



The  
University  
Of  
Sheffield.

# **High Efficiency Video Coding (HEVC) tools for next generation video content**

By:

**Yanxiang Wang**

A thesis submitted in partial fulfilment of the requirements for the degree  
of Doctor of Philosophy

The University of Sheffield  
Department of Electronic and Electrical Engineering

April 2016



# ACKNOWLEDGEMENTS

I am grateful to my parents for providing me with a good education environment and supporting me studying abroad. I wish to thank my supervisor Dr Charith Abhayaratne for his supervision of my research. I would like to thank Engineering and Physical Sciences Research Council (EPSRC) and BBC R&D for the financial support of this i-CASE project. I would like to thank Dr Marta Mrak and Dr Rajitha Weerakkody for their supervision during my internship at BBC R&D. I wish to thank my colleagues Mr Wattanit Hotrakool and Mr Sandipan Pal for helping me with the software testing for the subjective evaluation experiments. I would also like to thank Mr Steve Marsden for technical support, Mrs Dianne Webster for providing risk assessment support, Mrs Hilary Levesley for her excellent services as a research student support officer, and colleagues at the Communication Group and BBC R&D.

Special thanks to 35 students from the University of Sheffield who volunteered to participant in the visual quality evaluation tests.

# Abstract

The emergence of high resolution video content together with an increasingly broad range of content types demands highly efficient solutions for video coding. The new types of video content are defined as next generation content. *High Efficiency Video Coding* (HEVC) is emerging as an effective video compression technique that has been developed for new formats and new types of video content. This thesis aims to develop coding tools in HEVC for two types of next generation video content: screen content and *Ultra High-definition* (UHD) content, to improve the coding efficiency or the overall visual quality. For screen content, several colour formats are discussed and evaluated by comparing against natural content. The purpose is to convert the input RGB 4:4:4 video sequence into a luma-chroma model and maximise the de-correlation of the colour components. On the basis of the obtained results, a new design of the lifting-based Predict-and-Update colour conversion scheme is proposed for screen content. One of the proposed method achieved PSNR BD-rate saving up to 8.86%. Contouring is one of the artefacts in HEVC compressed UHD content. To reduce the contouring artefacts in UHD content, there are three methods proposed in this thesis. Multi-scale dithering is a contouring removal method applied as a post-processing method. Pixel mapping is another algorithm designed for contouring artefacts removal. It can be implemented as either a pre-codec or an in-codec approach. Both objective and subjective quality metrics are used to evaluate the performance of the proposed coding solutions. The results show that the proposed dithering method achieved the MOS BD-rate saving up to 85%, and one of the proposed pixel mapping method achieved the MOS BD-rate saving from 23% to 78%.

## Nomenclature

AI	<i>All Intra</i>
AMVP	<i>Adaptive Motion Vector Prediction</i>
AVC	<i>Advanced video coding (H.264)</i>
B	<i>Bi-predictive</i>
BBC R&D	<i>British Broadcasting Corporation, Research and development</i>
BD-PSNR	<i>Bjontegaard Delta PSNR</i>
BD-rate	<i>Bjontegaard Delta rate</i>
BO	<i>Band Offset</i>
CABAC	<i>Context Adaptive Binary Arithmetic Coding</i>
CAVLC	<i>Context Adaptive Variable-Length Coding</i>
CTU	<i>Coding Tree Unit</i>
CU	<i>Coding Unit</i>
DCR	<i>Degradation Category Rating</i>
DCT	<i>Discrete Cosine Transform</i>
DSIS	<i>Double Stimulus Impairment Scale</i>
DST	<i>Discrete Sine Transform</i>
EBU	<i>European Broadcasting Union</i>
EO	<i>Edge Offset</i>
HD	<i>High-Definition</i>
HDR	<i>High Dynamic Range</i>
HDTV	<i>High-Definition Television</i>
HEVC	<i>High Efficiency Video Coding</i>
HVS	<i>Human Visual System</i>
I	<i>Intra</i>
ITU	<i>International Telecommunications Union</i>
ITU-R	<i>Radio communication Sector of ITU</i>
ITU-T	<i>ITU Telecommunication Standardisation Sector</i>
JCTVC	<i>Joint Collaborative Team on Video Coding</i>

KLT	<i>Karhunen–Loeve Transform</i>
LDB	<i>Low Delay with B slices</i>
MOS	<i>Mean Opinion Score</i>
P	<i>Predicted</i>
P-DCT	<i>Pseudo Discrete Cosine Transform</i>
PMIC	<i>Pixel Mapping In-codec</i>
PPU	<i>Proposed colour space transform (prediction-and- update based)</i>
PSNR	<i>Peak Signal to Noise Ratio</i>
PU	<i>Prediction Unit</i>
QP	<i>Quantisation Parameter</i>
RA	<i>Random Access</i>
RD	<i>Rate Distortion</i>
RGB	<i>Red, green and blue colour components</i>
RQT	<i>Residual Quad Tree</i>
SAO	<i>Sample Adaptive Offset</i>
SC	<i>Screen Content</i>
SDTV	<i>Standard Definition Television</i>
SHV	<i>Super Hi-Vision</i>
SJTU	<i>Shanghai Jiao Tong University</i>
SSIM	<i>Structural Similarity</i>
TU	<i>Transform Unit</i>
UHD	<i>Ultra High-Definition</i>
UHDTV	<i>Ultra High-Definition Television</i>
VCEG	<i>Video Coding Experts Group</i>
VCEG	<i>Moving Picture Experts Group</i>
VCL	<i>Variable-Length Coding</i>
VQM	<i>Video Quality Metric</i>
YCbCr	<i>Colour space consist of luminance Y and two chrominance channels Cb and Cr</i>
YCoCg	<i>Colour space consist of luminance Y and two chrominance channels Cb and Cg</i>

## List of Symbols

$k_r, k_b$	<i>Luma coefficients</i>
$\mathbf{R}_{AA}$	<i>Covariance matrix</i>
$\mathbf{R}_{KK}$	<i>Autocorrelation matrix</i>
$\mathbf{K}$	<i>De-correlated matrix</i>
$\mathbf{V}$	<i>Modal matrix, columns are the Eigenvectors</i>
$\mathbf{M}$	<i>Diagonal matrix with three Eigenvalues on the main diagonal</i>
$\mathbf{V}^T$	<i>Transposed <math>\mathbf{V}</math></i>
$\alpha, \beta$	<i>Updating parameters</i>
$\mu_x (x = A, \dots, I)$	<i>Average pixel value of each block (<math>A, \dots, I</math>)</i>
$\bar{\mu}$	<i>Average of <math>\mu_x</math>.</i>
$\lambda_x$	<i>Difference between <math>\mu_x</math> and <math>\bar{\mu}</math></i>
$c$	<i>Local contrast</i>
$\mathcal{M}_{32}, \mathcal{M}_4$	<i>Contouring maps</i>
$N$	<i>Block size</i>
$\delta$	<i>Adjustment value in pixel dithering</i>
$G(\delta)$	<i>The gradient variation function</i>
$E(\delta)$	<i>The block boundary smoothness function</i>
$h, v$	<i>Sum of pixel differences in horizontal and vertical directions</i>
$A$	<i>Original pixel value</i>
$B$	<i>Mapped pixel value</i>
$a_1$	<i>Threshold values (minimum)</i>
$a_2$	<i>Threshold values (maximum)</i>
$b_1$	<i>Mapped value of <math>a_1</math></i>
$b_2$	<i>Mapped value of <math>a_2</math></i>
$d$	<i>Duration of pixel range (<math>a_2 - a_1</math>)</i>
$d'$	<i>Duration of pixel range after Pixel Mapping (<math>b_2 - b_1</math>)</i>
$P$	<i>Peak value of histogram of contouring area</i>
$M(), M^l()$	<i>Pixel mapping and inverse pixel mapping functions</i>

$P()$                       *Prediction operation in lifting scheme*  
 $U()$                       *Updating operation in lifting scheme*

## List of Figures

<i>Figure 1.1 Improvements in coding performance introduced by new standards [18].</i>	7
<i>Figure 2.1 Macroblock partitioning in H.264/AVC.</i>	13
<i>Figure 2.2 Block diagram of HEVC encoder and decoder.</i>	16
<i>Figure 2.3 CTU splitting and related quadtrees.</i>	17
<i>Figure 2.4 Partitioning modes for PUs.</i>	18
<i>Figure 2.5 A CTU split with top-right CU split into two PUs using PART_Nx2N partitioning and one level of TU split (dotted black lines).</i>	19
<i>Figure 2.6 Luma intra prediction modes of HEVC and H.264/AVC.</i>	20
<i>Figure 2.7 Basis patterns for 4 × 4 DCT.</i>	21
<i>Figure 2.8 Basis patterns for 8 × 8 DCT.</i>	22
<i>Figure 2.9 Basis patterns for 4 × 4 DST.</i>	23
<i>Figure 2.10 Screen content frames (Slide Showing).</i>	27
<i>Figure 2.11 Screen content frames residuals analysis.</i>	28
<i>Figure 2.12 Natural content frames (Basketball Pass)</i>	29
<i>Figure 2.13 Natural content frames residuals analysis.</i>	30
<i>Figure 2.14 Pixel change activity comparisons in screen content and natural content.</i>	31
<i>Figure 2.15 BD-rate saving comparisons (Random Access-Main).</i>	36
<i>Figure 2.16 BD-rate saving comparisons (on average).</i>	37
<i>Figure 2.17 Comparison of HEVC and H.264/AVC coding performance (all intra). (1) People On street (Class A), (2) Basketball Drive (Class B), (3) Race Horses (Class C), (4) Kimono (Class D), (5) Four People (Class E), (6) Slide Editing (Class F).</i>	41
<i>Figure 2.18 BD-rate results of reduced block sizes (16 × 16).</i>	44
<i>Figure 2.19 BD-rate results of reduced block sizes (16 × 16) on average.</i>	45
<i>Figure 2.20 BD-rate results of reduced block sizes (8 × 8).</i>	47
<i>Figure 2.21 BD-rate results of reduced block sizes (8 × 8) on average</i>	48
<i>Figure 2.22 Comparison of HEVC with different maximal CU sizes on LDB-Main configuration. (1) Kimono (Class B), (2) BQ Mall (Class C), (3) Basketball Pass (Class D), (4) Slide Show (Class F).</i>	50

<i>Figure 2.23 Comparison of HEVC without TS and HEVC standard (all intra). (1) Basketball Drill Text, (2) China Speed, (3) Slide Editing, (4) Slide Show.</i>	54
<i>Figure 3.1 Flow graphs of the RGB ↔ YCbCr colour conversion.</i>	58
<i>Figure 3.2 Flow graphs of the RGB ↔ YCoCg colour conversion.</i>	61
<i>Figure 3.3 Lifting scheme of the wavelet transform.</i>	68
<i>Figure 3.4 Flow graph of 2-channel lifting scheme.</i>	70
<i>Figure 3.5 Flow graph of 3-channel lifting scheme.</i>	71
<i>Figure 3.6 Flow graph of the RGB ↔ 3-channel PPU lifting colour conversion.</i>	73
<i>Figure 3.7 First frames of each test sequence.</i>	78
<i>Figure 3.8 An example of test sequence configuration (Kimono).</i>	80
<i>Figure 3.9 Block diagram of experiments.</i>	81
<i>Figure 3.10 R-D performance for “Kimono”.</i>	88
<i>Figure 3.11 R-D performance for “Park Scene”.</i>	88
<i>Figure 3.12 R-D performance for “Ducks&amp;Legs”.</i>	89
<i>Figure 3.13 R-D performance for “SC_map”.</i>	89
<i>Figure 3.14 R-D performance for “SC_pab layout”.</i>	90
<i>Figure 3.15 R-D performance for “SC_ppt_doc_xls”.</i>	90
<i>Figure 3.16 BD-rate comparisons of colour transforms.</i>	92
<i>Figure 3.17 First frame of each screen content test sequence.</i>	94
<i>Figure 3.18 BD-rate comparisons of colour transforms for screen content (negative value indicate BD-rate saving).</i>	95
<i>Figure 4.1 Example of contouring artefacts in Ningyo sequence.</i>	100
<i>Figure 4.2 Three contouring reduction approaches.</i>	102
<i>Figure 4.3 Experiment result by using method proposed in [55].</i>	104
<i>Figure 4.4 Test result of applying method proposed in [57].</i>	105
<i>Figure 4.5 Original and compressed frames in Ningyo sequence (luma component).</i>	107
<i>Figure 4.6. Luminance signal from original frame and compressed frame.</i>	107
<i>Figure 4.7 Relative locations of current block E and its neighbouring blocks.</i>	108
<i>Figure 4.8 Decoded frames and contouring maps.</i>	110
<i>Figure 4.9 Test result of pixel dithering on 4 × 4 blocks only</i>	113
<i>Figure 4.10 Overall procedure of the proposed contouring removal algorithm.</i>	114
<i>Figure 4.11 An example of high frequency component dithering (4 × 4).</i>	115
<i>Figure 4.12 Decoded frame (a) and processed frame after dithering (b).</i>	115
<i>Figure 4.13 Test room setup.</i>	120

<i>Figure 4.14 Presentation structure of test session and each video clip pair.</i>	121
<i>Figure 4.15 Frames before (left) and after dithering (right).</i>	122
<i>Figure 4.16 The results of Jin's method (left) and proposed method (right) of No Sleep sequence (cropped frame).</i>	123
<i>Figure 4.17 Objective metric PSNR results.</i>	124
<i>Figure 4.18 Objective metric SSIM results.</i>	124
<i>Figure 4.19 Subjective evaluation results.</i>	127
<i>Figure 4.20 MOS difference at various bit rates.</i>	128
<i>Figure 4.21 The horizontal profile (part) of the luminance signal for the selected line draws from Ningyo sequence.</i>	130
<i>Figure 5.1 The structure of pixel mapping.</i>	131
<i>Figure 5.2 Histogram of the pixels in the contouring regions (X axis: pixel value; Y axis: number of pixels).</i>	133
<i>Figure 5.3 The correlation of pixel values before and after reconstruction.</i>	135
<i>Figure 5.4 Contouring maps of original and decoded frames.</i>	138
<i>Figure 5.5 Overview of proposed pre-codec method.</i>	139
<i>Figure 5.6 Block diagram of HEVC.</i>	140
<i>Figure 5.7 Block diagram of proposed PMIC- A method.</i>	141
<i>Figure 5.8 Block diagram of proposed PMIC-B method.</i>	143
<i>Figure 5.9 Subjective test results of proposed PMIC-A method.</i>	145
<i>Figure 5.10 Subjective tests result of proposed PMIC-B method.</i>	146
<i>Figure 5.11 Subjective evaluation results.</i>	147
<i>Figure 5.12 Objective test results.</i>	150
<i>Figure 5.13 Subjective test results.</i>	151
<i>Figure 5.14 PSNR and MOS BD-rate results comparison.</i>	153
<i>Figure 5.15 Residual map analysis (Ningyo).</i>	154
<i>Figure 5.16 Residual map analysis (Badminton).</i>	155
<i>Figure 5.17 Residual values analysis from <math>8 \times 8</math> blocks (Badminton).</i>	156
<i>Figure 5.18 Residual map analysis (Bund Nightscape).</i>	157

## List of Tables

<i>Table 1.1 Characteristics comparisons.</i>	5
<i>Table 2.1 Comparisons of video compression standards.</i>	15
<i>Table 2.2 BD-rate results of luma component for HEVC compared to H.264/AVC.</i>	35
<i>Table 2.3 BD-rate average results for luma component for HEVC compared to H.264/AVC.</i>	37
<i>Table 2.4 BD-rate results for luma component for HEVC with maximal sizes of CU of 16 × 16 pixels, compared to typical HEVC settings with maximal CU size of 64 × 64 pixels.</i>	43
<i>Table 2.5 Average BD-rate for luma component for HEVC with maximal sizes of CU of 16 × 16 pixels, compared to typical HEVC settings with maximal CU size of 64 × 64 pixels.</i>	45
<i>Table 2.6 BD-rate results for luma component for HEVC with maximal sizes of CU of 8 × 8 pixels, compared to typical HEVC settings with maximal CU size of 64 × 64 pixels.</i>	46
<i>Table 2.7 Average BD-rate for luma component for HEVC with maximal sizes of CU of 8 × 8 pixels, compared to typical HEVC settings with maximal CU size of 64 × 64 pixels.</i>	48
<i>Table 2.8 BD-rate results for luma component for HEVC without transform skip compared to HEVC with transform skip (common test conditions)</i>	52
<i>Table 3.1 KLT matrix for each video sequence.</i>	65
<i>Table 3.2 KLT approximate matrices.</i>	66
<i>Table 3.3 Parameter setting of four PPU transforms.</i>	74
<i>Table 3.4 Test sequences.</i>	78
<i>Table 3.5 Results of colour transform comparison (kimono).</i>	82
<i>Table 3.6 Results of colour transform comparison (Park Scene).</i>	82
<i>Table 3.7 Results of colour transform comparison (Ducks&amp;Legs).</i>	82
<i>Table 3.8 Results of colour transform comparison (SC_map).</i>	83
<i>Table 3.9 Results of colour transform comparison (SC_pcb_layout).</i>	83
<i>Table 3.10 Results of colour transform comparison (SC_ppt_doc_xls).</i>	83

<i>Table 3.11 BD-rate of RGB components for HEVC with YCoCg compared to HEVC with YCbCr.</i>	85
<i>Table 3.12 BD-PSNR of RGB components for HEVC with YCoCg compared to HEVC with YCbCr.</i>	85
<i>Table 3.13 BD-rate of RGB components for HEVC with KLT compared to HEVC with YCbCr.</i>	85
<i>Table 3.14 BD-PSNR of RGB components for HEVC with KLT compared to HEVC with YCbCr.</i>	86
<i>Table 3.15 BD-rate of RGB components for HEVC with 3-channel PPU-1 compared to HEVC with YCbCr.</i>	86
<i>Table 3.16 BD-PSNR of RGB components for HEVC with 3-channel PPU compared to HEVC with YCbCr.</i>	86
<i>Table 3.17 BD-rate of RGB components for HEVC with 3-point P-DCT compared to HEVC with YCbCr.</i>	87
<i>Table 3.18 BD-PSNR of RGB components for HEVC with 3-point P-DCT compared to HEVC with YCbCr.</i>	87
<i>Table 3.19 Comparisons of BD-rate of different colour transforms in HEVC (YCbCr is the reference).</i>	91
<i>Table 3.20 Comparisons of BD-PSNR(dB) of different colour transforms in HEVC (YCbCr is the reference).</i>	92
<i>Table 3.21 KLT matrix MSE</i>	93
<i>Table 3.22 Comparisons of BD-rate of different colour transforms for screen content in HEVC (YCbCr is the reference).</i>	95
<i>Table 4.1 Contouring block analysis (luma component).</i>	101
<i>Table 4.2 The specifications of test UHD sequences.</i>	111
<i>Table 4.3 Test conditions.</i>	119
<i>Table 4.4 Objective evaluation results.</i>	123
<i>Table 4.5 Test sequence contouring features.</i>	125
<i>Table 4.6 Subjective evaluation results of the experiments.</i>	126
<i>Table 4.7 The processed result of each step and the corresponding noise map.</i>	129
<i>Table 5.1 Symbols and descriptions.</i>	132
<i>Table 5.2 Peak pixel value of histogram in contouring area.</i>	134
<i>Table 5.3 The pixel mapping algorithm.</i>	134
<i>Table 5.4 The pixel inverse mapping.</i>	136

<i>Table 5.5. BD-rate results of Badminton sequence with different parameter setups. ...</i>	137
<i>Table 5.6 Test conditions and parameter setup. ....</i>	144
<i>Table 5.7 Subjective test results MOS BD-rate. ....</i>	148
<i>Table 5.8 BD-rate results of PMIC. ....</i>	152
<i>Table 5.9 Cluster configurations. ....</i>	158
<i>Table 5.10 BD-rate results of different parameters. ....</i>	158

# Contents

Nomenclature .....	I
List of Symbols .....	III
List of Figures .....	V
List of Tables .....	VIII
Chapter 1 Introduction.....	1
1.1 Video compression background.....	2
1.1.1 Video coding history.....	2
1.1.2 Next generation video content .....	4
1.1.3 New market requirements .....	6
1.2 Aims and objectives.....	7
1.3 Original contribution.....	9
1.4 Author's publications.....	11
1.5 Organisation of the thesis.....	11
Chapter 2 HEVC analysis and evaluations.....	12
2.1 The overview of H.264/AVC.....	12
2.2 The Overview of HEVC .....	14
2.2.1 Picture partitions .....	16
2.2.2 Intra prediction.....	19
2.2.3 Transforms and transform skipping.....	20
2.2.4 In-loop filter .....	24
2.2.5 HEVC range extension .....	25

2.3	Screen content analysis .....	26
2.3.1	Residual frames .....	27
2.3.2	Pixel change activity .....	30
2.4	Performance evaluation .....	31
2.4.1	Objective quality evaluation .....	32
2.4.2	Subjective quality evaluation .....	32
2.4.3	Experiments and results .....	33
2.5	Summary .....	54
Chapter 3 Colour transforms for HEVC range extensions .....		55
3.1	Conventional Colour transform .....	56
3.2	Related work .....	59
3.2.1	YCoCg .....	60
3.2.2	3-point Pseudo DCT (P-DCT) .....	62
3.3	Karhunen-Loève Transform .....	63
3.4	Lifting-based design of new colour transforms .....	68
3.4.1	Two channel lifting scheme .....	68
3.4.2	Three channel lifting scheme .....	70
3.4.3	Proposed three channel lifting-based colour transforms .....	72
3.5	Performance evaluation .....	77
3.5.1	Test sequences .....	77
3.5.2	Software reference configurations .....	79
3.5.3	Evaluation metrics .....	80
3.6	Experiments and results .....	81
3.6.1	Colour transform error analysis .....	81
3.6.2	Colour transform for HEVC .....	84
3.6.3	Colour transforms for screen content .....	93
3.7	Discussion .....	96

