Thesis structure

The rest of the thesis is organized as follows:

- Chapter 2 presents a thorough literature review of different DVC architectures and the state-of-the-art DVC codecs. It critically evaluates their performance and presents the context for the overarching research question. It also surveys different transform and quantisation methods employed by these codecs and critically analyses their characteristics. It provides a comprehensive study of different channel coding techniques employed by DVC codecs and investigates more recent advanced channel decoding algorithms.

- Chapter 3 provides the justification and rationale of the research methodology adopted in this thesis. It also discusses the choice of the testing platform,
validation and simulation criteria and the evaluation metrics for critically assessing the performance of the developed DVC algorithms.

- Chapter 4 details the CAQ mechanism developed to fulfil objective #2. The CAQ can adapt to any bandwidth constrains and endeavours to produce the best achievable output quality. It is accomplished by introducing dynamic quantisation of transform coefficients. Unlike existing quantisation mechanism, CAQ is scalable and can be used with any transform block size. A critical analysis of the performance of the CAQ is presented. Work from this chapter has been published in (Mahmood et al., 2017a).

- Chapter 5 investigates the effect of larger transform block sizes on RD performance of the proposed DVC-HR framework at higher spatial resolution. It uses the CAQ mechanism developed in Chapter 4 for flexible rate control and compares its RD performance to existing DVC codecs at various bit-rate ranges. It critically analyses RD performance of the framework and identifies potential application scenarios which will derive the most benefit from this advance. Work from this chapter has been published in (Mahmood et al., 2017b).

- Chapter 6 presents a novel approach to reduce the latency issue of the DVC-HR framework that escalates at higher spatial resolution and larger DCT block sizes. The approach involves adopting an advanced Belief Propagation (BP) decoding algorithm, namely the mNWRBP algorithm that affords faster decoding of channel codes. The algorithm is modified to leverage from this property within a DVC context and is implemented in the DVC-HR framework. Critical analysis of the performance results confirms the mNWRBP algorithm can significantly reduce the decoding time of each
attempt and therefore consistently provided lower overall latency for sequences with differing spatial resolutions and DCT block sizes.

- Chapter 7 explore potential enhancements and extensions of the proposed DVC-HR framework and more broadly identifies fertile avenues for future research in the DVC domain.

- Chapter 8 presents some conclusions reflecting on the original contributions made in the thesis and their potential significance.

1.7 Summary

This chapter presented the motivation and challenges of today’s video coding and formulated the overarching research question. It also presented the three primary research objectives and a concise overview of the thesis structure. The next chapter presents a comprehensive literature review of DVC codecs and other associated technologies related to this thesis.
2 Survey of DVC literature

2.1 Introduction

DVC has attracted a considerable attention among the community since its inception and numerous research works have been conducted to bring its strong theoretical basis into practical realisations. Several DVC architectures were initially proposed and while impractical, they paved way for further investigations and the development of many practical state-of-the-art DVC frameworks. The majority of these frameworks share a common underlying architecture consisting of different building blocks responsible for various aspects of the coding process. This chapter presents the theoretical foundations of DVC, a review of various DVC architectures, an in-depth analysis of each building block of the common DVC architecture and a thorough literature review to highlight contributions in each building block as well as to identify gaps for potential research opportunities. The goal of this chapter is to offer to the reader the necessary knowledge and understanding of the DVC workflow along with an up-to-date overview of relevant literature and provide the requisite context for the overarching research question and objectives.

2.2 Theoretical foundation of DVC

2.2.1 Distributed source coding theories

DVC is built upon the fundamentals of the information theoretic results known as Distributed Source Coding (DSC). In general, Source Coding (SC) is the process of symbolising a signal for specific application purposes. The primary objective of SC is to compress a signal, i.e. represent a signal with the minimum number of symbols while keeping the distortion within acceptable limit for the target application.
The encoder exploits statistical redundancies within the source signal to achieve compression. The compression can be lossless, where coded symbols can be used to accurately reconstruct the original signal at the decoder. If, however, exact reconstruction of the original signal is not possible, then the compression is known as lossy and in this case, some information is permanently lost, and the decoded signal becomes an approximation of the original source signal.

Shannon’s (2001) theory defines the rate distortion bound of lossless SC through its entropy function. According to the theorem, if a source \( X \) is to be lossless coded at a rate \( R_X \), it must be greater than or equal to the entropy of source. This is presented in formal notation as:

\[
R_X \geq H(X)
\]

where \( H(X) \) refers to the Shannon’s entropy function.

For coding of multiple sources, a better combined rate can be achieved which is limited by their joint entropy. This is formally expressed for two correlated sources \( X \) and \( Y \) as:

\[
H(X) + H(Y) \geq R_X + R_Y \geq H(X,Y)
\]

Traditionally, this has been achieved through employing a joint encoding and joint decoding strategy. In this setup, both the encoder and the decoder have access to all sources and can exploit correlations between sources to achieve further compression. This however requires communication between sources and encoders which may not be feasible in some application scenarios. DSC resolves this issue by eliminating the need for communication between sources. The encoder setup in traditional SC and DSC are respectively depicted in the block diagrams in Figure 2.1 for two correlated sources.
An alternate approach to the problem of coding two correlated sources is to limit the scope to only one of the sources $X$. The other correlated source $Y$ is assumed to be available at the decoder. In a traditional coding setup, both encoder and the decoder for $X$ have access to $Y$, thus can code $X$ at a rate $R_X$ such that:

$$R_X \geq H(X|Y)$$

(2.3)

where $H(X|Y)$ denotes the conditional entropy of $X$ given the presence of $Y$. In this setup, $Y$ acts as a Side-Information (SI) for coding $X$. According to DSC theory, the same rate can be achieved even if the SI is available only at the decoder. Traditional and distributed setups for coding with SI are depicted in the two block diagrams in Figure 2.2:

The basic proposition of DSC is that while coding multiple correlated sources, there is no rate loss incurred if their correlation is exploited by only the decoder with respect to the case where both the encoder and the decoder exploit the correlation. The SW theorem proves the DSC case for lossless coding (Slepian and Wolf, 1973) while the
WZ theorem extends the former’s result to lossy coding scenarios (Wyner and Ziv, 1976).

2.2.2 DSC realisation

Practical implementations of DSC system involve employing channel codes. Channel codes are specially designed codes to protect a source signal against channel noise, thus preserving the fidelity of a source signal through transmission over noisy medium. The idea is to add redundant information to the source data which can be used to validate correctness and possibly fix errors of the data at the receiver. However, this redundant information inflates the payload and increases transmission overhead. Thus, channel coding is the antithesis of source coding whose primary purpose is to compress the source data to reduce the payload and transmission overhead. Interestingly, certain characteristics of the former can be utilised to achieve the latter in the DSC context.

A very basic example of adding redundant information is the parity bit. This bit represents the evenness of the number of 1-bit in the source symbol and can be used to detect a single bit error. Advanced channel codecs can generate parity bits which can be used to detect as well as correct limited number of errors. Practical DSC systems utilizes this error correcting property of channel codecs. The encoder of a typical DSC system generates parity bits from $X$ using a channel encoder and sends only these parity bits to the decoder, which uses the corresponding symbols of $Y$ and correlation information to approximate $X$. It then simulates the error correcting operation of the channel decoder using the parity bits to correct the estimation errors. Thus channel codes in DSC protect the data across a virtual channel whose noise is characterised by the lack of correlation between the two sources $X$ and $Y$ (Girod et al., 2005).
Initial attempts of implementing a practical lossless DSC system i.e. SW coder include block and trellis codes (Pradhan and Ramchandran, 2003), turbo codes (Aaron and Girod, 2002) and Low-Density Parity-Check (LDPC) codes (Liveris et al., 2002) among others. The results achieved for the systems employing sophisticated channel codes are more impressive and the compression levels achieved for these codes are very close to the SW bound. Practical WZ coder is implemented by cascading a quantiser around a SW coder to simulate the lossy coding scenario, where information loss i.e. distortion, is a function of the quantiser coarseness. A minimal block diagram of such a system is shown in Figure 2.3. Here, the codewords $Q$ are produced by quantising $X$ and acts as the input source for the SW encoder. The latter performs lossless coding and $Q$ is accurately decoded by the SW decoder. The final output $\hat{X}$ is thus a distorted version of $X$ and can be reconstructed from $Q$ using the SI.

![Figure 2.3: WZ coder conceptual block diagram comprising of a quantiser and a SW coder](image)

2.3 DVC architectures

DVC came as a natural extension of DSC and several architectures of the former have been proposed after practical realisations of the latter. A generic DVC architecture block diagram has been shown in Figure 1.1 (b). DVC architectures shares several unique properties that were found to be crucial for new and emerging application scenarios. Firstly, the architectures featured simple encoders compared to traditional video codecs by shifting the responsibility of exploiting temporal redundancies to the decoder. Thus, it advances towards the performance of complex inter-frame codecs at
the cost of simple intra-frame codecs. The underlying DSC system provides inherent protection against channel noise; hence it is robust against subsequent distortions at the decoder and thus they are especially attractive for low power devices and applications that require multimedia transmission over wireless networks. DVC is also attractive for multiview scenarios comprising multiple encoders and a single central decoder where a traditional setup requires communication between encoders for effective exploitation of inter-view redundancies which demands complex network infrastructure. In contrast, in a DVC setup, multiple encoders transmit to the central decoder and do not need to communicate between themselves. Since the central decoder has access to all encoders, it can efficiently exploit inter-view redundancies which can be conveniently achieved by replacing the temporal SI generation block by an inter-view SI generation module.

Amongst the initial DVC architecture proposals, two have gained notable prominence within the community, namely the ‘Berkeley architecture’ and the ‘Stanford architecture’ signifying the respective institutions from which the proposals originated.

2.3.1 The Berkeley architecture
The Berkeley DVC architecture was proposed by Puri and Ramchandran (2002) and styled as PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) (Puri et al., 2007; Puri and Ramchandran, 2003b, 2003a). The underlying SW coder is a syndrome based coder using trellis codes (Pradhan and Ramchandran, 1999, 2003). This is a block-based coding technique i.e. each frame is partitioned into non-overlapping blocks which are encoded individually. The PRISM encoding and decoding processes are respectively depicted in Figure 2.4:
Firstly, the residual of a block with respect to its co-located block in the previous frame is calculated. The residual acts as an estimate of the Virtual Correlation Channel (VCC) noise and the block is classified into a set of predefined classes based on its noise estimate value. A block can be intra-coded if the noise estimate is very high or skipped if the noise estimate is very low. The block is WZ coded if it belongs to a class in-between the aforementioned extrema. Each such block is first transformed using the DCT and quantised according to desired output quality before being syndrome coded using a predefined code associated with the corresponding WZ class. These predefined codes associated with various classes have different rates to reflect different VCC noise characteristics. Finally, the resulting bit-stream is accompanied by a Cyclic Redundancy Checksum (CRC) hash to detect any decoding errors.

At the decoder, a candidate block among several predictors obtained through motion search is selected as the SI. The decoder then uses the SI to obtain the quantised coefficients and the decoded coefficients are verified using CRC and the next
candidate predictor is selected upon failure. Upon successful decoding, the quantised coefficients are de-quantised and inverse transformed to produce the decoded block.

The prime benefit of the PRISM architecture is the shifting of processing complexity from the encoder to the decoder. The encoder does not perform any complex motion search, though it requires the storing of a previous frame to calculate block residuals. Thus, it offers near inter-frame high compression performance at low intra-frame complexity. It is also inherently robust against channel losses due to underlying SW coder and subsequent issues like the drift problem that can appear in predictive coding. The block-based coding model is also suitable for application scenarios where only a particular portion of the scene contains motion, while the rest remains static.

2.3.2 The Stanford architecture

The Stanford DVC architecture was proposed by Girod et al. (2005) featuring a turbo code based SW coder (Aaron and Girod, 2002). It adopted a frame-based coding (Aaron, Rane and Girod, 2004; Aaron, Rane, Setton, et al., 2004; Aaron et al., 2002, 2003) as opposed to the block-based coding of the PRISM architecture. Consecutive frames are split into KF and WZF with the former being intra-coded. A WZF is first transformed using 4×4 block-based DCT and quantised using a predetermined Quantisation Matrix (QM) matching the desired output quality. The quantised coefficients are grouped into coefficient bands and converted into bit-planes before being processed by a turbo encoder (Rowitch and Milstein, 2000) to produce parity bits. The original payload is discarded, and the parity bits are stored in a buffer to be transmitted to the decoder upon request.

The KF is assumed to be available at the decoder which are used to generate SI frame using Motion Compensated Temporal Interpolation (MCTI) techniques. The SI is
then similarly transformed and quantised, and the bit-planes are extracted. These SI bit-planes act as the corrupted bit-planes due to VCC noise at the turbo decoder. The noise characteristics are assumed to be Laplacian and perfectly known at the decoder. The turbo decoder progressively requests more parity bits from the encoder through a feedback channel and attempts error correction using the available parity bits, so ensuring minimum transmission. After all the bit-planes have been successfully decoded, a de-quantisation operation followed by an inverse DCT generates the decoded WZF. The complete process is shown in Figure 2.5:

![Figure 2.5: Simplified block diagram of the Stanford DVC architecture](image)

The Stanford architecture shares the low-complexity encoder and robustness against channel loss characteristics of the Berkeley architecture. Both architectures are suitable for scenarios where many low complexity encoders transmit video data to a central decoder. The frame-based coding model is more suitable for high motion scenarios or scenes that contain global motion. In the latter case, the block-based PRISM architecture may classify each block to be intra-coded, thus essentially reducing the processing to that of a traditional intra-frame codec. The feedback channel of the Stanford coder and progressive transmission of parity bits ensure a minimum bit-rate. It also opens pathways to attractive future upgrades (Girod et al., 2005). For example, the central decoder can be upgraded to employ more
sophisticated SI generation method to generate more accurate SI. The upgraded SI will have stronger correlation to the WZF, thus less corrupted and will require less parity bits to decode its bit-planes. The feedback channel will ensure that only the required parity bits are transmitted, reducing overall bit-rate. Thus, superior RD performance can be achieved by only upgrading the decoder of the Stanford DVC architecture. On the other hand, the feedback channel based decoding method also introduces latency in the system and forces it to perform encoding and decoding concurrently. Therefore, it is not suitable for low-delay applications or scenarios where encoded bit-stream needs to be stored. The PRISM architecture would thus be a better choice in such circumstances.

The PRISM architecture does not follow the DSC principles in the strictest sense since the encoder needs access to potential SI to classify the blocks. Moreover, the classification is performed through the co-located block in the previous frame which is essentially a zero-motion predictor, and subsequently coded at pre-defined rate. On the other hand, the SI is generated via full motion search so a discrepancy exists between them which can lead to an over or underestimation of the bit-rate. This is compounded in a multiview scenario where the SI is generated by exploiting inter-view redundancies, but a PRISM encoder does not have access to co-located blocks in other views. The lack of feedback channel means there is no possibility of correcting such issues. Since the Stanford architecture does not suffer from these issues, and easily extensible and upgradable through improving SI generation techniques, it is widely considered for further research in any practical DVC framework proposal.
2.4 The DISCOVER codec

Several other DVC frameworks have subsequently been proposed based on the aforementioned pair of architectures to develop a practical DVC solution. Among them is the DISCOVER codec (Artigas et al., 2007) which has become popular due to its practicality, consistent coding performance and extensive documentation. It is built upon the Stanford architecture but removes several of its key limitations and crucially produces improved RD performance. A functional block diagram of the DISCOVER codec is given in Figure 2.6:

![Figure 2.6: Block diagram of the DISCOVER codec](image)

2.4.1 The encoder

The DISCOVER encoder consists of several functional blocks which will now be described.

- **Frame Splitting**: Frames of a video sequence are first split into KF and WZF according to the well-known Group Of Pictures (GOP) configuration. The first frame of a GOP is a KF which is encoded using a traditional intra-frame encoder like the AVC (Wiegand et al., 2003). The remaining frames are WZF to be coded in a distributed way. The codec allows GOP consisting of 2, 4, and 8 frames and adopts a hierarchical transmission order for the WZF.
• **Transform and quantisation:** Each WZF is transformed using a 4×4 block based DCT and the transform coefficients are quantised using a predetermined QM. The codec defines a set of eight predetermined QM (Brites et al., 2006; Aaron, Rane, Setton, et al., 2004) from which one is chosen at the beginning of encoding based on the desired output quality. DC coefficients are quantised using a fixed-width uniform scalar quantiser while AC coefficients are quantised using a variable-width dead-zone scalar quantiser. The quantisation step size is varied based on the dynamic range of the AC coefficients to minimise information loss. The quantised coefficients are then grouped together into coefficient bands and bit-planes are extracted from each band to then be processed by the channel encoder. Thus, the selection of a QM provides rate-control features for different QoS requirements. The dynamic range of each coefficient band is also sent to the decoder to ensure that the same quantisers will be used for decoding.

• **Channel encoding:** The DISCOVER codec adopted a *Low-Density Parity-Check Accumulate* (LDPCA) code for the underlying SW coder (Varodayan et al., 2006). The LDPCA code can be described as a cascade of regular LDPC codes of decreasing rate. Initially a higher rate code is used and the resulting syndrome (parity bits) is transmitted to the decoder. If the decoding attempt fails, the encoder moves to a lower rate code and transmits corresponding syndrome. Interestingly, the codes are designed in such a way that a syndrome of a higher rate code is a subset of a lower rate one, therefore transmitting only the additional bits is sufficient. The encoder generates parity bits using the lowest rate code, stores them in a buffer, and transmits them progressively to simulate different coding rates. An 8-bit CRC hash is also transmitted to verify decoding of each bit-plane.
• **Feedback channel and rate estimation:** The DISCOVER codec inherits the feedback channel of Stanford architecture and exploits it to provide reliable decoding. The decoder requests additional parity bits through the feedback channel if the already received amount is insufficient. Thus, it ensures only the minimum required amount of parity bits are transmitted. However, this repeated transmission protocol adds latency to the system and to alleviate this, the encoder is equipped with a minimum rate estimation module that estimates the minimum required parity bits for a bit-plane. The required amount depends upon the VCC noise, i.e. the difference between the WZF and SI bit-planes, though the SI bit-plane is unavailable at the encoder and so needs to be estimated. The encoder transmits the estimated minimum parity bits to the decoder at once and transmits further bits if necessary. This setup substantially reduces both the number of feedback requests and the resulting latency.

2.4.2 The decoder

• **KF decoding:** The KF are decoded using a matching traditional intra decoder to be used by the SI generation module. Since SI for a WZF is generated through MCTI techniques, both temporally preceding and following KF are required. Therefore, temporal order of frames differs from their transmission and decoding order, and the KF from subsequent GOP is decoded prior to decoding of a WZF.

• **SI generation:** The decoder generates an approximation of the WZF by applying MCTI techniques to previous and next KF to be used as the SI. The DISOCVER codec has adopted a bi-directional motion interpolation approach (Ascenso et al., 2005) for this purpose.

• **Transform and quantisation:** The SI is then transformed and quantised in a similar manner to the WZF at the encoder and the dynamic ranges received from
the encoder are used to warrant usage of the exact quantisers. Coefficient bands
are formed from quantised coefficients and bit-planes are extracted. These SI bit-
planes act as the noisy payload at the channel decoder to be recovered using the
transmitted parity bits.

- **Correlation Noise Modelling (CNM):** The CNM module is responsible for
modelling the VCC noise. The VCC is modelled as a Binary Symmetric Channel
(BSC) whose noise characteristics are modelled using a Laplacian distribution.
For this purpose, the CNM module calculates the residual between two candidate
frames generated using forward and backward motion interpolation from the
previous and next KF respectively. This residual is considered as an estimation of
the residual between the SI and the WZF (Brites and Pereira, 2008). The model is
applied for each SI bit-plane to produce the corresponding soft inputs for the
channel decoder. The model parameters are also transmitted back to the encoder
via the feedback channel to assist minimum rate estimation. The CNM module is
an important addition to the Stanford architecture and alleviates the latter’s
unrealistic assumption of perfect a priori knowledge of the VCC noise.

- **Channel decoding:** The LDPCA decoder takes the soft inputs corresponding to a
SI bit-plane as input and performs channel decoding using the received syndrome.
It selects a matching LDPC code and tries to recover corresponding WZF bit-plane
by employing an iterative Belief Propagation (BP) algorithm. The algorithm
terminates on either convergence or after 100 iterations and the resulting decoded
bit-plane is verified using the CRC hash. If verification fails, more bits are
requested through the feedback channel.

- **Reconstruction and inverse transform:** Once all bit-planes are successfully
decoded, they are rearranged into quantised coefficient bands. DCT coefficients
are reconstructed in an RD optimal way (Kubasov, Nayak, et al., 2007) before applying Inverse DCT (IDCT) to produce the decoded WZF.

- Frame merging: At the end of decoding process, the decoded KF and WZF are merged together according to GOP structure to form a continuous video sequence.

2.4.3 Evaluation

The DISCOVER codec is one of the earliest practical DVC solutions based on the Stanford DVC architecture. It eliminates several impractical assumptions of the Stanford DVC architecture and offers a relatively complete set of features with respect to its contemporary alternatives to provide a practical and functional DVC codec. Extensive documentation and availability of a reference implementation of the codec (DISCOVER, 2010; Brites and Pereira, 2008; Artigas et al., 2007; Kubasov, Lajnef, et al., 2007; Kubasov, Nayak, et al., 2007; Brites et al., 2006; Ascenso et al., 2005) established the notion of DVC being better than traditional intra-frame codec, but worse than traditional complex inter-frame codecs in terms of RD performance. However, there are a number of significant limitations in the DISCOVER codec that prevent its deployment in practical real-world applications. Most notable among these are:

- **RD performance:** The superior performance of the DISCOVER codec with respect to traditional intra-frame codecs is consistent for low motion sequences only. For sequences with high motion content such as Soccer, its performance is inferior to that of traditional intra-frame codecs. Moreover, this performance gap extends when higher spatial resolution sequences are encoded (Shen et al., 2017).

- **Latency:** The DISCOVER codec incurs significant latency between its encoder and decoder which is unfeasible for practical realisations. The reference codec decoding time has been observed to be 100 times greater than the corresponding
encoding time for identical software and hardware platforms (DISCOVER, 2010). This puts an unrealistic burden on the encoder buffer and can become a notable framework bottleneck unless handled effectively.

- **Effective rate-control**: Though the DISCOVER codec provides a functional rate-control options through the choice of a QM from a set of eight predetermined QM, its effect is minimal and is difficult to maintain a stable QoS provision. To illustrate this issue, Table 2.1 shows the bit rates and *Peak Signal-to-Noise Ratio* (PSNR) achieved by the DISCOVER codec for Hall and Soccer QCIF sequences. These sequences have very different scene characteristics with the former being a low motion indoor sequence while the latter being a high motion, multi-object sporting sequence. It can be observed by comparing the bit-rates and PSNR of both sequences that the choice of QM does not directly determine either the resulting bit-rate or PSNR. Moreover, the progression of bit-rate and PSNR from one QM to the next is not uniform which means it is extremely difficult to effectively achieve a desired bit-rate or output quality by selecting a QM when coding of an unknown sequence.

<table>
<thead>
<tr>
<th>QM index</th>
<th>Hall Bit-rate (kbps)</th>
<th>Hall PSNR (dB)</th>
<th>Soccer Bit-rate (kbps)</th>
<th>Soccer PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>83.72</td>
<td>31.46</td>
<td>59.56</td>
<td>27.73</td>
</tr>
<tr>
<td>2</td>
<td>93.30</td>
<td>32.03</td>
<td>71.75</td>
<td>28.34</td>
</tr>
<tr>
<td>3</td>
<td>96.37</td>
<td>32.07</td>
<td>83.87</td>
<td>29.11</td>
</tr>
<tr>
<td>4</td>
<td>131.46</td>
<td>34.35</td>
<td>145.42</td>
<td>31.72</td>
</tr>
<tr>
<td>5</td>
<td>134.43</td>
<td>34.36</td>
<td>151.81</td>
<td>31.77</td>
</tr>
<tr>
<td>6</td>
<td>168.57</td>
<td>35.93</td>
<td>200.11</td>
<td>32.83</td>
</tr>
<tr>
<td>7</td>
<td>197.28</td>
<td>37.33</td>
<td>270.02</td>
<td>34.81</td>
</tr>
<tr>
<td>8</td>
<td>298.48</td>
<td>40.66</td>
<td>469.91</td>
<td>38.61</td>
</tr>
</tbody>
</table>

It is important to stress many of these limitations are not unique to the DISCOVER codec and most existing DVC solutions exhibit them to some extent (Chen et al., 2017; Shen et al., 2017; Park and Jeon, 2016; Brites et al., 2013). Subsequently, new and
improved DVC solutions have been proposed to minimise and alleviate these limitations, with the majority of them benchmarking their performance with DISCOVER, thus making the codec the standard choice for DVC performance evaluation. Most of this research has focused upon improving specific functional blocks of the DISCOVER codec and which will be investigated further in the next section.

2.5 Innovations in DVC

Though DSC theory suggests that DVC performance can be comparable to traditional complex inter-frame codecs, existing DVC solutions falls short of that goal in practice. Numerous research works have been conducted (Brites and Pereira, 2015; Brites et al., 2013) to reduce the performance gap and a detailed review of the DVC literature is now presented.

2.5.1 SI generation and refinement

SI quality is the primary determining factor for RD performance of any DVC solution since SI acts as the VCC noise corrupted WZF and a closer approximation to the original frame implies less noise. Since the feedback channel ensures only the required parity bits are transmitted, improving SI quality reduces the bit-rate, thus directly affecting the RD performance of the framework. Thus a majority of DVC literature is aimed at improving SI quality by proposing more advanced and sophisticated SI generation methods (Brites and Pereira, 2015; Brites et al., 2013). In general, SI can be extrapolated or interpolated from already decoded KF and WZF. While extrapolation methods are necessary for low-delay applications (Li et al., 2013; Natário et al., 2005), interpolation methods produce superior quality SI so most literature has adopted the latter approach. Early attempts to generate SI primarily consists of block-based motion compensated frame interpolation (Ascenso et al.,
2005; Girod et al., 2005). These relatively simpler SI generation methods assumed a linear motion model and are suitable for low-motion sequences, but do not work so effectively for sequences with high and more complex motion characteristics. To improve SI quality for high motion sequences, higher-order motion models have been found to be better suited (Akinola et al., 2010; Petrazzuoli et al., 2010). Mesh-based SI generation methods have been observed to create better SI for sequences containing global motion activity (Kubasov and Guillemot, 2006). Superior SI have also been obtained by applying optical flow theories (Luong et al., 2014; Huang et al., 2011).