3.8	Summary .....	97
Chapter 4	Contouring artefacts removal for HEVC compressed UHD content.....	98
4.1	The overview of contouring artefacts .....	98
4.2	Related work .....	103
4.3	Contouring detection.....	106
4.3.1	Contouring artefacts analysis .....	106
4.3.2	Proposed contouring detection method.....	108
4.4	The proposed contouring removal method .....	112
4.4.1	Overall procedure.....	112
4.4.2	High frequency component dithering .....	114
4.4.3	Block average dithering .....	116
4.5	Performance evaluation .....	118
4.5.1	Objective evaluation .....	119
4.5.2	Subjective evaluation .....	120
4.6	Experiments and results .....	121
4.6.1	Objective evaluation results.....	123
4.6.2	Subjective results .....	125
4.7	Discussion.....	128
4.8	Summary.....	130
Chapter 5	Contouring artefacts prevention for UHD content in HEVC.....	131
5.1	Pixel mapping algorithm.....	131
5.1.1	Pixel mapping .....	131
5.1.2	Inverse pixel mapping.....	135
5.1.3	Parameters setup .....	136
5.2	Implementation .....	137
5.2.1	Pixel mapping pre-codec.....	139

5.2.2	Pixel mapping in-codec.....	139
5.3	Performance evaluation .....	143
5.4	Experimental results and analysis.....	144
5.4.1	Subjective test results.....	144
5.4.2	Objective and subjective metrics comparisons.....	148
5.4.3	Pixel mapping in-codec applied on intra blocks only.....	152
5.5	Discussion.....	153
5.5.1	Residual analysis of pixel mapping algorithm.....	154
5.5.2	Complexity comparisons .....	157
5.5.3	Sensitivity of defined parameters.....	158
5.6	Summary.....	159
Chapter 6	Conclusions.....	160
6.1	HEVC analysis and evaluation .....	160
6.2	Colour transforms for HEVC range extensions.....	161
6.3	Contouring artefacts removal for UHD content in HEVC.....	162
6.4	Future work.....	163
	Bibliography .....	165

## Chapter 1 Introduction

*High Efficiency Video Coding* (HEVC) [1] [2] was developed by the *Joint Collaborative Team on Video Coding* (JCT-VC), which is collaboration between *International Telecommunications Union* (ITU) *Telecommunication Standardisation Sector* (ITU-T) *Video Coding Experts Group* (VCEG) and *ISO/IEC Moving Picture Experts Group* (MPEG). The motivation for development of HEVC came from the need to compress videos more efficiently, with the target to deliver at half the data rate (bit-rate) achieved by H.264/AVC standard [3] [4] while retaining the same objective video quality. H.264/AVC, the best performing video compression standard prior to HEVC, had been widely used since it was published by the Joint Video Team of MPEG and VCEG in 2003. Since then intensive research activities have resulted in a mature technology capable of underpinning a new video compression standard. Following the Call for Proposals [5] issued in January 2010, the development of the HEVC has gained great progresses to achieve the bit-rate reduction target set by the JCT-VC. The first version of the HEVC main profile was released in 2013 [6], about 10 years after H.264/AVC was released.

While H.264/AVC enabled the distribution of *High-Definition* (HD) video broadcasting, it is expected that HEVC is able to boost and improve such services. The ITU has announced a recommendation that represents the new television broadcasting environment, termed *Ultra High-Definition Television* (UHDTV) [7]. Due to its superior compression performance, HEVC paves the way for UHDTV, enabling a significantly improved viewing experience. The resolution of current *HD Television* (HDTV) offers about 2.1 Megapixels ( $1920 \times 1080$  pixels/frame). UHDTV systems support up to 16 times this resolution. Additionally, basic support for *Screen Content* (SC) is another new feature that needs to be addressed by next generation video compression standards. More details of UHD and SC content are introduced in Section 1.1.2. Since each new content type has different statistical and visual properties, new video compression solutions are needed to achieve high compression gains for such content.

Image and video compression has been developed and achieved significant improvement over the last 20 years [8]. Many different coding systems and algorithms have been proposed and some are widely used after being adopted into international standards. Reviewing video compression history, most standards were aimed at achieving higher coding efficiency considering the trade-off between video quality and bit-rate. This means minimising the bit-rate for a certain equivalent video quality or maximise the video quality level within a given bit-rate. Higher resolution video content is even more demanding so the compression efficiency must be further improved in order to store and transmit the video streams successfully within the given disk space and bandwidth constraints. This, in conjunction with the emergence of new generation content and its related requirements, has prompted standardisation bodies to develop a new high efficiency video compression standard - HEVC.

## **1.1 Video compression background**

Video applications have a history of innovations in quality starting from monochrome TV and then moving to colour TV, HDTV and now UHD TV. Due to the advances in consumer electronics, the affordability of the equipment and the convergence of the telecommunications, networking, computing and television technologies, there has been a growth in high quality user generated content demanding higher resolutions and new content formats. This includes UHD and screen content. Both UHD video and screen content demand higher standard compression and these requirements have driven the features and performance of HEVC. Before looking in detail at HEVC we first examine the history that had underpinned its development.

### **1.1.1 Video coding history**

Previous video compression standards have been mainly developed by ITU-T and ISO/IEC [8]. The ITU-T VCEG H.261 and H.263; ISO/IEC MPEG produced MPEG-1 and MPEG-4 Visual. These two standardisation organisations also jointly cooperated in developing H.262/MPEG-2 Visual, H.264/MPEG-4 AVC and now HEVC.

H.261 [9] was published in 1988 by ITU-T. It was originally designed for supporting CIF (with a resolution of  $352 \times 288$ ) and QCIF (with a resolution of  $176 \times 144$ ) video frame sizes in the 4:2:0 sampling format. The standard supports integer-accuracy motion compensation. H.261 first adopted macroblocks coding structure which remained in the later standards. More details of macroblocks are introduced in Section 2.1. ITU-T then developed the H.263 standard to improve the compression performance at low bit-rates (below 30 kbit/s) [10]. H.263 started using half-pixel motion compensation and improved motion vector coding which used the reconstructed neighbouring blocks as the predictors. The first version of H.263 contained four annexes and the version 2 added several optional coding modes such as a deblocking filter and an advanced intra coding mode [10].

H.262/MPEG-2 Video [11] (widely known as MPEG-2) was developed as a compression system for digital multimedia content. It was extended and developed from the prior MPEG-1 video standard and aims at *Standard Definition Television* (SDTV) and HDTV coding. Video content was coded at lower resolutions than standard television broadcasting until MPEG-2 was standardised in 1995. It was widely used for DVD storage [8]. MPEG-2 also uses macroblocks with a maximum block size of  $16 \times 16$ . The standard defines three picture types: *Intra* (I), *Predicted* (P) and *Bi-predictive* (B) [11]. The macroblocks in an I picture are coded as a block in a still image without using any prediction with respect to reference frames. The macroblocks in a P picture are coded using forward predictions from other macroblocks. Macroblocks in a B picture are coded either as an I type, P type or using bi-directional prediction (forward and backward prediction) from two frames on either side of the current frame. Motion compensated prediction is used to form residuals, i.e. the differences between the original and the predicted blocks. MPEG-2 uses the  $8 \times 8$  *Discrete Cosine Transform* (DCT) transform to decorrelate the data in I blocks and the prediction residuals in P and B blocks.

MPEG-4 [12] was developed as an improvement upon MPEG-2. Similar to MPEG-2, the inter prediction can be done with  $16 \times 16$  or  $8 \times 8$  blocks. MPEG-4 added support for motion compensation with quarter-pixel precision motion vector and bilinear interpolation [8]. The transform in MPEG-4 is similar to previous standards, using the  $8 \times 8$  DCT-based integer transform. The quantized transform coefficients are zig-zag scanned and coded using three-dimensional run-level *Variable-Length Coding* (VLC).

H.264/AVC [3] is the second video compression standard developed jointly by ITU-T and ISO/IEC. It was designed to provide good video quality at lower bit rates video compression than those in the previous MPEG-2, MPEG-4 and H.263 standards. Before HEVC was proposed, H.264/AVC has been used as the main video compression for more than ten years. With the development of new types of video content, the market needs more efficiency video compression system. More coding tools of H.264/AVC are detailed and compared to HEVC in Chapter 2.

### 1.1.2 Next generation video content

There are some new types of video content (also named next generation video content in this thesis) generated in the market which require advanced tools in video compression, such as UHD content, screen content, *High Dynamic Range* (HDR), high frame rate content and high bit depth content. This thesis is focused on two types of the next generation video content: UHD content and screen content.

#### 1.1.2.1 UHD video content

UHD content defines two video resolutions, up to  $7680 \times 4320$  pixels/ frame. It provides a wider viewing angle both horizontally and vertically. UHD TV is specified in the *Radio communication Sector of ITU* (ITU-R) Recommendation BT.2020 (or Rec. 2020) [6]. Comparing with HDTV, which is specified in Rec. ITU-R BT.709 [13], UHD TV contains many advanced features. There are two resolutions of UHD content:  $3840 \times 2160$  and  $7680 \times 4320$ , as defined in Rec. 2020. With aspect to HD content, Rec. 709 only includes the  $1920 \times 1080$  resolution. Both systems support picture resolution with the aspect ratio of 16:9. UHD TV has a viewing angle of  $100^\circ$  due to the large size of the picture that gives a better viewing experience compared to HDTV, which has a viewing angle of  $30^\circ$ . HDTV supports frame rates up to 60 Hz. UHD TV extends these frames rates up to 120 Hz. *Table 1.1* shows the different characteristics between UHD TV and HDTV.

## CHAPTER 1. INTRODUCTION

Table 1.1 Characteristics comparisons.

Characteristics	UHDTV	HDTV
Number of pixels	7680 × 4320 3840 × 2160	1920 × 1080
Aspect ratio	16: 9	16: 9
Viewing angle	100°	30°
Frame rate (Hz)	120, 119.88, 100, 60, 59.94, 50, 30, 29.97, 25, 24, 23.976	60, 59.94, 50, 30, 29.97, 25, 24, 23.976

The *Super Hi-Vision* (SHV) [14] is the first system that demonstrates UHDTV functionalities. It provides higher viewing experience by increasing sense of reality. Another feature of SHV is that it adopts a 22.2 multi-channel three-dimensional sound system that provides an immersive experience. SHV applications were applied during the London Olympic Games in 2012 [15].

Before HEVC, there were some earlier video compression standards optimised for lower resolutions. They are unlikely to achieve the best performance for higher resolutions, as shown in Section 2.4.3.2. HD or UHD contents are now widely used in television broadcasting and cinema, on mobile phones and for internet streaming. Compression efficiency must be improved in order to save storage space and transmit the video streams successfully within the limited space and bandwidth. Chapter 2 introduces improved coding tools in HEVC that provide better video coding performance for UHD content.

On aspect of visual experience, the advanced coding tools are needed to improve the UHD video content. False contouring is one of the visual artefacts in compressed UHD video sequence which affect the visual experience. Chapter 4 proposes a contouring detection method and a dithering contouring removal method for HEVC compressed UHD content. Chapter 5 designs a pixel mapping algorithm to reduce the contouring artefacts during video compression.

### 1.1.2.2 Screen content

In addition, basic support for *screen content* (SC) [16] [17] is another new feature that needs to be addressed by next generation video compression standards. SC, which is widely used in mobile and internet streaming, refers to an image or video that contains computer generated content in addition to elements that are captured by a camera. Camera captured content, often known as natural images, consists of homogeneous regions as well as texture regions. It also inherits the camera sensor noise. However, for non-camera captured content, such as text, there are frequent sharp edges leading to fine details. The features comparisons between natural content and screen content are presented in Section 2.3. The blind use of standard coding tools, like fixed block partitions, fixed prediction structures or fixed coding strategies does not lead to an optimum rate distortion performance nor does it enable delivery of the highest quality required by users. In this case, screen content requires a highly content-adaptive approach in all aspects of video coding.

### 1.1.3 New market requirements

With ultra-high definition content, the resolution of videos could be more than 30 Megapixels per frame. It is a challenge for transmission system which leads to the motivation of producing a higher efficiency video compression. HEVC, as the successor to H.264/AVC, aims to halve the bit-rate for a given level of visual quality compared to that achieved by its predecessor. Figure 1.1 shows the video compression development trend from MPEG-2 to H.264/AVC to HEVC, with bit-rates as suggested in [18]. Both H.264/AVC and HEVC have achieved the final goal of halving the bit-rate compared to the previous video compression system. As illustrated in Figure 1.1, the development of a compression standard does not always follow a smooth curve in reality. The history of the previous coding system improvements indicates that a significant change of performance can be considered once a decade.

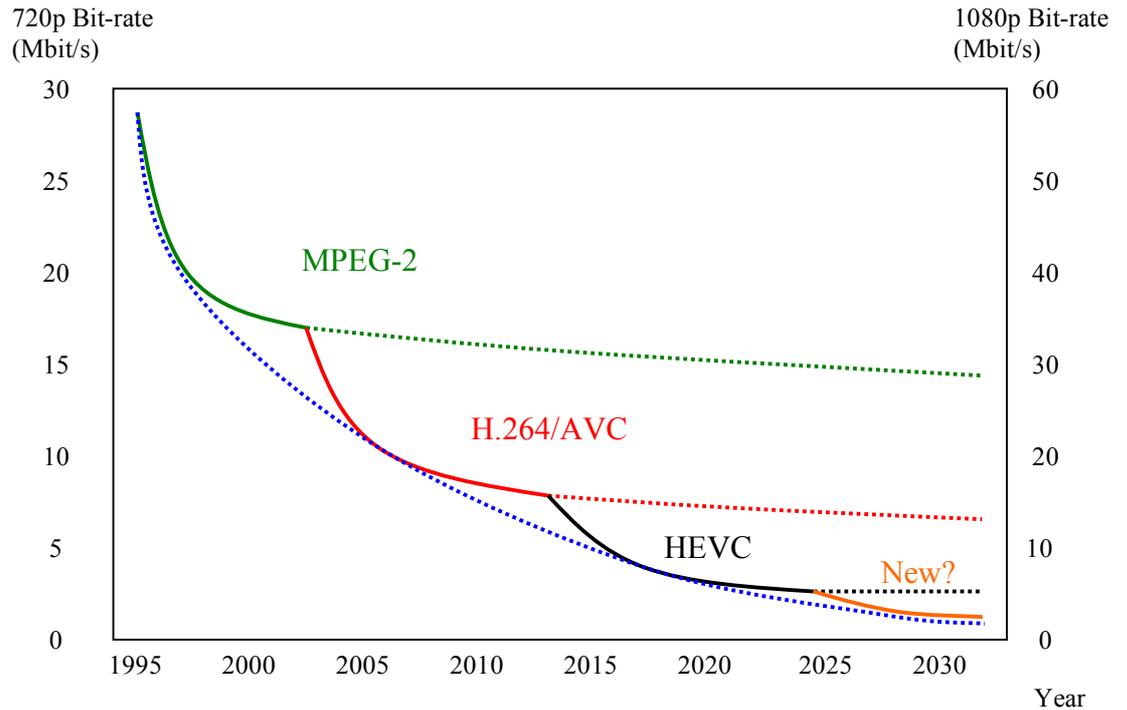


Figure 1.1 Improvements in coding performance introduced by new standards [18].

## 1.2 Aims and objectives

The aim of this study is to research and develop novel technologies and advanced tools that can improve HEVC coding efficiency for next generation video content.

The specific objectives to achieve the main aim are:

- To study the usability of HEVC for the next generation video content (screen content and ultra-high definition content).
- To research and develop novel colour space conversion methods in HEVC range extensions.
- To develop methods for improving video quality of HEVC compressed UHD content.

- To evaluate the proposed methods based on both objective metrics and subjective metrics.

The aim is achieved by the following four tasks:

1. HEVC review and evaluations

HEVC is designed for next generation video content for high efficient compression. Firstly, the basic architecture and key coding tools are introduced by comparing with H.264/AVC. Additionally, experiments of block partition and transform skipping for screen content are implemented. The basic principles behind HEVC and related experiments are presented in chapter 2.

2. Colour transforms for HEVC range extensions

The latest standards support colour format YCbCr with chroma sampling 4:2:0. However it is not optimal in video compression. We aim to investigate and propose colour de-correlation methods for efficient compression. This is started firstly on several commonly used methods by converting original RGB format videos into 4:4:4 colour space formats. Based on the results of different types of content, we propose a novel method to achieve maximum de-correlation for colour components and design optimal colour transform for next generation R-G-B content. This work is shown in Chapter 3.

3. Contouring artefacts in compressed UHD content

Compared to HD content, UHD video provides higher visual quality. However, artefacts in UHD content are more apparent due to increased spatial resolution. Contouring is one of the visual artefacts in compressed UHD video sequences which affect the visual experience. There are three proposed method using concepts. The proposed dithering de-contouring method is used as a post-processing. The proposed pixel mapping method can be applied as pre-codec or in-code HEVC. These two proposed methods are represented in Chapter 4 and Chapter 5, respectively.

4. Visual quality evaluations

Both subjective and objective quality metrics are used to evaluate the performance of the developed new technologies. Especially for UHD content which is claimed to provide a visual experience to viewers, a more efficient subjective evaluation metric needs to be developed to conduct and to accurate evaluation results.

### 1.3 Original contribution

The author considers the following elements of the study form an original contribution to the next generation content coding solutions and HEVC literature.

- Chapter 2
  - HEVC review and evaluation.
  - Comparison with H.264/AVC with all classes of video sequences.
  - Discovery that HEVC achieves higher BD-rate saving with higher definition sequences.
  - Analysis of HEVC block partitions.
  - Implementation of HEVC coding with reduced block.
  - Review and implementation of transform skipping for screen content.
  
- Chapter 3
  - Colour transform review.
  - Development of 3-point DCT for colour transform.
  - Analysis of *Karhunen–Loeve Transform* (KLT) and propose KLT approximations, named KLT-A, B and C.
  - Review lifting scheme and extend to three channel lifting structure
  - Design and implementation of 3-channel lifting based colour transform, named PPU.
  - Relationship between pixel bit depth and colour transform errors.
  - Comparison of the performance of the above colour transforms with different types of video content.
  - Discovery that the best colour transforms for natural content and screen screen used in HEVC are different.

## CHAPTER 1. INTRODUCTION

- Analysis of the performance of the PPU colour transforms for screen content with various parameters.
  
- Chapter 4
  - Literature review of contouring artefacts removal.
  - Design of three approaches of contouring artefacts removal: pre-codec, post processing and in-codec.
  - Analysis of pixels in contouring area in UHD content.
  - Design and implementation of block based contouring detection.
  - The multi-scale dithering contouring removal method.
  - Both objective and subjective quality evaluation on UHD content on respect of contouring artefacts.
  - Discovery that subjective evaluation metric is more reliable than objective evaluation on respect of contouring artefacts in UHD content.
  - Discovery that proposed dithering contouring removal method improved the video quality up to 56% (*Mean Opinion Score (MOS) difference*).
  
- Chapter 5
  - Pixel mapping and inverse pixel mapping.
  - Discovery that contouring detection conducted on original picture and compressed pictures results in the same contouring maps.
  - Design and implementation of pre-codec and in-codec contouring reduction methods using pixel mapping.
  - Subjective evaluation metrics on proposed methods
  - Discovery that one of the proposed in-codec methods achieves MOS *Bjontegaard Delta rate (BD-rate)* saving up to 78%.
  - Discovery that post processing contouring removal approach over performances the other approaches.

## 1.4 Author's publications

### Book chapter

1. Yanxiang Wang, Charith Abhayaratne and Marta Mrak, "High Efficiency Video Coding (HEVC) for next generation video application," in *Electronic Reference Signal Processing*, Vol.5, D. Bull, eds. Elsevier, 2014.

### Conference proceedings

1. Y. Wang, C. Abhayaratne, R. Weerakkody and M. Mrak, "Colour space transforms for improved video compression," in *Proc. International conference on Systems, Signals and Image Processing (IWSSIP)*, May 2014.
2. Y. Wang, C. Abhayaratne, R. Weerakkody and M. Mrak, "Multi-scale dithering for contouring artefacts removal in compressed UHD video sequences" in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP 2014)*, December 2014.

### In progress

1. Contouring detection and contouring removal in HEVC compressed UHD video (journal paper).
2. Content adaptive pixel mapping contouring prevention method for UHD content in HEVC (journal paper).

Part of the work presented in this thesis is being considered for protection by sponsor BBC R&D.

## 1.5 Organisation of the thesis

The rest of this thesis is organised as follows: Chapter 2 reviews HEVC by introducing improved coding tools in HEVC that provide better video coding performance and compared with H.264/AVC. The performance on screen content is shown in Chapter 3 which investigates the colour conversions in video compression and presents an evaluation performance of both natural content and screen content. Contouring artefacts removal in compressed UHD video sequences are studied in Chapter 4 and Chapter 5. Finally, the conclusions are drawn in Chapter 6.

## Chapter 2 HEVC analysis and evaluations

Chapter 2 provides an overview of HEVC analysis and performance evaluations. HEVC is compared with reviews of previous video compression standard H.264/AVC in Section 2.1. Section 2.2 provides a brief overview of coding tools in the first version of HEVC based on the HEVC text specification draft 8 [6] followed by a performance evaluation and experimental analysis presented in Section 2.3. The results show that HEVC halved the bit-rate over the predecessor video coding standard H.264/AVC at certain equivalent video quality. The results also confirm the efficiency of advanced tools in HEVC for different types of video content.

### 2.1 The overview of H.264/AVC

H.264/AVC is the best performing video compression standard prior to HEVC. Before HEVC was released, H.264/AVC was the most widely used in video compression.

H.264/AVC uses a macroblock-based coding structure [3]. It allows multi-layers of block partitioning. The maximum size of coding unit defined in H.264/AVC is  $16 \times 16$ . Each frame is firstly partitioned into  $16 \times 16$  macroblocks, which can be further partitioned in four ways, as shown in Figure 2.1. Directional intra prediction in the spatial domain is used in order to enhance prediction leading to reduced residual energy. Different prediction block sizes are used depending on the content statistics in the region covered by the macroblock:  $16 \times 16$  intra prediction is used in highly homogeneous blocks, while  $4 \times 4$  intra prediction is used in regions containing fine details, *i.e.*, high spatial frequency components. In motion compensated prediction, both past and future frames can be chosen as references from which to predict the current frame. Instead of I, P and B pictures,

H.264/AVC defines five different types of picture slices: I, P, B, Switching P (SP) and Switching I (SI). I, P and B slices are the same as in previous standards. An SP slice contains P and/or I type macroblocks switching between encoded streams. An SI slice contains a special type of I type macroblock switching between coded streams [3]. In H.264/AVC,  $4 \times 4$  and  $8 \times 8$  DCT-based integer transforms are used for de-correlating the prediction residuals prior to quantisation and entropy coding. Typically, the smaller transform block size is used in rich texture regions, while the larger transform block size is used in smooth regions.

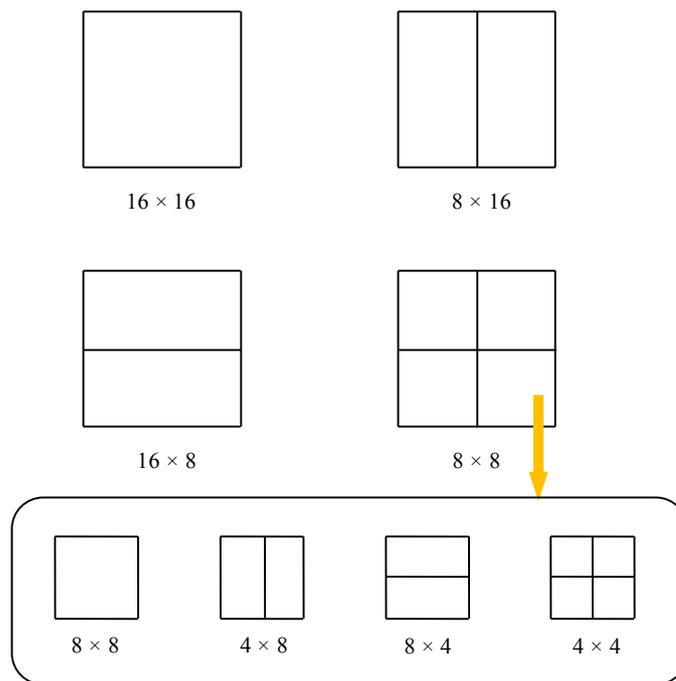


Figure 2.1 Macroblock partitioning in H.264/AVC.

Entropy coding in H.264/AVC is primarily performed by using *Context Adaptive Variable-Length Coding* (CAVLC). Alternatively, *Context Adaptive Binary Arithmetic Coding* (CABAC) can be used for enhanced performance. CAVLC is based on Huffman coding principles and uses adaptively generated coding contexts for encoding the symbols efficiently. CABAC converts symbols into binary codes and uses binary arithmetic coding dependent on the coding context. In addition, H.264/AVC also uses an interim processing step, known as in-loop filtering, within the motion compensated inter prediction steps. The in-loop filter is applied on the reconstructed pixels after inverse quantisation, the inverse transform and the synthesis. It is applied in a form of a de-

blocking filter adaptively at block boundaries to remove the blocking artefacts. This has led to improved rate distortion performance.

## 2.2 The Overview of HEVC

In this section, the key coding tools in the HEVC design are presented based on the HEVC text specification draft 8 [6].

HEVC relies on dividing a video into multiple *Coding Tree Units* (CTUs) [19]. Each coding process, including inter/intra prediction, quantisation, transformation, de-blocking and entropy coding, is performed on a CTU-basis or on its partitions. Related coding parameters (e.g. partitioning structure and coding modes) are defined by an encoder and are used at the reconstruction side. The intra prediction process aims to select the best directional intra mode among the 35 modes defined in HEVC (Planar, DC and 33 angular prediction modes). For this process, mode dependent smoothing of reference samples is applied to increase prediction efficiency. The three most probable modes are derived for each block to increase symbol coding efficiency. Additionally, prediction from the luma mode is applied to improve efficiency of chroma mode coding. The inter prediction process uses *Adaptive Motion Vector Prediction* (AMVP) which aims to select the best motion predictors among several candidates. Quarter sample luma interpolation is used to derive the texture prediction. In both intra and inter prediction cases, the residuals are generated by subtracting the prediction from the input and then they are spatially transformed and quantised. The DCT-based transforms are used in both intra and inter modes with the transform unit sizes of  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$ , and  $4 \times 4$ . The two-dimensional transform is implemented as two one-dimensional transforms applied horizontally first and then vertically. Additionally, a transform based on the  $4 \times 4$  *Discrete Sine Transform* (DST) is used for intra predicted luma residuals only [1]. Two in-loop filtering processes, de-blocking filter and Sample Adaptive Offset (SAO), are applied on the reconstructed samples to remove blocking artefacts and to reduce the distortion between the original frames and the reconstructed frames. Reconstructed CTUs are assembled to construct a picture. In typical configurations which involve motion

compensation, such frames are stored into a decoder picture buffer to be used for prediction of the next picture of input video. HEVC uses CABAC, which relies on binarization, context modelling and binary arithmetic coding, as the single entropy coding method. A comparison of features used in H.262/MPEG-2, H.264/AVC and HEVC is given in Table 2.1.

Table 2.1 Comparisons of video compression standards.

Comparison	H.262/MPEG-2	H.264/AVC	HEVC
Max block size	16 × 16	16 × 16	64 × 64
Intra prediction	None	Up to 9 modes	Up to 35 modes
Motion vector accuracy	Half-pixel	Quarter-pixel	Quarter-pixel
Motion compensation minimum size	8 × 8	4 × 4	4 × 4
In-loop filters	None	De-blocking filter	De-blocking filter SAO
Transforms	8 × 8 DCT	4 × 4 and 8 × 8 integer DCT	4 × 4 ~ 32 × 32 integer DCT 4 × 4 integer DST
Entropy coding	VLC	CAVLC CABAC	CABAC

Although HEVC has more flexible block structure and improved coding tools, the video coding flow is similar to previous video compression standards [1]. Figure 2.2 shows the block diagram of the basic HEVC design including encoder and decoder. The following subsections briefly describe the key features involved in HEVC.

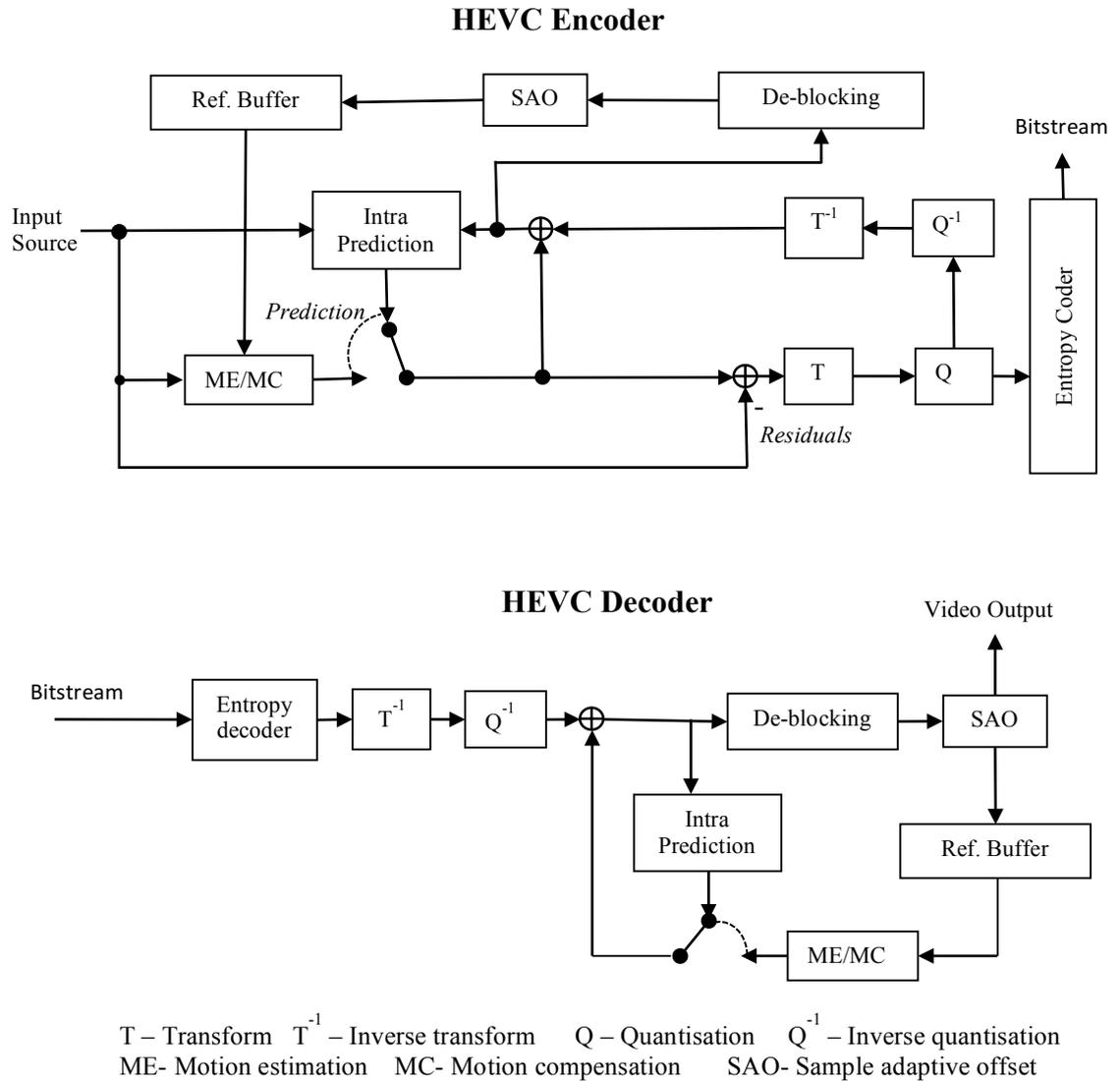


Figure 2.2 Block diagram of HEVC encoder and decoder.

### 2.2.1 Picture partitions

Similar to the techniques used in H.264/AVC, HEVC uses adaptive block partitioning. Additional flexibility is introduced by application of quadtree partitioning [1]. Blocks of pixels in a CTU can be further split into prediction and transform blocks. The decisions for each split are sent in the bit stream.

### 2.2.1.1 Coding tree unit (CTU)

In HEVC, each frame is divided into a number of squared *Coding Tree Units* (CTUs). Typically, a CTU consists of an  $N \times N$  block of single component samples (e.g. luma samples) and two sets of chroma samples (e.g. Cb and Cr). In the 4:2:0 format, each chroma component related to a CTU consists of  $N/2 \times N/2$  samples. In the main profile of the standard, the maximum allowed  $N$  is 64. In contrast to the macroblock structures used in previous standards, such as H.264/AVC, CTUs in HEVC are not just larger but also have different options for their divisions into smaller blocks.

### 2.2.1.2 Coding unit structure

A CTU consists of a sequence of *Coding Units* (CUs). The size of CU can be the same of related CTU block. Or it can be further split into four blocks recursively which is defined by a quadtree. The smallest CU size for luma sample is  $8 \times 8$ . The CU is the basic unit splitting used for predictions or transforms. Several examples of CTU splitting into CUs are given in Figure 2.3.

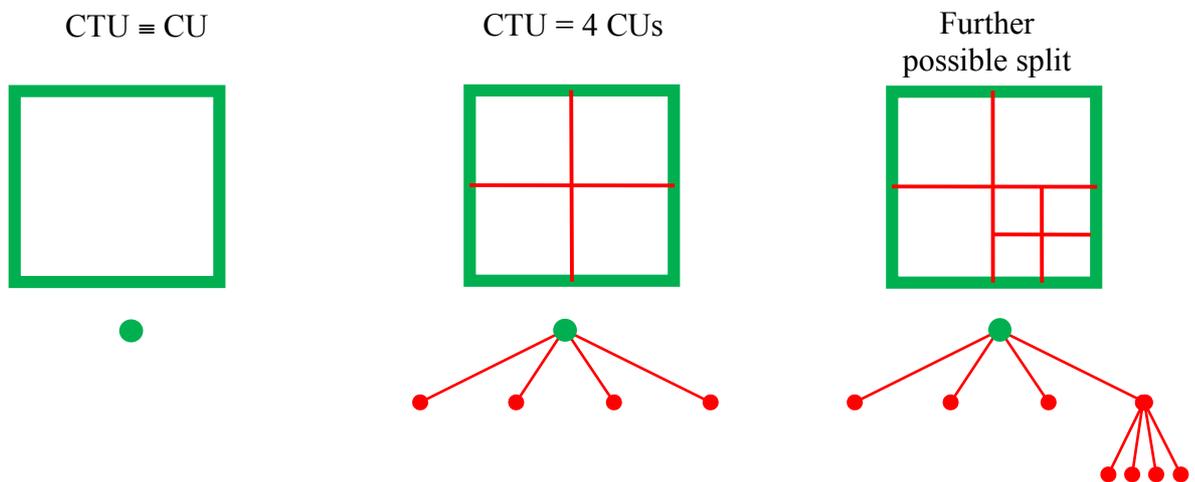


Figure 2.3 CTU splitting and related quadtrees.

### 2.2.1.3 Prediction unit structure

The *Prediction Unit* (PU) is the basic unit for prediction processes. As illustrated in Figure 2.4, a CU can be split into 1, 2, or 4 PUs. The PU partitioning can be asymmetric or symmetric, which aims to match the different boundaries or edges of objects in pictures. An inter-coded CU can be split into all of the eight possible partition modes. On the contrary, an intra-coded CU can be only split symmetrically [1]. CU with PART\_2Nx2N has only one PU while CU with PART\_NxN has 4 PUs. The remaining prediction modes have 2 PUs. The main information related to each PU are prediction details, for example, if a CU is an intra-coded CU then a prediction mode is defined for each PU.

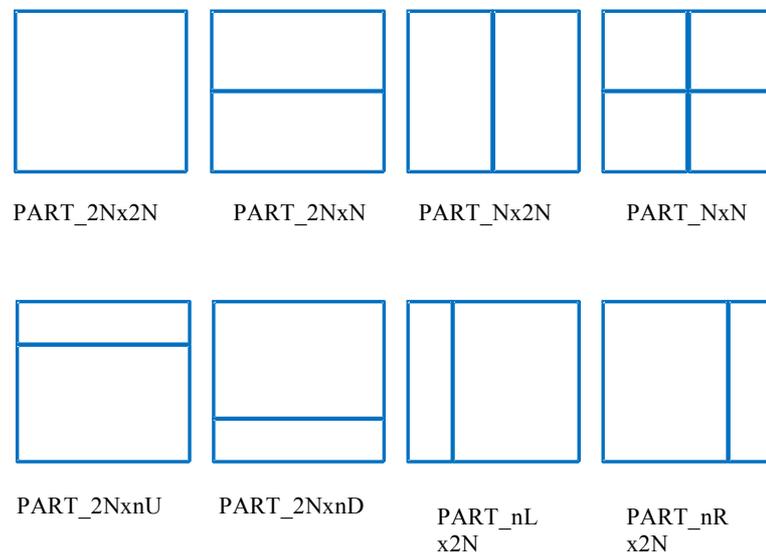


Figure 2.4 Partitioning modes for PUs.

### 2.2.1.4 Transform unit structure

The *Transform Unit* (TU) is the basic unit used for the transform and quantisation processes. Each CU contains one or more TUs. While the PU split defines prediction in different parts of CU, TU split defines transforms within a CU. Each TU consists of square blocks ranging from  $32 \times 32$  to  $4 \times 4$  pixels. Similar to the CTU split into CUs, a CU split into TUs is described by a quadtree structure. In this case the split structure is called *Residual QuadTree* (RQT). Some specific restrictions are applied depending on

the prediction mode, PU split, given component (luma or chroma) and block sizes. An example demonstrating the RQT position relative to CTU/PU split is shown in Figure 2.5. Actual transforms and quantisation are performed on each TU.

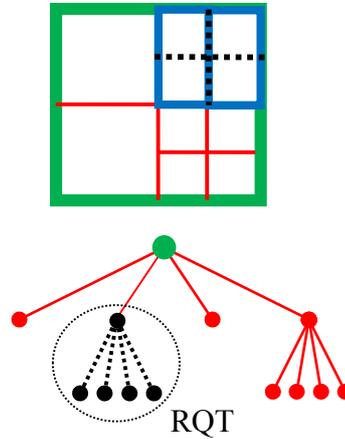


Figure 2.5 A CTU split with top-right CU split into two PUs using *PART\_Nx2N* partitioning and one level of TU split (dotted black lines).

### 2.2.2 Intra prediction

HEVC and H.264/AVC use similar block based intra prediction to predict current sample from neighbouring reconstructed samples. Since HEVC introduced larger block sizes there are 35 intra prediction modes which can support blocks up to  $32 \times 32$ . This makes HEVC more flexible than H.264/AVC as HEVC has more intra prediction modes (35 vs. 9, shown in Figure 2.6). It also increases the prediction accuracy as more neighbour blocks which have been previously decoded are used as reference [8].

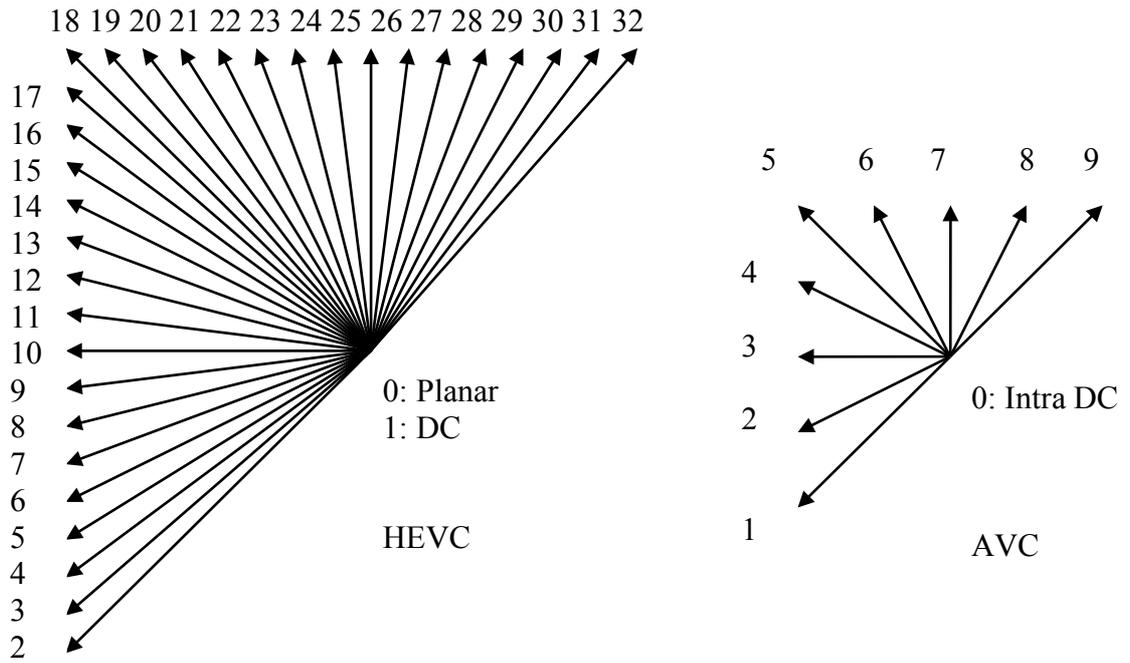


Figure 2.6 Luma intra prediction modes of HEVC and H.264/AVC.

### 2.2.3 Transforms and transform skipping

HEVC specifies five block transforms to code prediction residuals: A  $4 \times 4$  DST-like and four DCT-like transforms of sizes  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$  and  $32 \times 32$ . DST- and DCT-like functions in HEVC use integer arithmetic and have been designed to have basis functions similar to the DCT and DST transforms. In this chapter, related transforms used in HEVC are simply referred to as DCT and DST transforms. All transforms in HEVC are separable, i.e., they can be applied to the columns and rows of a block separately.

The 4-point DCT transform in HEVC is defined by the following matrix:

$$\begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}.$$

The transform basis patterns for its 2D equivalent are shown in Figure 2.7.