SI refinement techniques have attracted considerable attention in the DVC community in recent years (Jun, 2019; Taheri et al., 2019; Shen et al., 2017), though several methods can be found in the earlier literature (Badem et al., 2009; Weerakkody et al., 2007; Artigas and Torres, 2005). The idea is to start with a tentative SI, and progressively refine it using already received information during decoding. Since the received information is certain, it can be utilised to augment the SI generation process to improve accuracy of the estimation. Although these methods essentially generate multiple SI for the same WZF, thus increasing the decoder complexity by manifold, their RD performance benefits outweigh their cost. However, it is important to stress that both sophisticated SI generation and refinement method-based DVC frameworks have yet to fulfil the theoretical coding performance promises and importantly their performance has only been critically evaluated on very-low resolution QCIF sequences which are not very practical for most modern applications. Their performance significantly deteriorates at higher resolution sequences which must be addressed for practical DVC deployment (Shen et al., 2017). This provided the key motivation and context for formulation of the research question in Section 1.4. Moreover, since the RD performance issue at higher spatial resolution exists in both
simple and sophisticated SI generation techniques, it can be reasonably argued that the key to resolving the issue lies elsewhere in the DVC framework. Inventing a more sophisticated SI generation method is therefore excluded from the research objectives and will not explored in the proposed DVC-HR framework.

2.5.2 Transform and quantisation

The transform operation is performed to exploit spatial redundancies present within a frame. In the initial Stanford DVC architecture design, individual WZF pixels were compressed and directly transmitted to the decoder (Aaron et al., 2002), with no transform operation performed so spatial redundancies were not exploited. Subsequently transform domain coding became prominent due to the superior RD performance achieved by employing a 4x4 block-based DCT (Brites et al., 2006; Aaron, Rane, Setton, et al., 2004). While some Discrete Wavelet Transform (DWT)-based DVC frameworks have been proposed (Bernardini et al., 2011; Guo et al., 2006), the vast majority of the transform-domain DVC models have employed the DCT.

In DCT-based image/video compression, the DCT coefficients must be quantised to achieve effective compression. Aaron et al. (2004) have proposed a set of seven QM, with a further one added later covering high bit-rates (Brites et al., 2006). These QM have been empirically determined by trial and error by simulating a range test sequences. Figure 2.7 shows the eight predetermined QM where each element in a QM denotes the number of quantisation steps. In DCT-based DVC, for instance, the DISCOVER quantiser for each DCT coefficient is chosen according to corresponding QM element so for example, the quantiser for a DC coefficient would have 64 steps given that QM6 is chosen. A QM is selected at the beginning and used throughout the coding of a sequence in an analogous manner to selecting the quantisation parameter.
of AVC (Wiegand et al., 2003). Both the encoder and the decoder then determine the appropriate quantiser for each coefficient band according to the selected QM.

![Quantisation Matrices](image)

Figure 2.7: QM₁ – QM₈ as defined in (Brites et al., 2006; Aaron, Rane, Setton, et al., 2004) are shown in (a) – (h) respectively. Each entry in a QM denotes the number of quantisation steps for the quantiser.

Since DCT concentrates information towards lower frequency coefficients, they are allocated a finer quantiser to reduce information loss. This also is congruent with the characteristics of Human Visual System (HVS), since the human eye is less perceptive of higher frequency coefficients. As a result, though larger distortion may appear in higher frequency coefficients due to coarse quantisers, they will not significantly affect perceptual quality. It is important to note that these QM are only suitable for the DCT block size of 4×4 pixels, and indeed this is the only block size used so far in almost all Stanford architecture-based transform domain DVC frameworks (Zhou et al., 2019; Shen et al., 2017; Ascenso et al., 2010; Artigas et al., 2007). If the DCT block size needs to be changed, it must be accompanied by defining an appropriately sized QM. Larger DCT block sizes within the DVC context will be methodically investigated in Chapter 5.
In both the Stanford and DISCOVER frameworks, a QM is chosen beforehand and then used throughout the coding process. However, as illustrated in Table 2.1, the bit-rate and output quality for the same QM varies between the test sequences, so it is difficult to achieve and maintain a desired bit-rate and output quality when coding an arbitrary sequence. Moreover, although each QM incurs a higher bit-rate and consequently yields a higher PSNR than the previous, progression from one QM to the next is non-uniform. This issue is compounded in bandwidth-limited scenarios as a higher bit-rate QM may fail to produce an output due to insufficient bandwidth, while concomitantly a lower bit-rate QM may produce a poor output quality but leaving scarce bandwidth unused. This issue can be alleviated by introducing adaptive quantisation and this solution will be critically investigated in greater depth in Chapter 4.

Chien and Karam (2009) have proposed an Adaptive QuanTisation (AQT)-DVC framework where a WZF is partitioned into non-overlapping blocks of 16×16 pixels. QM for each partition is chosen independently of the others based on its local noise characteristics estimation. The basis is that each QM has a certain bit-rate requirement and reduces by a certain amount of distortion. The AQT-DVC framework estimates both the bit-rate requirement and resulting distortion reduction of each QM for a local partition and selects the one which is lower than the target bit-rate and also provides the greatest reduction. Since each local partition can have a different QM, the mapping information of a QM index to a local partition is transmitted to the encoder via the feedback channel, the corollary of which is that the bit-planes will have variable lengths, so multiple LDPCA coders of corresponding lengths need to be used.

The AQT-DVC framework shows superior RD performance than the DISCOVER for low motion sequences, but fails to surpass the latter for high motion sequences,
because the bit-rate and distortion reduction estimation is based upon the SI and since high motion sequences usually have a low-quality SI, the estimation accuracy is decreased. It also incurs significant feedback channel overheads to transmit the QM map to the encoder. Moreover, since each local partition can be quantised with a different QM, perceptual quality can vary across different parts of a frame which may be perceptually uncomfortable for the viewer. It also requires multiple LDPCA coder to account for the variable bit-plane lengths, leading to increased encoder complexity.

Park and Jeon (2016) have proposed a Region Of Interest (ROI)-based adaptive quantisation method. In this method, a 4×4 DCT is applied to a WZF, but only blocks within the ROI bound are quantised and transmitted while the blocks outside the ROI are discarded. To avoid variable bit-plane lengths, the bits corresponding to discarded blocks are zero-padded. Both the encoder and the decoder are assumed to have a priori knowledge of the ROI and it is shown that such knowledge reduces the parity bit-requirement of the bit-plane despite containing redundant padding bits. Similar to the AQT-DVC framework, the best predetermined QM is chosen for the ROI blocks that has a lower than a target bit-rate requirement. The proposed method increases output quality within the ROI under limited available bandwidth scenarios, but often the overall output quality is decreased due to inaccurate ROI definition. It also relies on an estimated bit-rate requirement for a QM, so exhibits poor performance in cases where the SI is of a low quality and thereby leads to inaccurate estimation. ROI-based methods are therefore only suitable for scenes that adhere to the ROI description and contain little motion activities outside the ROI. For unknown scenes with different ROI and/or scenes which contain diverse global and object motion activity, this ROI approach is inappropriate.
Zhang et al. (2017) have proposed a novel method for adaptive quantisation in DVC which considers both HVS perception and RD performance. Interestingly, an 8×8 block based DCT is adopted with the corresponding QM dynamically derived for each frame. For this purpose, a low-complexity Encoder-side SI (ESI) is generated at the encoder and perceptual distortion probability model is produced. The quantisation steps necessary to achieve the target perceptual distortion probability for each DCT coefficient band are determined online in an RD optimised way. The newly determined QM is used for quantisation and thus needs to be transmitted to the decoder. Results show an over 10% bit-rate reduction with similar perceptual quality compared to the AQT-DVC framework, though it is not apparent if the improvement is due to the novel adaptive quantisation or using the larger 8×8 DCT block size. Moreover, the proposed method incurs increased encoder complexity due to ESI generation.

The transform and quantisation blocks are the key to exploiting spatial redundancies in coding frames, however these are generally unexplored with most DVC frameworks adopting a general 4×4 block sized DCT for the transform operation. Larger DCT block sizes have shown improved performance in the traditional coding domain (Sullivan et al., 2004, 2012; Wallace, 1992) and thus this merits further investigation within the DVC context.

Quantisation also facilitates bit-rate control options to provide various QoS requirements. The existing predetermined QM mechanisms is defective in achieving desired bit-rate and output quality, an issue which becomes exaggerated in bandwidth limited scenarios where adaptive quantisation is more suitable. However, existing solutions rely on the approximation of several system parameters, so inaccuracies can arise while in high motion sequences where SI quality is poor, so the overall bit-rate

57
control performance degrades. This judgement leads to a clear aim for novel adaptive quantisation solutions to be developed which will allow more flexible and efficient bit-rate control.

2.5.3 Correlation noise modelling

The CNM block is crucial to DVC performance due to its role in providing soft inputs to the channel coder. An accurate estimate of the VCC noise leads to improved RD performance by requiring fewer parity bits. It is also utilised in estimating the required bit-rate in the aforementioned adaptive quantisation schemes. Most existing DVC frameworks assume a Laplacian distribution for the VCC noise whose parameters are estimated online (Huang and Forchhammer, 2009; Brites and Pereira, 2008). Several other models can be found in the literature (Taheri et al., 2018; Yin et al., 2015; Huang and Forchhammer, 2012) with minor RD performance impact with respect to the former.

Another key aspect of CNM is its role in estimating the minimum bit-rate at the encoder. This is important because the initial transmission contains the estimated number of parity bits. If the minimum bit-rate is underestimated, it will cause the decoder to request more parity bits, thus increasing the latency. Conversely, overestimation of the minimum bit-rate decreases RD performance by transmitting redundant parity bits, so accurate modelling of the VCC noise is essential. Interestingly, although Taheri et al. (2018) claims more accurate model through online refinement, with the improved accuracy coming after the initial transmission so it does affect the minimum rate estimation.
2.5.4 Channel coder and feedback channel

The channel coder is the underlying SW coder within the DVC framework. It decodes the source data using transmitted parity bits and the SI and its error correcting performance directly affects the overall RD performance of the DVC framework. Several alternatives for channel coding are available. The PRISM architecture uses trellis codes while the Stanford architecture uses punctured turbo code however, LDPC codes have been proven to perform better than turbo codes (Liveris et al., 2002), while a rate-adaptive LDPCA code has been proposed by Varodayan et al. (2006). LDPCA codes have been more lately introduced into DVC frameworks due to their superior performance (Taheri et al., 2018; Shen et al., 2017; Zhang et al., 2017; Ascenso et al., 2010; Artigas et al., 2007). Several alternatives including the BCH code (Salmistraro et al., 2013) and arithmetic code (Zhou et al., 2019; Grangetto et al., 2009) have also been developed, though these have not been widely incorporated for DVC purposes.

One significant drawback of the LDPCA code-based DVC is the trial-and-error nature of its decoding scheme which uses the feedback channel (Hu et al., 2018; Chen et al., 2017). This scheme is inherently slow and incurs significant latency when a decoding attempt fails due to insufficient parity bits. This is because the LDPCA decoder consists of a cascade of LDPC code of varying lengths. A suitable LDPC code is selected according to the amount of received parity bits and an iterative BP decoding algorithm is employed that involves message generation and passing between nodes of the underlying factor graph corresponding to the chosen LDPC code. The factor graph of a LDPC code is a bipartite graph with each node corresponding to either one bit in the bit-plane or a parity bit. The former node is called a variable node while the latter is called a check node. Each variable node is connected to a set of check nodes and each
check node is connected to a set of variable nodes according to the parity check matrix. Figure 2.8 shows an example factor graph and the assorted connections between 8 variable (\(v_j\)) and 4 check (\(c_i\)) nodes.

A message from a variable node can be passed to only the check nodes that are connected to it. Similarly, a message from a check node can be propagated to only the variable nodes connected to it. A message corresponds to the Log-Likelihood Ratio (LLR), whose sign and magnitude represent respectively the probable value and reliability of the bit. In a DVC context, each bit in an SI bit-plane corresponds to a variable node and its respective LLR value is determined from the soft-input generated by the CNM module. The BP algorithm terminates if the messages converge or a predefined number of iterations is reached and the decoding attempt is verified using the CRC hash. If the attempt fails, additional parity bits are requested, and the BP algorithm restarted for the newly matching LDPC code.

![Factor graph of an LDPC code of 8 variable nodes (\(v_j\)) and 4 check nodes (\(c_i\)).](image)

Figure 2.8: Factor graph of a LDPC code of 8 variable nodes (\(v_j\)) and 4 check nodes (\(c_i\)).
Due to the iterative nature of LDPCA decoding, each failed attempt is costly as it incurs significant latency which can become a serious issue for applications that require a low delay system. To alleviate this, Kuo et al. (2016) have proposed a DVC framework that limits the number of feedback request allowed for the decoder. The proposed framework combines feedback requests for all bit-planes instead of decoding them sequentially as standard, further reducing total number of feedback requests. Although the limited number of feedback request does not affect RD performance at low bit-rates, the performance significantly degrades at higher bit-rates due to the lower correlation between the least significant bit-planes.

There are a number of feedback-free solutions proposed for low-delay applications (Zhou et al., 2019; Hu et al., 2018; Chen et al., 2017; Qiu et al., 2013) that estimate the required bit-rate at the encoder and transmit the estimated parity bits just once. However, this requires accurate estimation of bit-rate at the encoder which increases encoder complexity. Moreover, if the required rate is underestimated, it will lead to a failure in the bit-plane decoding, thus reducing output quality. Moreover, they generally perform poorly for high motion sequences due to increased disparity between WZF and the SI.

The slow convergence of the iterative BP algorithm and resulting overhead incurred due to failed decoding attempts are the primary causes of the latency issue which intensifies at higher bit-rates due to additional bit-planes needing to be decoded. This is counter-intuitive since adding more bandwidth increases the latency instead of decreasing it. Moreover, at higher spatial resolutions, more data need to be processed, resulting in increased burden for the channel decoder. Existing solutions suffer from undesirable drawbacks, such as, increased encoder complexity and inferior RD performance, therefore necessitates further investigation of the issue. An advanced BP
algorithm that improves the latency by reducing the overhead for each failed decoding attempt will be analytically explored in Chapter 6.

2.6 Gap identification and challenges

Though DVC research has advanced substantially since the development of the DISCOVER codec, the issues identified in Subsection 2.4.3 remain, with the primary limitation being the inferior RD performance compared to traditional codecs like AVC or HEVC. Though DISCOVER shows improved performance with respect to low-complexity intra-frame codecs, it has been consistently verified for only low resolution QCIF sequences, which is unsuitable for modern coding applications (Jun, 2019; Taheri et al., 2018, 2019; Zhou et al., 2019; Hu et al., 2018; Shen et al., 2017). The performance of DVC, however, significantly deteriorates for sequences with higher spatial resolutions, such as CIF and 4CIF, and this provided the main motivation to critically investigate how novel coding solutions could be developed to address this central DVC limitation.

The following critical assessments have also been made from the detailed literature review:

- Most DVC literature focus on improving SI quality due to its obvious effect on the RD performance. Since the RD performance issue at higher spatial resolution exists in both simple and sophisticated SI generation techniques, it can be reasonably argued that the key to resolving the issue lies elsewhere in the DVC framework. Therefore, it will not be investigated further in this work and the proposed DVC-HR framework will adopt the DISCOVER codec’s SI generation method (Ascenso et al., 2005).
• The transform and quantisation blocks are the key to exploiting spatial redundancies in coding frames, however these are generally unexplored with most DVC frameworks adopting a general 4×4 block sized DCT for the transform operation. Larger DCT block sizes have shown improved performance in the traditional coding (Sullivan et al., 2004, 2012; Wallace, 1992) as well as in the DVC (Zhang et al., 2017) domain and thus this merits further investigation within the DVC context.

• Quantisation also facilitates bit-rate control options to provide various QoS requirements. The existing predetermined QM mechanisms is unreliable in achieving desired bit-rate and output quality, as issue which becomes exaggerated in bandwidth limited scenarios where adaptive quantisation is more suitable. However, existing adaptive solutions rely on the approximation of several system parameters, so inaccuracies can arise, while in high motion sequences where SI quality is poor, so the overall bit-rate control performance degrades. This judgement leads to a clear aim for novel adaptive quantisation solutions to be developed which will allow more flexible and efficient bit-rate control.

• Latency is a significant issue in the feedback channel of most DVC frameworks. While feedback channel-free alternatives sacrifice encoder simplicity and RD performance, the primary contributors of the latency are the iterative BP decoding algorithm of the channel coder, the inherently slow protocol for transmitting parity bits upon request and inaccurate modelling of VCC noise, which need to be investigated.

These assessments characterise the most significant limitations of existing DVC frameworks that must be resolved in order to realise a practical DVC solution for
modern application scenarios. This provided the motivation for the research question
and objectives formulated in Section 1.4. A novel DVC-HR framework is developed
in this work to answer the research question whose technical contributions to be
presented in the following chapters. The research methodology adopted will be
presented first in Chapter 3.
3 Research Methodology

3.1 Introduction

This chapter presents the research methodologies adopted to critically evaluate the performance of the DVC-HR framework to address the issues identified in Section 2.6. Following the practice of many DVC frameworks reviewed in Section 2.5, the proposed framework is also based on the DISCOVER codec. However, the novel framework is substantially different from the basis as it introduces a new module in the form of Content Aware Quantisation (CAQ), modifies the existing transform module to enhance its efficacy as well as replaces the traditional BP decoding algorithm with an advanced and better performing one for the LDPCA coder. Throughout the development of the new framework, various decisions have been taken and assumptions made in order to design, develop and evaluate its performance. Their justifications, implications and effectiveness are also discussed in this chapter along with a critical analysis of the performance evaluation criteria and key performance metrics.

3.2 Development methods

Design, development and evaluation of a video coding framework is challenging due to several practical constraints. Development of a framework can be done through software simulation and hardware implementation, both having their own set of advantages and limitations. After thorough analysis and comparison of both methods, a software-based simulation method has been adopted in this work in preference to a hardware prototype-based method.
In general, hardware prototype-based methods provide more accurate and realistic results while software-based methods are flexible and cost-effective. Coding tools used for video compression are generally computationally intensive and often require the tuning of a multitude of parameters. Moreover, framework performance in different scenarios needs to be measured to test its robustness. It is challenging for a hardware implementation to cover such a range of diverse application scenarios. Besides, both the encoder and the decoder performances are constrained by the computational resource capacity of the device hardware. Since encoding and decoding devices vary significantly in terms of computational resources, building hardware prototypes representing a comprehensive range of devices is unrealistic. On the other hand, software-based simulation is both cost-effective and convenient to imitate different scenarios. The framework software can simulate the coding of test sequences with different scene characteristics to measure its robustness across different application scenarios.

Another important factor is the capacity to update, modify, extend and repair the framework. Designing a framework is often a long-term process and many modules and coding tools need to be changed and updated over time. Sometimes, it is essential to compare performances of alternative coding tools in different scenarios before deciding upon one. A software-based simulation in this case saves a lot of development and testing time, and resources.

Software components are often reusable while their hardware counterparts are generally disposable. Since a lot of revision is done during the designing phase of a framework development, the hardware cost can accumulate over time.
Due to these advantages in terms of superior flexibility of designing and testing, and cost effectiveness of software simulation with respect to hardware prototype-based counterpart, the former is the de-facto choice for developing a video coding framework. This is reflected in the development of traditional codecs such as AVC (Wiegand et al., 2003) and HEVC (Sullivan et al., 2012), as well as practically all DVC solutions available in the literature as surveyed in Section 2.5. Hardware implementation usually comes after finalisation and standardisation of a codec development and it is very rare to find any during the designing phase. A rare hardware DVC implementation is found in (Yang et al., 2015) which focuses on the implementation details instead of codec performance.

A software simulation-based approach was also adopted in this thesis for design and performance evaluation of the proposed DVC-HR framework due to aforementioned advantages. A testbed software had been newly developed and several existing coding tools had been adopted and integrated for convenience and conformance. The software was then used to simulate the DVC-HR framework in various application scenarios using a diverse range of test sequences. The framework performance in each scenario has been measured and several relevant and standard performance metrics are calculated. The results were compared with corresponding metrics from the DISCOVER codec to evaluate the framework performance.

3.3 Overview of the framework development process

The development process of the DVC-HR framework is of iterative nature that requires several iterations of planning, implementation and evaluation procedures. The various steps of the process are summarised via a block diagram in Figure 3.1. It is important to note that these steps are not strictly distinctive and may be taken simultaneously in parallel to one another.
• **Literature review**: Critical review of extensive DVC literature is performed and gaps and opportunities for further research are identified. The overarching research question is formulated with several key objectives defined.

• **Scoping & Planning**: The aims and scopes of the framework are determined according to the research objectives. A candidate solution is proposed and the requirements and specifications of the framework are defined.

• **Design & implementation**: The candidate solution is designed and implemented within the testbed software following the specifications to meet the requirements set in the previous step.

• **Testing & validation**: Rigorous testing and debugging is done to ensure that the testbed software accurately models the candidate solution.
• **Experimentations:** Once the testbed software is validated, simulation experiments are conducted using multiple test video sequences to measure the candidate solution’s performance in various application scenarios.

• **Evaluation & Assessment:** The experimental results are quantified and evaluated. The candidate solution’s performance is assessed with respect to its scope and requirements and viability of the solution is determined. It is discarded if not viable and the process returns at the beginning to initiate the next iteration. If a viable solution is determined, the corresponding objective is considered achieved and the framework is updated to include the assessed solution. The process continues with the next objective.

After achieving all the objectives through several iteration of the development process, the novel DVC-HR framework is proposed to answer the overarching research question formulated in Section 1.4.

### 3.4 Implementation of the testbed

This section describes the implementation details of the testbed software in order to provide necessary information to replicate the process. The testbed software enables the development and evaluation of the DVC-HR performance. It is a prerequisite for the design & implementation, testing & validation, experimentations and evaluation & assessment steps of the development process described in the previous section.

#### 3.4.1 Programming language and platform

The testbed software was developed in C++ language in the *Microsoft Visual Studio Community 2015* development environment on a computer running a *Windows 7 Professional* operating system. The platform and tools used for developing are summarised in Table 3.1:
<table>
<thead>
<tr>
<th>Platform/tool category</th>
<th>Platform/tool specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Language</td>
<td>C++</td>
</tr>
<tr>
<td>Development environment</td>
<td>Microsoft Visual Studio Community 2015</td>
</tr>
<tr>
<td>Operating system</td>
<td>Microsoft Windows 7</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel Core i7</td>
</tr>
<tr>
<td>RAM</td>
<td>16 GB</td>
</tr>
</tbody>
</table>

Each platform and tool used in Table 3.1 provides some benefits and also has some limitations. Each one is selected after careful deliberation of its advantages and disadvantages compared to its alternatives. C++ was chosen instead of its alternatives, for instance MATLAB, due to several practical advantages.

- C++ is a general-purpose programming language and a C++ source code can be compiled for a diverse range of devices. On the other hand, MATLAB provides a high-level simulation environment which requires powerful computing resources and lacks support for many non-standard embedded devices. It would be easier to import the framework into a custom hardware prototype implementation if the original software is written in C++.

- Though functionalities of both C++ and MATLAB can be extended using various libraries and toolboxes respectively, however, C++ generally provides more control and flexibility than its counterpart. A library function can be easily extended to accommodate desired variation which is often difficult to achieve using MATLAB toolboxes.

- Another important factor is the cost. MATLAB is an expensive professional software suite and has a separate price tag for each of its toolboxes. Requiring functionalities of multiple toolboxes is very frequent which adds to the cost. In contrast the Visual Studio Community 2015 is a free software. The several libraries used in the testbed development, for instance, Open Computer Vision (OpenCV) is also free and opensource.
• Various coding tools relevant to a DVC framework is available only in C++. For example, the LDPCA coder software developed by (Varodayan et al., 2006) is written in C++.

Microsoft Visual Studio Community was chosen as an integrated development environment primarily due to its easy-to-use, powerful and versatile debugging tools. It makes testing and debugging the software tremendously easier compared to its alternatives. It is also free to use for non-commercial projects. As it is only available for Microsoft Windows operating systems, Windows 7 was selected as it was the most stable and suitable version for development. However, the source code is portable to other platforms through appropriate compiler and has been tested to work on a Debian Linux distribution. The computer hardware was provided by the institution and the information only serves as a reference, since the testbed can be run on other hardware configurations having a similar software platform.

Several external libraries have been used while developing the testbed software for convenience and conformance. Functions and tools provided by these libraries have been tested and validated through synthetic data prior to their application in the testbed.

• **OpenCV**: OpenCV is an open source image processing and computer vision library written in C++. It provides a wide range of powerful and optimised image processing tools. In particular, the DCT and corresponding inverse transform functionalities have been used in the testbed. Other data structures functions were also used when necessary primarily for easier and consistent inter-operability between different modules. For example, each frame of a test video sequence was stored in OpenCV’s Mat data structure.
- **LDPCA**: The LDPCA coder software from (Varodayan et al., 2006; Varodayan, 2005) was used for encoding and decoding of bit-planes.

- **CRC generator**: A 8-bit CRC generator is adopted from (Shen et al., 2017) to calculate CRC of a bit-plane at the encoder. The 8-bit checksum is used at the LDPCA decoder to validate successful decoding of a bit-plane.

### 3.4.2 Implementation of DVC modules

Figure 3.2 reproduces Figure 1.1 (b) showing different modules of a generic DVC framework. To develop the testbed, requirements of each module have been defined first before designing and implementation of the module. During the designing phase, specifications of the DISCOVER codec as detailed in Section 2.4 have been followed to mimic its performance for benchmarking. More importantly, the testbed has been designed such that several alternative realisations of each module can coexist and conveniently activated through configuration parameters. This ensures that while experimenting any proposed scheme, only the corresponding modules differ from the benchmark and rest of the framework remain identical to enable equitable comparison.

![Figure 3.2: Block diagram a generic DVC framework for testbed implementation](image-url)
Requirements for each module and corresponding realisation according to the DISCOVER codec is given below:

- **SI generation module:** This module takes previous and next KF of a WZF and generates the SI necessary to decode the WZF frame. Following the DISCOVER codec, it performs a *Motion Compensated Temporal Interpolation* (MCTI) on two input frames and creates an estimate of the frame temporally between the input frames (Ascenso et al., 2005).

- **Transform module:** This module performs a block based DCT operation on a WZF (at the encoder) or an SI (at the decoder) using a specified block size. The DCT coefficients are then grouped into bands for quantisation. The 4×4 block size employed by the DISCOVER codec has been implemented for this module.