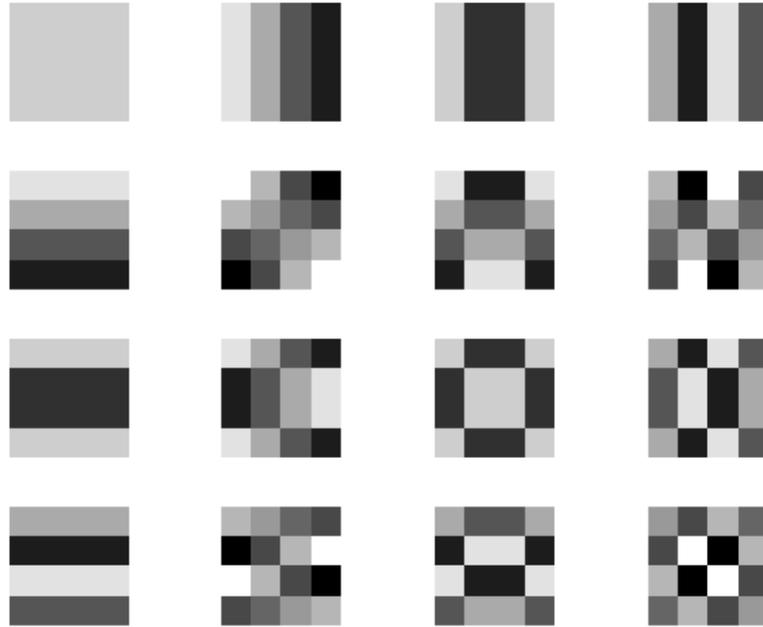


Figure 2.7 Basis patterns for  $4 \times 4$  DCT.

Similarly, the 8-point DCT-like transform is defined by the following matrix:

$$\begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix}$$

The transform basis patterns for its 2D equivalent are shown in *Figure 2.8*. For the definition of larger transforms in HEVC, see [6].

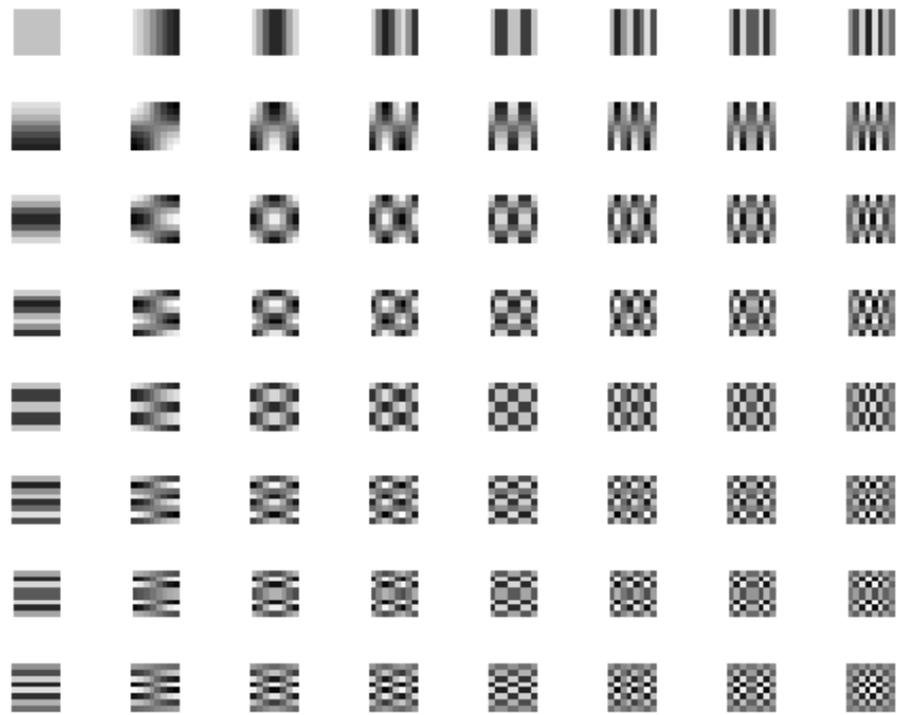


Figure 2.8 Basis patterns for  $8 \times 8$  DCT.

The DST has some compression advantages over the DCT, especially for intra  $4 \times 4$  residuals. When using intra prediction, the samples from the upper and left boundaries are used to predict the current pixel. Therefore, the pixels which are closer to the top-left corner in the frame can be predicted more accurate than the pixels which are closer to the bottom-right corner. In other words, residuals of pixels which are away from the top/left neighbours are usually larger than pixels near neighbours. It can be shown that for such residuals the DST is an optimal choice [20].

HEVC also uses  $4 \times 4$  DST for intra modes. The transform matrix of 4-point DST is:

$$\begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}$$

The basis patterns for the  $4 \times 4$  DST are shown in Figure 2.9.

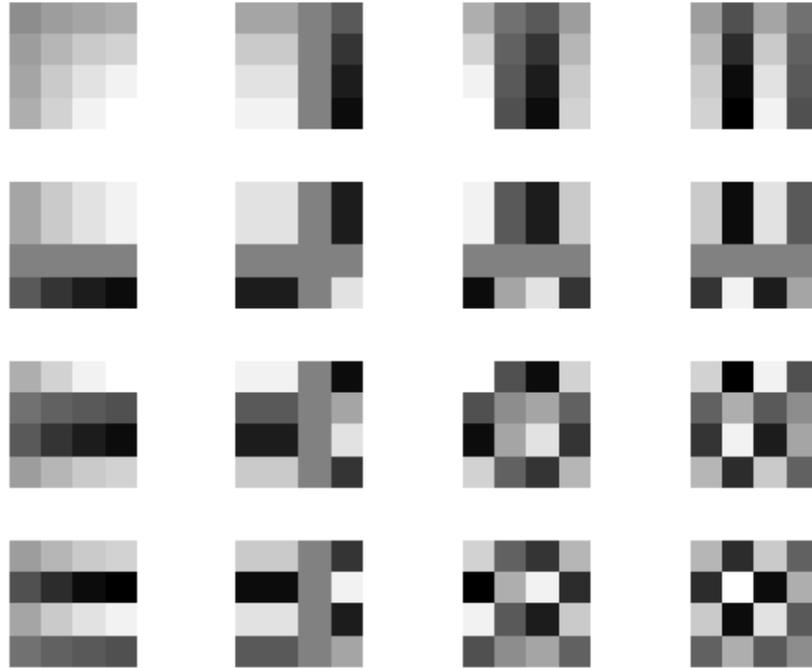


Figure 2.9 Basis patterns for  $4 \times 4$  DST.

The transform process concentrates the energy of the signal into fewer coefficients reducing the correlation in the residual signals. However, for certain types of residuals, the compression efficiency could be improved by completely skipping the transform. For instance, screen content can be noiseless, intra prediction can lead to much smaller or even zero residuals, compared to coding of camera-captured content. In cases where sharp lines are preserved in the residuals, a transform might be inefficient and might even decrease the compression efficiency. Transform skipping also benefits inter coding, especially for motion compensated residuals which more often consist of sharp edges and uncorrelated pixels, compared to those in intra-predicted blocks. Transform skipping has been used in lossless video coding using hybrid frameworks. Since the reference frames are coded losslessly the residuals are low. Also sequences where the motion (change) is low the residuals are low. According to the studies in [21] and [22], it is noted that in such cases the transform can be skipped and the residual is just entropy coded.

To address such coding features, HEVC enables full skip of the 2D transform for  $4 \times 4$  blocks. In the presence of transform skip all other processing steps remain unchanged. However, additional normalisation of signals is required in order to preserve effective application of unchanged quantisation and entropy coding. In Section 2.3, the performance improvements introduced by application of transform skipping in HEVC are demonstrated.

## 2.2.4 In-loop filter

In HEVC Loop filters are applied to the reconstructed samples before they are used for predicting other pictures. In-loop filtering includes De-blocking filters and Sample Adaptive Offsets (SAO). De-blocking filter is similar to that in the H.264/AVC, which aims to reduce the blocking artefacts due to block based coding. SAO is newly introduced in HEVC and is applied adaptively to all samples by adding a simple offset value to each pixel based on the category.

### 2.2.4.1 De-blocking filter

Blocking is one of the most visible and identifiable artefacts in block-based image/video compression. The de-blocking filter in HEVC is similar to the one used by H.264/AVC but simplified and improved for better support (especially for parallel processing structure). HEVC limits the de-blocking filtering to the edges lying on  $8 \times 8$  blocks while H.264/AVC applies filtering to  $4 \times 4$  blocks. De-blocking filter uses  $8 \times 8$  blocks since it caused no noticeable degradation. Another change is that the de-blocking filter first applies horizontal filtering for vertical edges. The filtered sample are then used in filtering horizontal edges. This supports multiple parallel implementations.

#### 2.2.4.2 Sample adaptive offsets

Sample adaptive offsets (SAO) is a process which modifies the samples by adding offset values to certain sample values to reduce distortion. It is applied to the reconstructed pixels after De-blocking filter. SAO can be easily implemented as there is a Lookup table which allows the offset value to be pre-determined before conducting any work. The lookup table includes six categories according to reconstructed pixel classification. It then adds either *Band Offset* (BO) or *Edge Offset* (EO) to the pixel depending on the pixel intensity or edge properties.

In the band offset, the offset value is selected according to the sample value. The whole sample amplitude range is divided into 32 equal bands (each band has its own offset) which are then divided into two groups, a central band that consists of 16 bands and another group containing the remaining bands. An encoder selects one group band to apply SAO.

EO uses four one-dimensional three-pixel patterns to classify pixels according to the edge directional information. It classifies each pixel by comparing with its two neighbouring pixels. It then adds offset of each category to compensate pixel values.

#### 2.2.5 HEVC range extension

HEVC range extension is designed to support several colour formats as well as increased bit depths. Some additional coding tools have been added in the August 2013 draft [21] of the range extensions amendment. The profiles support bit depths beyond 10 bit (including 10/12/14 bit) per sample and separate the bit depths for luma and chroma with YUV colour spaces. The banding artefact which occurs with 8-bit depth in plain areas can be compensated when using 10-bit. Although 10-bit operation upgrades pixels into 1024 levels, it does not cost bitrate. For gradually changing content, 8-bit quantisation of the prediction causes residual errors whereas 10-bit quantisation makes a better predictor, leaving less residual to be coded.

Chroma sampling is another change made in HEVC range extensions. HEVC test condition specifies 4:2:0 as the main format. The range extension profiles added 4:2:2 and 4:4:4 chroma sampling. 4:4:4 format is full chroma resolution that three components have the same sample rate. 4:2:0 format is supported in HEVC main profile, which halved two chroma component both horizontally and vertically. More details are discussed in Chapter 4. HEVC range extension also supports original 4:4:4 RGB format. HEVC range extension support different colour formats and chroma sampling.

Additionally, there are some other improved coding tools added in HEVC range extension profile [23]. These include:

- Intra prediction angle adjustment for 4:2:2 format sequences.
- Removing rectangular transform blocks, and using square transform and square intra prediction.
- New tools for 4:4:4 screen content coding.

Although the HEVC range extension profile is not implemented in this chapter, it is applied in Chapter 3 for the colour space transform study.

### 2.3 Screen content analysis

Screen content is non-camera captured content which consists of a mixture of different types (text, graphics, and natural images). As the development of the new technology, screen content related applications becomes more important in our daily life, e.g., desktop screen, video conferencing, game on demand. Screen content has different properties compared to natural content:

- It has sharper edges. Sequences such as desktop screen, slide showing and slice editing are more likely to contain texts which have sharper edges.
- The difference between two neighbouring frames are larger (see Section 2.3.1).
- In screen content sequence, frame to frame can have a fast motion activity (see Section 2.3.2).
- Screen content has less or no camera sensor noise since it is non-camera captured.

### 2.3.1 Residual frames

Figure 2.10 to Figure 2.13 present frames from screen content video sequence (Slide Showing) and camera-captured natural video sequence (Basketball Pass). The residual frames are calculated from each sequence to compare the motion activity. The histogram indicates the difference between the two neighbouring frames. From the results it is shown that for screen content frames the changes are more dramatic but less frequent.

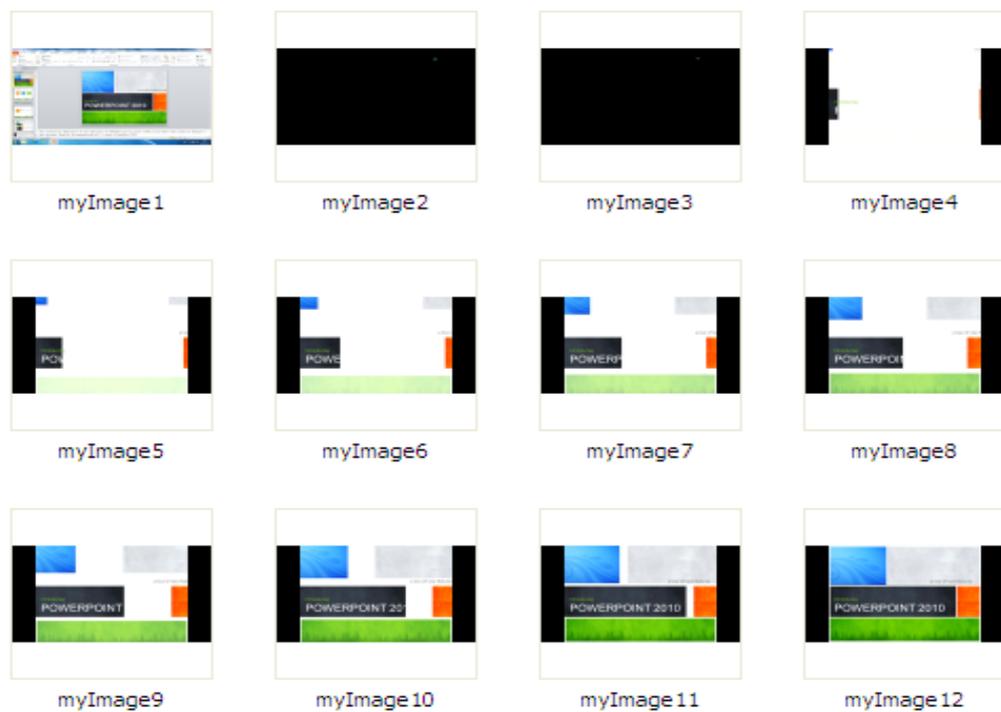
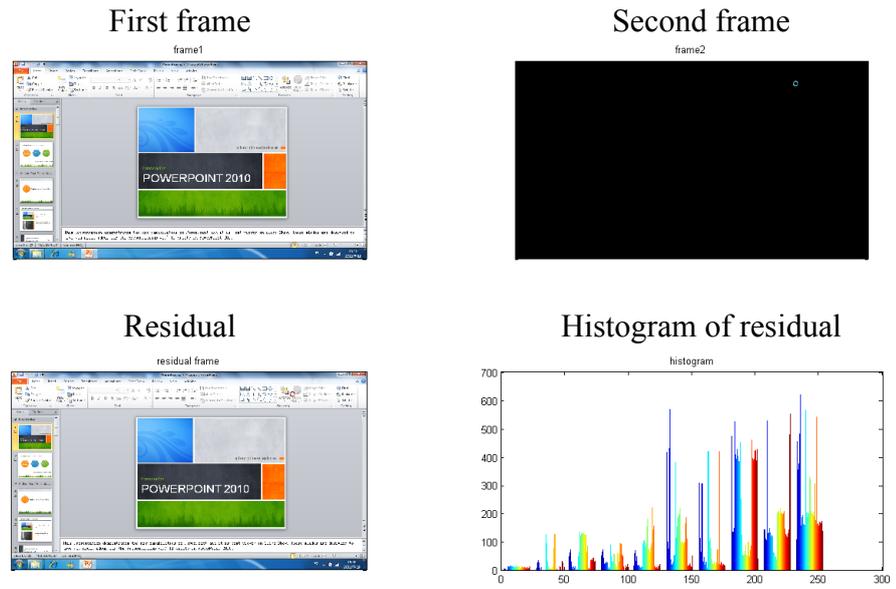
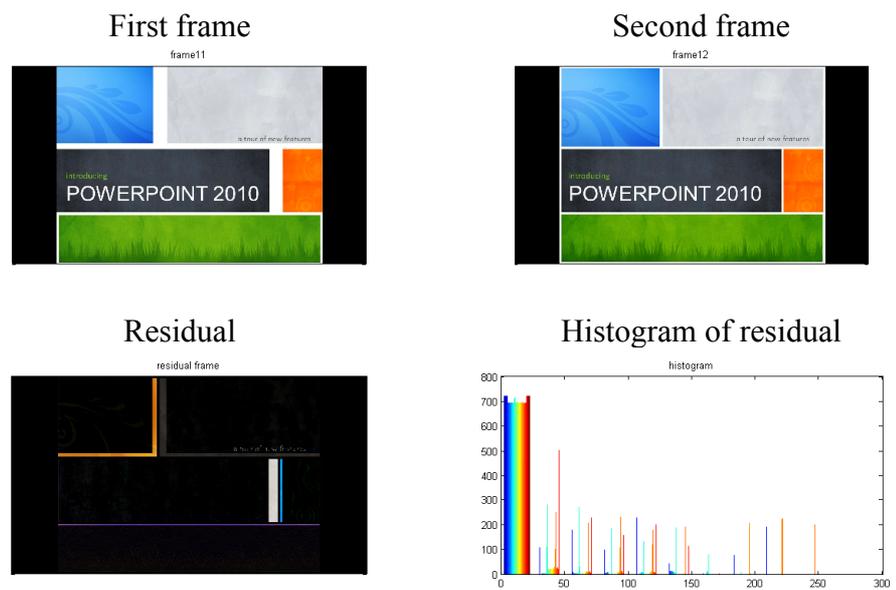


Figure 2.10 Screen content frames (Slide Showing).



1) 1<sup>st</sup> and 2<sup>nd</sup> frames



2) 10<sup>th</sup> and 11<sup>th</sup> frames

Figure 2.11 Screen content frames residuals analysis.

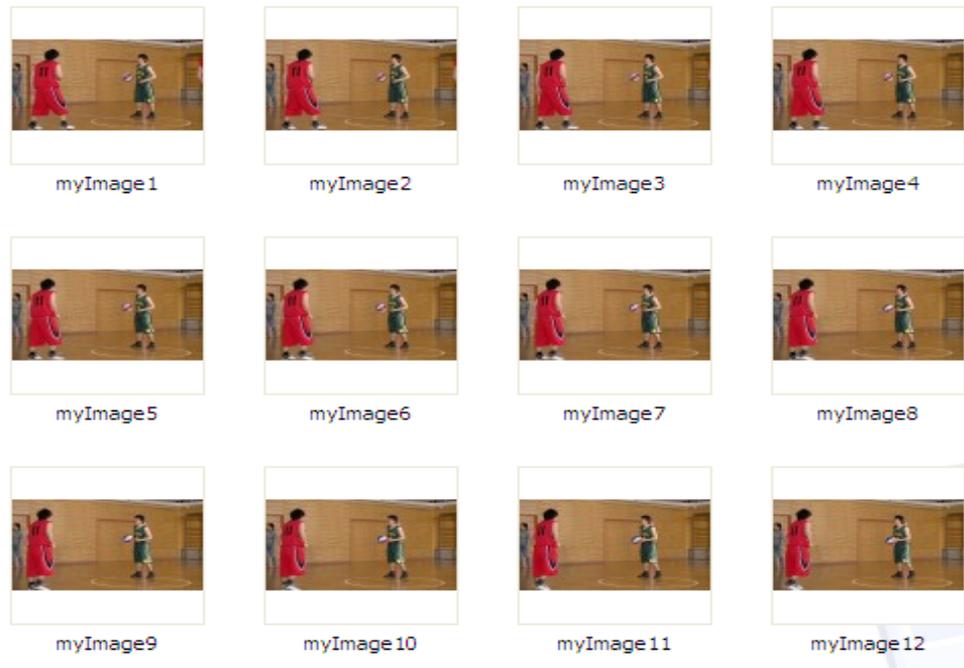
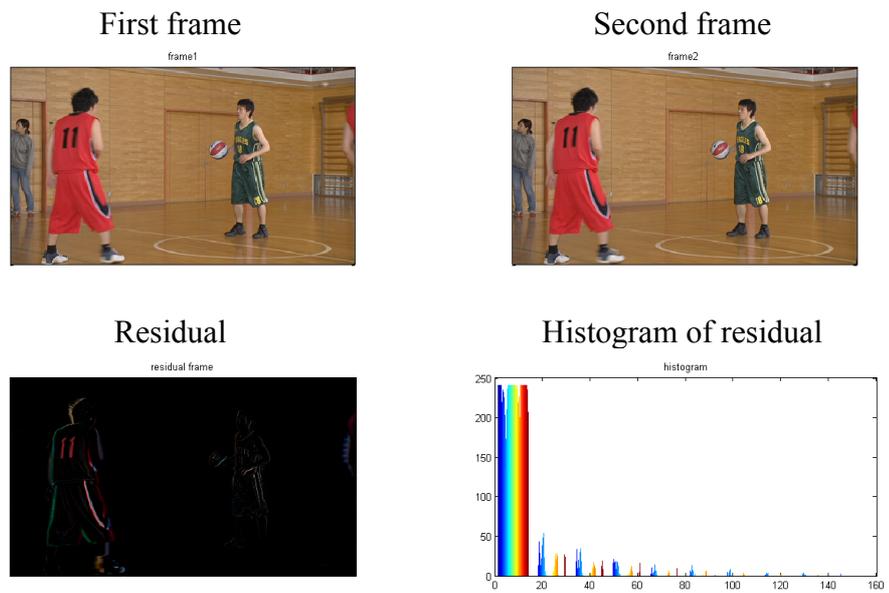
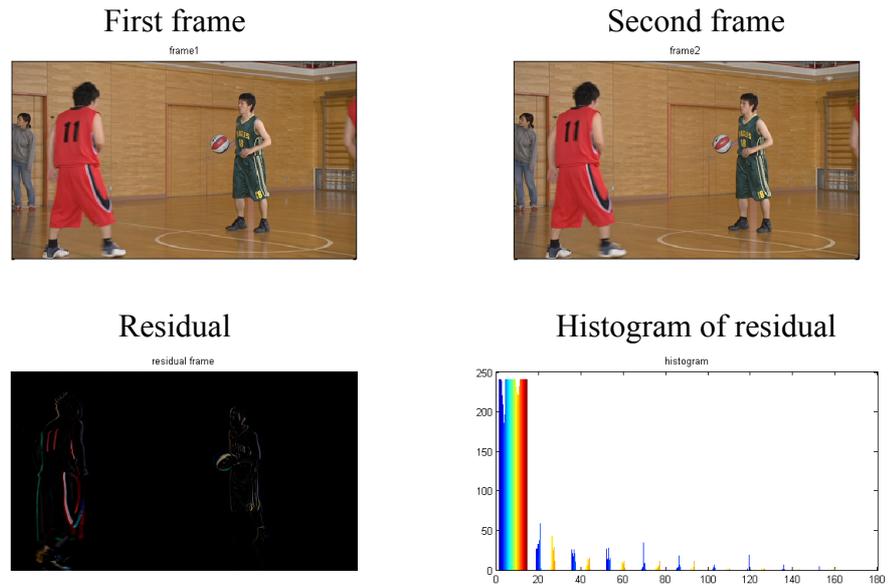


Figure 2.12 Natural content frames (Basketball Pass)



1) 1<sup>st</sup> and 2<sup>nd</sup> frames



1) 10<sup>th</sup> and 11<sup>th</sup> frames

Figure 2.13 Natural content frames residuals analysis.

### 2.3.2 Pixel change activity

Another feature of screen content is the pixel change activity between two neighbouring frames. *Figure 2.14* shows pixel change activity between neighbouring frames in both screen content and natural content. The blue lines represent the screen content sequence, and the red lines represent the natural content sequence. The blue lines change faster while the red lines are more stable. In the test we used Mean Square Error (MSE) to represent the average pixel difference. The natural content sequence is camera captured, which has gradient pixel change activity from a current frame to the next frame. However, the MSE between two neighbouring frames in screen content fluctuates from frame to frame more dramatically.

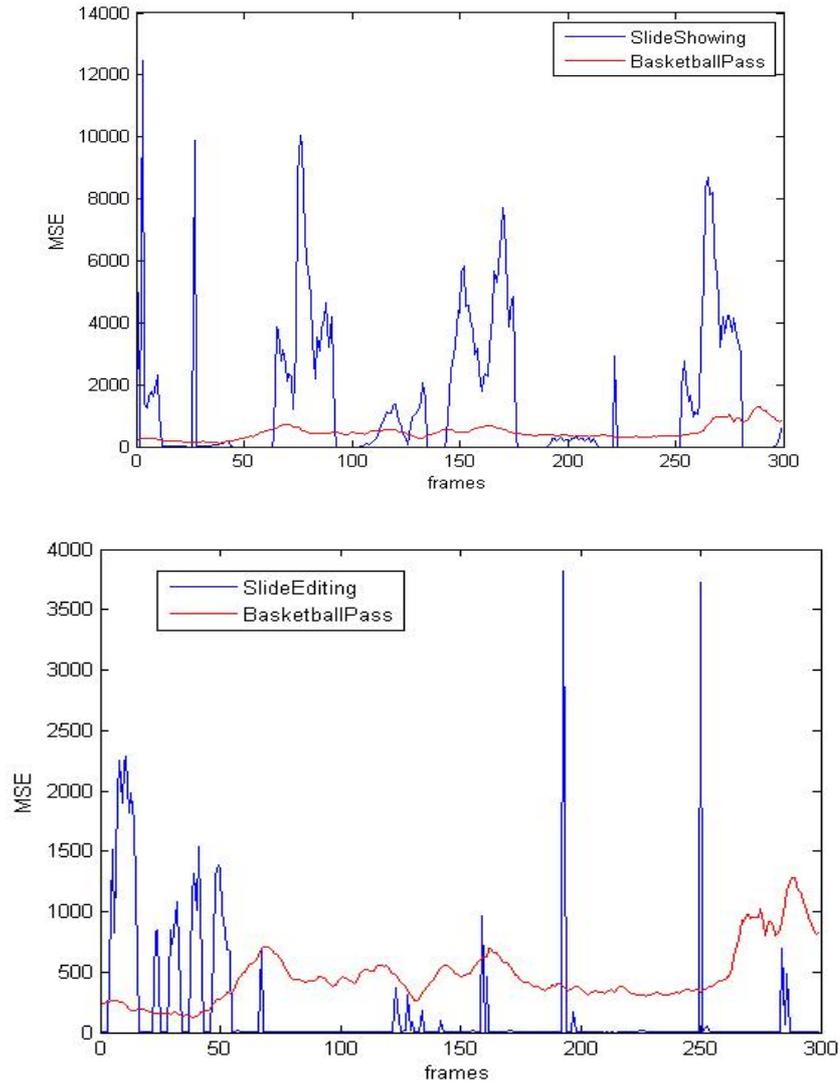


Figure 2.14 Pixel change activity comparisons in screen content and natural content.

## 2.4 Performance evaluation

This section presents a performance evaluation of HEVC and a comparison with that for H.264/AVC. Additionally, the features of HEVC that introduce significant gains for coding of high resolution and screen content are evaluated. While the detailed results presented here are based on an objective evaluation study, it should be noted that limited subjective test results reported in the literature already indicate superior subjective quality of HEVC, compared to H.264/AVC [24].

### 2.4.1 Objective quality evaluation

In objective video quality evaluations, the bit-rate reduction is typically calculated using *Bjontegaard Delta rate* (BD-rate) metric [25]. The distortion can be measured using *Bjontegaard Delta PSNR* (BD-PSNR) metric.

There are several configurations defined in JCT-VC common test conditions [26]. Common test conditions are desirable to conduct experiments during the development of HEVC. The configurations used in the experiments in this chapter include *All Intra* (AI), *Random Access* (RA) and *Low Delay with B slices* (LDB). Each frame is coded as I slice in AI. The RA configuration is designed for broadcasting environment which uses random access picture orders. The LDB configuration is typically applied in video conferences only the first frame is encoded as an intra frame. The common test conditions define four *Quantisation Parameters* (QP): 22, 27, 32 and 37. For each sequence, a *Rate Distortion* (RD) curve can be plotted by using the results of four pairs of bitrate and PSNR. According to the RD curves, BD-rate and BD-PSNR are computed to present the bit-rate reduction and distortion, respectively.

In addition, there are some other objective evaluation metrics that can be used in video quality evaluation. *Structural Similarity* (SSIM) [27] measures the similarity between two images by computing the local pixel information. *Video Quality Metric* (VQM) closely predicts the subjective quality ratings [27]. Zhang et al. proposed a human visual system perception-based model for video quality assessment [28]. These are not included in the experiments in this Chapter but may be used in future work.

### 2.4.2 Subjective quality evaluation

A subjective quality study reported in [24] involved 36 subjects that visually examined three complex video sequences compressed using both H.264/AVC and HEVC. The three video sequences, namely, *People On Street* ( $3840 \times 2160$ , 30 fps), *Traffic* ( $3840 \times 2048$ ,

30 fps) and Sintel2 ( $3840 \times 1744$ , 24 fps), are all near 4K resolution with frame rates of either 24 fps or 30 fps.

The video sequences are encoded at various bit-rates in a Random Access configuration. Considering the different content features and different spatio-temporal characteristics, the bit-rates are set separately for each sequence. The research uses *The Double Stimulus Impairment Scale* (DSIS) method which presents video sequences continuously to subjects and asks them to rate visual quality on a rating scale with a maximum score of 100. The distance between the monitor and the subjects was set approximately 3.5 times the height of the screen.

Some sequences used in the visual test are relatively hard to encode because of related higher spatial information and temporal information indexes. For example, in *PeopleOnStreet* sequence the content was smoothed in the HEVC reconstructed sequences while blocking was perceived in H.264/AVC. For this content, HEVC achieved a high bit-rate reduction up to 74%. The sequence Traffic is easier to encode but also benefits from high bit-rate reductions introduced by HEVC. In subjective tests BD-MOS values of HEVC over H.264/AVC range from 51% to 74%. The corresponding BD-PSNR values range from 28% to 68%.

The study presented in [29] evaluated the compression performance for HEVC standard compared with its predecessor AVC on UHD content. The subjective evaluation metric *Degradation Category Rating* (DCR) was used in the tests. The results verified that HEVC achieved bit rate saving of more than 50% compared to AVC for the same quality. The DCR subjective evaluation is used in Chapter 4 and Chapter 5.

### 2.4.3 Experiments and results

The performance of the HEVC test model version 8.0 (HM-8.0) was evaluated and compared to H.264/AVC (JM-18.4). All JCT-VC test video sequences specified in [26] were used in tests, including Class F, which consists of the screen content video. In the test that compares H.264/AVC and HEVC performance; six configurations were used for HEVC: AI-Main, RA-Main, LDB-Main, AI-HE, RA-HE and LDB-HE, where HE stands

for High Efficiency. The High Efficiency setting defined in [26] differs from the Main setting in terms the internal bit-depth used. While the Main setting used 8-bit internal processing precision, the HE setting uses 10-bit internal bit-depth. For evaluations of different HEVC settings, only the Main configurations were used.

### 2.4.3.1 Test sequence

The JCT-VC common test condition provides six classes of test sequences. In general, Class A to E test sequences are camera captured content and Class F contains screen content sequences. All sequences are in 4:2:0 chroma sampling format. There are four Class A sequences with the resolution of  $2560 \times 1600$  pixels and 30 and 60 fps. Class A sequences have the highest resolution among all JCT-VC test sequences. Classes B (5 sequences,  $1920 \times 1080$ , 24 to 60 fps), C (4 sequences,  $832 \times 480$ , 30 to 60 fps) and D (4 sequences,  $416 \times 240$ , 50 and 60 fps) are tested in all configurations. Class E is video conferencing material which is not tested in RA configurations. All three sequences ( $1280 \times 720$ , 60 fps) are tested in AI and LDB configurations. Class F ( $832 \times 480$  to  $1280 \times 720$ , 20 to 50 fps) contains four sequences that combined camera captured and text/ graphical content, such as graphical overlays, a computer game and computer desktop content. Class G ( $3840 \times 2160$ , 30 to 60 fps) have four UHD video sequences with different frame rates. Class G is not contained in common test condition but listed in the table to compare the performance evaluations of different temporal resolutions.

### 2.4.3.2 Comparisons of HEVC and H.264/AVC

A summary of BD-rate results of the luma component for all JCT-VC test sequences comparing HEVC performance (Main setting) to the performance of H.264/AVC are listed in *Table 2.2*. Negative values indicate bit-rate savings. The analysis of random access results is shown in *Figure 2.15*.

Table 2.2 BD-rate results of luma component for HEVC compared to H.264/AVC.

Class	Sequence name	Resolution	Bit depth	Frame rate (fps)	All Intra-Main	Random Access-Main	Low-Delay-B Main
A	Traffic	2560×1600	8	30	-21.6%	-36.8%	N/A
A	People On Street	2560×1600	8	30	-22.4%	-23.4%	N/A
A	Nebuta	2560×1600	10	60	-25.8%	-30.6%	N/A
A	Steam Locomotive	2560×1600	10	60	-24.8%	-56.0%	N/A
B	Kimono	1920×1080	8	24	-28.6%	-44.5%	-44.1%
B	Park Scene	1920×1080	8	24	-16.4%	-31.5%	-34.3%
B	Cactus	1920×1080	8	50	-22.2%	-35.9%	-38.2%
B	BQ Terrace	1920×1080	8	60	-19.4%	-44.1%	-49.5%
B	Basketball Drive	1920×1080	8	50	-26.9%	-42.9%	-44.9%
C	Race Horses	832×480	8	30	-17.3%	-25.3%	-27.5%
C	BQ Mall	832×480	8	60	-19.5%	-32.4%	-33.2%
C	Party Scene	832×480	8	50	-11.6%	-28.3%	-31.8%
C	Basketball Drill	832×480	8	50	-30.3%	-35.0%	-38.1%
D	Race Horses	416×240	8	30	-18.4%	-23.2%	-24.6%
D	BQ Square	416×240	8	60	-12.9%	-38.3%	-40.4%
D	Blowing Bubbles	416×240	8	50	-12.7%	-23.9%	-26.6%
D	Basketball Pass	416×240	8	50	-21.7%	-25.9%	-27.3%
E	Four People	1280×720	8	60	-23.3%	N/A	-34.5%
E	Johnny	1280×720	8	60	-34.7%	N/A	-52.8%
E	Kristen And Sara	1280×720	8	60	-28.6%	N/A	-44.6%
F	Basketball Drill Text	832×480	8	50	-27.1%	-32.9%	-37.0%
F	China Speed	1024×768	8	30	-27.6%	-32.0%	-34.4%
F	Slide Editing	1280×720	8	30	-28.3%	-27.7%	-31.8%
F	Slide Show	1280×720	8	20	-30.4%	-30.5%	-31.6%
G	BT709Birthday	3840×2160	8	50	N/A	-62%	N/A
G	Book	3840×2160	8	50	N/A	-59%	N/A
G	Manege	3840×2160	8	60	N/A	-33%	N/A
G	Traffic2	3840×2160	8	30	N/A	-41%	N/A

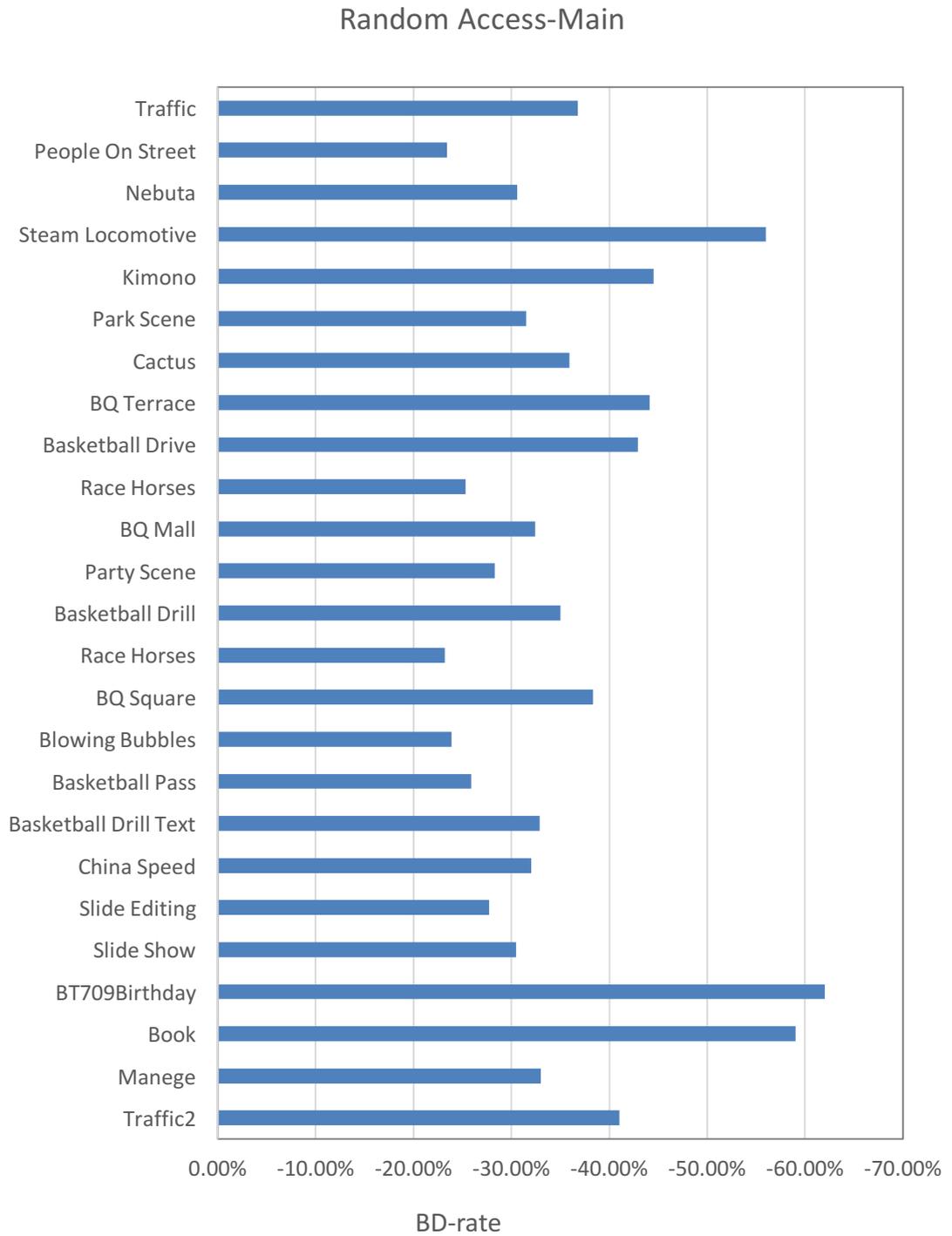


Figure 2.15 BD-rate saving comparisons (Random Access-Main).

The BD-rate average results is summarised in *Table 2.3* and *Figure 2.16*.

Table 2.3 BD-rate average results for luma component for HEVC compared to H.264/AVC.

Class	AI-Main	AI-HE	RA-Main	RA-HE	LDB-Main	LDB-HE
A	-23.7%	-25.2%	-36.7%	-38.1%	N/A	N/A
B	-22.7%	-23.8%	-39.8%	-41.0%	-42.2%	-43.5%
C	-19.7%	-20.2%	-30.3%	-31.0%	-32.6%	-33.3%
D	-16.4%	-16.8%	-27.8%	-28.3%	-29.7%	-30.2%
E	-28.9%	-30.2%	N/A	N/A	-44.0%	-46.4%
F	-28.4%	-28.3%	-30.8%	-30.9%	-33.7%	-32.9%
G	N/A	N/A	-48.75%	N/A	N/A	N/A

BD-rate saving comparisons

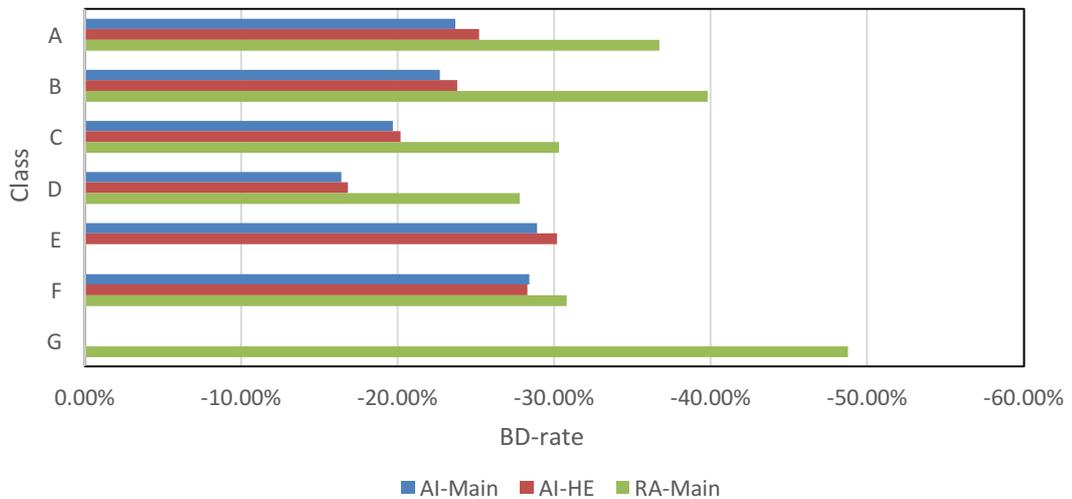
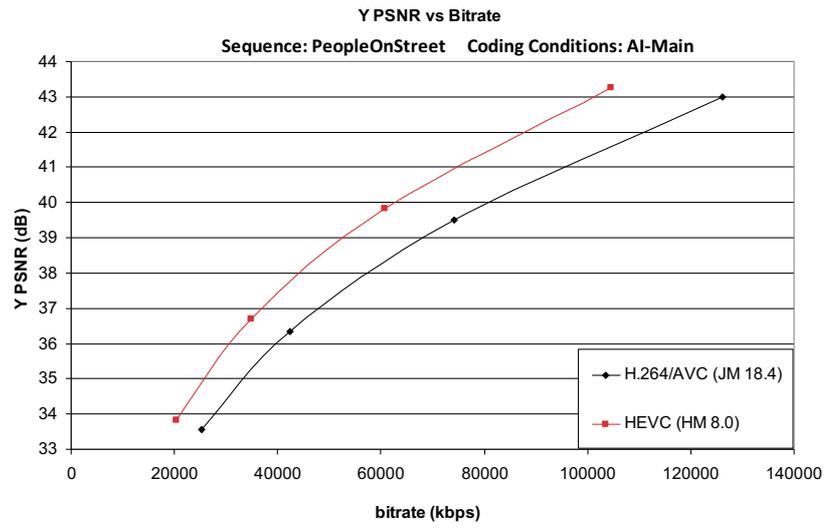


Figure 2.16 BD-rate saving comparisons (on average).