- **Quantisation module:** This module performs quantisation operation on DCT coefficient bands according to a provided QM before converting into bit-planes to be processed by the channel coder. The eight predetermined QM detailed in Figure 2.7 have been realised in the testbed software.

- **Correlation Noise Modelling (CNM) module:** This module models the probability distribution of the VCC noise and determines soft-inputs for the channel decoder. This module requires an estimate of the difference between the WZF and the SI which is provided by the SI generation module. The noise model for DCT coefficient bands proposed by Brites and Pereira (2008) has been implemented.

- **Channel encoder module:** This module generates parity bits from bit-planes produced by the quantisation module. These parity bits are stored in a buffer
before being progressively sent to the decoder. The aforesaid LDPCA library
(Varodayan et al., 2006; Varodayan, 2005) has been integrated for this module.

- **Channel decoder module:** This module is used at the decoder and decodes
an SI bit-plane using soft-inputs from the CNM module and parity bits from
the encoder buffer. The decoded bit-planes are transferred to the reconstruction
module. To complement the LDPCA encoder, the LDPCA decoder provided
by the LDPCA library has been integrated. However, a few necessary
modifications, for example, CRC verification of a decoded bit-plane has been
performed.

- **Reconstruction and inverse transform module:** This module is responsible
for generating the final output of the framework. It first restores quantised
DCT coefficients from decoded bit-planes before reconstructing the DCT
coefficients using the SI (Kubasov, Nayak, et al., 2007). Inverse DCT is then
performed to produce the decoded frame.

The aforementioned modules are the primary ones developed and integrated in the
testbed. Apart from these, several other secondary modules were developed to ensure
smooth inter-operability between modules. For example, there is a module for parsing
the test sequence and splitting the WZF and KF, while another module supervises
encoding and decoding of a KF using AVC intra-frame configuration.

The testbed software is designed such that it can simulate the DISCOVER codec for
benchmarking purpose. This allows critical evaluation of performance of the
framework in scenarios which are not available in the reference DISCOVER codec
software. For example, the reference software can code only QCIF and CIF test
sequences and cannot work with any other, for example 4CIF, test sequences. It also
has a predetermined set of configurations for a few selected test sequences and
behaviour for a test sequence excluded in the selection is undetermined. Therefore, the performance results of the DISCOVER codec for comparison and benchmarking in this thesis were obtained using the simulated version instead of the reference software to be consistent across different scenarios.

The performance of the simulated DISCOVER codec is tested for a diverse range of scenarios and it was observed that it performed reasonably similar to the reference software. The slight difference can be attributed to variations in configuration parameters. Apart from enabling certain scenarios, using the simulated DISCOVER codec more importantly also ensures equitable and precise evaluation of the DVC-HR framework. For instance, while assessing performance impacts of the CAQ mechanism presented in Chapter 4, it was ensured that the only difference between the two is the QM used where the benchmark DISCOVER codec used a predetermined QM and the DVC-HR used a dynamic QM generated by the CAQ module.

3.4.3 Software testing and validation

Individual modules have been tested and validated before being integrated into the testbed software. The testing has been done through performing corresponding operations on synthetic data and comparing the results with ground truth. In general, the following steps comprise the testing of a module:

a. Synthesise the test data by generating random numbers and corresponding ground truth by evaluating the module’s definition.

b. Simulate the module’s operation with the synthetic data from Step a as input.

c. Compare the module’s output with the ground truth and validate its functionality.
For example, while validating the transform module, a random sequence representing pixel values have been generated and corresponding DCT coefficients have been computed as the ground truth by applying the DCT equation on the sequence. In Step b, the sequence has been taken as the input and the DCT module has been simulated to produce coefficients. In the last Step c, the output of the module is compared with the ground truth from Step a. An exact match was observed and thus the DCT module’s functionality was verified.

Producing the ground truth in Step a is challenging for some modules. For example, the SI generation module estimates a WZF by performing MCTI on two neighbouring KF. Due to extensive scale of operations required for this task, producing and comparing a ground truth is not feasible. For this module, the generated SI has been contrasted with the two KF as well as the corresponding WZF and it was verified that a reasonable estimate of the WZF was produced. More importantly, both the benchmark DISCOVER codec and the DVC-HR framework share the same SI generation module, thus equally affected by its operation.

After all the modules have been tested and validated, they are integrated into the testbed software. Sufficient care has been taken to ensure accurate and smooth communication between different modules. Finally, another set of testing and validation has been performed to confirm collective functionalities of the modules. For instance, a sample frame has been transformed with the DCT module followed by inverse transformed using the IDCT module. The output of the IDCT module has been compared and found to be identical to the input of the DCT module. It was thus verified that both the DCT and the IDCT module is working properly after being integrated into the testbed software.
3.5 Simulation criteria

After the testbed software is tested and validated, simulation experiments were conducted using a range of test sequences to measure its performance in various application scenarios. Several configuration parameters were selected and a few scenario conditions were assumed throughout the simulation experiments.

3.5.1 Video format

Most of the test sequences used in simulation experiments are CIF that have a spatial resolution of 352×288 pixels and a frame rate of 30fps. Each frame of a video sequence is encoded in YUV 4:2:0 planar format, where Y is the luminance channel and U and V are the chrominance channels. Planar format indicates that pixel values of each channel are grouped together instead of being interleaved and 4:2:0 means that the spatial resolution of both chrominance channels U and V is half of the luminance channel Y in each dimension. Only Y channel is coded while U and V channels are ignored during experimentations following the norm of DVC solutions available in the literature including the DISCOVER codec (Taheri et al., 2019; Hu et al., 2018; Shen et al., 2017; Artigas et al., 2007). Apart from CIF test sequences, QCIF and 4CIF test sequences were also used which have spatial resolutions of 176x144 pixels and 704x576 pixels respectively.

3.5.2 Configuration parameters

Various configuration parameters were used to select and control behaviour of the testbed software which are listed in Table 3.2:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Accepted Values</th>
<th>Used Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOP size</td>
<td>2, 4, 8</td>
<td>2</td>
</tr>
<tr>
<td>DCT block size</td>
<td>2, 4, 8, 16, 32</td>
<td>4, 8, 16</td>
</tr>
<tr>
<td>QM</td>
<td>1, 2, 3, 4, 5, 6, 7, 8</td>
<td>1, 2, 3, 4, 5, 6, 7, 8</td>
</tr>
<tr>
<td>Available Bit-Planes (ABP)</td>
<td>Any positive integer</td>
<td>Depends on DCT block size</td>
</tr>
<tr>
<td>Advanced LDPCA Decoding</td>
<td>TRUE, FALSE</td>
<td>TRUE, FALSE</td>
</tr>
</tbody>
</table>
The *GOP size* parameter determines the frequency of KF and WZF and a value of \( n \) means that for a group of \( n \) consecutive frames, first frame is a KF while remaining \( n-1 \) frames are WZF. Although the testbed software allows various GOP sizes, only the GOP size of 2 was used during experimentations as it is the most commonly used configuration in the DVC literature for comparing competitive codecs’ performances.

The *DCT block size* parameter controls the block size of the DCT operation and the values 4, 8, 16 indicating block sizes of 4×4, 8×8 and 16×16 respectively were used during simulations. The 8×8 and 16×16 block sizes are presented in Chapter 5. While simulating the DISCOVER codec, the DCT block size of 4×4 is used and the *QM* parameter is selected at the beginning of a simulation and corresponding predetermined QM is used. While simulating the CAQ performance, the *ABP* parameter needs to be provided instead of the QM parameter. Although any positive integer is a valid value for the ABP parameter, a very low or very high value can be redundant and impractical. Practical values for the ABP parameter depend on the DCT block size and will be discussed in Section 4.4 while analysing its impact on the codec performance. The last parameter in the table indicates whether to use the advanced LDPCA decoding algorithm presented in Chapter 6 instead of the traditional decoding algorithm used in the DISCOVER codec.

The following assumptions were made during simulations:

- Communications between encoder and decoder is flawless. The decoder receives data from the encoder in a perfect condition and the encoder also receives decoder feedbacks perfectly.
- The KF are coded using AVC intra-frame configuration and available at the decoder for SI generation.
• The decoder has sufficient processing resources to provide accurate and timely decoding of a WZF.

These assumptions are necessary for the software simulation-based methodology adopted in this thesis. The experimentations can be simplified through these assumptions as many potential hardware issues can be ignored. Instead, answering the research question and fulfilling the research objectives outlined in Section 1.4 can be focused on by investigating various coding tools and developing the DVC-HR framework.

3.6 Performance benchmark

The performance of DVC-HR framework is measured and compared to the performance of DISCOVER codec for a range of test sequences. Although many DVC solutions in the literature present their performances in comparison to AVC intra-frame and inter-frame codecs, the DISCOVER codec is chosen as the benchmark due to several practical reasons:

• Majority of the recent and advanced DVC solutions analyse and discuss their performances compared to the DISCOVER codec, while present the RD curves of AVC codecs as a backdrop (Taheri et al., 2019; Zhou et al., 2019; Shen et al., 2017).

• It is also well established that the performance of the DISCOVER codec at low spatial resolution is consistently between AVC intra-frame and inter-frame codecs. Therefore, presenting RD curves of the AVC codecs adds little value to the discussion. It is also noteworthy that the AVC reference software is publicly available as well as the test sequences used in this thesis, therefore
the effort required for an interested reader to generate respective RD curves and compare their performances is trivial.

- Although several advanced DVC solutions available in the literature which report significant performance improvements over the DISCOVER codec as reviewed in Section 2.5, they have the same performance issue at higher spatial resolution. Moreover, these solutions focus on improving the SI module, thus their advantages do not stack, rather competes and replaces one another. On the other hand, the contributions in this thesis focuses on other modules of the architecture, thus can be easily transferred to a more advanced codec. Therefore, the DISCVOER codec is chosen as the benchmark since it is the basis of both the DVC-HR framework presented in this thesis and advanced DVC solutions available in the literature.

- The reference software of the DISCOVER codec is also publicly available, therefore an interested reader can easily verify the performance metrics presented in this thesis.

### 3.7 Key performance metrics

During each simulation experiment, several measurements were taken to quantify the DVC-HR performance. Following key performance metrics were considered to quantify its performance:

- **Bit-rate**: Bit-rate is the average number of bits transmitted from the encoder to the decoder in unit time. It is measured in \textit{bits-per-second} (bps), though prefix multipliers like \textit{kilo} are often used for convenience which is denoted by kbps. The number of transmitted parity bits and bits required to transmit
secondary data, for instance, range of AC coefficient bands and CRC checksum of a bit-plane contributes to the number of transmitted bits.

- **Peak Signal-to-Noise Ratio (PSNR):** PSNR is a measurement of distortion and is a widely used metric to quantify quality of a signal. In the testbed software, PSNR of a frame is computed from the corresponding differences in individual pixel values between an original WZF and its decoded version. PSNR of a sequence is determined by taking average PSNR of all WZF for performance comparison.

- **RD curve:** The PSNR of a sequence depends on the bit-rate and higher bit-rate generally yields better PSNR. Therefore a single bit-rate and PSNR pair is often unrepresentative and the progression of bit-rate and PSNR is more significant. This progression is depicted through a bit-rate versus PSNR curve which is popularly known as RD curve. To produce a RD curve for a codec, a sample number of bit-rate and PSNR pair values are obtained using different configuration, each denoting an RD point in the graph. Successive points in a RD graph is usually connected by a straight line to indicate estimated RD performance between them. This provides an effective visual comparison of the performance of multiple codecs and have been widely adopted in both DVC and traditional domain (Taheri et al., 2019; Shen et al., 2017; Brites et al., 2013; Sullivan et al., 2012; Wiegand et al., 2003).

In some cases, connecting subsequent RD points may be misleading and disconnected set of RD points better represents the rate-control options of a codec. In case of the DISCOVER codec, only 8 rate-control configurations are defined and connecting corresponding RD points mis-represents the rate-control options since there is no configuration between two consecutive RD
points. This issue is illustrated in Chapter 4 where the DVC-HR is shown to offer superior and flexible rate-control options compared to the DISCOVER codec.

- **Bjontegaard Delta (BD-) PSNR and rate:** The BD-PSNR and BD-rate metrics (Bjontegaard, 2001) quantify the performance gap between the RD curves of two codecs. They basically signify the average PSNR and bit-rate differences across the RD points of both graphs and are especially useful if one codec performs better than the other at certain bit-rate ranges, but worse in other bit-rate ranges. This is also a common metric used for benchmarking DVC solutions (Jun, 2019; Shen et al., 2017; Park and Jeon, 2016).

The calculation of these metrics assumes exponential growth in bit-rate with respect to PSNR between successive RD points which can lead to anomalous figures for RD curves which contrast with the assumption. One such case is explored in Section 5.3 where the RD curve for *Mother & Daughter* sequence generates inconsistent BD-rate metric. Further investigation reveals irregular bit-rate progression at very low bit-rates causing the issue, which subsequently is fixed by considering RD points at higher bit-rates.

- **Bit-rate progression across frames:** While the RD curve provides useful insights of overall codec performance, it is also significant to know the progression of expended bit-rates across the frames of a sequence. This is displayed by a line graph depicting bit-rates expended for each frame. This graph shows the uniformity and stability of bandwidth consumption and also displays the required bandwidth for a smooth transmission to the decoder.

It is important to stress that all the above performance metrics are only computed for the WZF in this thesis. This is an established practice in the community (Taheri et al.,
2019; Park and Jeon, 2016) that offers better insights on the DVC performance over its alternative of computing metrics for all frames. This is because KF are separately encoded and decoded using a dedicated AVC intra-frame codec, and while this impacts upon the SI quality to some extent, both DVC-HR framework and the benchmark DISCOVER codec benefit equally from this. Thus, in order to avoid any confusion or misinterpretation of the results, KF measurements are not considered in the critical evaluation of the comparative assessment of the coding performance of the various DVC models.

### 3.8 Test sequences

To critically evaluate the performance of the DVC-HR framework, a set of community-accepted test sequences exhibiting a range of diverse characteristics were employed. These included varying degrees of global and local motion, complex texture, multiple moving objects, occlusions and disocclusions, and pragmatically covered an extensive set of application scenarios. While the DVC community mostly has been using QCIF sequences, CIF sequences are primarily used for evaluating the DVC-HR performance at a higher spatial resolution. Some 4CIF sequences like *Crew* and *Soccer* having even higher spatial resolution is also used to evaluate the scalability and robustness of the DVC-HR framework. Table 3.3 summarises the key features of the chosen sequences, which are publicly available and downloaded from (Xiph.org, 2017).

A sample frame from each test sequence is presented in Figure 3.3. The testbed software was simulated to encode and decode these test sequences and corresponding performance metrics were computed. For comparison, equitable performance metrics were also computed for the simulated DISCOVER codec. The discussion and analysis of experimental results are presented in the following chapters.
Table 3.3: Summary of test sequences selected for evaluation

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Total frames</th>
<th>Format</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>150</td>
<td>QCIF, CIF</td>
<td>Global motion, high local motion, multiple moving objects and occlusion.</td>
</tr>
<tr>
<td>Coastguard</td>
<td>300</td>
<td>QCIF, CIF</td>
<td>Global motion, local motion, multiple moving objects, occlusion and complex texture.</td>
</tr>
<tr>
<td>Crew</td>
<td>300</td>
<td>QCIF, CIF, 4CIF</td>
<td>Global motion, high local motion, multiple moving objects and occlusion.</td>
</tr>
<tr>
<td>Football</td>
<td>260</td>
<td>QCIF, CIF</td>
<td>Global motion, very high local motion, multiple moving objects and occlusion.</td>
</tr>
<tr>
<td>Foreman</td>
<td>300</td>
<td>QCIF, CIF</td>
<td>Global motion, local motion, varying motion activity</td>
</tr>
<tr>
<td>Hall</td>
<td>300</td>
<td>QCIF, CIF</td>
<td>Local motion</td>
</tr>
<tr>
<td>Ice</td>
<td>240</td>
<td>QCIF, CIF, 4CIF</td>
<td>High local motion, multiple moving objects and occlusion.</td>
</tr>
<tr>
<td>Mother &amp; Daughter</td>
<td>300</td>
<td>QCIF, CIF</td>
<td>Local motion</td>
</tr>
<tr>
<td>Soccer</td>
<td>300</td>
<td>QCIF, CIF, 4CIF</td>
<td>Global motion, very high local motion, multiple moving objects and occlusion.</td>
</tr>
</tbody>
</table>

Figure 3.3: A sample frame from the test sequences used in DVC-HR evaluation: (a) Bus, (b) Coastguard, (c) Crew, (d) Football, (e) Foreman, (f) Hall, (g) Ice, (h) Mother & Daughter and (i) Soccer
4 Content-Aware Quantisation of Transform Coefficients

4.1 Introduction

To reduce spatial redundancy, transform-based DVC frameworks perform their basic transforming and quantising operations on WZF pixels. While some information is lost during quantisation, the resulting bit-rate is significantly lower making this trade-off advantageous from an RD perspective. The transform and quantisation operations also enable the framework to generate a variety of output qualities and bit-rates to better match the QoS requirements of different applications. This is generally realised by changing the quantiser for transform coefficients, so fine quantisation is used if superior output quality is required and conversely, a coarse quantiser is applied in bandwidth critical scenarios.

As evidenced in Chapter 2, most DVC frameworks employ a 4×4 block based DCT and offer several preset QM for rate control purposes, though these are not adequate for practical applications since the quantisation mechanism is inflexible and offers few opportunities to control either the output quality or bit-rate. Moreover, it is difficult to obtain a desired output quality and bit-rate because the progression from one QM to another is non-uniform. At higher spatial resolutions, these restrictions compound the design problem, so that fulfilling a desired QoS and bit-rate is extremely challenging. Existing solutions attempt to resolve these issues by developing adaptive quantisation methods (Zhang et al., 2017; Park and Jeon, 2016; Chien and Karam, 2009), however, the price is to compromise encoder complexity, incur a higher transmission overhead or use an undesirable partial coding scheme as reviewed in Subsection 2.5.2. Thus, an
innovative solution that does not suffer from the aforementioned drawbacks has considerable potential and merit in terms of overall DVC performance.

This chapter presents a novel Content-Aware Quantisation (CAQ) mechanism to provide an efficient and flexible rate-control solution for the new DVC-HR framework while crucially minimising the impact of the drawbacks in existing alternatives. In Section 4.2, the challenges and restrictions of designing a QM within the DVC context are discussed. The new CAQ algorithm is then presented in Section 4.3, while Section 4.4 presents a rigorous analysis of the experimental results to evaluate the performance of CAQ. Some concluding remarks are then provided in Section 4.5.

4.2 Limitations and design challenges

The quantisation mechanism of the benchmark DISCOVER codec has been reviewed in Section 2.4. To recap, a 4×4 block based DCT is employed together with a predetermined QM based quantisation to remove spatial redundancies. An LDPCA coder then processes these quantised coefficients which is known to perform better on longer sequences (Varodayan et al., 2006). To exploit this property, transform coefficients are arranged into bands according to their corresponding frequency component. Each coefficient band is then collectively quantised, with the number of steps derived from the QM. The DISCOVER codec utilises eight predetermined QM where each entry in a QM denotes the number of quantisation steps. Bit-planes are extracted from quantised coefficient bands to be processed by the LDPCA coder. Each QM entry is set to a power of 2 for maximum bandwidth utilisation, therefore, it essentially denotes the number of bit-planes to be extracted from each quantised coefficient band. The eight QM has been presented in Figure 2.7, which is reproduced here in terms of number of bit-planes in Figure 4.1.
One of the prime limitations of this set of eight predetermined QM is that the bit-rate progression from one QM to the next is not uniform as discussed in Subsection 2.4.3. This becomes more apparent when presented in terms of total number of bit-planes extracted by employing a QM as detailed in Table 4.1. For example, the bandwidth requirement for QM4 is almost twice that of QM3, while switching to QM6 from QM7 saves only 10% of the bandwidth, assuming equi-compression of the bit-planes. Therefore, it is extremely difficult to reliably achieve the desired output quality or bandwidth.

### Table 4.1: Number of bit-planes produced by QM1 – QM8

<table>
<thead>
<tr>
<th>QM</th>
<th>Bit-planes</th>
<th>QM</th>
<th>Bit-planes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>6</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>8</td>
<td>63</td>
</tr>
</tbody>
</table>

The erratic progression of bit-rate across different QM is also affected by the length of a bit-plane which is determined by 1) the spatial resolution of the video and 2) the DCT block size. Table 4.2 shows bit-plane lengths corresponding to different spatial

![Figure 4.1: QM1 – QM8 used in the DISCOVER codec are shown in (a) – (h) respectively in terms of number of bit-planes](image)
resolution for 4×4 block size. It can be seen that while coding higher resolution sequences, the length of each bit-plane is increased manifold compared to lower spatial resolution, making the actual bit-rate difference between successive QM larger. As a result, utilising the available bandwidth and maintaining a desired QoS level becomes harder. Thus, the inflexible rate-control issue intensifies at higher spatial resolutions.

Table 4.2: The length of a coefficient band and subsequent bit-planes corresponding to different spatial resolutions for 4×4 DCT block size

<table>
<thead>
<tr>
<th>Format</th>
<th>Resolution (pixels)</th>
<th>Bit-plane length</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCIF</td>
<td>176x144</td>
<td>1584</td>
</tr>
<tr>
<td>CIF</td>
<td>352x288</td>
<td>6336</td>
</tr>
<tr>
<td>4CIF</td>
<td>704x576</td>
<td>25344</td>
</tr>
</tbody>
</table>

Since there are large gaps between some predetermined QM, new QM needs to be introduced if the target QoS requires anything in between. However, it is a challenging task due to several properties of DCT. DCT tends to aggregate information within the block towards lower frequency components, therefore, magnitudes of lower frequency coefficients are usually greater than magnitudes of higher frequency coefficients. Therefore, allocating a bit-plane to a lower frequency coefficient band yields greater RD benefit than allocating to a higher frequency counterpart. This is also reflected in the eight predetermined QM where each lower frequency coefficient has equal or greater number of bit-planes allocated than its higher frequency counterparts. Moreover, the 2D DCT is a composition of 1D DCT along its rows and columns and the coefficient at a certain row and column has equal significance to the coefficient at the transposed position. The eight predefined QM are all symmetric with respect to their main diagonal to reflect this property. There are only a few potential QM arrangements possible excluding the existing ones which respect both aforementioned properties. Moreover, allocating an additional bit-plane to a coefficient not on the
main diagonal breaks this symmetry, putting extra significance to the coefficient band. Since each scene has different DCT coefficient distribution, an asymmetric QM does not perform well for every scene. For these reasons, determining a new QM within the DVC context is challenging.

The erratic progression of QM causes another problem when scene characteristics or QoS requirement changes during coding. Bit rates for the DISCOVER codec primarily depend on the quality of SI, so using the same QM to code sequences with disparate scene characteristics can result in significantly different in the resulting bit-rates. The QM needs to be adjusted in order to match QoS changes. However, since there are large and irregular gaps between QM, and little can be done to decrease these gaps, seamless transition from one QM to another is not possible in the DISCOVER codec.

In summary, the predetermined QM approach of the DISCOVER codec has several design constraints which make it exceedingly difficult to control bit-rates and to closely match QoS requirements. Especially in bandwidth constrained scenarios, properly utilising the available bandwidth to produce the best output is highly challenging. Moreover, it is also problematic to switch QM during coding in order to adapt to changing scene characteristics and QoS requirements. The following section presents the CAQ mechanism for the DVC-HR framework in order to overcome these limitations.

4.3 The CAQ mechanism

The CAQ mechanism analyses the DCT coefficient distribution and allocate bit-planes to coefficient bands accordingly to derive a QM. Thus, it endeavours to minimise quantisation error for each coefficient band and improve output quality by adaptively producing the QM according to the content of the frame. A new QM is
generated for each WZF; thus, it implicitly adapts to changes in scene characteristics and derives the most suitable QM for the frame.

The number of *Available Bit-Planes* (ABP) is first determined at the beginning of CAQ mechanism. In a bandwidth limited scenario, it can be computed from the bandwidth and correlation noise model information. The process is similar to the rate estimation methods of encoder rate control based DVC schemes (Hu et al., 2018; Chen et al., 2017; Brites and Pereira, 2011). The ABP can also be pre-determined in scenarios with no bandwidth constraint. The latter case is broadly analogous to the existing quantisation mechanism where a predetermined QM is selected prior to coding of a video sequence in order to achieve a desired output quality and bit-rate.

To introduce the QM derivation process of CAQ mechanism, let us consider the state where DCT has been applied to a WZF and the coefficients are arranged into bands. If no quantisation is applied, i.e. DCT coefficient bands are directly converted into bit-planes, then the number of bit-planes generated from each band depends on the dynamic range of the coefficients within that band. More specifically, if the maximum magnitude among all coefficients in the DC coefficient band is, say 493 (binary: 1 1110 1101), it would require 9 bit-planes. Adding more bit-planes would be redundant. AC coefficient bands require an additional bit-plane to indicate the sign of each coefficient since AC coefficients can have positive as well as negative values. Therefore, If the maximum magnitude in an AC coefficient band is 493, it would require 10 bit-planes. This way the number of *Required Bit-Planes* (RBP) for all coefficient bands can be calculated.

If sufficient bandwidth is available, i.e. ABP is greater than RBP, then no quantisation is necessary and bit-planes can be extracted directly from the coefficient bands.
However, if ABP is less than RBP, then quantisation is needed to reduce RBP. This can be done by scaling down each coefficient by a *Scaling Factor* (SF). The value of SF is in the range \([0,1]\) and after scaling, magnitude of all coefficients will decrease and the maximum magnitude will be reduced consequently. For example, if 0.25 is used as SF, the maximum value of 493 from the previous example will be reduced to 123, which requires 7 bit-planes and 8 bit-planes for DC and AC coefficient bands respectively. This way, an SF can be used to effectively reduce number of bit-planes required for each coefficient band. As a result, overall RBP will also reduce.

Now the problem of finding appropriate QM can be reformulated to the one of finding appropriate SF such that the total RBP for all coefficient bands is less than the ABP. Naturally, finding the maximum value for the RBP is desired in order to reduce quantisation error and resulting distortion. Since RBP is a non-decreasing function of SF, the binary search algorithm (Cormen et al., 2009, p. 39) can be employed to efficiently find the most appropriate SF. The SF can be applied to each coefficient to reduce corresponding RBP and a QM can be generated by allocating the reduced RBP to all coefficient bands. The CAQ algorithm facilitates these observations to find the best QM effectively and efficiently. The detailed description of the CAQ algorithm is given next.