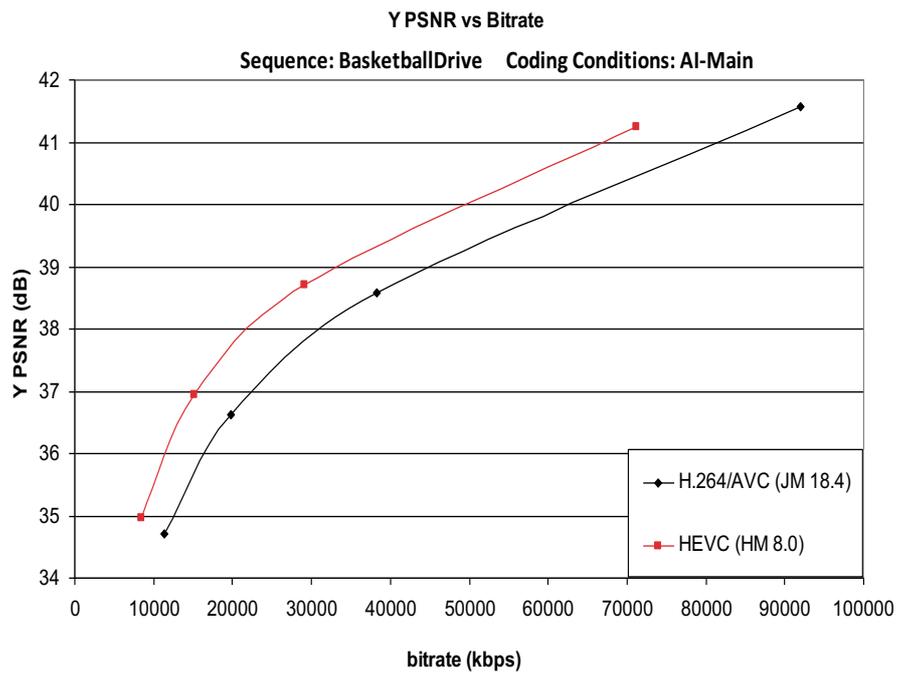
The results indicate:

1. HEVC achieves larger bitrate saving on higher resolution videos (Class A, B and E) than lower resolutions (Class C and D). It can be seen from BD-rate saving comparisons in *Figure 2.15* and *Figure 2.16*.
2. The largest bitrate saving gains are observed for the low-delay configuration. Due to the new coding tools used in HEVC (adaptive large block partition, motion compensation and SAO loop filter), frames can be more accurately predicted from inter coded frames, and therefore the gain is mostly related to inter coding configurations (RA and LDB).
3. In the High Efficiency settings, gains are larger confirming the benefits of using higher internal bit-depth processing settings.
4. For screen content (Class F), HEVC achieves gains in the range of 28.3% to 35% BD-rate.
5. For UHD content (Class G), HEVC achieves the highest BD-rate saving (up to 62%).

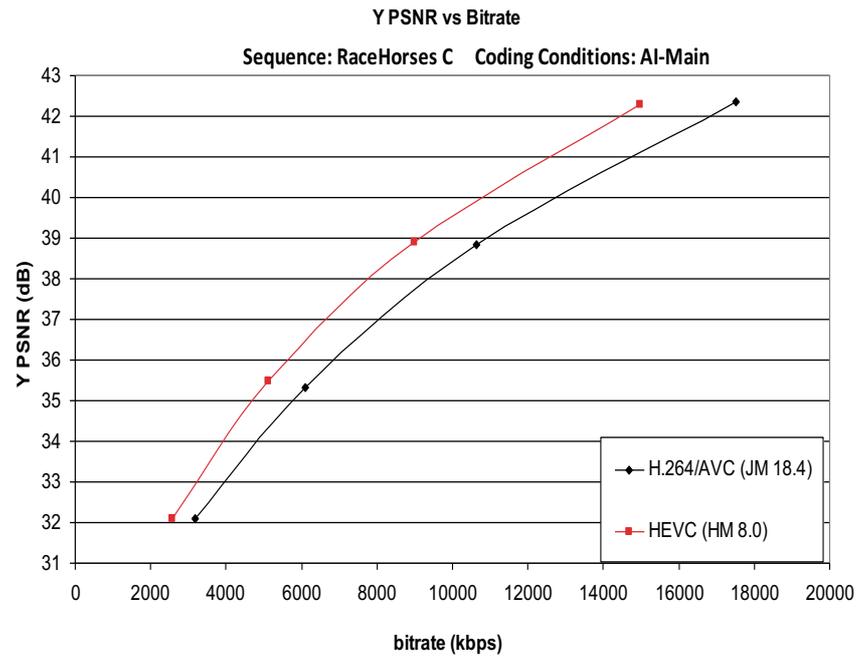
To demonstrate the PSNR and bit-rate ranges involved in BD-rate computations, sequences are selected from each Class (from A to F) providing a selection of content with different resolution and frame rates. Figure 2.17 shows the rate-distortion (RD) curves for the selected test sequences from each Class in AI-Main configuration.



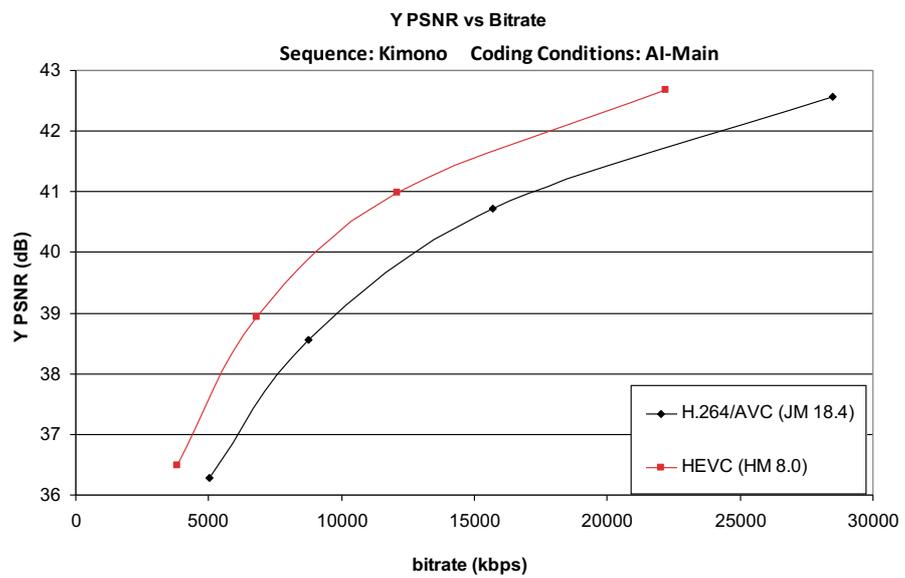
(1)



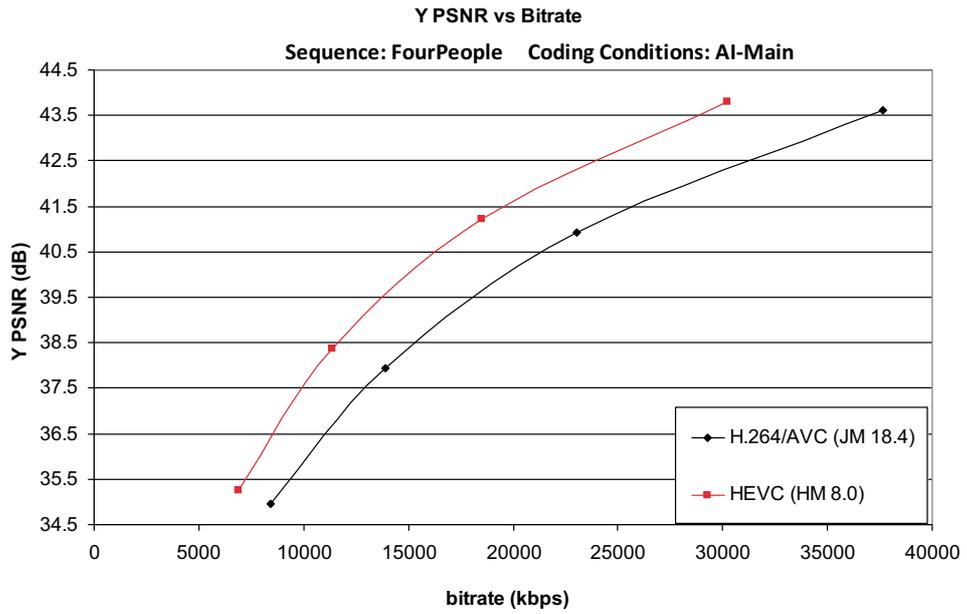
(2)



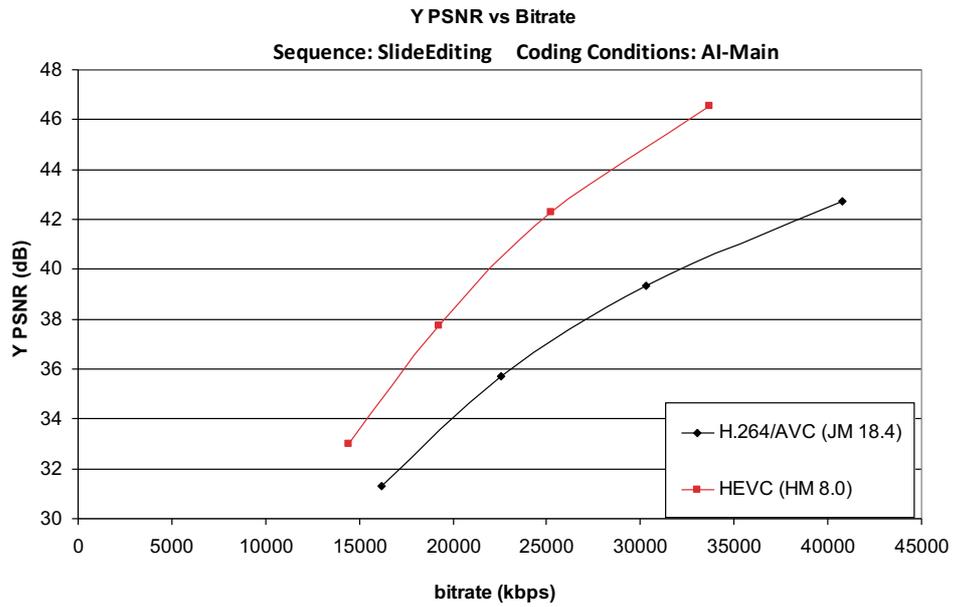
(3)



(4)



(5)



(6)

Figure 2.17 Comparison of HEVC and H.264/AVC coding performance (all intra). (1) People On street (Class A), (2) Basketball Drive (Class B), (3) Race Horses (Class C), (4) Kimono (Class D), (5) Four People (Class E), (6) Slide Editing (Class F).

### 2.4.3.3 Coding with reduced block sizes

The results from Section 2.4.3.2 indicate that HEVC achieves larger gains on higher resolution videos (Class A, B and E) and smaller gains on lower resolutions (Class C and D). This behaviour comes partially from the application of larger block sizes whose performance is evaluated in this subsection.

Under HEVC common test conditions, each frame is split into a number of CTUs with luma block size of  $64 \times 64$  pixels. Each CTU can be recursively split into smaller CUs depending on the applied coding conditions and rate-distortion decisions. In order to evaluate the influence of different block sizes, two tests were performed. While keeping the common test conditions settings with CTUs with  $64 \times 64$  blocks, in the first test the maximal CU size was limited to  $16 \times 16$  pixels (equivalent to macroblock size in H.264/AVC), while in the second test the maximal CU size was limited to  $8 \times 8$  pixels. The results are compared with HEVC results reported in the previous section (where maximal CU size equals CTU size).

The results obtained are summarised in Table 2.4 to Table 2.7. Positive BD-rate values indicate a bit-rate increase introduced by the limitation of block sizes available for coding. The figure bars in *Figure 2.18* to *Figure 2.21* shows the related graph comparisons. The results show that the compression of JCT-VC sequences with larger resolution significantly benefits from the application of larger blocks for inter coding configurations (RA and LDB).

CHAPTER 2. HEVC ANALYSIS AND EVALUATIONS

Table 2.4 BD-rate results for luma component for HEVC with maximal sizes of CU of  $16 \times 16$  pixels, compared to typical HEVC settings with maximal CU size of  $64 \times 64$  pixels.

Class	Sequence name	AI-Main	RA-Main	LDB-Main
A	Traffic	1.1%	13.5%	N/A
A	People On Street	0.7%	3.0%	N/A
A	Nebuta	5.3%	11.7%	N/A
A	Steam Locomotive	7.0%	56.4%	N/A
B	Kimono	5.8%	21.8%	16.6%
B	Park Scene	1.1%	10.7%	9.2%
B	Cactus	1.3%	10.6%	9.3%
B	BQ Terrace	0.9%	14.9%	18.3%
B	Basketball Drive	2.7%	18.8%	16.4%
C	Race Horses	0.9%	5.7%	4.6%
C	BQ Mall	0.8%	8.7%	8.6%
C	Party Scene	0.1%	4.3%	4.0%
C	Basketball Drill	0.7%	8.4%	8.7%
D	Race Horses	0.5%	2.8%	2.2%
D	BQ Square	0.3%	4.5%	4.8%
D	Blowing Bubbles	0.0%	3.2%	3.4%
D	Basketball Pass	1.0%	4.1%	3.8%
E	Four People	1.2%	N/A	16.0%
E	Johnny	6.6%	N/A	50.0%
E	Kristen And Sara	3.5%	N/A	32.8%
F	Basketball Drill Text	0.5%	6.6%	6.8%
F	China Speed	0.7%	6.1%	6.3%
F	Slide Editing	0.5%	4.1%	12.9%
F	Slide Show	3.2%	9.7%	11.9%

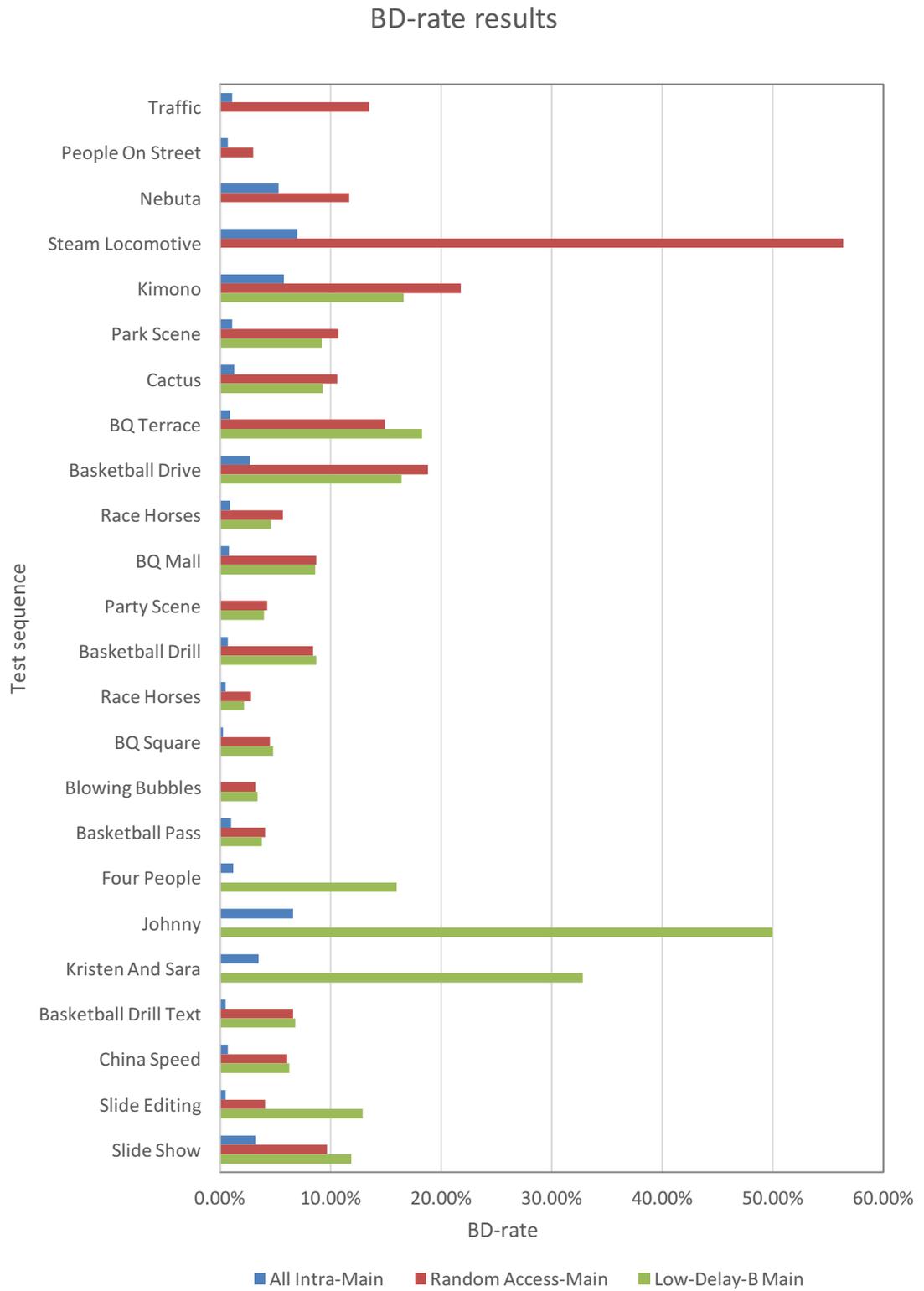


Figure 2.18 BD-rate results of reduced block sizes ( $16 \times 16$ ).

Table 2.5 Average BD-rate for luma component for HEVC with maximal sizes of CU of  $16 \times 16$  pixels, compared to typical HEVC settings with maximal CU size of  $64 \times 64$  pixels.

Class	AI-Main	RA-Main	LDB-Main
A	3.5%	21.1%	N/A
B	2.4%	15.4%	14.0%
C	0.6%	6.8%	6.5%
D	0.5%	3.7%	3.5%
E	3.8%	N/A	32.9%
F	1.2%	6.7%	9.5%

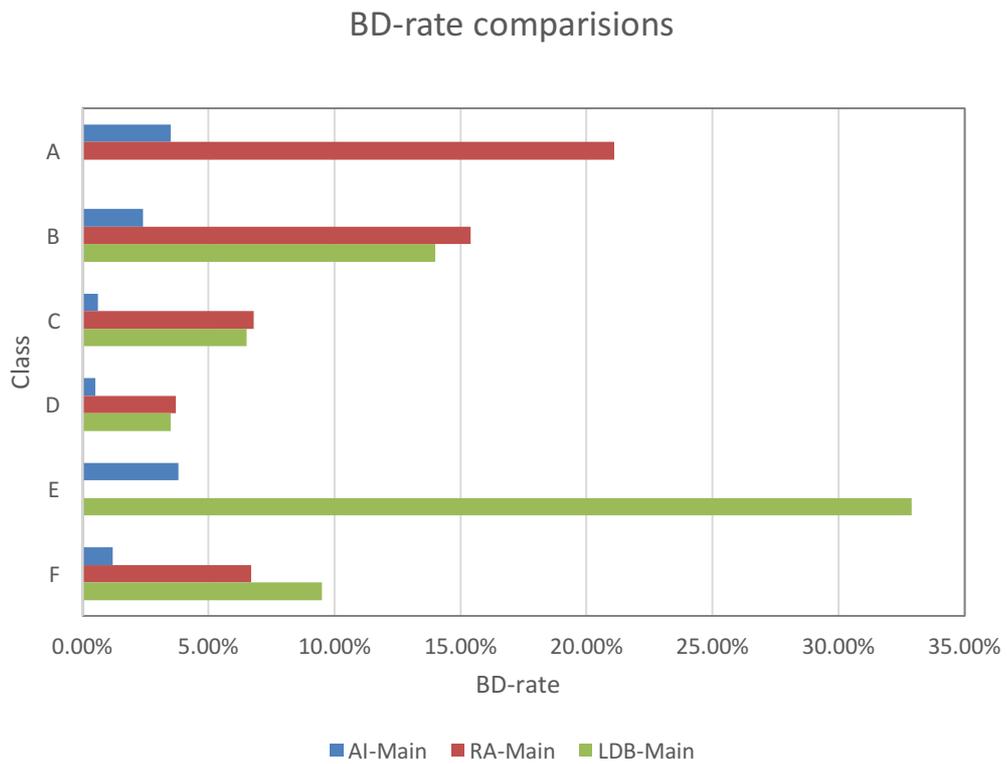


Figure 2.19 BD-rate results of reduced block sizes ( $16 \times 16$ ) on average.

Table 2.6 BD-rate results for luma component for HEVC with maximal sizes of CU of  $8 \times 8$  pixels, compared to typical HEVC settings with maximal CU size of  $64 \times 64$  pixels.

Class	Sequence name	All Intra-Main	Random Access-Main	Low-Delay-B Main
A	Traffic	7.1%	34.7%	N/A
A	People On Street	5.4%	17.8%	N/A
A	Nebuta	17.0%	34.3%	N/A
A	Steam Locomotive	28.2%	165.8%	N/A
B	Kimono	23.2%	71.1%	63.3%
B	Park Scene	6.2%	29.0%	30.4%
B	Cactus	7.2%	37.7%	37.2%
B	BQ Terrace	4.7%	38.3%	48.1%
B	Basketball Drive	14.3%	62.7%	55.7%
C	Race Horses	4.6%	27.1%	23.3%
C	BQ Mall	4.8%	28.0%	29.4%
C	Party Scene	0.8%	14.2%	17.4%
C	Basketball Drill	4.2%	27.7%	28.3%
D	Race Horses	3.2%	17.7%	15.1%
D	BQ Square	1.8%	14.6%	19.9%
D	Blowing Bubbles	0.7%	12.3%	16.1%
D	Basketball Pass	6.2%	19.0%	17.8%
E	Four People	8.7%	N/A	47.8%
E	Johnny	24.0%	N/A	126.2%
E	Kristen And Sara	16.4%	N/A	87.8%
F	Basketball Drill Text	3.3%	23.0%	24.4%
F	China Speed	4.4%	23.2%	25.5%
F	Slide Editing	2.9%	10.2%	32.4%
F	Slide Show	16.5%	32.8%	37.3%

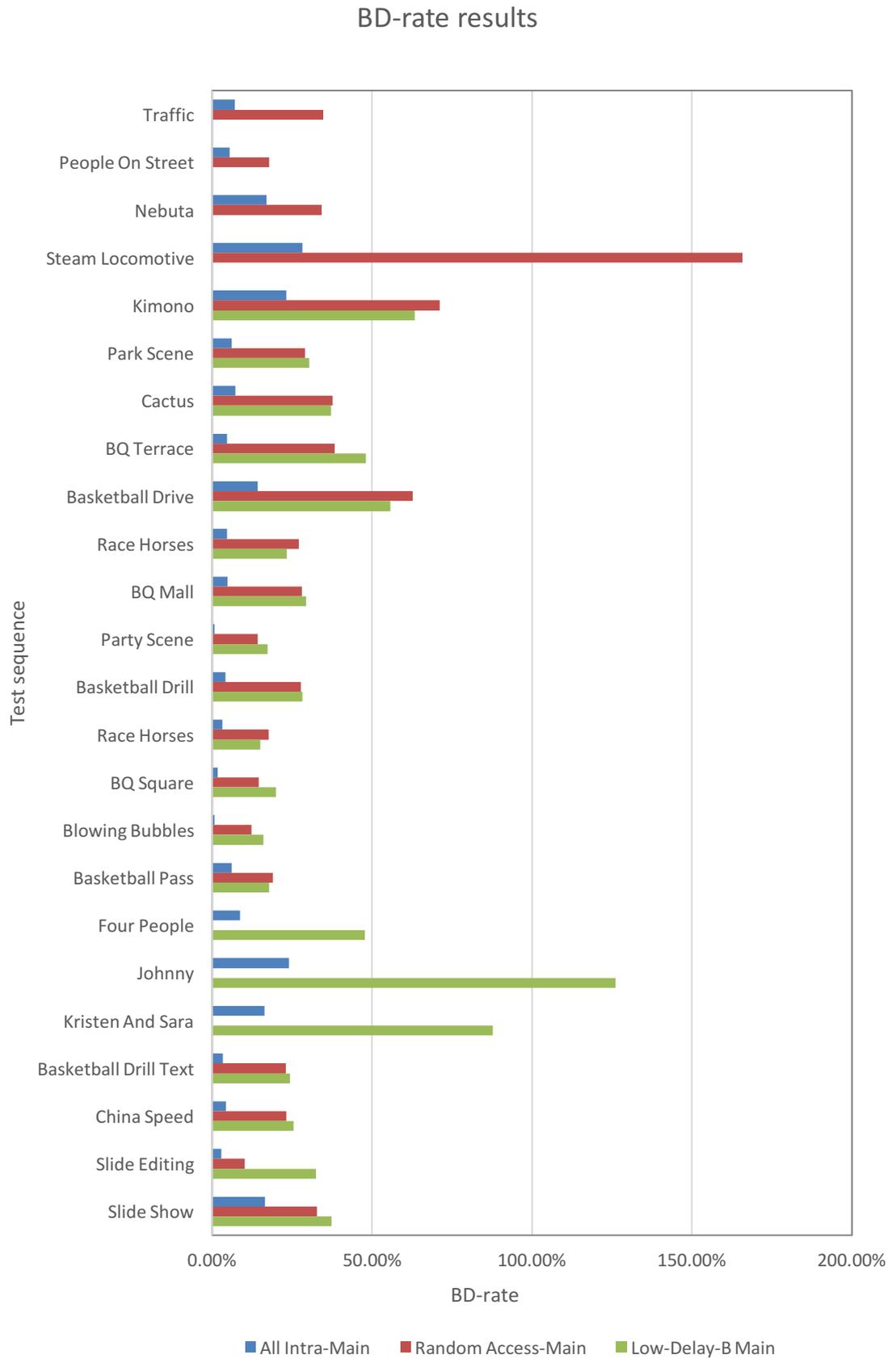


Figure 2.20 BD-rate results of reduced block sizes ( $8 \times 8$ ).

Table 2.7 Average BD-rate for luma component for HEVC with maximal sizes of CU of  $8 \times 8$  pixels, compared to typical HEVC settings with maximal CU size of  $64 \times 64$  pixels.

Class	AI-Main	RA-Main	LP-Main
A	14.4%	63.1%	N/A
B	11.1%	47.8%	46.9%
C	3.6%	24.3%	24.6%
D	3.0%	15.9%	17.2%
E	16.4%	N/A	87.3%
F	6.8%	22.3%	29.9%

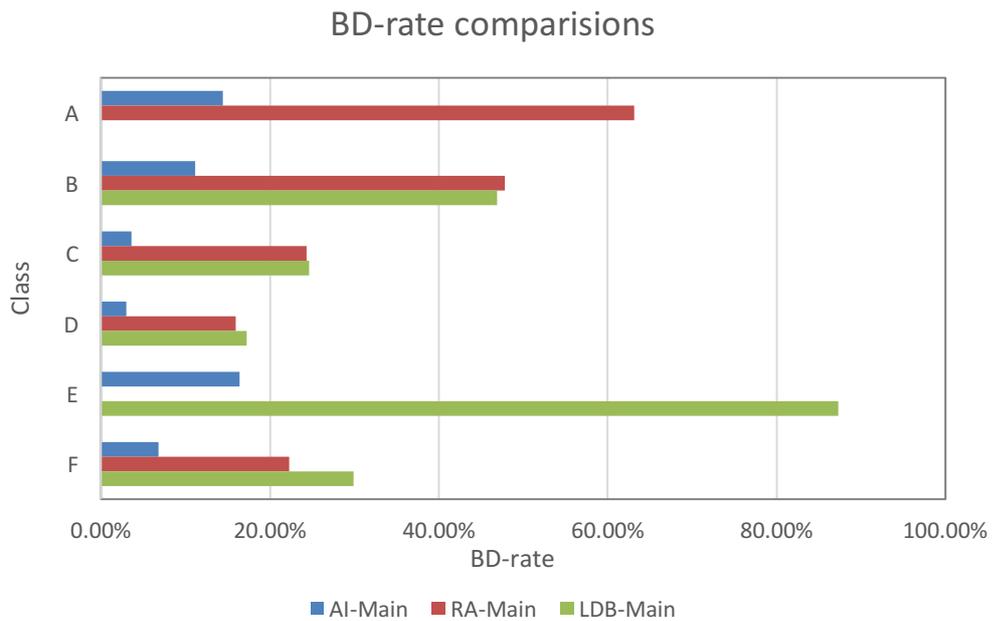
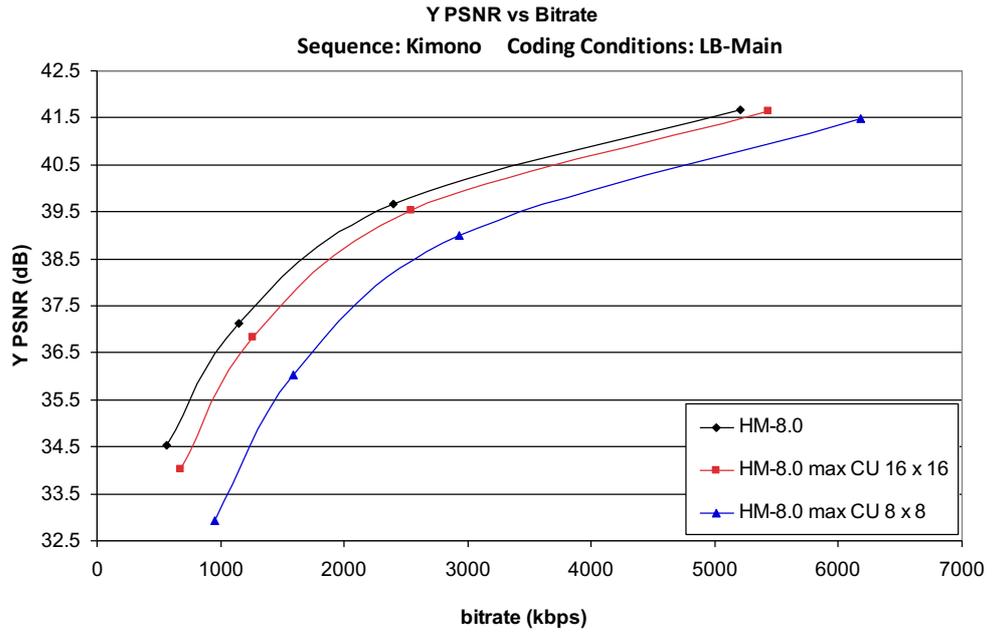
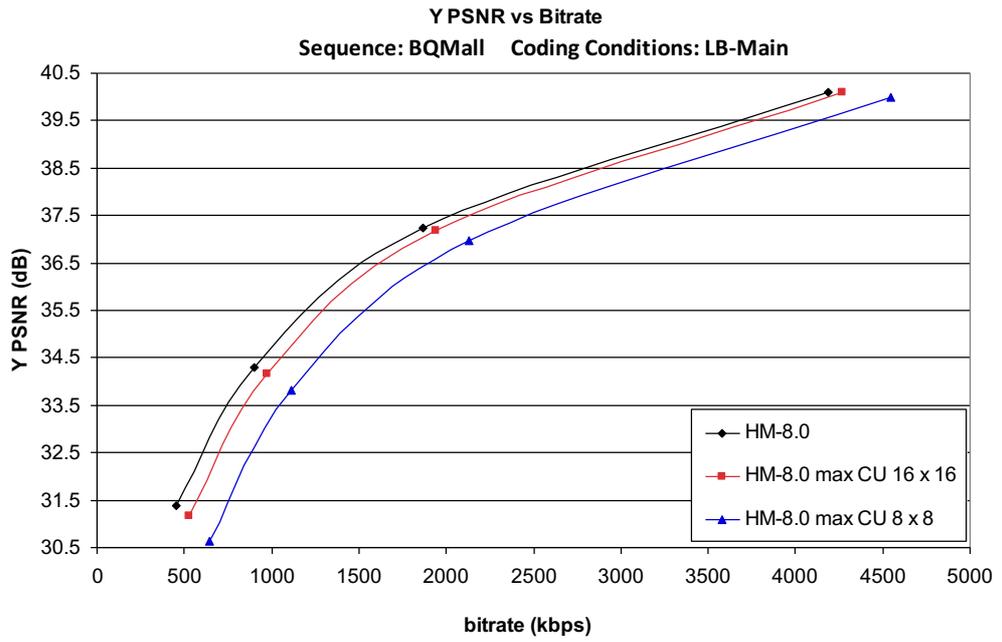


Figure 2.21 BD-rate results of reduced block sizes ( $8 \times 8$ ) on average

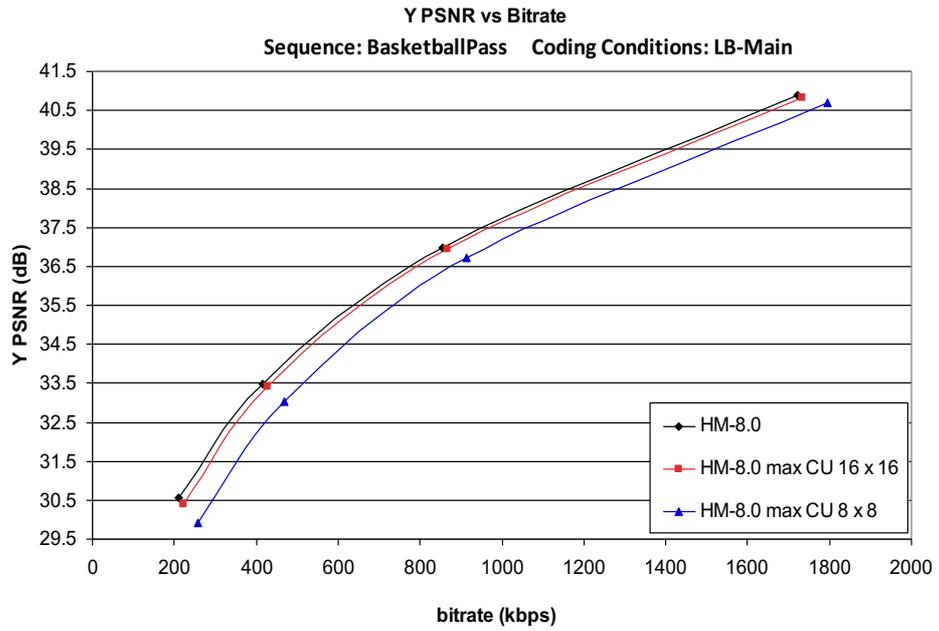
The rate-distortion curves for a selection of sequences of different resolutions coded in the LDB-Main configuration are shown in *Figure 2.22*. From the RD curves it is evident that adaptive block partition in HEVC with larger maximal block sizes contributes to efficient compression at all tested bit-rates.



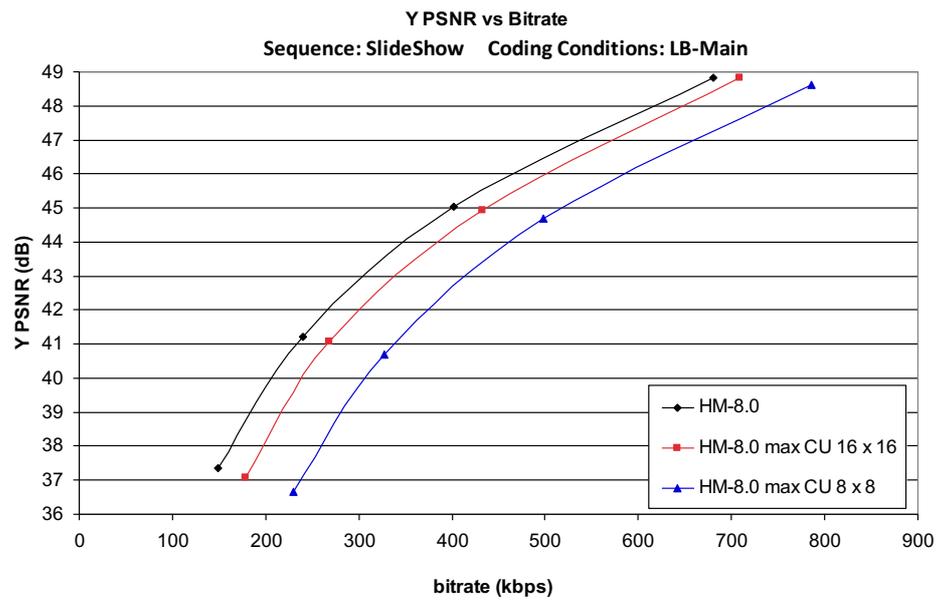
(1) Kimono



(2) BQ mall



(3) Basketball



(4) Slide Show

Figure 2.22 Comparison of HEVC with different maximal CU sizes on LDB-Main configuration. (1) Kimono (Class B), (2) BQ Mall (Class C), (3) Basketball Pass (Class D), (4) Slide Show (Class F).

#### 2.4.3.4 Transform skipping for screen content

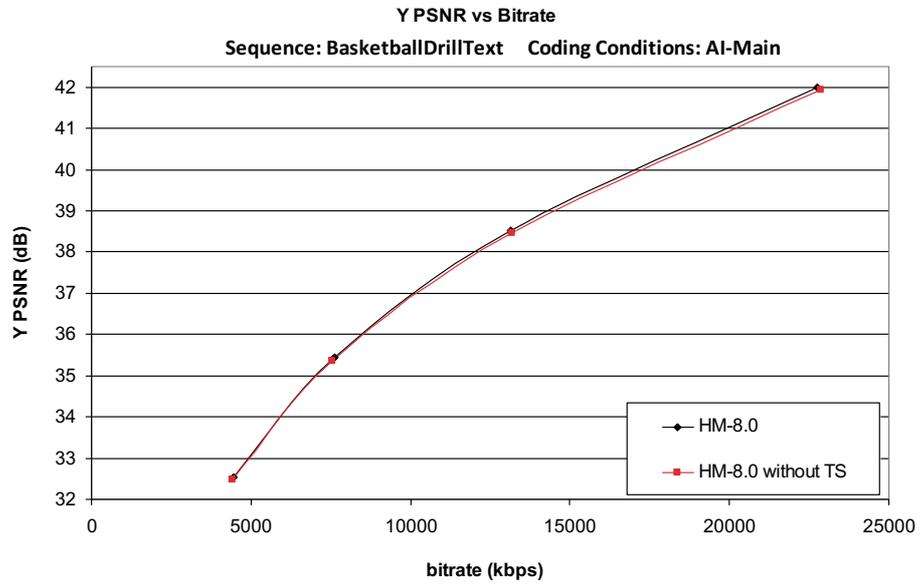
As discussed in Section 2.2.3, transform skipping (TS) can be efficiently used on certain content types, namely, on screen content that contains computer generated elements. In this test the efficiency of transform skipping is evaluated by disabling the normative transform skip option at the encoder, i.e., by forcing application of transforms on each coded block.

While HM-8.0 and common test conditions are used, as in the other evaluations reported in this chapter, the focus of this experiment is on screen content (Class F). However, it should be noted that the effect of disabling transform skip does not lead to significant change in coding performance for Class A to E sequences.

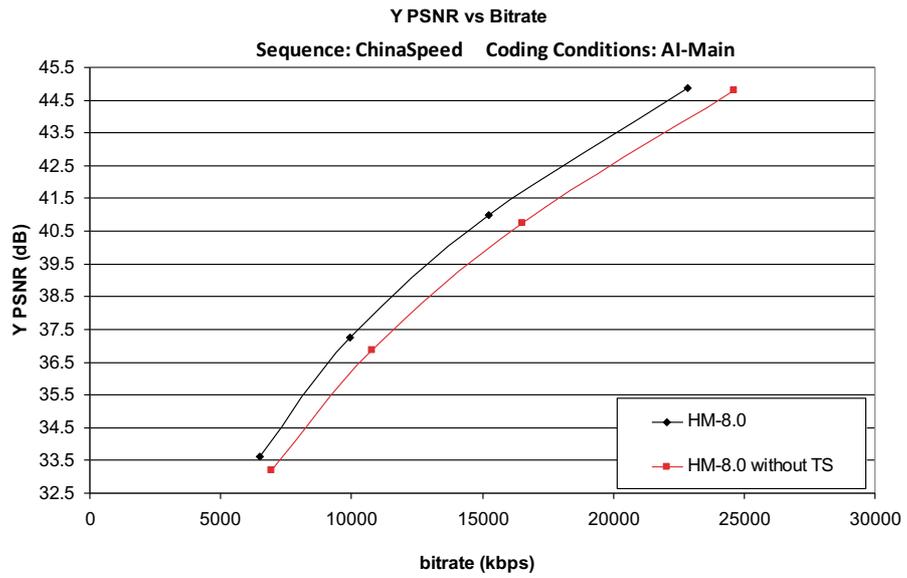
The results for four screen content sequences are summarised in *Table 2.8*, where HEVC without transform skip is compared to HEVC with transform skip (common test conditions). Positive values of BD-rate indicate losses introduced when transform skip is not an option for compression of given content. The resulting BD-Rate ranged from 0.4 % to 16.2% with an average of 8.2%, 7.3% and 6.4% in All Intra, Random Access and Low Delay configurations, respectively. It should be noted that the Basketball Drill Text sequence mainly consist of camera captured content and therefore its coding does not greatly benefit from transform skipping. Figure 2.23 shows the RD curves for the four test sequences in AI-Main configuration. The results indicate that the compression efficiency can be improved for screen content when using transform skipping in HEVC.

Table 2.8 BD-rate results for luma component for HEVC without transform skip compared to HEVC with transform skip (common test conditions)

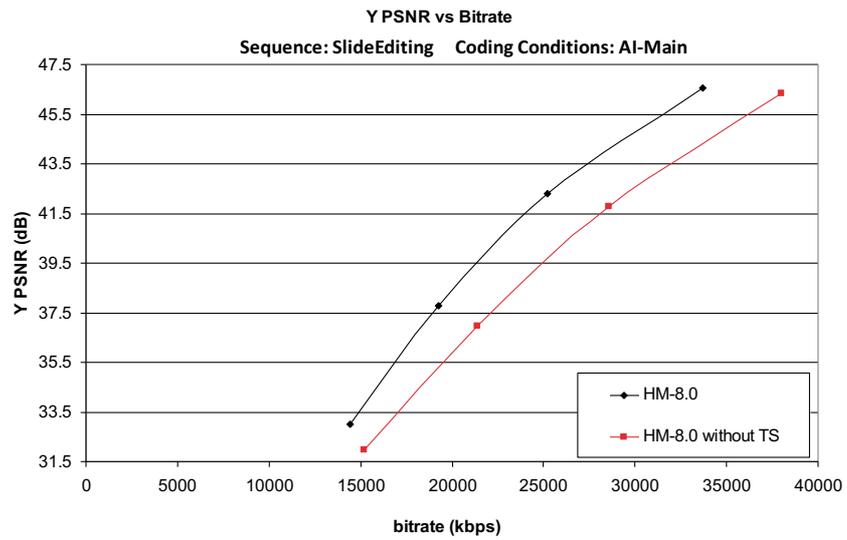
Class	Sequence name	All Intra-Main	Random Access-Main	Low-Delay-B main
F	Basketball Drill Text	0.7%	0.6%	0.4%
F	China Speed	11.6%	11.1%	10.2%
F	Slide Editing	16.2%	13.3%	11.5%
F	Slide Show	4.1%	4.3%	3.4%



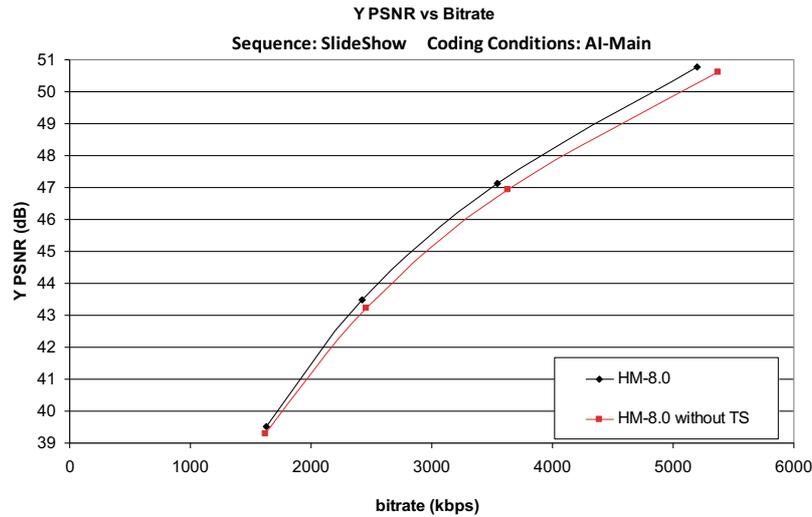
(1) BasketballDrillText



(2) ChinaSpeed



(3) Slide Editing



(4) Slide Show

Figure 2.23 Comparison of HEVC without TS and HEVC standard (all intra). (1) Basketball Drill Text, (2) China Speed, (3) Slide Editing, (4) Slide Show.