### 4.3.1 CAQ algorithm

To illustrate the CAQ mechanism, consider the scenario where \(N\) coefficient bands are formed from the DCT coefficients, with each band containing \(L\) coefficients. \(N\) depends on the DCT block size, while \(L\) can be derived from \(N\) and the spatial resolution of the sequence as illustrated in Table 4.2. For example, to encode a CIF sequence with 4×4 DCT block sizes, 16 coefficient bands, each containing 6336 coefficients will be formed i.e., \(N = 16\) and \(L = 6336\). If we use a different spatial
resolution sequence or a different DCT block size, these values will change accordingly as will be revealed in Section 5.2. More generally, if \(dct_{ij}\) is a DCT coefficient in coefficient band \(j\), and \(M_j\) is the corresponding number of bit-planes for this band, then \(j\) will have values in the range \([1, N]\) and the corresponding range for \(i\) will be \([1, L]\). QM can now be framed in terms of the number of bit-planes allocated to each coefficient band, i.e.,

\[
QM = \{M_1, M_2, ..., M_N\} \tag{4.1}
\]

The maximum magnitude in the coefficient band \(j\) is denoted by \(\text{MaxVal}_j\), and given by:

\[
\text{MaxVal}_j = \max_{1 \leq i \leq L} |dct_{ij}|
\tag{4.2}
\]

And the number of required bit-plane, \(\text{rbp}_j\) is computed using:

\[
\text{rbp}_j = \begin{cases} 
\lceil \log_2 \text{MaxVal}_j \rceil, & j = 1 \text{ (DC)} \\
\lceil \log_2 \text{MaxVal}_j \rceil + 1, & j > 1 \text{ (AC)} 
\end{cases}
\tag{4.3}
\]

Note that an additional bit-plane is required for AC coefficient bands to represent the sign of each coefficient. DC coefficients do not need the sign-bit since they take only positive values.

The overall RBP is calculated by aggregating \(\text{rbp}_j\) from all coefficient bands:

\[
\text{RBP} = \sum_{j=1}^{N} \text{rbp}_j
\tag{4.4}
\]

The appropriate value for the SF, \(s\) is found by employing the binary search algorithm which will be used to lower the RBP below ABP. The upper bound, \(s_u\) is set to 1.0 and the lower bound, \(s_l\) is set to 0 initially. The mid-point between \(s_u\) and \(s_l\) is initially
assigned to $s$, i.e. $s = (s_u + s_l)/2$ and $\text{MaxVal}_j$ for each coefficient band is scaled accordingly. $\text{rbp}_j$ is calculated from the scaled $\text{MaxVal}_j$ for each coefficient band using:

$$\text{rbp}_j = \begin{cases} 
\lceil \log_2 (s.\text{MaxVal}_j) \rceil, & j = 1 \text{ (DC)} \\
\lceil \log_2 (s.\text{MaxVal}_j) \rceil + 1, & j > 1 \text{ (AC)}
\end{cases}$$

(4.5)

The overall RBP is then calculated using (4.4). If this value is greater than the ABP, then $s_u$ is decreased and set to the current value of $s$ else $s_l$ is increased and set to the current value of $s$. This process is repeated until the stopping condition is met.

After the binary search algorithm finishes, the latest value of $s$ is used to scale the $\text{MaxVal}_j$ for all coefficient bands before calculating $\text{rbp}_j$ using (4.5). This value is then assigned to the corresponding element of the QM, i.e.

$$M_j = \text{rbp}_j$$

The complete CAQ algorithm for finding a QM is presented in pseudo-code format in Algorithm 4.1. Steps 1–13 represents the binary search algorithm to find appropriate SF while steps 14–17 derives the QM. Specifically, a new SF is calculated at the start of each iteration from the upper and lower bounds at Step 3. The required number of bit-planes for a band is computed from the scaled maximum magnitudes of a coefficient band at Step 5. This is repeated for each coefficient band before calculating the overall RBP value at Step 7. This is then compared to the ABP value and the upper or lower bound is adjusted accordingly at Steps 8–12. After the stopping condition is satisfied, the CAQ algorithm then proceeds to allocating bit-planes to each coefficient band. It uses the final value of SF after binary search to find required number of bit-plane for a coefficient band at Step 15 and allocates the value to the corresponding
QM entry at Step 16. After every coefficient band is processed, the resulting QM is considered as the output of the CAQ algorithm.

Algorithm 4.1: Customised QM from a given number of available bit-planes

<table>
<thead>
<tr>
<th>Input:</th>
<th>Number of available bit-planes ABP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maximum magnitudes for each coefficient band, ( \text{MaxVal}_j )</td>
</tr>
<tr>
<td>Output:</td>
<td>Quantisation matrix ( \text{QM} = { M_1, M_2, M_3, ..., M_N } )</td>
</tr>
</tbody>
</table>

1: \( s_u \leftarrow 1.0, s_l \leftarrow 0 \) 
2: \( \text{while} \) stopping condition is not met 
3: \( s \leftarrow (s_u + s_l)/2 \) 
4: \( \text{for} \ j \leftarrow 1 \ \text{to} \ N \ \text{do} \) 
5: \( \text{Calculate} \ \text{rbp}_j \ \text{using} \ (4.5) \) 
6: \( \text{end for} \) 
7: \( \text{Calculate} \ \text{RBP} \ \text{using} \ (4.4) \) 
8: \( \text{if} \ \text{RBP} > \text{ABP} \ \text{then} \) 
9: \( s_u \leftarrow s \) 
10: \( \text{else} \) 
11: \( s_l \leftarrow s \) 
12: \( \text{end if} \) 
13: \( \text{end while} \) 
14: \( \text{for} \ j \leftarrow 1 \ \text{to} \ N \ \text{do} \) 
15: \( \text{Calculate} \ \text{rbp}_j \ \text{using} \ (4.5) \) 
16: \( M_j \leftarrow \text{rbp}_j \) \( \ \text{// allocate bit-plane} \) 
17: \( \text{end for} \)

A stopping condition is necessary for successful termination of Algorithm 4.1. The stopping condition for Algorithm 4.1 is set such that it is satisfied when RBP becomes equal to ABP. An additional condition is needed to prevent probable infinite-loop circumstances. This can happen when two or more coefficient bands have equal maximum magnitude, as a result a single change in the RBP for such a band translates into multiple changes in the overall RBP. In this case RBP may never be equal to ABP for any value of the SF. To detect and avoid such scenario, RBP value of last few iterations are recorded and compared. If it does not change for a predefined number of iterations, then the stopping condition is considered satisfied and the binary search terminates. The algorithm then proceeds to allocating bit-planes according to RBP for each coefficient band.

The QM generated by the CAQ algorithm is used for quantisation of DCT coefficients. It is important to note that the SF is discarded and only the generated QM is considered
as the output of the algorithm. The DCT coefficient are not scaled, rather the quantisation operation is performed on the original DCT coefficients in the same way as fixed QM approach of the DISCOVER codec, using the generated QM instead of a predetermined QM.

To demonstrate the procedure of the CAQ algorithm presented in Algorithm 4.1 let us consider a scenario where we will compute a QM for a WZF. For this example, we use Frame #42 from the Foreman CIF sequence and ABP value of 45. The choice of the sequence, frame and ABP value is not crucial for this example and primarily chosen to highlight the differences between CAQ and fixed QM approach. We apply 4×4 DCT to get 16 coefficient band, each having 6336 coefficients. Figure 4.2 shows the maximum magnitude for each coefficient band for the frame.

<table>
<thead>
<tr>
<th>977</th>
<th>236</th>
<th>179</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>281</td>
<td>115</td>
<td>127</td>
<td>39</td>
</tr>
<tr>
<td>122</td>
<td>91</td>
<td>62</td>
<td>36</td>
</tr>
<tr>
<td>45</td>
<td>23</td>
<td>17</td>
<td>11</td>
</tr>
</tbody>
</table>

Figure 4.2: Maximum magnitudes of coefficient bands for Frame #42 of Foreman CIF sequence

We then follow the steps in Algorithm 4.1. The appropriate SF is determined using binary search algorithm following Steps 1–13 which are illustrated in Table 4.3. The initial value of the SF results in an RBP larger than the given ABP. The algorithm therefore adjusts the upper bound and finds a smaller value for the SF. This process continues until the 11th iteration when the RBP becomes equal to the ABP, therefore, the stopping condition is satisfied and the while loop at Step 1–13 terminates. The algorithm proceeds with the final value of SF, \( s = 0.0275879 \) and allocates bit-planes to coefficient bands at Step 14–17. The resulting QM is given in
Figure 4.3 along with QM$_6$ of the DISCOVER codec for comparison since both QM has 45 bit-planes.

Table 4.3: Step by step procedure of Algorithm 4.1 for frame #42 of Foreman CIF sequence

<table>
<thead>
<tr>
<th>Iteration $s_l$</th>
<th>$s_u$</th>
<th>$s$</th>
<th>MaxVal</th>
<th>rbp$_j$</th>
<th>RBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.0</td>
<td>1.0</td>
<td>0.5</td>
<td></td>
<td></td>
<td>107</td>
</tr>
<tr>
<td>2 0.0</td>
<td>0.25</td>
<td>0.125</td>
<td>4.3</td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>3 0.0</td>
<td>0.125</td>
<td>0.0625</td>
<td>5.3</td>
<td></td>
<td>62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 0.0273438</td>
<td>0.0283203</td>
<td>0.027832</td>
<td>4.4</td>
<td></td>
<td>46</td>
</tr>
<tr>
<td>11 0.0273438</td>
<td>0.027832</td>
<td>0.0275879</td>
<td>4.3</td>
<td></td>
<td>45</td>
</tr>
</tbody>
</table>

Figure 4.3: (a) QM$_6$ and (b) QM generated by the CAQ algorithm for frame #42 of Foreman CIF sequence

One of the major drawbacks of the quantisation mechanism of the DISCOVER codec as outlined in Section 4.2 is that all predetermined QM are symmetric, while DCT
coefficient distribution is often asymmetric. Therefore, some coefficient bands can be over-quantised, leading to increased quantisation error. On the other hand, CAQ avoids this issue by scaling all coefficient bands uniformly using the SF before allocating the required number of bit-planes, thus coefficient bands with greater maximum magnitude are allocated more bit-planes than bands with lesser maximum magnitude. This is apparent in Figure 4.3 as the CAQ algorithm generates an asymmetric QM by allocating more bit-planes to coefficient bands with greater maximum magnitude. As a result, overall quantisation error is minimised.

4.3.2 Enhancement-I: Exploiting human visual system in CAQ

Algorithm 4.1 treats all DCT coefficient bands equally and minimises the overall quantisation error in the transform domain. However, human eye is more sensitive to errors at lower frequencies than higher frequencies, so by giving equal weight to all coefficient bands, CAQ may allocate some bit-planes to higher frequency bands that have greater maximum magnitudes than a lower frequency band. It is thus possible for coefficient bands to have lower quantisation error in the transform domain, while the corresponding decoded output has more visible artefacts.

This uneven sensitivity reflects the way the HVS operates and so in order to make the DCT coefficients perceptually equal, a set of weights are applied to the different DCT frequency components, in an analogous manner to the quantisation tables (Q-tables) employed in JPEG (Wallace, 1992). JPEG however, uses 8×8 DCT block sizes so these Q-tables are not applicable to the 4×4 block sizes used in the DISCOVER codec. To incorporate the HVS into CAQ, a customised weighting matrix for 4×4 block sizes has been derived based on (Wang et al., 2001), and this is displayed in Figure 4.4.
In CAQ, the maximum magnitude in each coefficient band is adjusted according to the HVS weight matrix, by scaling it by the corresponding weight in Figure 4.4 prior to applying Algorithm 4.1. The enhancement process can be expressed mathematically, as follow.

Let \( HVS = \{ w_1, w_2, w_3, \ldots, w_N \} \) denote the HVS weight matrix where each element \( w_j \) represents the weight corresponding to coefficient band \( j \). Then the adjustment is done according to the following equation:

\[
\text{MaxVal}^{\text{hws}}_j = w_j \cdot \text{MaxVal}_j
\]  

These weighted maximum magnitudes, \( \text{MaxVal}^{\text{hws}}_j \) are then passed to Algorithm 4.1 as input instead of original maximum magnitudes, \( \text{MaxVal}_j \). As a result, calculation of \( \text{rbp}_j \) is affected at Step 5 and Step 15. Although the algorithm itself is unaware of the HVS weighting, the actual formula to determine \( \text{rbp}_j \) in (4.5) essentially becomes:

\[
\text{rbp}_j = \left\{ \begin{array}{ll} 
\left\lceil \log_2 \left( s \cdot \text{MaxVal}^{\text{hws}}_j \right) \right\rceil, & j = 1 \text{ (DC)} \\
\left\lceil \log_2 \left( s \cdot \text{MaxVal}^{\text{hws}}_j \right) \right\rceil + 1, & j > 1 \text{ (AC)} 
\end{array} \right.
\]  

Figure 4.5 illustrates the HVS-weighted maximum magnitudes and resulting QM generated by Algorithm 4.1 for the selected frame in the aforementioned example.
The original magnitudes from Figure 4.2 is also presented for convenient comparison.

The difference between the two QM before and after applying this enhancement is apparent from Figure 4.3 (b) and Figure 4.5 (c).

<table>
<thead>
<tr>
<th>977</th>
<th>236</th>
<th>179</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>281</td>
<td>115</td>
<td>127</td>
<td>39</td>
</tr>
<tr>
<td>122</td>
<td>91</td>
<td>62</td>
<td>36</td>
</tr>
<tr>
<td>45</td>
<td>23</td>
<td>17</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>977</th>
<th>236</th>
<th>172</th>
<th>51</th>
</tr>
</thead>
<tbody>
<tr>
<td>281</td>
<td>110</td>
<td>105</td>
<td>26</td>
</tr>
<tr>
<td>117</td>
<td>74</td>
<td>34</td>
<td>15</td>
</tr>
<tr>
<td>34</td>
<td>15</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>5</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4.5: (a) original maximum magnitudes, (b) HVS-weighted maximum magnitudes and (c) resulting QM generated by the CAQ algorithm for frame #42 of Foreman CIF sequence after Enhancement-I

The Enhancement-I attempts to reduce perceptually significant quantisation error by pre-processing the DCT coefficients prior to applying the CAQ algorithm. It is designed as an extension of the CAQ algorithm, therefore affects only the generation of a QM. It does not influence the actual quantisation operation which is performed on original un-weighted DCT coefficient bands. This separation of QM generation and quantisation operation provides the opportunity for additional processing of the DCT coefficients germane to an application or domain.

4.3.3 Enhancement-II: Maximising the QM

An opportunity to further maximise the derived QM exists due to the particular quantisation method used in the DISCOVER framework which is the basis of the DVC-HR framework. In the DISCOVER framework, the AC coefficients are quantised using a dead-zone quantiser where the quantisation step size is doubled for the 0th quantisation step. Therefore, in the case of only a single bit-plane being allocated to an AC coefficient band in the generated QM, the widened quantisation step covers the whole range and every coefficient is quantised to zero as a result.
Therefore, allocating a single bit-plane to an AC coefficient band is redundant. Specifically, all ‘1’s corresponding to AC coefficient bands in the generated QM can be replaced with ‘0’. Since the DC coefficient is quantised using a uniform scalar quantiser, this enhancement cannot be applied to the DC coefficient band. To demonstrate this enhancement, the generated QM in Figure 4.5 is maximised as shown in Figure 4.6.

![QM](image)

Figure 4.6: QM derived at different stages of the CAQ algorithm (a) original, (b) Enhancement-I only and (c) both Enhancements I & II for Foreman CIF sequence Frame #42

The QM in Figure 4.6 (c) has 4 redundant bit-planes removed, as a result, although a value of 45 as ABP was passed to Algorithm 4.1, the actual number of bit-planes transmitted to the decoder becomes 41. The obvious effect of this enhancement is a decrease in the bit-rate without affecting output quality due to avoiding transmission of redundant bit-planes.

Since Enhancement-II removes some bit-planes from the derived QM, the QM may become identical to a QM derived using another ABP value. This phenomenon is more probable if ABP is very low. Since a few bit-planes are available, most higher frequency bands will be allocated a single bit-plane or less which will be subsequently removed by Enhancement-II. In such scenario, increasing ABP will not result in greater bandwidth usage or superior output quality.
The effect of *Enhancement-II* on the RD performance is difficult to evaluate directly. Since the bit-planes removed by this enhancement are comprised of only 0, the LDPCA decoder usually successfully decodes on the first trial using initially transmitted parity bits and no additional transmission is necessary. However, number of parity bits initially transmitted depends on the correlation noise model. If probability of noise is high, i.e. SI quality is low, then more parity bits are transmitted during initial transmission. On the other hand, only a few parity bits are transmitted initially if noise probability is very low. Therefore, actual bit-rate savings by *Enhancement-II* depends on the sequence being coded.

Another important property of the CAQ algorithm is that it can be used with any DCT block size to produce a suitable QM. In contrast, the pre-determined QM of the DISCOVER codec are only suitable for 4×4 block size and cannot be used with any other DCT block size. Thus, CAQ can be utilised to produce suitable QM for larger block sizes which is investigated in Chapter 5. The *Enhancement-I* needs to be adjusted accordingly by determining appropriate weights for the new block size. However, the *Enhancement-II* is not affected by the block size and can be readily incorporated.

The CAQ algorithm along with *Enhancement-I* and *Enhancement-II* forms the CAQ module. *Enhancement-I* acts as pre-processing while *Enhancement-II* post-processes the output of the CAQ algorithm. The relationship among *Enhancements I, II* and the CAQ algorithm is shown in Figure 4.7.
The CAQ module generates a QM for every WZF, thus seamlessly adapts to varying scene characteristics. The ABP value can be changed for each WZF, therefore it can also adapt to changing bandwidth limitations and match different QoS requirements. More importantly, since ABP can have a wide range of values, it is much more flexible compared to the DISCVOER codec which has only 8 QM options. This also makes it easier to control the output quality and bit-rate by changing the ABP value. Additionally, QM generated by two consecutive ABP values differ in a single bit-plane, the transition from one ABP to the next is much smoother and more uniform than a transition from one QM to the next in the DISCOVER codec.

We propose to integrate the CAQ module at the DVC-HR decoder as shown in Figure 4.8. Although the WZF is not available at the decoder, the maximum magnitudes for all coefficient bands are transmitted to the decoder which can be used to generate the QM for the WZF. Alternatively, the CAQ module can work with DCT coefficient bands of the SI. The generated QM is then transmitted back to the encoder through the feedback channel.

This design has several advantages over its alternative of integrating the CAQ module at the DVC encoder, while its drawbacks are limited. Firstly, the CAQ algorithm is simple and computationally efficient and generates a QM for every WZF. It requires maximum magnitudes of coefficient bands and bandwidth information in terms of
ABP. While maximum magnitudes are available at both the encoder and the decoder, determining ABP can be impractical for the encoder, especially in bandwidth constrained scenarios.

Secondly, this design avoids adding processing overhead to the encoder. The generated QM is transmitted back to the encoder via the feedback channel which is used by the quantisation module. Since the QM is represented in terms of number of bit-planes allocated to each coefficient band, each entry can be represented using 4 bits. Therefore, to transmit a QM for $4 \times 4$ block size, only 64 bits are necessary. The transmission overhead of the feedback channel is thus negligible.

Finally, CAQ is incorporated as a separate module at the DVC-HR decoder and does not replace any existing module. Therefore, it can be seamlessly transferred to any state-of-the-art DVC framework derived from the DISCOVER codec. Moreover, the CAQ module can be integrated in both monoview and multiview DVC architectures to extend their functionality. The critical evaluation of qualitative and quantitative performance of the CAQ module is presented next.
4.4 Critical evaluation of CAQ performance

The CAQ module has been implemented in the testbed software as a part of the DVC-HR decoder. A series of experiments have been conducted to critically assess RD performance of the DVC-HR framework containing the CAQ module in bandwidth constrained scenarios. The rate-control flexibility enabled by CAQ is also analysed in terms of the number of available RD points, and uniformity between successive RD points compared to existing quantisation mechanism.

To critically evaluate the comparative CAQ performance, a set of diverse test sequences has been encoded by both the DVC-HR framework, and the simulated DISCOVER codec. The test sequences exhibit a variety of challenging features including different types of motion, multiple objects, textures and occlusions. While QCIF sequences are commonly used for DVC testing, higher resolution CIF sequences have instead been applied to appraise CAQ robustness. All sequences used GOP size of 2, i.e. alternate KF and WZF, and the corresponding bit-rate and PSNR performance measured. In addition, since CAQ has no RD performance impact on KF, only results for different quantisation methods for the WZF are analysed in this discussion. The test sequences, simulation criteria and measurement techniques have been discussed in broader detail in Chapter 3.

4.4.1 Rate-control flexibility

Figure 4.9 displays the comparative RD performance of the DVC-HR employing CAQ to generate QM (denoted as CAQ) and the simulated DISCOVER codec using predetermined QM (denoted as DISCOVER) for the selected test sequences. Both frameworks have been simulated using the testbed under the same test conditions and all elements of the encoding/decoding processes except the CAQ module are kept identical to ensure an equitable comparison. The graphs show disconnected RD points
instead of the commonly used line graph format where two consecutive RD points are connected through a straight line. This is to showcase the limitations of fixed QM quantisation method of the DISCOVER codec while highlight the superior rate-control flexibility of the CAQ mechanism.

The erratic RD progression of DISCOVER between QM is evident from these graphs. There are only 8 RD points available, but these points are also not uniformly spaced. The gaps between the first three QM and that between QM\(_4\) and QM\(_5\) are very small. There are comparatively bigger gaps between QM\(_3\) and QM\(_4\) and between QM\(_5\), QM\(_6\)
and QM\textsubscript{7}. However, the last QM is substantially further apart from all other QM, whose bit-rate is almost double of the bit-rate produced by the QM\textsubscript{7}. This characteristic can be observed for all test sequences. In contrast, the RD points produced by CAQ is almost uniformly distributed over the bandwidth range. Therefore, it is much more convenient to adjust bit-rate or output quality by changing ABP value. The transition from one RD point to the next will also be smoother than DISCOVER.

While comparing in terms of RD performance, both quantisation mechanism produces similar output quality and bit-rates. For every RD points produced by DISCOVER, an RD point can be found from the output of CAQ with similar performance. These RD points might seem to be superimposed on each other at times for several test sequences. CAQ does improve RD performance when comparing to DISCOVER for Mother & Daughter sequence where up to 0.5dB gain can be observed. However, this is not reflected for other test sequences.

4.4.2 Bandwidth utilisation

The flexible rate-control characteristic of CAQ can lead to better bandwidth utilisation in bandwidth constrained scenarios. Since RD points generated by CAQ are near uniformly distributed over the bandwidth range, an RD point closer to the bandwidth limit can be easily chosen. On the other hand, it is much more unlikely to find a pre-determined QM whose RD point is close to the bandwidth limit. The situation worsens if the bandwidth limit changes during coding. Moreover, bit-rate requirement of a QM largely depends on the sequence being coded. This can be seen from Figure 4.9 where the bit-rate of QM\textsubscript{7} in Subfigure (f) is greater than the bit-rate of QM\textsubscript{8} in Subfigure (d). Therefore, if scene characteristics change, bit-rate requirement will also change which may lead to bandwidth wastage or overflow. Since a new QM is generated for every
WZF in CAQ, bit-rate requirement can be easily adjusted according to changing scene characteristics.

![Figure 4.10: Bandwidth utilisation comparison of CAQ and pre-determined QM quantisation mechanism of DISCOVER](image)

To demonstrate superior bandwidth utilisation of the CAQ mechanism, several hypothetical bandwidth limits are assumed and RD points closest to these limits are selected from both CAQ and DISCOVER. The selection procedure is illustrated in Figure 4.10 using the RD point graph of *Football* sequence. The solid vertical line at 1600kbps denotes the hypothetical bandwidth limit. Among the pre-determined QM, QM\textsubscript{1}–QM\textsubscript{6} have bit-rates lower than the bandwidth limit while QM\textsubscript{7} and QM\textsubscript{8} (not shown in the graph) require higher bit-rates, therefore cannot be selected. As a result, QM\textsubscript{6} has been selected since it has the best performance among the QM with lower bit-rate requirement than the bandwidth limit. Similarly, RD point of the QM generated by the CAQ algorithm using ABP value of 46 has been selected. By comparing these two RD points, it can be observed that CAQ exhibits substantially superior bandwidth utilisation which results in improved output quality with significantly higher PSNR compared to DISCOVER.
Table 4.4 summarises the bandwidth utilisation results for various test sequences encoded by the two quantisation mechanisms for several hypothetical bandwidth limits. Each entry in the table has been obtained by following the procedure stated above. The bit-rate requirements are shown as a percentage of the bandwidth limit. Corresponding PSNR values are also presented.

Table 4.4: Bandwidth utilisation statistics for CAQ and DISCOVER quantisation methods

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Quantisation Method</th>
<th>200 kbps</th>
<th>400 kbps</th>
<th>800 kbps</th>
<th>1600 kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bit-rate (%)</td>
<td>PSNR (dB)</td>
<td>Bit-rate (%)</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>Football</td>
<td>CAQ</td>
<td>95.41</td>
<td>26.22</td>
<td>98.24</td>
<td>27.82</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td></td>
<td></td>
<td>92.82</td>
<td>27.11</td>
</tr>
<tr>
<td>Soccer</td>
<td>CAQ</td>
<td>100</td>
<td>30.43</td>
<td>92.69</td>
<td>31.95</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td>86.17</td>
<td>27.01</td>
<td>93.97</td>
<td>26.68</td>
</tr>
<tr>
<td>Bus</td>
<td>CAQ</td>
<td>95.96</td>
<td>34.10</td>
<td>98.20</td>
<td>35.11</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td>87.77</td>
<td>34.10</td>
<td>97.35</td>
<td>34.74</td>
</tr>
<tr>
<td>Foreman</td>
<td>CAQ</td>
<td>91.08</td>
<td>33.15</td>
<td>96.79</td>
<td>34.07</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td>78.85</td>
<td>33.11</td>
<td>62.05</td>
<td>33.70</td>
</tr>
<tr>
<td>Coastguard</td>
<td>CAQ</td>
<td>92.46</td>
<td>42.53</td>
<td>96.95</td>
<td>43.43</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td>81.67</td>
<td>42.51</td>
<td>84.92</td>
<td>42.90</td>
</tr>
<tr>
<td>Hall</td>
<td>CAQ</td>
<td>93.23</td>
<td>36.90</td>
<td>95.53</td>
<td>38.76</td>
</tr>
<tr>
<td></td>
<td>DISCOVER</td>
<td>99.40</td>
<td>37.12</td>
<td>92.57</td>
<td>37.80</td>
</tr>
</tbody>
</table>

The Table 4.4 results consistently reveal superior bit-rate control of CAQ compared to the DISCOVER, with more than 90% of the available bandwidth capacity utilised in all scenarios except for Bus at 200kbps limit displaying 86% utilisation. Conversely, bandwidth utilisation for DISCOVER is inconsistent, so while it uses ≈99% of the 200kbps capacity for Hall, this drops to only 62% for the 400kbps bound for Coastguard. In some cases, DISCOVER does surpass bandwidth utilisation of CAQ by a slight margin. For example, bandwidth utilisation of DISCOVER is 6% more than CAQ for Hall sequence at 200kbps. However, as discussed above, these are random occurrences and cannot be reliably reciprocated for other combinations of test sequences and bandwidth limits.
The superior bandwidth utilisation of CAQ leads to better output quality measured by PSNR gain. In some cases, this PSNR gain can be considerably large, for example, CAQ utilises 24% more bandwidth than DISCOVER and gains 1.83dB PSNR for *Football* sequence at 1600kbps limit. This can also be observed for other test sequences and bandwidth limit combinations, for example, *Football* at 800kbps, *Bus* at 1600kbps, *Foreman* at 400kbps and 1600kbps, *Coastguard* at 200kbps, 400kbps and 1600kbps, *Mother & Daughter* at 200kbps and 400kbps and *Hall* at 1600kbps bandwidth limits. In some rare cases, despite using more bandwidth, CAQ failed to gain PSNR with respect to DISCOVER. This can be seen from Table 4.4 for *Foreman* sequence at 200kbps limit where CAQ uses 8% more bandwidth than DISCOVER but produces equal output quality. In some other cases however, CAQ produces better output quality while occupying less bandwidth than DISCOVER. An example of such cases can be seen for *Mother & Daughter* sequence at 1600kbps limit.

A significant advantage of CAQ over DISCOVER becomes apparent while analysing the data in Table 4.4 at 200kbps limit. For *Bus*, *Football* and *Soccer* sequences, the lowest RD point produced by QM₁ from the pre-determined QM of DISCOVER is actually greater than the bandwidth limit. As a result, the fixed QM method of DISCOVER fails to produce any output in this scenario. CAQ not only successfully produces output at such low bit-rate, it can produce output even if the bandwidth limit were lower, as can be seen from the RD point graphs in Figure 4.9. This is also true for very high bandwidth limit scenarios where CAQ can produce better output at bit-rates beyond that of QM₈.

### 4.4.3 Adaptation to scene changes

The RD points for CAQ in Figure 4.9 have been produced using a fixed ABP value for all frames within a sequence. This has been done to make equitable comparison
with fixed QM method of the DISCVOER codec where a QM is selected at the beginning and fixed throughout coding of the whole sequence. Although the design allows using a different ABP for every sequence, this feature has not been utilised in our experiments. Therefore, the results presented so far does not adapt to changes in scene characteristics.

CAQ can adapt to scene changes by changing the ABP value. However, finding appropriate ABP value is a complex task. Firstly, the decoder needs the bandwidth capacity information. It then has to accurately estimate number of parity bits required for LDPCA decoding of each bit-plane from the correlation noise model. The correlation noise model itself needs to reflect the scene changes properly. The length of each bit-plane can be derived from spatial resolution of the sequence and DCT block size as shown in Section 4.3. The ABP can be derived from the length of bit-planes, corresponding estimated compression ratio, and available bandwidth information. These tasks, especially correlation noise model estimation and estimation of number of required parity bits for each bit-plane are beyond the scope of this work. However, to demonstrate the potential of this feature, we provide a simulation of adapting to scene changes by changing the ABP value.

For this demonstration, a 1600kbps bandwidth limit is assumed and the Foreman CIF sequence is selected for simulation. This sequence has mixed motion activity across the frames. Initially the scene is focused on the face of a person (foreman) talking towards the camera. There is very low motion activity at this point. Then scene changes occur as the camera pans towards a building under construction. The sequence shows very rapid motion activity. Due to the varying motion activity present in this sequence, it is an ideal choice for this demonstration purpose. An additional sequence, namely Soccer is also selected to evaluate the robustness of this simulation.
From the bandwidth limit and frame rate information, bits available for each frame can be determined. Since both Forman and Soccer sequences have a frame rate of 30fps, there are 15 WZF each second for GOP 2. Therefore, for each frame we have

\[
\frac{1600 \text{kilo-bits-per-second}}{15 \text{frames-per-second}} = 106666.67 \text{ bits-per-frame}
\]

Therefore, 106,666 bits are available for each frame. Now, to simulate the adaptive ABP feature, for each frame, we select the maximum ABP value for which the number of bits is less than the available quantity. The selection procedure is shown in Figure 4.11 for the first few frames of Foreman sequence. For each WZF the result of the selected ABP value is highlighted.

![Figure 4.11: Simulation of adaptive ABP for Foreman CIF sequence](image)

The best ABP value has been selected in this way for every WZF. Afterwards, the bit-rate and average PSNR are calculated using appropriate formulae. These measurements are then compared to corresponding measurements of fixed ABP value having bit-rate requirement close to the bandwidth limit of 1600kbps. Additionally, Standard Deviation (SD) of bits-per-frame is also calculated to demonstrate stability of the adaptive ABP simulation. The measurements are presented for comparison in Table 4.5.

![Table 4.5: Performance comparison of fixed ABP and adaptive ABP simulation for Foreman CIF sequence](table)
Both bit-rate and PSNR measurements of adaptive ABP and specific ABP values very similar to each other. However, adaptive ABP produces noticeably better output at slightly less bit-rate than the output produced using ABP value of 58. However, the most contrasting difference between them is the SD measurement where the adaptive ABP exhibits nearly 10 times less deviation than the other three. This means that the bandwidth is more uniformly distributed among the frames for adaptive ABP than a fixed ABP value. Similar trend can be observed for measurements for Soccer sequence presented in Table 4.6.

Table 4.6: Performance comparison of fixed ABP and adaptive ABP simulation for Soccer CIF sequence

<table>
<thead>
<tr>
<th>ABP</th>
<th>Bit-rate (kbps)</th>
<th>PSNR (dB)</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive</td>
<td>1546.115</td>
<td>36.915</td>
<td>2785.08</td>
</tr>
<tr>
<td>49</td>
<td>1487.269</td>
<td>36.877</td>
<td>21831.29</td>
</tr>
<tr>
<td>50</td>
<td>1565.019</td>
<td>37.103</td>
<td>22114.50</td>
</tr>
<tr>
<td>51</td>
<td>1648.868</td>
<td>37.304</td>
<td>23259.92</td>
</tr>
</tbody>
</table>

To further illustrate the significance of low SD of adaptive ABP simulation, the bits consumed by each frame is plotted in a line graph for adaptive ABP as well as fixed ABP as shown in Figure 4.12 for both Foreman and Soccer sequences. The stability and robustness of adaptive ABP is evident from these graphs. The bits consumed by a frame varies significantly throughout the sequence for fixed ABP according to changes in scene characteristics. On the other hand, the consumption is steady and stable across all frames for adaptive ABP simulation regardless of the scene changes.

The adaptive ABP simulation above is performed in a bandwidth limited scenario where the bandwidth is distributed equally across the frames of a sequence. As a result, each frame requires almost equal number of bits irrespective of changes in scene characteristics. However, this causes the output quality to vary according to scene variations. In order to make the output quality stable despite variation in scene characteristics, a converse approach may be taken. Since output quality is determined
by the degree of quantisation, a simple and straightforward method would be to fix
the SF in Algorithm 4.1 and allocate as many bit-planes as necessary. This is
analogous to selecting the quality parameter in JPEG. However, this is not covered in
this thesis due to time constraints but is a future work suggestion.

Figure 4.12: Bandwidth usage of adaptive ABP and fixed ABP simulations across all frames of CIF
sequences (a) Foreman (b) Soccer
Figure 4.13: Perceptual comparison of frame #104 for *Football* decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM.

4.4.4 Qualitative evaluation

To complement the CAQ numerical analysis, a qualitative assessment was also undertaken to corroborate its enhanced performance in bandwidth constrained scenarios. Figure 4.13 shows an example frame (#104) of *Football* using both quantisation mechanisms at the 1600kbps limit, with the respective QM for each frame included. It is visually apparent that the CAQ output has significantly lower perceptual artefacts than decoded DISCOVER output. Note the asymmetric QM generated by the CAQ module leading to reduced quantisation error and superior perceptual quality. This trend has been observed for most frames throughout the sequence.
Figure 4.14: Perceptual comparison of frame #184 for Foreman decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM.

Similar comparison can be made between decoded frames of other test sequences. Figure 4.14 compares a sample frame #184 of Foreman sequence decoded by different quantisation methods at 1600kbps bandwidth limit. Some visual artefacts can be noticed, for instance, below the left eye of the foreman, in the frame decoded using a fixed QM.
Figure 4.15 compares a sample frame #90 of Foreman sequence decoded by CAQ and DISCOVER at a bandwidth limit of 800. Although PSNR measurement of both frames are very close, meaning similar amount of quantisation error, more visual artefacts are present in the frame decoded by DISCOVER. This can be observed by comparing, for instance, the face and finger regions of the decoded frame. The HVS weighting provided by Enhancement-I of the CAQ algorithm is primarily responsible.

Decoded version of a sample frame #78 of Coastguard decoded by different quantisation methods are compared in Figure 4.16 at 1600kbps. Interestingly, though PSNR gain of CAQ over DISCOVER for this particular frame is 1.92dB, it is difficult
to judge perceptually better one due to unique and complex texture of the images. Similar behaviour has been found for all frames of the sequence.

![Figure 4.16: Perceptual comparison of frame #78 for Coastguard decoded using different quantisation mechanisms at 1600kbps bandwidth limit (a) CAQ and (b) DISCOVER along with their corresponding QM](image)

In summarising, both quantitative and qualitative results confirm the key advantage of CAQ is flexible rate control, which is especially beneficial in bandwidth limited scenarios where it can more reliably utilise the available bandwidth leading to superior decoded output quality. The adaptive nature of CAQ can be particularly useful in scenarios where scene characteristics or bandwidth constraints change over time. This can be facilitated to ensure consistent bandwidth usage or stable output quality regardless of variations in scene characteristics or bandwidth limits.
4.5 Summary

CAQ is an adaptive quantisation mechanism to extend the rate-control functionality of the DISCOVER codec. It provides superior rate-control flexibility in contrast to existing pre-determined QM based quantisation mechanism. At the core of the CAQ mechanism is an efficient and effective algorithm to dynamically determine the best QM for the frame being coded in a given bandwidth circumstances. This algorithm is further enhanced to reduce perceptible artefacts due to quantisation error by incorporating HVS leading to superior output quality. The resulting QM is then maximised by removing redundant bit-planes resulting in reduced bit-rate. To avoid adding workload to the resource limited encoder, the CAQ module is incorporated into the decoder of the DVC-HR framework and produces a QM for every WZF. The QM is then transmitted to the encoder through the feedback channel incurring negligible overhead. The DVC-HR framework with the CAQ module integrated at the decoder is then simulated with several standard test sequences and its performances were measured and compared with the original codec. The rigorous analysis of both quantitative and qualitative results confirmed that CAQ offers far superior and flexible rate control options, which is especially beneficial in bandwidth limited scenarios. It more reliably utilised the available bandwidth leading to superior decoded output quality. More crucially, it successfully produced output at very low bandwidth limits where the DISCOVER codec failed to produce any output. The adaptive nature of CAQ was demonstrated through a simulation where it stably utilised available bandwidth regardless of changes in scene characteristics. Since CAQ is incorporated as a new module at the DVC-HR decoder and does not replace any existing module, it can be embedded into any current state of the art transform domain DVC framework based upon the DISCOVER codec to improve its rate-control performance in both
mono and multiview scenarios. CAQ crucially can produce a QM for any transform block size, so it can be exploited in coding larger block sizes. The DVC performance of larger DCT block size will be presented in the next chapter.
5 Investigation of larger transform block sizes for higher resolution DVC

5.1 Introduction

The motivation for investigating larger DCT block sizes within the new DVC-HR framework is that many prominent image and video codecs use the DCT to lower spatial redundancies and achieve robust compression performance (Sullivan et al., 2012; Wiegand et al., 2003; Wallace, 1992). In the traditional coding paradigm, employing larger DCT block sizes has been observed to exhibit favourable impact on the RD performance. While increasing the block size requires more computational resources, the added complexity is deemed tractable for current computing hardware and thus its benefits outweigh its drawbacks (Bossen et al., 2012). This is evident through the fidelity range extension of the AVC which introduced 8×8 block size in addition to the original 4×4 (Sullivan et al., 2004), and up to 32×32 block size configuration options available in the HEVC (Sullivan et al., 2012). In contrast, DVC frameworks available in the literature have used a block size of 4×4 pixels for the DCT operation, as confirmed in Subsection 2.5.2. It is therefore desirable to investigate the benefits of using larger DCT block sizes within a generic DVC framework.

This chapter investigates the effect and impact of applying larger DCT block sizes on DVC complexity and performance by implementing 8×8 and 16×16 block sizes within the DVC-HR framework. One of the important considerations to implementing larger block sizes is how to be able to obtain an appropriately sized QM because existing QM are only suitable for the prevailing 4×4 DCT block size. This requirement can be
fulfilled by the new CAQ module introduced in Chapter 4 which is able to effectively produce a suitable QM for any block size. The implementation however, also affects subsequent modules of the DVC-HR framework which need to be addressed, with the necessary modifications to the framework being discussed in Section 5.2. The framework is then critically analysed using a range of standard test sequences of differing spatial resolutions and scene characteristics. Rigorous evaluation of experimental results is presented in Section 5.3 along with a critical evaluation of the findings, with Section 5.4 making some concluding remarks.

5.2 Modifications of the DVC-HR framework

The DCT is employed at both the encoder and decoder of the DVC-HR framework. At the encoder, a WZF is first partitioned into non-overlapping blocks of pixels and a 2D DCT is applied to each block to generate the corresponding coefficients. These coefficients are then arranged into bands before being quantised according to a designated QM which is generated by the CAQ module and then sent to the encoder via the feedback channel as described in Section 4.3. The quantised coefficient bands are converted into bit-planes and processed by an LDPCA encoder to produce the parity bits. A similar corresponding procedure is applied at the decoder side, with the WZF being replaced by the SI and the bit-planes being fed into the LDPCA decoder. Thus, implementing larger DCT block sizes requires some modification to other framework modules which use DCT output. Since each element of the QM defines the quantiser to be used for quantisation of the corresponding coefficient band, the QM size needs to be updated according to the new block size. In addition, the length of coefficient bands and thereby the length of bit-planes are changed so that appropriately-sized LDPCA coders to process these bit-planes are required. The next two subsections will discuss the rationale for these two modifications.
5.2.1 QM modification

Quantisation is an integral part of DCT-based image and video codecs as it discards less important information to achieve better compression. Moreover, the DCT paired with suitable quantisers make it substantially easier to control the bit-rate and thus provide different QoS levels. Existing DVC solutions achieve this by a set of eight predetermined QM which are only suitable for the prevalent 4×4 block size as detailed in Subsection 2.5.2. This means a suitably-sized QM must firstly be obtained before other block sizes can be implemented. More specifically, different QM suitable for both 8×8 and 16×16 block sizes are needed. The new CAQ mechanism presented in Chapter 4 can be profitably utilised for this purpose due to its capability of producing a QM for any block size, though additional processing stages need to be introduced to complete this procedure.

Enhancement-I of the CAQ algorithm described in Subsection 4.3.2 integrates HVS to prioritise the reduction of perceptible distortion. A customised HVS weight matrix for 4×4 block size has been determined to apply appropriate perceptual weights to different coefficient bands. To generate the respective QM for the 8×8 and 16×16 DCT block sizes using CAQ, the corresponding HVS weight matrices must firstly be determined. Using the procedure proposed in (Wang et al., 2001), the two HVS weight matrices in Figure 5.1 (a) and (b) are produced.

Each DCT coefficient is first multiplied by the corresponding HVS weight value before the CAQ algorithm is applied to calculate the most suitable QM. This step ensures that less perceptible coefficients are suppressed while the more perceptible coefficients receive a greater proportion of the bandwidth. The full procedure for applying HVS weights and determining a QM is detailed in Section 4.3.
Figure 5.1: HVS weight matrices for (a) 8×8 and (b) 16×16 DCT block sizes
DCT coefficients generated by larger block sizes have wider dynamic ranges than coefficients produced by a smaller block size, so the coefficient bands can generate more bit-planes. Table 5.1 displays the dynamic ranges of both the DC and AC coefficients in terms of number of bit-planes that can be extracted from the respective bands in relation to the DCT block size. This disparity notably impacts the quantisation of the DC coefficient band because the corresponding range is assumed to be $2^{10} = 1024$ respective to the 4×4 block size shown in Table 5.1. As a consequence, a fixed quantiser is used which needs updating to accommodate the increased dynamic range. In contrast, since the quantiser for the AC coefficients is determined during quantisation based on the actual range of the corresponding band, these do not need to be updated.

<table>
<thead>
<tr>
<th>DCT block size</th>
<th>DC dynamic range</th>
<th>AC dynamic range</th>
</tr>
</thead>
<tbody>
<tr>
<td>4×4</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>8×8</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>16×16</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

5.2.2 Modification of the LDPCA coder

DCT coefficients are arranged into coefficient bands before being quantised using a QM. Bit-planes are then extracted and processed through a LDPCA coder. The length of a coefficient band and consequently, the length of a bit-plane depends on the spatial resolution and the DCT block size. For any specific resolution, increasing the DCT block size shortens the length of a bit-plane and vice-versa. The bit-plane length for different combinations of spatial resolutions and DCT block sizes is presented in Table 5.2. It is important to note that though employing a larger block size shortens the bit-plane length, the number of bit-planes increases compared to a smaller block size. More specifically, 16×16 block sizes produce a larger number of shorter bit-planes than either 8×8 or 4×4 block sizes.
Table 5.2: Bit-plane length for various spatial resolutions and DCT block sizes combinations

<table>
<thead>
<tr>
<th>DCT block size</th>
<th>QCIF</th>
<th>CIF</th>
<th>4CIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>4×4</td>
<td>1584</td>
<td>6336</td>
<td>25344</td>
</tr>
<tr>
<td>8×8</td>
<td>396</td>
<td>1584</td>
<td>6336</td>
</tr>
<tr>
<td>16×16</td>
<td>99</td>
<td>396</td>
<td>1584</td>
</tr>
</tbody>
</table>

Since the LDPCA coder requires a block of input bits, the length of a bit-plane must match the intended length of the LDPCA coder. If a bit-plane is too long, it needs to be split into appropriately sized chunks, though this increases complexity and decreases the LDPCA coding performance. If it is too short, padding with dummy bits have to be inserted which impacts the bit-rate overhead. Thus, a better strategy is to employ LDPCA coders which are suitable for desired bit-plane lengths.

To produce the LDPCA coders with desired lengths, the accompanied tools of the LDPCA coder software (Varodayan, 2005) have been used. The first step is to generate the highest rate LDPC code before progressively generating the lower rate ones. Due to the limitations of the tools, the code length must always be a multiple of 66 and the highest rate LDPCA code has a rate of 2/66, i.e. it produces 2 parity bits for every 66 input bits. The remaining codes have rates of 3/66, 4/66, and so forth up to 66/66. Since the bit-plane length of 99 corresponding to the 16×16 block size and QCIF resolution is not a multiple of 66, the tools cannot generate a LDPCA code of matching length, so 16×16 block sizes are omitted from the critical testing of QCIF sequences.

It is important to stress that the bit-plane length can affect the LDPCA coder performance. Generally, for a LDPCA coder, long bit-planes perform better than coder for shorter bit-planes (Varodayan et al., 2006), and while the performance gap is small, it can accumulate over time and become noticeable while decoding a large number of bit-planes.
The two modifications have been implemented in the DVC-HR framework testbed so larger DCT block sizes are now able to be managed. In encoding a test sequence, the framework determines the most appropriate LDPCA coder from the spatial resolution and DCT block size configurations and then uses the selected LDPCA coder throughout the coding process. It is not currently feasible to dynamically switch DCT block sizes while coding a sequence.

5.3 Results discussion

Larger DCT block sizes have been implemented in the DVC-HR framework testbed after satisfying the pre-requisites, with the specific implementation details described in Section 3.4. The framework has been extensively tested with a range of standard test sequences of different spatial resolutions and scene characteristics. QCIF, CIF and 4CIF sequences have been selected to test its performance across different spatial resolutions. The selected sequences also exhibit a range of challenging scene characteristics including object motion, global motion, occlusion, disocclusion and texture as explained in Section 3.8. During each simulation run, the framework performance is critically evaluated using several key metrics as defined in Section 3.7. To determine the RD performance, the average bit-rate and average PSNR of all WZF frame are measured for different QoS configurations and then formally presented as RD curves. Additionally, both the BD-PSNR and BD-rate metrics are applied to evaluate the impact of larger DCT block sizes with respect to the benchmark DISCOVER framework. A critical performance evaluation of larger DCT block sizes in the DVC-HR framework is presented in the next subsections using both quantitative and qualitative performance assessments.

To simulate 16×16, 8×8 and 4×4 DCT block size configurations, appropriate QM have been produced by the CAQ module. Four different RD points are necessary to produce
and compare the respective RD curves which are obtained by changing the ABP value configurations. Since larger block sizes produce more bit-planes than their smaller counterparts, the corresponding ABP values need to be increased accordingly. For the 16\times16 block size, ABP values of 375, 450, 525 and 600 have been used to generate the requisite four RD points. Similarly, ABP values of 100, 125, 150 and 175 have been used for the 8\times8 block size while for the 4\times4 block size, the corresponding ABP values were 20, 30, 40 and 48. Comparative performance results for the DISCOVER codec are also included comprising 4\times4 DCT block size and a predetermined QM, with the four necessary RD points being obtained using QM\textsubscript{1}, QM\textsubscript{3}, QM\textsubscript{5} and QM\textsubscript{7}. Table 5.3 summarises the configuration parameters used for the simulations:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>DCT block size</th>
<th>QM source</th>
<th>RD Points (QM/ABP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCOVER</td>
<td>4\times4</td>
<td>Pre-determined</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>DCT-16</td>
<td>16\times16</td>
<td>CAQ</td>
<td>375 450 525 600</td>
</tr>
<tr>
<td>DCT-8</td>
<td>8\times8</td>
<td>CAQ</td>
<td>100 125 150 175</td>
</tr>
<tr>
<td>DCT-4</td>
<td>4\times4</td>
<td>CAQ</td>
<td>20 30 40 48</td>
</tr>
</tbody>
</table>

The first column reflects the nomenclature adopted for the different test configurations used for performance comparison. DCT-16, DCT-8 and DCT-4 respectively indicate 16\times16, 8\times8 and 4\times4 DCT block size configurations of the DVC-HR framework with the QM being produced by the CAQ module, while for the DISCOVER codec, a 4\times4 DCT block size and predetermined QM are used.

The ABP values have been adjusted so that the generated RD points have similar bit-rates to enable an equitable quantitative comparison. For example, the bit-rate of each RD point produced by DCT-16 has been found to be comparable to the bit-rate of corresponding RD point produced by DCT-8 during coding and while actual bit-rates can vary across the different test sequences, this similarity is broadly maintained.
across all sequences. This will be corroborated when evaluating the RD curves produced for various test sequences in the next subsections.

5.3.1 4CIF sequences

To critically evaluate the DVC-HR framework performance at higher spatial resolutions, three different 4CIF test sequences, namely *Soccer*, *Ice*, and *Crew* have been selected and their corresponding performances measured. Figure 5.2 shows the respective RD performance curves for the three 4CIF sequences. The same nomenclature is adopted in these graphs as in Table 5.3.

![Figure 5.2: RD curves for different DCT block size configurations for (a) Crew, (b) Ice and (c) Soccer 4CIF test sequences](image)

For each of the test sequences, DCT-16 and DCT-8 employing larger 16×16 and 8×8 block sizes respectively exhibit significantly superior RD performance to DCT-4 and DISCOVER with the 4×4 block size counterpart. The performance gain is consistent across the various bit-rate ranges. DCT-16 provides slightly better performance over
DCT-8, though this improvement is more pronounced in the lower bit-rate range and converges at higher bit-rates.

In the simulations of different configurations for a specific test sequence, only the DCT block sizes are varied along with the requisite modifications to the QM and LDPCA coder. The QM determines the target bit-rate range while the DCT block size and LDPCA coder both impact the compression performance of the DVC-HR framework, so a critical analysis of the performance of latter two elements is mandated in order to better understand and interpret the Figure 5.2 RD curves.

In general, a large block size DCT exhibits better compression performance than smaller block sizes, though it then is accompanied by a shorter LDPCA coder, which as detailed in Table 5.2, leads to performance degradation with the truncated coder length. This trade-off fundamentally governs the RD performance of the DVC-HR framework. If the performance gain of using larger DCT block sizes exceeds the deficit of a shorter LDPCA coder, the framework consistently provides superior RD performance and vice versa. Following this reasoning, the gain secured by using 16×16 and 8×8 block sizes with respect to 4×4 block sizes is much more significant than any deficit incurred by using shorter LDPCA coders, so substantial RD performance gains are observed for DCT-16 and DCT-8 over DCT-4 for all the test sequences analysed.

The diminishing advantage of DCT-16 over DCT-8 at higher bit-rates can be similarly justified. A larger DCT block size produces a significantly greater number of shorter bit-planes than its smaller block size counterpart. This is evident from both the data presented in Table 5.3 and the RD curves in Figure 5.2. Larger block sizes use more ABP values than smaller equivalents to produce outputs at comparable bit-rates. A
bit-plane produced by a larger block size needs to be processed by a shorter, thus inferior LDPCA coder. At higher bit-rates, the number of bit-planes increases substantially leading to an accumulated performance deficit, which can tend to dominate the performance gain by the larger block size resulting in lower performance than the smaller block size counterpart. This is evident in Figure 5.2 (b), where DCT-16 incurs an approximate 0.9dB loss in comparison to DCT-8 at higher bit-rates.

Figure 5.2 also includes comparative RD curves for the DISCOVER codec which uses 4×4 DCT block size with predetermined QM for quantisation. During the experiments, DISCOVER exhibited nearly identical performance to DCT-4 for all test sequences, with the results confirming the findings of Chapter 4 that CAQ introduces more flexible bit-rate control whilst crucially maintaining RD performance.

To quantify the RD performance benefit of DCT-16 and DCT-8 over DISCOVER, the average PSNR gain and average bit-rate savings have been calculated using the BD-PSNR and BD-rate metrics and their values are presented in Table 5.4. It is observed that DCT-16 gained more than 2dB PSNR over the DISCOVER for Soccer at the same bit-rate and required a 42% lower bit-rate to achieve the same PSNR. DCT-8 attained the best average gain of ≈1.9dB for Ice with a bit-rate saving of 38% to achieve the same PSNR compared to DISCOVER.

<table>
<thead>
<tr>
<th>Sequences (4CIF)</th>
<th>DCT-16</th>
<th></th>
<th></th>
<th>DCT-8</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>BD-rate (%)</td>
<td>BD-PSNR (dB)</td>
<td>BD-rate (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crew</td>
<td>1.26</td>
<td>-36.88</td>
<td>0.98</td>
<td>-29.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ice</td>
<td>2.05</td>
<td>-41.23</td>
<td>1.89</td>
<td>-38.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soccer</td>
<td>2.22</td>
<td>-41.81</td>
<td>1.64</td>
<td>-33.69</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The performance of larger DCT block sizes has been critically evaluated using the RD curves which reveal a consistent and significant improvement over the predominant
4×4 block size. Of the newly implemented block sizes, DCT-16 is marginally better than DCT-8 especially at lower bit-rates, with a substantial performance gain with respect to DISCOVER being evidenced with an over 2dB average PSNR gain secured by DCT-16 for the high motion Soccer sequence.

5.3.2 CIF sequences
A number of CIF test sequences including Mother & Daughter, Crew and Foreman have been simulated to critically evaluate the performance of being able to select larger DCT block sizes in the encoding. Their respective RD curves are presented in Figure 5.3, from which it is evident that both DCT-16 and DCT-8 substantially outperform DCT-4 and DISCOVER for CIF sequences as well. In all the sequences tested, DCT-16 and DCT-8 exhibited significantly superior RD performance to DCT-4 and DISCOVER with DCT-16 being slightly superior to DCT-8.

Quantitative data of the average performance gains by DCT-16 and DCT-8 with respect to DISCOVER are presented in Table 5.5. Both DCT-16 and DCT-8 consistently achieve better average PSNR than DISCOVER, with the highest gain observed for Football with 1.68dB and 1.39dB achieved by DCT-16 and DCT-8 respectively. Conversely, they have been able to save up to 37% bit-rate in achieving the same PSNR compared to DISCOVER. Interestingly, DCT-16 and DCT-8 required a higher bit-rate for Mother & daughter despite the average PSNR gain. This is due to an inherent limitation in the BD-rate metric whereas the RD curves are assumed to be logarithmic, this is not valid for this particular sequence as evidenced in Figure 5.3 (e) which leads to an inconsistency. This will be investigated more depth along with a similar anomaly for Mother & daughter QCIF sequence later this section.
Figure 5.3: RD curves for different DCT block size configurations for (a) Coastguard, (b) Crew, (c) Football, (d) Foreman, (e) Mother & Daughter and (f) Soccer CIF test sequences

Table 5.5: Average performance gain by DCT-16 and DCT-8 with respect to DISCOVER for CIF sequences

<table>
<thead>
<tr>
<th>Sequences (CIF)</th>
<th>DCT-16</th>
<th>DCT-8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD-PSNR (dB)</td>
<td>BD-rate (%)</td>
</tr>
<tr>
<td><strong>Bus</strong></td>
<td>1.40</td>
<td>-33.02</td>
</tr>
<tr>
<td><strong>Coastguard</strong></td>
<td>0.94</td>
<td>-32.17</td>
</tr>
<tr>
<td><strong>Crew</strong></td>
<td>0.89</td>
<td>-23.24</td>
</tr>
<tr>
<td><strong>Football</strong></td>
<td>1.68</td>
<td>-28.32</td>
</tr>
<tr>
<td><strong>Foreman</strong></td>
<td>1.00</td>
<td>-33.11</td>
</tr>
<tr>
<td><strong>Hall</strong></td>
<td>0.81</td>
<td>-37.12</td>
</tr>
<tr>
<td><strong>Ice</strong></td>
<td>1.32</td>
<td>-23.56</td>
</tr>
<tr>
<td><strong>Mother &amp; daughter</strong></td>
<td>0.69</td>
<td>8.34</td>
</tr>
<tr>
<td><strong>Soccer</strong></td>
<td>1.42</td>
<td>-28.06</td>
</tr>
</tbody>
</table>
The trend of larger block sizes exhibiting superior RD performance compared with their smaller counterparts for 4CIF test sequences is maintained for CIF test sequences as well. Both DCT-16 and DCT-8 show significant performance gain over DCT-4 and DISCOVER with DCT-16 slightly outperforming DCT-8. Comparative analysis with respect to DISCOVER confirms this with up to 1.68dB average PSNR gain observed for Football when coding with a 16×16 DCT block size.

5.3.3 QCIF sequences

Similar experiments have been conducted with a number of QCIF test sequences including Coastguard, Hall and Soccer whose respective RD performance curves are presented in Figure 5.4. The RD curve for DCT-16 has been omitted since, as explained in Subsection 5.2.2, 16×16 block sizes cannot be used for QCIF sequences. A similar trend of larger block sizes consistently outperforming their smaller counterparts can be observed from these RD graphs.

The average PSNR gain and average bit-rate savings by DCT-8 with respect to DISCOVER in terms of the BD-PSNR and BD-rate metrics for various QCIF sequences are presented in Table 5.6. In all experiments concerning QCIF sequences, DCT-8 consistently achieved better average PSNR than DISCOVER. The former also consistently required a lower bit-rate to achieve the same quality, though as in the CIF equivalent, the BD-rate metric for QCIF Mother & daughter sequence also exhibits the same anomaly that DCT-8 requires more than twice the bit-rate of DISCOVER despite showing PSNR gain in corresponding BD-PSNR metric.
Figure 5.4: RD curves for different DCT block size configurations for (a) Coastguard, (b) Crew, (c) Foreman, (d) Hall, (e) Mother & daughter and (f) Soccer QCIF test sequences.

Table 5.6: Average performance gain by DCT-8 with respect to DISCOVER for QCIF sequences

<table>
<thead>
<tr>
<th>Sequences (QCIF)</th>
<th>DCT-8 BD-PSNR (dB)</th>
<th>BD-rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>0.33</td>
<td>-17.41</td>
</tr>
<tr>
<td>Crew</td>
<td>0.29</td>
<td>-9.00</td>
</tr>
<tr>
<td>Foreman</td>
<td>0.82</td>
<td>-26.54</td>
</tr>
<tr>
<td>Hall</td>
<td>0.33</td>
<td>-12.68</td>
</tr>
<tr>
<td>Ice</td>
<td>0.27</td>
<td>-8.81</td>
</tr>
<tr>
<td>Mother &amp; daughter</td>
<td>0.50</td>
<td>213.89</td>
</tr>
<tr>
<td>Soccer</td>
<td>0.46</td>
<td>-13.04</td>
</tr>
</tbody>
</table>

To investigate the anomaly further, the BD-rate metric calculation procedure must be studied. It consists of 3 steps:
1. Plot both curves with PSNR on the x-axis and log (bit-rate) on the y-axis.

2. Calculate a 3rd order polynomial expression for each curve using curve-fitting.

3. Find the integrals of both expressions and calculate average difference between the integrals.

It is assumed that for a ‘normal’ RD curve, the bit-rate progression for the same amount of PSNR gain is exponential. For example, consider the case of 3 RD points where the PSNR differences between first and second points and between the second and third points are equal. If the bit-rate of second point is 20% greater than that of the first point, according to the assumption, the bit-rate of the third point will be 20% more than bit-rate of the second point. This makes the bit-rate of the third point 44% more than bit-rate of the first point. Thus, PSNR progression in a ‘normal’ RD curve is steeper in the lower bit-rate range than in the higher bit-rate range.

However, in Figure 5.4 (e), it can be observed that the PSNR difference between the first two RD points of DISCOVER is relatively small compared to the remaining portion of the curve, which makes the above assumption invalid. Figure 5.5 displays this anomaly by plotting the PSNR vs log(bit-rate) curve for the Mother & daughter QCIF test sequence. As a result, the polynomial and subsequent integral calculation become skewed resulting in an abnormal BD-rate value. This hypothesis is proven by replacing the RD point corresponding to QM1 with another RD point corresponding to QM6 which produces a more ‘normal’ RD curve and PSNR vs log(bit-rate) curve in Figure 5.5 (b) and (c). The corresponding BD-PSNR and BD-rate metrics then become 0.58dB and –37.54% respectively as shown in Table 5.7 along with the updated corresponding values for the CIF sequence.
Figure 5.5: RD curves and PSNR vs log(bit-rate) curves for Mother & Daughter QCIF sequence for calculation of BD-rate metric: (a) the original curve when plotting PSNR vs log(bit-rate), (b) RD curve for DISCOVER when QM6 is used instead of QM1, and (c) resulting ‘normal’ PSNR vs log(bit-rate) curve.

Table 5.7: Updated BD-PSNR and BD-rate metric for Mother & daughter test sequence when DISCOVER RD curves include QM6 instead of QM1

<table>
<thead>
<tr>
<th>Resolution</th>
<th>DCT-16 BD-PSNR (dB)</th>
<th>DCT-16 BD-rate (%)</th>
<th>DCT-8 BD-PSNR (dB)</th>
<th>DCT-8 BD-rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIF</td>
<td>0.83</td>
<td>-49.75</td>
<td>0.62</td>
<td>-40.27</td>
</tr>
<tr>
<td>QCIF</td>
<td>0.58</td>
<td>-37.54</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The reason for the slow PSNR progression at the beginning of the Mother & daughter sequence is due to its low motion characteristics. Since the scene is mostly static, the resulting SI quality is very high. As a result, the first few QM containing mostly higher order bit-planes do not add any new information as they are already present in the SI. Only when the lower order bit-planes begin to be transmitted, does the decoder then receive new detail which contributes to the PSNR gain. For this reason, this anomaly is not observed in either moderate or high motion sequences like Soccer, where due to the lower SI quality, each additional bit-plane contains substantial additional new
information which directly contributes to the PSNR gain starting from very low bit-rate.

5.3.4 Performance evaluation across spatial resolutions

It can be observed from Tables 5.4, 5.5 and 5.6 that the corresponding gains derived are greater at higher spatial resolutions by using larger DCT block sizes. For instance, for Soccer, DCT-16 exhibits BD-PSNR metrics of 2.22dB, 1.42dB and 0.46dB respectively for 4CIF, CIF and QCIF sequences, with similar observations observed for other sequences. The level of improvement is directly related to the spatial resolution, so as the resolution is reduced, the lengths of corresponding LDPCA coders shorten as shown in Table 5.2, which incurs a more substantial performance loss. For instance, 4CIF, DCT-16 and DCT-4 respectively use LDPCA coders of lengths 1584 and 25344 while for CIF sequences, the corresponding LDPCA coders have lengths of 396 and 6336 respectively. Thus, the performance deficit of the LDPCA coder of length 396 will be greater than that of its 6336 counterpart and likewise for the coder of length 1584 with respect to its 25344 counterpart, so the overall performance gain for 4CIF sequences will always be greater than for CIF sequences.

5.3.5 Performance evaluation across scene characteristics

The test sequences used contain a variety of different scene characteristics in terms of local motion, global motion, occlusion and texture. From the results in the previous subsections, DCT-16 and DCT-8 exhibited better RD performance for all sequences compared to both DCT-4 and DISCOVER. For CIF sequences, the performance gain by DCT-16 was higher in high-motion sequences like Football and Soccer than low-motion sequences such as Mother & daughter and Hall. This pattern is not however, replicated at other spatial resolutions because the DCT-8 results for example, follow
a very different pattern. An interesting observation from the RD curves is that the performance of DCT-16 and DCT-8 is very close to DISCOVER at the low bit-rates for sequences like *Mother & daughter*. While it may intuitively appear that DISCOVER will exceed DCT-16 and DCT-8 if the bit-rate is further lowered, this deduction is invalid because this RD point of DISCOVER is generated by the lowest QM1 and is the minimum QoS the codec offers, so if the bit-rate is lowered any further, DISCOVER will not be able to produce an output.

![Original WZF vs DCT-16, DCT-8 vs DISCOVER](image)

Figure 5.6: A sample frame #150 of *Crew* 4CIF sequence decoded by different configurations of the DVC-HR framework

5.3.6 Qualitative performance evaluation

To complement the quantitative performance evaluation, a number of sample frames from various test sequences decoded by the competing DVC-HR configurations in Table 5.3 are presented for qualitative comparison. Figure 5.6 shows frame #150 from
the *Crew* 4CIF sequence decoded by DCT-16, DCT-8 and DISCOVER. Corresponding PSNR and bit-rate information are also shown along with the original WZF. In these results, while there are all encoded at similar bit-rates, frames decoded by DCT-16 and DCT-8 contain notably fewer artefacts and have a superior perceived quality compared to the DISCOVER decoded frame.

Figure 5.7 shows a similar perceptual comparison for sample frame #164 of the *Ice* 4CIF sequence. The DCT-16 decoded frame includes minor artefacts around some object edges, but in contrast both the DCT-8 and DISCOVER decoded frames contain conspicuous visible artefacts especially on the icy ground. Also, the respective bit-rates of both DCT-16 and DCT-8 are significantly lower (≈30%) than DISCOVER, a trend which is consistently observed across all frames in this sequence.

Figure 5.7: A sample frame #164 of *Ice* 4CIF sequence decoded by different configurations of the DVC-HR framework
Figure 5.8 compares the decoded output quality of frame #148 of the Soccer 4CIF sequence. It is difficult to perceive artefacts in both the DCT-16 and DCT-8 decoded frames due to the inherent motion blur in the original WZF. The DISCOVER decoded frame has visibly inferior quality with numerous perceptible artefacts, so while the corresponding bit-rates for both DCT-16 and DCT-8 is greater than that of DISCOVER, they both provide superior subjective qualities as corroborated by PSNR values which are at least 7dB higher. Although the PSNR difference is not as substantial for every frame across the sequence, both DVC-16 and DVC-8 again consistently produced perpetually superior output quality compared with DISCOVER.

Figure 5.8: A sample frame #148 of Soccer 4CIF sequence decoded by different configurations of the DVC-HR framework
Figure 5.9 displays the visual comparison of frame #76 from the *Coastguard* CIF sequence decoded by various configurations of the DVC-HR framework. This is a challenging sequence containing multiple moving objects, occlusion, disocclusion and complex motion characteristics caused by the dynamic textures created by the waves and water rippling. This means SI quality is generally far lower for this sequence and so all three decoded frames display some artefacts when compared with the original. While fine detail such as edges becomes distorted in the decoded frames, due to the unique continual varying textural structure in this scene, it is difficult to identify specific areas containing visible artefacts. This observation is also observed in many other frames in this sequence.

![Original WZF](image1.png) ![DCT-16, 29.4dB @ 227.97kbps](image2.png)

![DISCOVER, 27.36dB @ 285.35kbps](image3.png) ![DCT-8, 29.16dB @ 230.86kbps](image4.png)

Figure 5.9: A sample frame #76 of *Coastguard* CIF sequence decoded by different configurations of the DVC-HR framework.
5.3.7 Transmission Overheads

Changing the DCT block size affects the transmission overheads of the DVC-HR framework in two ways. Firstly, the feedback channel overhead due to the transmission of QM from the CAQ module is impacted. It is clear from Table 5.1 that despite having increased dynamic range, each element of the QM corresponding to both 8×8 and 16×16 block sizes can still be coded with 4 bits — exactly the same as the 4×4 block size. Crucially however, since the relationship between the number of coefficients and block size is quadratic, larger block sizes will produce significantly more coefficients and consequently, the corresponding QM will have more elements which need to be coded and transmitted back to the encoder. Larger block sizes thus incur higher feedback channel overheads than its smaller equivalents. Following the same calculation as in Section 4.3, the CAQ module incurs 256 bits and 1024 bits feedback channel overhead for each QM transmission to the encoder for 8×8 and 16×16 block sizes respectively.

Larger block sizes also produce larger number of bit-planes, each of which is accompanied by an 8-bit CRC hash to assist channel decoding, so another overhead is incurred due to transmission of CRC hashes. Conversely, each bit-plane produced by a larger block size is shorter so its CRC hash can be more effective in detecting potential incorrect decoding of the LDPCA coder.

5.3.8 Parallel processing advantages

Aside from offering superior RD performance, larger DCT block sizes are more suitable for parallel processing applications. In the DVC-HR framework, DCT coefficients are grouped into coefficient bands before being converted into bit-planes. Bit-planes from the same band are interdependent and must be processed sequentially while bit-planes from different bands are independent and so can be processed in
parallel. Since a larger DCT block size produce more coefficient bands than its smaller counterparts, it leads to more independent bit-planes which can be processed simultaneously by parallel processing. Moreover, since each bit-plane is now shorter, it can be processed more efficiently before moving to the next bit-plane from the same coefficient band. It thus makes a parallel processing implementation more attractive and efficient.

5.3.9 Latency considerations

Using a larger DCT block size is computationally more complex than their smaller counterparts and though the increased complexity is deemed tractable for modern hardware and software platforms, it contributes to both the overall encoder and decoder complexity and consequent coding time. However, since both the encoder and the decoder employ the DCT, using larger block sizes will not cause any latency drift between them so their relative latency does not increase. However, larger block sizes produce significantly more bit-planes than their smaller counterpart, so sequential processing of the additional bit-planes can be inefficient, leading to accumulative latency issues.

In addition, the increased dynamic range of the coefficient bands produced by larger DCT block sizes can escalate the latency issue, especially in parallel processing scenarios. Since the DCT compacts most of information in the DC coefficient, the corresponding coefficient band will have a comparatively wide dynamic range. Moreover, the HVS weighting applied to the DC coefficient is the highest and consequently the DC coefficient band is generally allocated the highest number of bit-planes by the CAQ module. Since bit-planes from the same coefficient band are processed successively, the overall processing speed can become compromised. In contrast, AC coefficients are generally allocated a smaller number of bit-planes which
allows for more efficient processing. From a parallel processing perspective, this poses a significant problem as the entire system must wait for a single component to complete its processing before starting to process the next WZF, with a net impact on the system latency between the encoder and decoder. This particular issue will be investigated in more depth in the next chapter and a novel solution proposed.

5.4 Summary

This chapter has rigorously investigated the effects of using larger DCT block sizes on DVC complexity and performance. Larger block sizes have previously been shown to provide improved performance in traditional video coding, and it has been proven similarly in this chapter for DVC, especially in higher resolution sequences. Both 16×16 and 8×8 block sizes have been embedded into the new DVC-HR framework, with the corresponding QM of appropriate sizes produced by the CAQ module with suitable modifications to both the CAQ and LDPCA coder modules of the framework. Critical analysis of both quantitative and qualitative results has confirmed the hypothesis that larger DCT block sizes consistently exhibit superior RD performance. The performance gain is more substantial for higher resolution sequences and the benefits of larger DCT block sizes are pervasive for sequences with varying scene characteristics. In addition, it is suitable for a parallel processing realisation since it produces more independent bit-planes that can be processed concurrently. The limitation is that larger block sizes both incur higher transmission overheads and increase the dynamic range of coefficient bands which can lead to greater latency between the encoder and decoder. This codec latency issue will be the main focus of the next contribution chapter.
6 Advanced Decoding Method for the Channel Coder

6.1 Introduction

This chapter investigates the accumulated latency of the DISCOVER codec that develops and intensifies when coding both higher resolution sequences and larger DCT block sizes as discussed in Subsections 2.5.4 and 5.3.9. To recap, the feedback channel-based trial-and-error decoding procedure of LDPCA coder is inherently slow where each failed decoding attempt incurs a latency overhead. This latency penalty increases while processing at higher resolutions since more data need to be processed. Additionally, increasing bandwidth requirement increases latency due to additional bit-planes needing to be decoded. Similarly, larger DCT block size produces more bit-planes compared to their smaller alternatives leading to increased latency. This can cause a significant latency which is detrimental for time sensitive applications, thus needs to be addressed.

A novel approach to address this issue is proposed in this thesis by integrating an advanced Belief Propagation (BP) decoding algorithm for the LDPCA coder into the DVC-HR framework. This chapter describes the proposed method and critically analyses experimental results to evaluate its effect on the overall latency. Section 6.2 investigates the cause of the latency issue in greater depth while Section 6.3 presents the advanced decoding algorithm adopted in the DVC-HR framework. The simulation criteria to assess its effectiveness is described in Section 6.4. Section 6.5 critically analyses the simulation results and some concluding remarks are provided in Section 6.6.
6.2 Background

The DISCOVER codec minimises the transmission of parity bits, and thus the bit-rate, by adopting an on-demand transmission protocol. The LDPCA encoder produces parity bits from bit-planes of a WZF which are stored in a buffer. For each bit-plane, the encoder initially transmits a bulk portion of parity bits based on its estimate of the entropy of the bit-plane under consideration. The entropy is estimated following the WZ theorem (Wyner and Ziv, 1976). The LDPCA decoder attempts to correct errors from the SI bit-plane using these parity bits and requests for more parity bits if the error correction process fails. Thus, the parity bits are progressively transmitted until a decoding attempt is successful. While this method ensures only the necessary number of parity bits are ever transmitted, it concomitantly introduces latency into the coding process due to the possible multiple decoding attempts for each bit-plane.

In each attempt, the LDPCA decoder employs a BP algorithm to decode the bit-plane under consideration using the available parity bits. The BP algorithm is iterative and converges an erroneous codeword to a valid codeword using the parity bits and the soft-input information from the Correlation Noise Modelling (CNM) module. Each iteration involves floating-point operations, so they are computationally expensive and the overall process is inefficient. The convergence is checked after each iteration using the parity check matrix of the corresponding LDPCA code and the algorithm terminates if all parity checks are satisfied. Otherwise, the algorithm is executed for a predetermined number of iterations which is 100 in the DISCOVER codec. This means for each unsuccessful decoding attempt, 100 iterations of the BP algorithm are performed before more parity bits are requested, so incurring potentially a notable delay between the encoder and decoder. This is aggravated at higher spatial
resolutions since the complexity of the BP algorithm is intensified due to the increased bit-plane lengths.

The use of larger block sizes for transform and quantisation decreases the length of bit-planes, so effectively counterbalancing the aforementioned problem that arises at higher resolution. However, another issue emerges due to the increased dynamic range of the transform coefficients produced at larger block sizes. As explained in Subsection 5.3.9, larger block sizes produce more bit-planes in a coefficient band than smaller block sizes and as bit-planes from the same band must be sequentially decoded from the Most Significant Bit (MSB) to the Least Significant Bit (LSB), a lower order bit-plane will need to wait for higher order bit-planes so compounding the latency issue.

The primary reason behind multiple decoding attempts is the underestimation of the entropy of a bit-plane. However, it is extremely difficult to guarantee accurate entropy estimation because neither the encoder nor the decoder has access to both the WZF and the SI, so the exact noise characteristics required to determine the entropy are unknown. On the other hand, overestimating of the entropy may help to reduce the latency by ensuring success of the first decoding attempt, however this can incur a significant bit-rate cost.

The latency issue can be detrimental for time critical applications and several solutions aiming to alleviate the issue have been proposed in the literature. For instance, Kuo et al. (2016) have attempted to reduce the delay by constraining the number of requests the decoder is allowed to make. This however sacrifices rate control flexibility and incurs significant RD performance loss in certain scenarios. The feedback channel-free solutions as reviewed in Subsection 2.5.4 undesirably compromises encoder
complexity and performs poorly for sequences with high motion activities. A more practical approach proposed by Shen et al. (2017) is to introduce parallel processing to simultaneously decode multiple bit-planes. Whilst it does reduce the latency to some extent, it does not address the root cause identified above, namely, latency incurred due to entropy underestimation and consequent failed decoding attempts.

A more pragmatic approach is therefore to reduce the decoding time for each attempt which intuitively means constraining the requisite number of BP iterations. If the number of iterations is reduced, each unsuccessful attempt will be recognised faster and the decoder can request more parity bits sooner so reducing the overall latency. However, limiting of the number of iterations this way will negatively impact upon the convergence of the BP algorithm as the decoder may prematurely judge an attempt unsuccessful after a smaller number of iterations, which otherwise could have succeeded given a sufficient number of iterations. In that case, the decoder unnecessarily requests additional parity bits and increases the bit-rate thus affecting the overall RD performance. It is apparent a more intelligent solution needs to be developed.

6.3 Advanced BP algorithms

This section describes the theory behind the proposed smart solution to reduce the latency, however the basic workings of the classic BP algorithm needs to be outlined first.

6.3.1 The classic BP algorithm

The LDPCA coder proposed by Varodayan et al. (2006) can be viewed as a cascade of multiple LDPC codes (Gallager, 1962) having the same number of input bits, but varying number of parity bits. During each decoding attempt, the LDPCA coder
selects the appropriate LDPC code according to the number of available parity bits. The parity bits transmitted from the encoder are in fact accumulated syndrome generated by the LDPCA encoder, which needs to be converted to plain parity bits suitable for the selected LDPC code. The soft input information is received from the CNM module and the LDPCA decoder initiates the classic BP algorithm within the selected LDPC code context.

![Factor graph of an LDPC code of 8 variable nodes (v) and 4 check nodes (c)](image)

Figure 6.1: Factor graph of an LDPC code of 8 variable nodes (v) and 4 check nodes (c)

An overview of the classic BP algorithm is given in Subsection 2.5.4 that involves passing *Log Likelihood Ratio* (LLR) information as messages between check nodes (c) and variable nodes (v) of the underlying factor graph. Figure 2.8 has been reproduced in Figure 6.1 showing an example factor graph of an LDPC code consisting of eight variable nodes and 4 check nodes. Each variable node corresponds to a bit in the bit-plane while each check node corresponds to a parity bit. The connections between variable nodes and a check node represent the parity check equation corresponding the check node. In the BP decoding algorithm, messages are
propagated along the edges between connected pairs of check and variable nodes and the message in each direction are generated according to:

\[ m_{v_j \rightarrow c_i} = \sum_{c_a \in \mathcal{N}(v_j) \setminus c_i} m_{c_a \rightarrow v_j} + C_{v_j} \]  \hspace{1cm} (6.1)

\[ m_{c_i \rightarrow v_j} = 2 \times \tanh^{-1} \left( \prod_{v_b \in \mathcal{N}(c_i) \setminus v_j} \tanh \left( \frac{m_{v_b \rightarrow c_i}}{2} \right) \right) \]  \hspace{1cm} (6.2)

Where \( m_{v_j \rightarrow c_i} \) and \( m_{c_i \rightarrow v_j} \) are the message from a variable node \( v_j \) to a check node \( c_i \) and the message from a check node \( c_i \) to a variable node \( v_j \) respectively; \( \mathcal{N}(v_j) \setminus c_i \) denotes the neighbours of \( v_j \) less \( c_i \) and \( \mathcal{N}(c_i) \setminus v_j \) denotes the neighbours of \( c_i \) less \( v_j \). \( C_{v_j} \) is the priori LLR corresponding to the variable node \( v_j \) as supplied by the CNM module.

In the classic BP algorithm, during each iteration, messages from all variable nodes are propagated to their neighbouring check nodes simultaneously according to (6.1), followed by messages from all check nodes to their neighbouring variable nodes according to (6.2). This continues until the stopping criteria is met. The stopping criteria generally consists of two conditions, whether all the parity check equations are satisfied, and whether the total number of iterations has reached the maximum \( I_M \).

Therefore, after each iteration, the algorithm generates a candidate bit-plane from the posteriori LLR \( C_{v_j}^{p} \) and evaluates the parity check equations. \( C_{v_j}^{p} \) is calculated by taking summation of all incoming messages to the variable node \( v_j \) as well as its priori LLR \( C_{v_j} \) according to:

\[ C_{v_j}^{p} = \sum_{c_i \in \mathcal{N}(v_j)} m_{c_i \rightarrow v_j} + C_{v_j} \]  \hspace{1cm} (6.3)
The value of a bit is determined from the sign of $C^P_{vj}$, namely, if $C^P_{vj}$ is negative, the value of the bit is set to ‘1’ and it is set ‘0’ otherwise. If all parity checks are satisfied, the decoding is deemed successful and the BP algorithm terminates. If it is not successful after $I_M$ iterations, the decoding attempt is considered a failure and additional parity bits are requested.

The method of simultaneous propagation of all variable node messages followed by simultaneous propagation of all check node messages is known as *flooding* and henceforth, it will be referred to as such to distinguish between different BP algorithms. A problematic property of the flooding algorithm is that it does not use messages that have already been propagated during the current iteration. Rather, it uses messages from the previous iteration to prepare new messages for propagation. For instance, after propagating $m_{vj \rightarrow c_i}$, all messages $m_{ci \rightarrow v_b}, v_b \in \mathcal{N}(c_i) \setminus v_j$ will change and consequently, all messages originating from a $v_b$ also change. However, in the flooding algorithm, this change is not considered and the message $m_{v_b \rightarrow c_i}$ is generated using the messages available prior to propagating $m_{vj \rightarrow c_i}$. A similar observation can be made about the propagation of $m_{ci \rightarrow v_j}$ from check nodes to variable nodes. This disrupts efficient propagation of messages across the factor graph and causes slower convergence.

### 6.3.2 Sequential BP algorithms

It has been shown that by adopting a *sequential* message-passing schedule instead of a simultaneous one, the messages from a node can be generated with more up-to-date information (Zhang and Fossorier, 2005; Hocervar, 2004). In these methods, a message is generated and propagated from a variable node to its neighbouring check nodes, followed by generating and propagating messages from those check nodes. While
generating messages for the next variable node, the algorithm uses recently received messages from connected check nodes, if available, instead of messages from the previous iteration. An alternate approach is to start by propagating messages from a check node followed by propagating updated messages from neighbouring variable nodes to other check nodes, before moving onto the next check node.

The common advantage reported for all these sequential BP algorithms is that they converge twice as fast and can even achieve a lower error floor than the flooding BP. In a DVC context, this means each attempt for decoding a bit-plane will converge faster, requiring fewer iterations and could be decoded with fewer parity bits. The first advantage is especially attractive for this work since if convergence of the BP algorithm can become more efficient, then the maximum number of iterations can be confidently lowered without incurring the risk of adversely affecting the performance.

6.3.3 Informed Dynamic Scheduling strategy for sequential BP

The question of how to best schedule the nodes arises with the adoption of the sequential BP algorithm. Casado et al., (2007) proposed a Node-Wise Residual Belief Propagation (NWRBP) technique which is an Informed Dynamic Scheduling (IDS) approach to this scheduling problem which prioritises messages to more error-prone nodes. The prioritisation is based on the residual, \( r(m) \) of a message which is defined as the absolute difference between its value before and after an update to the originating node. If a message has a large residual, i.e. the message to a variable node changes substantially before and after an update, it signifies that node is more likely to be connected to an unsatisfied check node and so is considered unstable and error-prone. The algorithm then prioritises such nodes so that they receive more message updates during an iteration and have more opportunities to correct any errors. Conversely, more stable nodes will receive fewer updates. Simulation results shows
that by adopting a dynamic scheduling strategy, convergence of the NWRBP is even faster than the sequential BP and can also achieve a consistently lower error floor (Aslam et al., 2017; Casado et al., 2007).

An IDS strategy like the NWRBP is especially attractive for DVC due to its unique design features. In a BP algorithm, the LLR of a variable node $C_{vj}$ is central to both message generation and stopping criteria which is the natural logarithm of the Likelihood Ratio (LR) probability of a bit in a bit-plane. LR is computed as the probability of a bit being a ‘0’ divided by the probability of the bit being a ‘1’. The LR for each variable node is determined from the channel parameters and its exact computation varies depending on the channel. We are interested in the Binary Symmetric Channel (BSC) since the virtual channel in the DVC architecture is assumed to be such. In a BSC, the value of a bit may be inverted due to noise and the relevant channel parameter is the crossover probability which denotes the probability of a bit being affected by noise. In a traditional context, LR is determined from this crossover probability and since it is fixed for the channel, all ‘0’ bits in the bit-plane share a single LR value, while all ‘1’ bits share a different LR value. Consequently, LLR being the natural logarithm of LR, many nodes will share a common LLR. This slows down the prioritisation of nodes at the beginning of NWRBP since many nodes have the same LLR value, thus the same residual, though the situation improves after a few iterations when the nodes have more diverse LLR values.

In a DVC context, the relationship between channel parameters and LR is reversed. In the DVC-HR framework, the CNM module calculates soft-inputs, i.e. LR values for each variable node and crossover probability is estimated using these LR values. Contrary to the traditional context, each variable node can have a unique LR value so
is therefore more distinctive. This harmonises with the prioritisation step of the NWRBP and can improve its convergence performance.

In this DVC work, it is proposed to integrate the NWRBP algorithm rather than the classic flooding BP algorithm in the LDPCA coder. In order to exploit its higher convergence rate and facilitate faster decoding time per attempt, the stopping criteria of the NWRBP is modified and will be referred to as modified NWRBP (mNWRBP) to differentiate from the original. A formal description of the original NWRBP algorithm is presented in the next subsection before the modified stopping criteria is introduced in Subsection 6.3.5.

6.3.4 The NWRBP algorithm

The approximate variant of the NWRBP algorithm (Casado et al., 2007) was chosen for the DVC framework because of its simplicity and reduced complexity. In this variant, an approximate of a check-to-variable message is used for residual computation. The sign of the message is computed by considering the signs of all incoming messages so that it conforms with the received parity bits. The magnitude of the message is then computed as:

\[ |m_{c_i \rightarrow v_j}| = \min_{v_i \in \mathcal{N}(c_i) \setminus v_j} |m_{v_b \rightarrow c_i}| \]  \hspace{1cm} (6.4)

According to (6.4) the magnitude of a message from a check node to a variable node is the minimum of the magnitudes of incoming messages from all other variable nodes. The residual of a message is then:

\[ r\left(m_{c_i \rightarrow v_j}\right) = |m_{c_i \rightarrow v_j} - m^{pre}_{c_i \rightarrow v_j}| \]  \hspace{1cm} (6.5)

where \(m^{pre}_{c_i \rightarrow v_j}\) denotes the message before the update. Note that all message computations in (6.5) are performed using (6.4) instead of (6.2).
The NWRBP algorithm is now presented in Algorithm 6.1 in pseudocode format:

**Algorithm 6.1 NWRBP decoding for LDPC codes**

1: Initialise all $m_{c_i \rightarrow v_j} \leftarrow 0$
2: Initialise all $m_{v_j \rightarrow c_i} \leftarrow c_{v_j}$
3: Compute all $r(m_{c \rightarrow v})$ using (6.4) and (6.5)
4: Let $m_{c_i \rightarrow v_j}$ be the message with largest residual
5: for every $v_x \in \mathcal{N}(c_i)$ do
6: \hspace{1em} Generate and propagate $m_{(c_i \rightarrow v_x)}$ using (6.2)
7: \hspace{1em} Set $r(m_{c_i \rightarrow v_x}) \leftarrow 0$
8: \hspace{1em} for every $c_a \in \mathcal{N}(v_x) \setminus c_i$ do
9: \hspace{2em} Generate and propagate $m_{v_x \rightarrow c_a}$
10: \hspace{2em} for every $v_b \in \mathcal{N}(c_a) \setminus v_x$ do
11: \hspace{3em} Compute $r(m_{c_a \rightarrow v_b})$ using (6.4) and (6.5)
12: \hspace{1em} end for
13: end for
14: if Stopping criteria is not satisfied then
15: \hspace{1em} go to Step 4.
16: end if

Figure 6.2 demonstrates an example of the NWRBP algorithm over a factor graph, where it is assumed after initialisation, the edge between $v_7$ and $c_4$ has the largest residual (Step 4). This is indicated by the red lines in Figure 6.2 (a). Now all messages associated with the check node $c_4$ will be generated and propagated including messages to $v_4$, $v_5$, $v_7$ and $v_8$ (Step 6). This is shown in Figure 6.2 (b) as blue lines. If the message to $v_4$ is propagated first, then the message from $v_4$ to its neighbouring check nodes, except $c_4$ will be generated and propagated. Since the only one in this case is the check node $c_2$, the message $m_{v_4 \rightarrow c_2}$ is generated and propagated (Step 9) and is indicated by the purple lines in Figure 6.2 (c). Now all the residuals associated with $c_2$ are recomputed except the one associated with $v_4$. This results in the residual computations along $v_1$, $v_2$ and $v_6$ (Step 10 – 12) and is represented by the green lines in Figure 6.2 (d). Algorithm 6.1 then returns to Step 4 to find the next largest residual and the process is repeated.
Figure 6.2: Example demonstrating message propagations in the LDPC code during an iteration of the NWRBP algorithm

6.3.5 The mNWRBP algorithm

The mNWRBP algorithm differs from the original in its stopping criteria which is modified to exploit the faster convergence property of the algorithm. The stopping
criteria in the original NWRBP algorithm (Casado et al., 2007) required two conditions to be assessed after each iteration, namely if:

a. All parity check equations have been satisfied

b. The maximum number of iterations, $I_M$ has been reached

The algorithm is then terminated if the message passing loop upholds either of these two criteria.

To better exploit the advantages of NWRBP in a DVC context, a new dynamic stopping condition is proposed. According to the new criterion, the candidate bit-plane is checked for changes after each iteration by comparing with the one before the iteration. If no change is observed after a certain amount of iterations $I_C$, the algorithm is assumed to have converged and since the parity check equations have not been satisfied in previous iterations, the convergence is at a wrong codeword. This signifies that the available amount of parity bits is insufficient and more need to be requested. The new stopping criteria is described as follows:

**mNWRBP stopping criteria:** Terminate the message passing loop if any of the following three conditions is true:

a. The candidate bit-plane has not changed in the recent $I_C$ iterations

b. All parity check equations are satisfied

c. The maximum number of iterations $I_M$ has been reached.

It is important to stress that if the mNWRBP terminates due to condition b, then the attempt is considered a success and deemed as a failure otherwise.

To illustrate the effect of $I_C$ and the newly introduced condition a, let us consider the case of decoding a bit-plane containing a single bit error. The following three cases
can happen depending on whether the amount of available parity bits is sufficient and the value of $I_C$:

- **Case 1**: First let us assume the original NWRBP algorithm is employed and the amount of parity bits is insufficient. Therefore, the algorithm will execute $I_M$ iterations before the decoding attempt fails. Throughout these $I_M$ iterations, the erroneous bit will not be corrected and consequently the candidate bit-plane will remain unchanged. If the mNWRBP were to be employed, it would notice the unchanged candidate bit-plane after $I_C$ iterations, thus satisfying condition $a$. As a result, the algorithm will be stopped and the decoding attempt will be declared as failure before additional parity bits being requested. In this case, employing the mNWRBP algorithm has saved $I_M - I_C$ iterations for this attempt.

- **Case 2**: Now, let us assume that the decoder has sufficient number of parity bits and the NWRBP algorithm successfully decodes the bit-plane after, say, 10 iterations. Now if $I_C \geq 10$, the erroneous bit will be corrected at the 10th iteration, thus will not satisfy condition $a$, and proceed to condition $b$ which is now satisfied. As a result, the decoding attempt will be declared as success and performance of mNWRBP will be identical to the original NWRBP.

- **Case 3**: The situation changes if $I_C < 10$. The mNWRBP algorithm will notice that the candidate bit-plane has been unchanged during last $I_C$ iterations, satisfying condition $a$, therefore will terminate the algorithm before requesting additional parity bits. In this case, the mNWRBP algorithm incurs a bit-rate cost with respect to the original NWRBP algorithm. However, the former can still save some iterations if there were more failed attempt prior to the successful one where it would save $I_M - I_C$ iterations for each failed attempt.
The value of $I_C$ is set between 1 and $I_M$, which is the maximum number of iterations in the flooding BP algorithm. If the value is too small, then the algorithm may prematurely stop before a sufficient number of iterations has been completed for successful decoding as in Case 3. If the value is too big, the saving of $I_M - I_C$ iterations per failed attempt will be small and there will be little overall improvement compared to flooding BP. The proposed modification becomes ineffectual when $I_C = I_M$, thus essentially making it identical to the original NWRBP algorithm. This means $I_C$ needs to be carefully and empirically determined.

Table 6.1: Total iterations and rate performance of the mNWRBP algorithm for various values of $I_C$ for Soccer QCIF sequence

<table>
<thead>
<tr>
<th>QM</th>
<th>$I_C = 5$</th>
<th>$I_C = 10$</th>
<th>$I_C = 15$</th>
<th>$I_C = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration</td>
<td>Rate (kbps)</td>
<td>Iteration</td>
<td>Rate (kbps)</td>
<td>Iteration</td>
</tr>
<tr>
<td>1</td>
<td>94226</td>
<td>81.4106</td>
<td>145526</td>
<td>81.4059</td>
</tr>
<tr>
<td>2</td>
<td>136187</td>
<td>130.3224</td>
<td>220835</td>
<td>130.3224</td>
</tr>
<tr>
<td>3</td>
<td>154057</td>
<td>140.0110</td>
<td>247763</td>
<td>140.0086</td>
</tr>
<tr>
<td>4</td>
<td>200975</td>
<td>203.9364</td>
<td>317610</td>
<td>203.9340</td>
</tr>
<tr>
<td>5</td>
<td>213553</td>
<td>212.4459</td>
<td>336713</td>
<td>212.4459</td>
</tr>
<tr>
<td>6</td>
<td>330576</td>
<td>275.8252</td>
<td>475186</td>
<td>275.8252</td>
</tr>
<tr>
<td>7</td>
<td>440623</td>
<td>414.3117</td>
<td>654778</td>
<td>414.3117</td>
</tr>
<tr>
<td>8</td>
<td>1117059</td>
<td>786.7329</td>
<td>1576254</td>
<td>786.7329</td>
</tr>
</tbody>
</table>

Several experimentations have been conducted with various test sequences and different values for $I_C$ to find the most suitable one. Table 6.1 presents the total number of iterations required by the mNWRBP at four different $I_C$ values along with corresponding bit-rates. The results have been obtained by simulating the Soccer QCIF sequence with the eight predetermined QM, so naturally 4x4 DCT block size has been used. It is important to note that the choice of Soccer sequence is not crucial for this example and similar results can be obtained for other test sequences. It can be seen from Table 6.1 that smaller $I_C$ values required significantly less iterations than greater $I_C$ values, though they exhibit identical rate performance except for QM3 and QM4 where $I_C = 5$ incurred slightly higher bit-rates than other $I_C$ values. It was
revealed upon further investigation that while coding the whole sequence using QM₃, decoding attempt of a single bit-plane from a WZF was prematurely terminated and more parity bits were requested as in Case 3, causing the higher bit-rate. Similar incident has been found for QM₄ as well as for other sequences where such cases have been noticed. However, their occurrences have been very rare and corresponding rate differences have been negligible. Since 𝐼ₐ = 5 required the least number of iterations with little to no rate cost, this value has been chosen over its greater alternatives for further experimentations.

6.3.6 mNWRBP algorithm complexity analysis

The mNWRBP algorithm can potentially save 𝐼ₘ − 𝐼ₐ iterations for each failed attempt compared to the flooding algorithm while decoding a bit-plane. However, the definition of an iteration is not identical for both algorithms and consequently, their complexity differ. As a result, reduction in number of iterations by the mNWRBP algorithm does not translate into proportionately reduced latency. Hence, the complexity analysis of the mNWRBP algorithm with respect to the flooding algorithm is necessary for equitable comparison between the two.

In the flooding algorithm, the definition of an iteration is well-defined and counted when messages from all check nodes to neighbouring variable nodes followed by messages from all variable node to neighbouring check nodes are propagated once. If the factor graph has a total of 𝐸 edges between variable and check nodes, then each iteration consists of 𝐸 check-to-variable messages followed by 𝐸 variable-to-check messages. An iteration in the mNWRBP algorithm is defined as the state when the number of check-to-variable messages is the same as in the flooding algorithm, i.e. 𝐸. So, the overall algorithm complexity depends on how computationally intensive each check-to-variable message propagation is in either algorithm.
For each check-to-variable message, the mNWRBP propagates messages from that variable node to neighbouring check nodes except the originating node of the check-to-variable message. Therefore, the number of variable-to-check messages depends on how many neighbours that variable node has. If $\overline{d}_v$ is the average degree of a variable node in the factor graph, then a variable node is on average connected to $\overline{d}_v$ check nodes, so for every check-to-variable message, $(\overline{d}_v - 1)$ variable-to-check messages are propagated, giving a total of $E(\overline{d}_v - 1)$ variable-to-check messages per iteration. Although the mNWRBP propagates more variable-to-check messages than the flooding BP algorithm, their generation only involves additions (see equation (6.1)) so incurs minimal complexity cost.

Apart from message generation, the mNWRBP algorithm contains two other tasks which contribute to its complexity. The first is the residual computation which is performed a number of times for each variable-to-check message. If $\overline{d}_c$ is the average degree of a check node in the factor graph, then a check node receives on average messages from $\overline{d}_c$ variable nodes. For each variable-to-check message, the residual of all messages related to the check node is computed except for that relating to the original variable node. Thus, $(\overline{d}_c - 1)$ residual computations are performed for each variable-to-check message so each iteration requires $E(\overline{d}_v - 1)(\overline{d}_c - 1)$ residual computations. From equation (6.4) and (6.5), each residual computation involves finding the least valued incoming message to the check node except from the variable node under consideration. This is performed by simple comparisons so no complex computational overheads are incurred.

The second task is the sorting of residuals. Both computing and sorting the residuals require comparison of real-valued, floating-point numbers and though these are more
costly than integer operations, their complexity is insignificant compared to the overall algorithm complexity. Additionally, there is some extra processing incurred with the new stopping criteria, which requires the comparison of two bit-planes. This is performed via bit-wise operations which is even faster than integer arithmetic operations. Thus, the overhead of the proposed additional stopping criterion is negligible. The most critical complexity factor is the check-to-variable message generation which is identical in both the mNWRBP and the flooding BP algorithms.

From the above discussion it is evident that though the mNWRBP is more complex than the flooding BP algorithm, their difference is not substantial. The added complexity depends on the length of the bit-plane and a shorter bit-plane will incur less complexity overhead than a longer one. It is noteworthy that actual reduction in latency depends on specific implementation of the algorithms. Due to the lack of availability of a reference NWRBP implementation, it was implemented along with the proposed modification from scratch following Algorithm 6.1. It was then tested and validated using randomly generated bit-planes before being integrated into the DVC-HR testbed software. It is important to note that the mNWRBP implementation was an inefficient and naïve one and can be potentially improved using advanced data structures and coding techniques.

For completeness, it is worth comparing the actual time taken to decode a bit-plane by both algorithms to gain practical insights about the mNWRBP performance. However, due to design complications of the testbed software, decoding time for individual bit-planes could not be measured. Instead, the total decoding time for a test sequence was observed which revealed that employing the mNWRBP algorithm resulted in significantly faster decoding compared to employing the flooding algorithm when using 16×16 DCT block size. Decoding time increased with the
decrease of block size where using 8×8 block size resulted in similar decoding time and using 4×4 block size caused slower decoding time for the mNWRBP algorithm compared to the flooding algorithm. Though actual decoding time varied across different test sequences, overall trend has been maintained for all sequences experimented. This corroborates the aforementioned analysis that mNWRBP complexity depends on the length of bit-planes and since larger block sizes produce shorter bit-planes, they are decoded faster than smaller block size counterparts.

6.4 Simulation criteria

The mNWRBP algorithm with the new stopping criteria has been implemented in the DVC-HR testbed software and experimented for a range of test sequences to test its performance and stability at different noise levels. Various spatial resolutions and DCT block sizes have been used for different bit-plane coder lengths. The simulation criteria employed are listed in Table 6.2:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial resolution</td>
<td>QCIF, CIF, 4CIF</td>
</tr>
<tr>
<td>DCT block sizes</td>
<td>4, 8, 16</td>
</tr>
<tr>
<td>ABP</td>
<td>Varies with DCT block size</td>
</tr>
<tr>
<td>( I_C )</td>
<td>5</td>
</tr>
</tbody>
</table>

Since larger DCT block sizes produce more but shorter bit-planes, different number bit-planes need to be allocated through the ABP value configurations for equitable comparison. Table 6.3 presents the different ABP value configurations for various DCT block sizes used during simulations. The nomenclature adopted in Section 5.3 to denote different block size configurations is conserved, i.e. DCT-16, DCT-8 and DCT-4 denote the 16×16, 8×8 and 4×4 block sizes respectively. Corresponding ABP values have been adjusted such that resulting RD points lie in similar rate regions. For example, bit-rate of the RD point generated by DCT-16 and 375 ABP value is found
to be roughly alike to the one generated by DCT-8 and 100 ABP value as well as DCT-4 and 20 ABP value while coding a sequence.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>RD points</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT-16</td>
<td>375 450 525 600</td>
</tr>
<tr>
<td>DCT-8</td>
<td>100 125 150 175</td>
</tr>
<tr>
<td>DCT-4</td>
<td>20 30 40 48</td>
</tr>
</tbody>
</table>

Table 6.3: ABP value configurations for different DCT block sizes

The mNWRBP performance has been critically evaluated using various performance metrics, with the total number of iterations required to decode a bit-plane being included alongside the usual RD-related (bit-rate and PSNR) metrics. The required number of iterations reflects the potential improvement in decoding latency that can be achieved by employing an IDS strategy like the mNWRBP instead of using the traditional flooding BP algorithm. Though as discussed in Subsection 6.3.6, actual improvement is smaller since each iteration of the mNWRBP is slightly more complex than an iteration of the flooding BP algorithm. Detailed simulation results are presented in the next section along with a critical analysis and discussion of the mNWRBP performance.

6.5 Critical evaluation of mNWRBP performance

Various test sequences have been coded using the DVC-HR framework containing the mNWRBP algorithm to test the robustness of the proposed solution in different application scenarios. The sequences were chosen based on various scene characteristics like motion activities, occlusion, texture as detailed in Section 3.8. Different combinations of transform block sizes and spatial resolutions were simulated to critically evaluate the mNWRBP performance for different bit-plane lengths. The total number of BP iterations performed during each simulation run was then recorded to estimate the potential latency reduction achieved, while the
corresponding bit-rate and PSNR values were calculated to enable its impact on the RD performance to be rigorously investigated.

6.5.1 Reduction of total iterations

Table 6.4 shows performance of the mNWRBP algorithm with respect to the flooding algorithm in terms of the number of iterations required to decode Crew, Ice and Soccer 4CIF sequences. The DCT-16 configuration has been used along with the four ABP values corresponding to four RD points as detailed in Table 6.3. Table 6.4 displays the total number of BP iterations performed with the mNWRBP and flooding BP algorithms being respectively denoted by ‘mNWRBP’ and ‘Flooding’, with the mNWRBP results being generated by using the \( I_c = 5 \) configuration. The results show the iteration reduction achieved by employing the mNWRBP algorithm as a percentage of the flooding algorithm, so for example, for the RD point 1 of Ice sequence, mNWRBP requires 84.23% fewer BP iterations to decode this sequence.

Table 6.4: Decoding performance of mNWRBP with respect to Flooding for various 4CIF sequences using DCT-16 configuration

<table>
<thead>
<tr>
<th>Sequence</th>
<th>RD point</th>
<th>mNWRBP</th>
<th>Flooding</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crew</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1434908</td>
<td>9040285</td>
<td></td>
<td>84.13</td>
</tr>
<tr>
<td>2</td>
<td>2983932</td>
<td>1818596</td>
<td></td>
<td>83.59</td>
</tr>
<tr>
<td>3</td>
<td>5345881</td>
<td>2917892</td>
<td></td>
<td>81.68</td>
</tr>
<tr>
<td>4</td>
<td>8080085</td>
<td>40255095</td>
<td></td>
<td>79.93</td>
</tr>
<tr>
<td>Ice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>401351</td>
<td>2545417</td>
<td></td>
<td>84.23</td>
</tr>
<tr>
<td>2</td>
<td>704313</td>
<td>4102467</td>
<td></td>
<td>82.83</td>
</tr>
<tr>
<td>3</td>
<td>1065279</td>
<td>5648718</td>
<td></td>
<td>81.14</td>
</tr>
<tr>
<td>4</td>
<td>1814272</td>
<td>7940507</td>
<td></td>
<td>77.15</td>
</tr>
<tr>
<td>Soccer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2003540</td>
<td>11483528</td>
<td></td>
<td>82.55</td>
</tr>
<tr>
<td>2</td>
<td>4040122</td>
<td>20221111</td>
<td></td>
<td>80.02</td>
</tr>
<tr>
<td>3</td>
<td>6881676</td>
<td>29884468</td>
<td></td>
<td>76.97</td>
</tr>
<tr>
<td>4</td>
<td>10388662</td>
<td>40369565</td>
<td></td>
<td>74.27</td>
</tr>
</tbody>
</table>

Table 6.4 exhibits consistent improvement achieved by mNWRBP with respect to Flooding across different test sequences and RD points. In each case, the former required a significantly fewer number of iterations than the latter. Therefore, employing the proposed algorithm saves a substantial amount of resources and
reduces latency. For example, assuming the latency overhead incurred per iteration is equal for both algorithms, employing the mNWRBP algorithm can reduce the latency by up to 84% while coding the aforementioned sequences.

The consistent and significant improvement achieved by mNWRBP can be explained by considering the corresponding estimated entropy of the bit-planes. Consider broadly categorising the bit-planes of a sequence into two classes according to their estimated entropy. Those bit-planes which underestimate the entropy are assigned to class I, while the remainder are in class II. For the class II bit-planes, since the estimated entropy is similar or larger than the actual, the initial parity bits will be sufficient for decoding in most cases though some may require a few additional requests for parity bits. This means the respective mNWRBP and Flooding performance will be similar for these bit-planes. On the other hand, for class I bit-planes, the initial parity bits will not be sufficient and this will mandate more requests for extra parity bits before decoding succeeds. mNWRBP excels in these circumstances because each failed attempt is detected after only a few iterations instead of waiting for the entire $I_M = 100$ iterations as in Flooding.

Similar results have been obtained for experimentations with different resolution sequences as well as different DCT block sizes. Table 6.5 shows improvement achieved by mNWRBP with respect to flooding in terms of the number of iterations for the aforementioned three 4CIF sequences. Contrary to Table 6.4, specific values of number of iterations for both algorithms have been omitted and only the resulting ‘Reduction (%)’ are shown in Table 6.5. Corresponding results for CIF and QCIF sequences are presented in Tables 6.6 and 6.7 respectively though DCT-16 has not been used for QCIF sequences due to limitations of the LDPCA coder as explained in Subsection 5.2.2.
Table 6.5: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several 4CIF sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>RD point</th>
<th>Reduction (%)</th>
<th>DCT-16</th>
<th>DCT-8</th>
<th>DCT-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crew</td>
<td>1</td>
<td>84.13</td>
<td>75.92</td>
<td>52.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>83.59</td>
<td>77.41</td>
<td>61.91</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81.68</td>
<td>76.37</td>
<td>63.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>79.93</td>
<td>72.46</td>
<td>65.64</td>
<td></td>
</tr>
<tr>
<td>Ice</td>
<td>1</td>
<td>84.23</td>
<td>67.96</td>
<td>33.78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>82.83</td>
<td>76.53</td>
<td>43.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81.14</td>
<td>77.02</td>
<td>70.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>77.15</td>
<td>83.18</td>
<td>79.03</td>
<td></td>
</tr>
<tr>
<td>Soccer</td>
<td>1</td>
<td>82.55</td>
<td>76.63</td>
<td>50.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>80.02</td>
<td>74.73</td>
<td>62.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>76.97</td>
<td>69.72</td>
<td>63.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>74.27</td>
<td>64.92</td>
<td>59.74</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.6: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several CIF sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>RD point</th>
<th>Reduction (%)</th>
<th>DCT-16</th>
<th>DCT-8</th>
<th>DCT-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>1</td>
<td>85.42</td>
<td>76.82</td>
<td>62.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>85.59</td>
<td>76.42</td>
<td>28.37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>89.32</td>
<td>75.16</td>
<td>38.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>84.70</td>
<td>73.27</td>
<td>53.21</td>
<td></td>
</tr>
<tr>
<td>Foreman</td>
<td>1</td>
<td>86.58</td>
<td>76.08</td>
<td>68.34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>86.20</td>
<td>79.01</td>
<td>18.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>85.73</td>
<td>77.53</td>
<td>38.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>84.52</td>
<td>76.81</td>
<td>58.28</td>
<td></td>
</tr>
<tr>
<td>Mother &amp; Daughter</td>
<td>1</td>
<td>91.57</td>
<td>87.28</td>
<td>10.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>90.40</td>
<td>85.64</td>
<td>72.19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>89.23</td>
<td>78.34</td>
<td>69.51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>87.50</td>
<td>81.22</td>
<td>64.97</td>
<td></td>
</tr>
<tr>
<td>Soccer</td>
<td>1</td>
<td>87.25</td>
<td>84.15</td>
<td>68.55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>86.43</td>
<td>82.30</td>
<td>72.87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>85.01</td>
<td>78.73</td>
<td>72.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>83.63</td>
<td>74.03</td>
<td>68.66</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.7: % Reduction of iterations achieved by mNWRBP with respect to Flooding for several QCIF sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>RD point</th>
<th>Reduction (%)</th>
<th>DCT-8</th>
<th>DCT-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>1</td>
<td>78.96</td>
<td>49.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>81.42</td>
<td>51.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>79.92</td>
<td>38.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>80.03</td>
<td>42.32</td>
<td></td>
</tr>
<tr>
<td>Foreman</td>
<td>1</td>
<td>85.47</td>
<td>72.58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>83.22</td>
<td>75.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>83.61</td>
<td>66.87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>81.15</td>
<td>64.48</td>
<td></td>
</tr>
<tr>
<td>Mother &amp; Daughter</td>
<td>1</td>
<td>91.57</td>
<td>76.88</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>90.74</td>
<td>89.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>88.69</td>
<td>85.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>87.66</td>
<td>84.30</td>
<td></td>
</tr>
<tr>
<td>Soccer</td>
<td>1</td>
<td>87.46</td>
<td>80.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>87.02</td>
<td>80.84</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>85.80</td>
<td>77.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>84.38</td>
<td>76.01</td>
<td></td>
</tr>
</tbody>
</table>
Based on the data from Tables 6.5, 6.6 and 6.7 above, effect of neither scene characteristics, nor spatial resolutions on the mNWRBP performance can be deduced. The proposed algorithm claimed improvement across all sequences with diverse scene characteristics and spatial resolutions. From these results, it can however be seen that mNWRBP achieved the best improvement at larger DCT block sizes. This is because, larger DCT blocks means more, but shorter bit-planes than the smaller block size counterparts so the improved performance is due to the additional bit-planes being in class II discussed above. From these tables, it can be concluded that the proposed mNWRBP algorithm can save substantial resources and improve latency through significantly reducing the number of iterations compared to the flooding BP algorithm.

6.5.2 RD performance

In certain cases, the mNWRBP algorithm may prematurely terminate the message passing loop and request additional parity bits as in Case 3 explained in Subsection 6.3.5. Since the available parity bits would have been sufficient given enough iterations, the requested parity bits are in fact redundant and their transmission increases the overall bit-rate. Hence an analysis of the RD performance is necessary to critically evaluate the performance of the mNWRBP algorithm.

Figure 6.3 compares the RD performance of mNWRBP and Flooding for the three 4CIF resolution sequences, namely Crew, Ice and Soccer at DCT-16, DCT-8 and DCT-4 configurations which corroborates the above hypothesis of mNWRBP needing more bit-rate than Flooding. However, the former achieved identical PSNR for each RD point compared to the latter, thus, the proposed method avoids the issue of poor output quality that exists in some prior arts as discussed in Subsection 2.5.4 and Section 6.2. In fact, in all cases, mNWRBP at DCT-16 shows superior RD
performance than Flooding at DCT-4 which has been shown to perform similar to the DISCOVER codec in previous chapters. Therefore, employing the mNWRBP algorithm with larger DCT block sizes is advantageous than using the DISCOVER codec from an RD performance perspective. Similar trend has also been observed for CIF and QCIF test sequences whose corresponding RD curves are given in Appendix A for completeness.

![Diagram](image1.png)

**Figure 6.3:** RD performance curves generated by mNWRBP and Flooding for (a) Crew, (b) Ice and (c) Soccer 4CIF resolution sequence at different DCT block sizes.

mNWRBP requires more bit-rate to achieve the same PSNR because it prematurely judges some attempts to be failed and requests for redundant parity bits as explained in Case 3 of Subsection 6.3.5. The primary reason for this poor judgement is because some errors exist in the bit-plane that are difficult to correct and requires many iterations. For these errors, propagation of reliable messages towards corresponding variable nodes are slower, resulting in small changes between iterations. Since the
mNWRBP algorithm prioritises the nodes that show greater changes in messages between iterations, these erroneous nodes are assigned a lower priority, thus have less opportunity to propagate a message. This is an inherent weakness of the NWRBP algorithm and can be alleviated by adopting more advanced IDS strategies (Aslam et al., 2017; Liu et al., 2016, 2017). Moreover, longer bit-planes are more vulnerable to this weakness as they may contain a greater number of erroneous bits that are difficult to correct. As a result, the bit-rate penalty is less for larger DCT block sizes that produces shorter bit-planes. This is corroborated by the RD curves in Figure 6.3 where the bit-rate penalty incurred by DCT-4 is substantially more significant than the penalty incurred by DCT-16.

To study the effect of employing mNWRBP at different spatial resolutions, Table 6.8 shows the BD-rate metric (Bjontegaard, 2001) of mNWRBP with respect to Flooding for Soccer 4CIF, CIF and QCIF sequences. Positive values reflect that mNWRBP requires higher bit-rate than Flooding to achieve the same PSNR as observed from the RD curves in Figure 6.3. The bit-rate overhead is less for smaller resolution variants than larger ones for a specific DCT block size. These results corroborate the aforementioned findings that the bit-rate penalty is lesser for shorter bit-plane length.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>BD-rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCT-16</td>
</tr>
<tr>
<td>4CIF</td>
<td>21.86</td>
</tr>
<tr>
<td>CIF</td>
<td>5.65</td>
</tr>
<tr>
<td>QCIF</td>
<td>5.82</td>
</tr>
</tbody>
</table>

It can also be observed from Table 6.8 that bit-rate penalty is the lowest for larger DCT block sizes than their smaller counterparts, which correspondingly shows the strongest improvement as evident from Tables 6.5, 6.6 and 6.7. Moreover, larger DCT block sizes show improved RD performance than the DISCOVER as revealed in
Section 5.3. Therefore, adopting the mNWRBP algorithm can be advantageous in certain scenarios despite the bit-rate overhead. The primary aim of the proposed algorithm is to reduce latency and it is evident from the results in Tables 6.5, 6.6 and 6.7 that the mNWRBP has considerable potential to fulfil this aim as it consistently reduces the number of BP iteration required to decode a sequence. For some configurations, the reduction is over 90% so the advantage of reduced latency can persuasively be argued to outweigh the cost of extra bandwidth especially for 16×16 block size at CIF resolution and 8×8 block size at QCIF resolution.

Table 6.9: % Reduction of iterations achieved by mNWRBP with respect to Flooding for bit-planes corresponding to DC coefficients of Soccer 4CIF sequences

<table>
<thead>
<tr>
<th>RD point</th>
<th>Reduction (%)</th>
<th>DCT-16</th>
<th>DCT-8</th>
<th>DCT-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>59.06</td>
<td>37.38</td>
<td>48.32</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>68.15</td>
<td>56.98</td>
<td>27.22</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>67.49</td>
<td>64.43</td>
<td>51.93</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>66.70</td>
<td>65.79</td>
<td>63.29</td>
</tr>
</tbody>
</table>

6.5.3 Effect on parallel processing scenario

Table 6.9 records the number of iterations required to decode bit-planes from only the DC coefficient band for the Soccer 4CIF sequence. This is to evaluate the advantage of the mNWRBP over flooding BP algorithm in parallel processing scenarios. Since bit-planes from the same coefficient band are sequentially decoded and the DC coefficient band is generally allocated the highest number of bit-planes, it signifies the decoding bottleneck that a parallel processing decoder must complete. Thus, the performance improvement can be estimated from the number of iterations incurred in decoding the DC coefficient band. It can be seen from Table 6.9 that the mNWRBP algorithm maintain its advantage of requiring a significantly smaller number of iterations compared to the flooding BP algorithm. Though this advantage is less than corresponding figures when comparing with Table 6.5, the proposed algorithm is more
suitable for parallel processing scenarios than the flooding BP due to its potential in reducing latency.

The mNWRBP algorithm requires a significantly smaller number of iterations compared to flooding BP algorithm to decode a bit-plane, thus potentially reduces latency. Rigorous quantitative analysis has proven this for a number of test sequences of varying spatial resolutions and DCT block sizes. Though each iteration of the mNWRBP algorithm is computationally costlier than the flooding BP algorithm, and incurs bit-rate overhead, its potential advantages in terms of reduced latency in both serial and parallel processing scenarios outweigh the drawbacks especially at larger DCT block sizes.

6.6 Summary

This chapter has presented a novel approach to address the latency issue of the DVC-HR framework that escalates at higher spatial resolution and larger DCT block sizes. The approach involves adopting an advanced BP decoding algorithm, namely the NWRBP algorithm that promises faster convergence of BP iterations. The algorithm is modified to leverage from the faster convergence property by proposing a new stopping criterion. The modified NWRBP (mNWRBP) is implemented in the DVC-HR framework and simulated with a range of test sequences and different DCT block sizes configurations. Critical analysis of the performance results shows that the mNWRBP algorithm consistently and significantly reduced the total number of iterations required to decode a sequence. Although each iteration is slightly more complex than an iteration of the traditional flooding BP algorithm, the reduction is much more substantial. This reduction comes at the cost of increased bit-rate, however the benefits outweights the limitations especially when employing larger DCT block sizes. Besides, the proposed method avoids the poor output quality issue that limits
applicability of existing solutions. Moreover, the mNWRBP algorithm is also suitable for parallel processing scenarios where it maintains its advantage of a significantly smaller number of iterations, thus potentially reduces latency.
7 Future Work

The preceding chapters have presented the DVC-HR framework featuring original contributions consisting of introduction of the new CAQ module and modifications of existing transform and channel coding modules of the benchmark DISCOVER codec. As a result, the novel framework exhibits significant RD performance improvement, lower latency and superior rate-control flexibility especially at higher spatial resolutions. The DVC-HR framework does have some limitations however, which can prevent its deployment in certain application scenarios. Firstly, following accepted community practice, DVC-HR considers only the luminance information of a video test sequence for coding and ignores the chrominance components which may be essential in some applications. Moreover, the QoS configuration is controlled by the ABP value which depends on the DCT block size, so it is not a readily intuitive relationship. In addition, while the mNWRBP algorithm can significantly reduce latency, it incurs a bit-rate overhead, especially for longer bit-planes which negatively impacts on RD performance. Above all, the testbed implementation of the framework has been developed for experimental purposes and so is currently not appropriate for real-time coding scenarios.

While collectively these may constrain the application of the DVC-HR framework, they also provide rich opportunities to enhance and extend the framework. This chapter explains some of the more promising avenues which merit further critical investigation.

1. The most important contribution of this thesis is the introduction of CAQ module which enables superior rate-control flexibility as well as larger DCT
block sizes in the DVC-HR framework. An interesting feature of the new design is discussed in Subsection 4.4.3 where it has been shown how the ABP in the novel framework can adapt to scene changes to achieve Constant Bit-Rate (CBR) coding and produce maximum quality output within the available bandwidth limit. The process however, is theoretical and involves manually selecting the ABP for the maximum quality of each WZF within the bandwidth limit from a results database. For a practical realisation of the framework, this procedure should be automated and to achieve this, accurate estimation of the compression ratio of each bit-plane of a WZF is essential. In DVC, compression ratio of each bit-plane and consequently, overall bit-rate depends on the difference between a WZF and corresponding SI. The primary challenge here is that neither the encoder nor the decoder has access to both WZF and SI and as a result, their difference needs to be estimated. The correlation information from the CNM module can help in this regard. A converse approach of encoder rate control based DVC schemes can also be adopted (Hu et al., 2018; Brites and Pereira, 2011). Once compression ratio of each bit-plane is obtained, the ABP value can then be iteratively adjusted to achieve desired overall bit-rate, thus replacing the manual selection process.

2. A converse scenario of the aforementioned CBR coding is where the output quality is crucial but the bit-rate requirement can be relaxed. This is analogous to the selection of quantisation parameter of AVC or the quality settings of JPEG coding. The basic idea is that since the distortion in output is a direct consequence of quantisation of transform coefficients, it can be regulated by controlling the degree of quantisation. The DVC-HR framework partially simulates this by supplying an ABP value to the CAQ module to determine a
QM. However, ABP does not directly control the degree of quantisation which varies depending on the content of the frame. In fact, the degree of quantisation is determined by the *Scaling Factor* (SF) which is a real number in the range $[0,1]$ as explained in Section 4.3. Therefore, a better way to accomplish this purpose is to use the SF instead of ABP while determining a QM. However, this is not trivial as the relation between the SF and the distortion is not linear. Moreover, output of the DVC-HR framework is reconstructed using the SI, thus the distortion also depends on the quality of the SI. Therefore, a new parameter needs to be defined which considers both the SF and SI quality, and a new algorithm to be developed to regulate the parameter to reliably control distortion.

3. The performance results presented in this thesis have been obtained by simulating the DVC-HR framework for various test sequences of YUV format as explained in Subsection 3.5.1. Following the community practice, only the luminance Y channel is simulated and the chrominance U and V channels are discarded. Colour information in the U and V channels is essential for many application scenarios, thus must be processed by the framework. However, special transformation strategy is required for this purpose since the chrominance channels are subsampled in test sequences. For instance, in the AVC the chrominance channels are Hadamard transformed using a smaller block size (Sullivan et al., 2004; Wiegand et al., 2003). This has not been possible in the DISCOVER codec (and its derivatives) due to only one block size being available. The subsampled chrominance channels produce very short bit-planes which may not be realised due to limitations of the LDPCA coder as detailed in Subsection 5.2.2. However, since CAQ allows any DCT
block size, the subsampled channels can be transformed using proportionally smaller blocks to produce consistent bit-plane length. This will require further modification of the framework similar to Section 5.2.

4. The mNWRBP algorithm integrated in the DVC-HR framework is a proof-of-concept to illustrate the benefits of a more efficient decoding of bit-planes. The potential reduction of the latency between encoder and decoder that hinders practical DVC applications is promising, hence merits further investigation. An inherent weakness of the original NWRBP algorithm is that it greedily prioritises message propagation in one portion of the bit-plane while the other portion remains silent. This may cause the mNWRBP algorithm to incorrectly judge a failed decoding attempt and request for redundant parity bits. As a result, the proposed scheme incurs greater bit-rate overheads especially for higher resolution sequences that produce longer bit-planes as discussed in Subsection 6.5.2. This can be alleviated by employing a more advanced IDS strategy available in the literature (Aslam et al., 2017; Liu et al., 2017) that are free from the aforesaid weakness. Their performances need to be evaluated and their applicability and effectiveness be studied before they can be integrated in the DVC-HR framework. Further research opportunity also exists in determining properties of the relationship between the CNM module and the mNWRBP algorithm. The former generates soft inputs which are used by the latter as the basis of its dynamic scheduling. The DVC-HR framework presented in this thesis uses a Laplacian model (Brites and Pereira, 2008) whose analysis may lead to predict the scheduling outcome. A different model, (Yin et al., 2015) for instance, may be recognised as more suitable for the mNWRBP algorithm. Furthermore, the lack of an efficient implementation of
the mNWRBP algorithm as pointed out in Chapter 6 hinders accurate evaluation of its effectiveness in reducing the latency between encoder and decoder, and also influences analysis of its real-time performance. A prospective future work would be to develop a reference implementation of the algorithm to facilitate precise evaluation and analysis of its performance.

5. The DVC-HR framework has exhibited crucial performance advantages over the DISCOVER codec by introducing adaptive quantisation through the CAQ module, better exploitation of spatial redundancies via employing larger DCT block sizes, and improved latency achieved by the mNWRBP algorithm. However, some key shortcomings must be resolved before it can be a practical DVC solution capable of replacing a traditional codec. Firstly, a practical implementation must be able to perform real-time and adopting a parallel processing architecture is essential for this purpose (Shen et al., 2017). All three original contributions can be expedited through parallel processing. The QM generation process of the CAQ module described in Section 4.3 can be parallelised to make it faster and more efficient. Since DCT blocks are non-overlapping, individual blocks can be transformed in parallel. Similar parallel processing can be performed for quantisation of DCT coefficients. The mNWRBP algorithm can also be implemented for parallel processing architectures in a similar manner explained by Casado et al. (2007) for the original NWRBP algorithm, thus further reducing the latency. Moreover, innovations in the DVC-HR framework does not cover two integral modules of the DVC architecture, namely, the SI generation and the CNM modules. Though several good solutions exist in the literature as reviewed in Section 2.5, they are verified for low resolution sequences only. Their performance
and scalability need to be validated for higher resolution sequences. One important point of concern is their complexity and accuracy in relation to granularity, which may become a bottleneck. Generally, increasing granularity improves accuracy, but also escalates the complexity of the process. A good balance between the two must be obtained. A parallel processing solution for the SI generation and the CNM modules also merits investigation in this regard.
8 Conclusion

In contrast to traditional video coding techniques, Distributed Video Coding (DVC) is an alternate paradigm that shifts the computational burden of exploiting inter-frame redundancies to the decoder, so making the encoder a simpler design, but at the cost of increased decoder complexity. This complexity redistribution is attractive for low-power applications where traditional encoders cannot be employed due to a scarcity of processing resources. Current DVC models however, have yet to achieve the same theoretically proven Rate-Distortion (RD) performance as traditional codecs and while they exhibit superior performance to traditional low-complexity, intra-frame codecs, this has only been shown for very low spatial resolution sequences which is unrealistic for modern applications and services. In higher resolution sequences, the RD performance of DVC deteriorates along with the emergence of other limitations such as inflexible bit-rate control and significant latency issues. This provided the motivation to investigate how new DVC techniques can enable the proven coding advantages to be retained at higher spatial resolutions while mitigating some of the inherent DVC framework limitations.

This thesis has presented a novel DVC for Higher Resolutions (DVC-HR) framework that consistently exhibits superior RD performance at higher spatial resolutions, by supporting enhanced bandwidth usage through flexible rate control options, and reducing the accumulated latency between encoder and decoder. Using the DISCOVER architecture as its origin, the DVC-HR framework makes the following three original scientific contributions to the field.
1. The most significant contribution is the development of a *Content-Aware Quantisation* (CAQ) mechanism to facilitate superior bit rate-control options by dynamically generating a *Quantisation Matrix* (QM) according to the available bandwidth. The CAQ module is embedded into the DVC-HR decoder to determine the best QM by analysing both the dynamic range distribution and perceptibility of DCT coefficients in the *Wyner-Ziv Frames* (WZF). The generated QM is then fed back to the encoder via the feedback channel to quantise the WZF, so CAQ does not impact upon encoder complexity while imposing negligible feedback channel overheads. CAQ offers a number of benefits over the existing pre-determined QM scheme. Firstly, it affords flexible bit-rate control which makes it significantly easier to uphold the desired QoS level. It also allows coding at extreme bit-rates where existing solutions fail and is inherently capable of adapting to changes in scene characteristics and bandwidth constraints, to fully utilise the available bandwidth and concomitantly provide maximum output quality. Critical analysis of the performance of the CAQ mechanism reveals consistent utilisation of the available bandwidth leads up to 1.8dB average PSNR gain for some sequences compared to existing QM schemes. Most importantly, CAQ can be used to generate a QM for different DCT block sizes, thus supporting the realisation of larger DCT block sizes within the DVC-HR framework.

2. The second contribution is a critical investigation of using larger DCT block sizes to better exploit WZF spatial redundancies and improve the coding performance of higher resolution sequences. While existing DVC frameworks use only 4×4 DCT block sizes, the DVC-HR framework supports both 8×8
and 16×16 block sizes. These larger blocks require appropriately sized QM which are generated by the CAQ module developed in contribution #1. Rigorous quantitative analysis of the results corroborates that larger DCT block sizes consistently outperformed their smaller counterparts with, on average, more than a 2dB PSNR gain. Furthermore, the performance improvement is more substantial at higher resolution variants in coding the same sequence. Additionally, larger block sizes increase the number of bit-planes while decreasing their length, so making them better suited for parallel processing realisations. Since the DCT is applied by both the encoder and decoder, using larger block sizes does impact on encoder complexity, however, the extra computations are very reasonable for current hardware and the coding benefits derived more than pragmatically offset this cost.

3. The third contribution addresses the accumulated latency issue between the encoder and decoder of the DVC-HR framework. The feedback-based design of the DVC architecture can be inherently prone to accumulative processing delays, which is compounded by the recursive nature of the LDPCA decoding process, particularly for larger DCT block sizes. To resolve this problem, a modified version of the fast converging Node-Wise Residual Belief Propagation (NWRBP) algorithm has been developed for the LDPCA decoder that intelligently detects a failed attempt earlier and efficiently requests, when necessary, further parity bits. This lowers the overheads incurred in failed decoding attempts so significantly reducing the overall latency. Rigorous critical analysis of results substantiates the superiority of the modified NWRBP (mNWRBP) algorithm over the existing classic belief propagation algorithm, requiring more than 90% fewer iterations to decode the same set of bit-planes.
The mNWRBP algorithm also performs comparatively better for both larger DCT block sizes and shorter bit-planes and while mNWRBP can sometimes lead to false positives by incorrectly identifying failed decoding attempts and thus requesting redundant parity bits, this was found to be infrequent. Indeed the advantages of reduced latency notably outweigh the very small extra bit-rate incurred, especially in processing larger DCT block sizes.

The new DVC-HR framework makes an important advance to the DVC field by offering an efficient and robust solution to narrow the oft-quoted coding performance gap. When set against alternatives, this facilitates an effective DVC realisation for processing higher spatial resolution sequences to meet the increasing demand for modern applications and services. The outcomes formulated focus on key components of the DVC pipeline, with the algorithms developed being portable so they can be incorporated into other DVC architectures, to enhance their coding functionality and performance.
References


Appendices

A. Supplementary Results for the mNWRBP RD Performance

This appendix presents RD curves comparing mNWRBP performance with Flooding for several QCIF and CIF test sequences in Figures A.1 and A.2 respectively to complement the results presented in Subsection 6.5.2.

![RD performance curves](image)

Figure A.1: RD performance curves generated by mNWRBP and Flooding for (a) Coastguard, (b) Foreman, (c) Mother & Daughter and (d) Soccer QCIF resolution sequence at different DCT block sizes
Figure A.2: RD performance curves generated by mNWRBP and flooding for (a) Bus, (b) Coastguard, (c) Crew, (d) Football, (e) Foreman, (f) Hall, (g) Mother & Daughter and (h) Soccer CIF resolution sequence at different DCT block sizes.