## 2.5 Summary

This chapter introduced the new video compression standard HEVC, which is being developed jointly by ITU-T and ISO/IEC. It also discussed the key features of HEVC and compared it to earlier coding standards. The performance evaluation results have confirmed that more than 50% bit-rate saving in HEVC for certain video quality compared to H.264/AVC [29]. This is more evident for high-definition video sequences. The Version 1 of HEVC has been released in 2013 [30] [31] and second version of HEVC has been released in 2014 which added 21 range extensions profiles [31]. The development of the HEVC is essential for the next generation video compression technology. This thesis focus on two types of next generation video content: screen content and UHD content. Since screen content has different features, further study on screen content coding tools are needed. Chapter 3 investigates and develops several colour space transforms for screen content compared with natural content. On aspect of visual artefacts, the contouring artefacts which show in HEVC compressed UHD content. There are three proposed contouring removal methods includes in this thesis which are detailed in Chapter 4 and Chapter 5.

## Chapter 3 Colour transforms for HEVC range extensions

HEVC, which was introduced in Chapter 2, is designed as a high efficiency video coding for next generation video content. This chapter investigates and develops novel colour space conversion methods for converting RGB colour space into a luminance and two chrominance colour space suitable for HEVC range extension.

RGB colour space is the common colour space used in TV or computer devices and video displays. The RGB image or video are usually represented by three values per pixel indicating the relative components of red, green and blue. These are the three most sensitive colours for human eyes. The RGB colour space weighs each of these three colour components equally and represents each component with the same precision. For instance, for an 8 bits signal, 24 ( $3 \times 8$ ) bits are required to represent one pixel. Each component is represented by the value ranging from 0 to 255. Scale the maximum pixel values up to higher bit depths if using more than 8 bits.

The colour space transform is one of the significant pre-processing steps that enables removal of the inter-spectral redundancy in RGB colour space. It does so by compacting the majority of the energy in one spectral band, usually in the luminance band, and by highly de-correlating two chrominance bands. The latter can be even subsampled further as in 4:2:2 and 4:2:0 chroma sampling to reduce the data rates.

The most commonly used colour transform in video compression is YCbCr (luma/ blue chroma/ red chroma) [32] [33]. YCbCr 4:2:0 is the basic colour space and colour sampling format described in HEVC main profile. Since some new types of content such as screen content and high bit depth (higher than 8 bits) content come out in the format of 4:4:4 RGB, we take into account whether this conventional colour transform method is still desirable in compression. Furthermore, the HEVC main profile is not sufficient for higher than 10 bits, which leads to the HEVC Range extension test model that not only

supports different video formats but also higher bit depth. The question rises whether the commonly used YCbCr is the most efficient colour space to be used in HEVC, especially for screen content.

The objective of this chapter is to study the colour coding transforms of different video content in HEVC range extension. There are several related work on colour transforms in HEVC, as reviewed in Section 3.2. YCoCg colour space [34] is also included in this chapter for comparisons. Another reversible colour transform included in this chapter is 3-point *Pseudo Discrete Cosine Transform* (P-DCT). The transform matrix is normalised from 3-point DCT [35]. *Karhunen-Loève Transform* (KLT) algorithm [33] [36] can maximum decorrelate the RGB components. It is proved that KLT is the best colour transform method for test images in [38]. However, it is inapplicable to be used in real-time video coding. The KLT approximations are proposed in Section 3.3. Another proposed colour space transform method in this chapter, namely 3-channel PPU, is based on prediction-and- update lifting scheme. PPU transforms are designed for screen content and can be extended to a series of transforms by setting the parameters, which is detailed in Section 3.4.3.

Test sequences and performance evaluation metrics are provided in Section 3.5. Section 3.6 presents the experimental results of compression performance with the above mentioned colour transforms. The results analysis and discussion presented in Section 3.7 followed by the summary in Section 3.8.

### 3.1 Conventional Colour transform

The most commonly used colour space in video compression is YCbCr, where luma component (Y) represents the brightness information and the chroma components (Cb and Cr) indicate colour difference. Y is computed as a weighted average of R, G and B as:

$$Y = k_r R + (1 - k_r - k_b)G + k_b B. \quad (3.1)$$

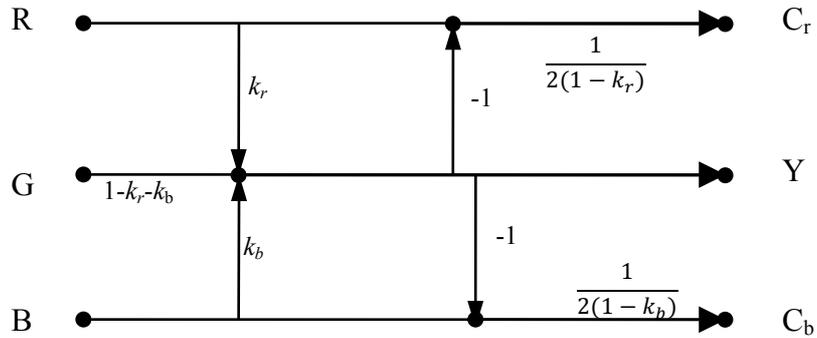
The lowercase letters,  $k_r$  and  $k_b$  represent the luma coefficients. Two chroma components  $C_b$  and  $C_r$  are proportional to colour difference  $(B - Y)$  and  $(R - Y)$ , respectively. The resulting components are nominalised so that  $Y$  is ranging from 0 to 255, and two chroma components are ranging from -128 to 128 converted from 8 bits RGB signals. The conversion of YCbCr colour space is given by the equations below [32]:

$$\begin{aligned} Y &= k_r R + (1 - k_r - k_b)G + k_b B, \\ C_b &= \frac{1}{2(1-k_b)}(B - Y), \\ C_r &= \frac{1}{2(1-k_r)}(R - Y). \end{aligned} \tag{3.2}$$

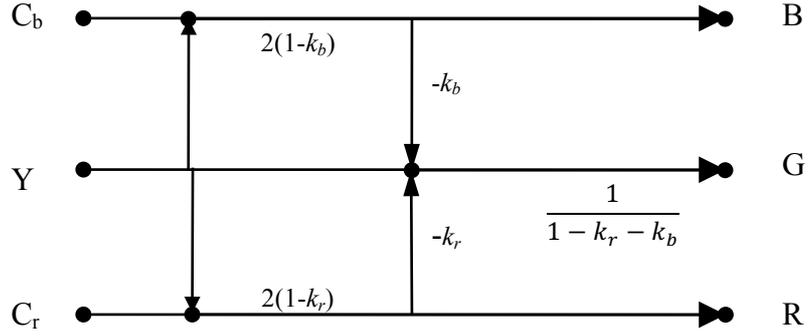
The inverse conversion is described by the following equations:

$$\begin{aligned} R &= 2(1 - k_r)C_r + Y, \\ B &= 2(1 - k_b)C_b + Y, \\ G &= \frac{1}{(1-k_r-k_b)}(Y - k_r R - k_b B). \end{aligned} \tag{3.3}$$

The flow graphs of forward and inverse transforms are depicted in Figure 3.1.



a) Forward transform



b) Reverse transform

Figure 3.1 Flow graphs of the RGB ↔ YCbCr colour conversion.

Another way to represent YCbCr colour space is to use transform matrix. There are several representations of the transform matrix, depending on the input video resolution.

For *Standard Definition* (SD) resolution, the coefficients are set as  $k_r = 0.299$  and  $k_b = 0.114$ . The conversion of RGB colours into full range YCbCr colour is described by the following equation (ITU\_R BT601 [39]):

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.4)$$

The other way round, to convert a full range YCbCr colour into RGB is described by the following equation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & -0.714 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}. \quad (3.5)$$

In ITU-R BT.709 [13], the colour transform is defined for HD content with coefficients  $k_r = 0.2126$  and  $k_b = 0.0722$  and the actual transform as:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3857 & 0.5000 \\ 0.500 & -0.4542 & -0.0468 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.6)$$

This is the corresponding inverse matrix to get the RGB colour components out of YCbCr colour:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.570 \\ 1 & -0.187 & -0.467 \\ 1 & 1.856 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}. \quad (3.7)$$

The test sequences in this chapter are all HD. The experiments on YCbCr colour space transform in this chapter are applied following ITU-T BT.709.

### 3.2 Related work

By the time we started the colour transform study, there was not too many research on colour transforms for HEVC. However, as the development of HEVC profiles, several improved colour transforms are proposed by other researchers. An in-loop adaptive colour transform was proposed in [40]. The colour transform is block based and applied on residuals. Each coding unit can adaptively select a set of the colour transforms which can achieve the lowest RD costs. This adaptive colour transform method is adopted in the HEVC range extension for screen content. However, only limit sets of colour transforms are selected. The colour transform proposed in [41] is another in-loop method designed for HEVC range extension. The method uncorrelated colour channels by using singular value decomposition. The colour space investigation in [42] includes a large set of colour transform matrices. It uses several possible low complexity calculations for luma and chroma components and derive the computation matrices as combinations. The methods was improved with an adaptive selection of colour transforms without increase the bit depth [43]. A similar method was proposed for applying in HEVC, as presented in [44].

The rest of this section reviews several reversible colour spaces which can be used in HEVC range extensions. The input RGB video format can be converted into a luma-chroma format and vice versa. All transform matrices are introduced as well as their inverse transform matrices.

### 3.2.1 YCoCg

YCoCg is another colour spaces that consist of luma and two chroma components. It can be implemented as a lossless transform since there are only two binary shifts involved. Modern video codec, such as AVC/H.264, also supports YCoCg colour space [37]. It reserves the same luma-chroma model but chroma involves different colour components: orange (Co) and green (Cg).

The following equations describe the conversion from RGB to YCoCg colour format. The transform matrix is [34]:

$$\begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 0 & -0.5 \\ -0.25 & 0.5 & -0.25 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.8)$$

And the inverse transform is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix}. \quad (3.9)$$

The main advantage of YCoCg transform over YCbCr transform is low computational complexity since the multiplication by 0.5 and 0.25 can be simply realised by a shift operation in binary arithmetic. Both direct and inverse transforms can be implemented by only additions and shifts, no multiplications are necessary. We can also convert RGB into YCoCg colour space with four additions and two shifts by using lifting [38]:

$$\begin{aligned} C_o &= R - B; \\ t &= B + C_o \gg 1; \\ C_g &= G - t; \\ Y &= t + C_g \gg 1. \end{aligned} \quad (3.10)$$

The inverse transform can be computed with only four additions:

$$t = Y - C_g \gg 1;$$

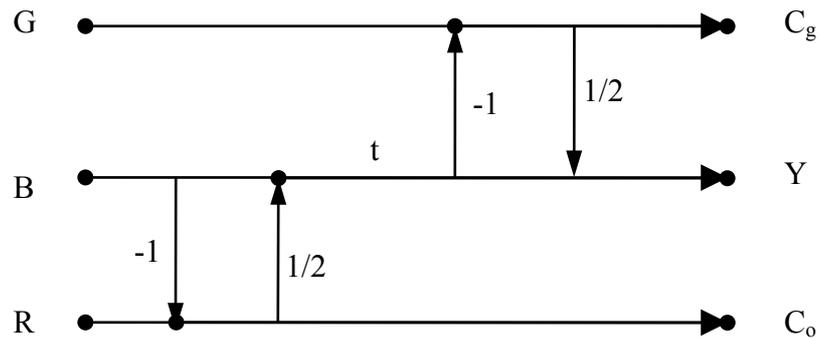
$$G = t + C_g;$$

$$B = t - C_o \gg 1;$$

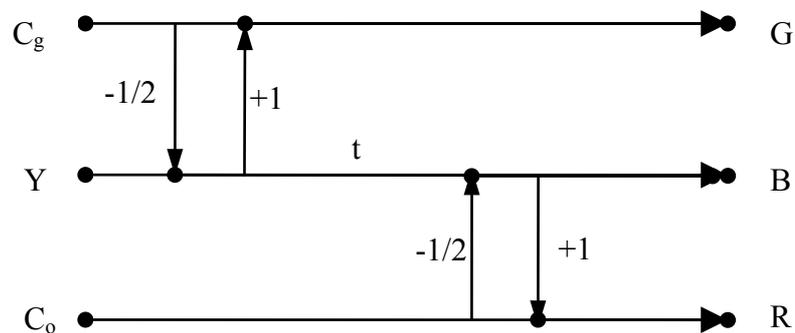
$$R = B + C_o.$$

(3.11)

The flow graphs of forward and inverse transforms are depicted in Figure 3.2 [38].



a) Forward transform



b) Reverse transform

Figure 3.2 Flow graphs of the  $RGB \leftrightarrow YCoCg$  colour conversion.

Since there are only two binary shifts involved, the transform can be implemented as a lossless transform. Moreover, the results of this specific transform can be represented by bit depths of  $C_o$  and  $C_g$  components that are only by 1 bit larger than the original RGB components. At the same time the lossless property the forward/inverse transforms are still preserved. For still image, the  $YCoCg$  transform has a better coding gain than that of the  $YCbCr$  transform [38]. Experiments in [38] computed statistics from the 24 Kodak

image set (768 × 512 pixels). Results showed that the improvement in coding gain is about 0.68 dB over YCbCr.

Although YCoCg has better coding gain for still images [38], it is still a question whether it is good to be used in video compression, especially for screen content. Experiments in Section 3.5 includes both natural content and screen content sequences.

### 3.2.2 3-point Pseudo DCT (P-DCT)

Some of the orthogonal linear reversible colour transform can also be applied for colour de-correlation, such as 3-point DCT. The transform matrix equation of 3-point DCT is defined as [36]:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ 0.7071 & 0 & -0.7071 \\ 0.4082 & -0.8165 & 0.4082 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.12)$$

We normalised the DCT transform matrix and derived a new colour conversion equation. In each row of the matrix, scale each vector is scaled down by the sum of three values. In this way, the transform matrix can be normalised to the following equation, which is named 3-point Pseudo DCT (P-DCT) in this thesis:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.13)$$

And the inverse transform equation is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2/3 \\ 1 & 0 & -4/3 \\ 1 & -1 & 2/3 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}. \quad (3.14)$$

This colour conversion preserves the same range of values for luma and chroma components as in YCbCr and YCoCg. Also, the descriptions of two chroma components are the same as in the YCoCg colour transform matrix.

In the experiments, we use the normalised 3-point P-DCT rather than the DCT to keep the normalisation rules the same as in YCbCr and YCoCg colour conversion. The rule is that the absolute sum of vectors in each row equals to 1. All colour conversion methods we mentioned in this chapter conform to this requirement. This also means that the conversion uses the same chroma offset values for all tested colour spaces, except for KLT which is introduced in Section 3.3.

### 3.3 Karhunen-Loève Transform

In image/ video compression, we aim to achieve the highest compression ratio that is possible with the lowest distortion. One of the key principles of achieving high compression is to decorrelate pixels. This can be done in the transformed domain by choosing a suitable colour space a high decorrelation of colour components a can be achieved. *Karhunen-Loève Transform* (KLT) is one of the methods which can provide maximum decorrelation [33]. The colour transform matrix used in KLT corresponds to the Eigenvector of the correlation matrix which can be obtained from the images.

The KLT for a video sequence can be derived via the following steps. First, a covariance matrix of three planes (R, G and B) is computed from the RGB values of all pixels in the sequence. Let  $\mathbf{A}$  be a matrix of three vectors of  $\mathbf{R}$ ,  $\mathbf{G}$  and  $\mathbf{B}$ , and each vector consists of  $n$  variables. Vectors of length  $n$  can be formed from the input sequences frame after frame by grouping pixels along rows and columns. Assuming each frame consists of  $N \times M$  pixels, then the value of  $n$  is  $N \times M \times f$  ( $f$  is the number of frames). Matrix  $\mathbf{A}$  is described as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{G} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \\ g_1 & g_2 & \cdots & g_n \\ b_1 & b_2 & \cdots & b_n \end{bmatrix}. \quad (3.15)$$

The aim is to find the matrix  $\mathbf{x}$  that can completely decorrelate matrix  $\mathbf{A}$ . The decorrelated matrix  $\mathbf{K}$  can be then computed as  $\mathbf{K} = \mathbf{x} \mathbf{A}$ . This implies that the covariance matrix of  $\mathbf{K}$  is diagonal. The autocorrelation matrix  $\mathbf{R}_{\mathbf{K}\mathbf{K}}$  of  $\mathbf{K}$  is derived from:

$$\begin{aligned}
 \mathbf{R}_{\mathbf{K}\mathbf{K}} &= E(\mathbf{K}\mathbf{K}^T) \\
 &= E(\mathbf{x}\mathbf{A}\mathbf{A}^T\mathbf{x}^T) \\
 &= \mathbf{x}E(\mathbf{A}\mathbf{A}^T)\mathbf{x}^T \\
 &= \mathbf{x}\mathbf{R}_{\mathbf{A}\mathbf{A}}\mathbf{x}^T.
 \end{aligned}
 \tag{3.16}$$

$\mathbf{R}_{\mathbf{A}\mathbf{A}}$  is the covariance matrix of  $\mathbf{A}$ .  $E()$  is the covariance function. Note that one could use the covariance matrix rather than correlation matrix. In general, the colour information of the given test sequence is unknown. In this case, the correlation matrix of  $\mathbf{A}$  has to be computed by converting the image into separate  $\mathbf{R}$ ,  $\mathbf{G}$  and  $\mathbf{B}$  vectors of length  $n$  and then compute the correlation matrix.

The transformed matrix  $\mathbf{K}$  need to be de-correlated, which means the covariance matrix  $\mathbf{R}_{\mathbf{K}\mathbf{K}}$  in equation (3.16) needs to be a diagonal matrix. The matrix that diagonalises  $\mathbf{R}_{\mathbf{K}\mathbf{K}}$  is shown to be the matrix whose column vectors are the Eigenvectors of  $\mathbf{R}_{\mathbf{A}\mathbf{A}}$ .

The next step is to determine the Eigenvectors and Eigenvalues from the covariance of matrix  $\mathbf{A}$ . This can be easily computed in MATLAB as [35]:

$$[\mathbf{V}, \mathbf{M}] = \text{eig}(\text{covariance}(\mathbf{A}')).
 \tag{3.17}$$

Matrix  $\mathbf{M}$  is a diagonal matrix with three Eigenvalues on the main diagonal. Matrix  $\mathbf{V}$  is the modal matrix as its columns are the Eigenvectors of  $\mathbf{A}$ . Since  $\mathbf{R}_{\mathbf{A}\mathbf{A}}$  is symmetric and positive semi definite, the three Eigenvalues and Eigenvectors of  $\mathbf{R}_{\mathbf{A}\mathbf{A}}$  are real. Matrix  $\mathbf{V}$  is then transposed so that each row implies one group of Eigenvectors. Then the matrix  $\mathbf{V}^T$  is normalised by scaling vectors so that the sum of each row is 1. Finally, the matrix of three Eigenvectors is rearrange so that the corresponding Eigenvalues are in decreasing order. After normalization and re-ordering we get the KLT matrix. The transform matrix

is unique for each sequence and depends on the features of the colour components in the images. The only shortcoming of this method is the high computational complexity. KLT can be applied to de-correlate the colour components, which is beneficial for compression but the additional computation of the transform matrix is required.

The test was applied on six video sequences (three screen content and three non-screen content) chosen from [45]. The KLT matrix for test each sequence was computed and derived by using the statistics from the first 20 frames. Assuming KLT colour space is called  $K_1K_2K_3$ . The transform is shown in the equation below and the transform matrix for each video sequence is listed in the *Table 3.1*.

$$\begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.18)$$

*Table 3.1 KLT matrix for each video sequence.*

Sequences	Transform matrix	Inverse transform matrix
Kimono	0.3374 0.3277 0.3349	1.0120 0.6328 1.0318
	0.2369 0.2651 -0.4980	0.9829 0.7080 -1.0130
	0.4930 -0.4840 -0.0231	1.0047 -1.3301 -0.0483
Park Scene	0.3563 0.3344 0.3092	1.0655 0.6121 0.9800
	0.2387 0.2336 -0.5277	0.9999 0.5990 -1.0319
	0.4839 -0.5095 -0.0066	0.9246 -1.3530 -0.0135
Ducks and Legs	0.3469 0.3395 0.3137	1.0387 0.8561 0.9138
	0.3555 0.1207 -0.5238	1.0165 0.2907 -1.2424
	0.3670 -0.4989 0.1341	0.9393 -1.2612 0.3339
SC_Map	0.3295 0.3179 0.3525	0.9868 0.9528 0.8824
	0.3732 0.1462 -0.4807	0.9520 0.3732 -1.2520
	0.3618 -0.5134 0.1248	1.0556 -1.2272 0.3043
SC_pcb_layout	0.3333 0.3406 0.3261	0.9995 -1.1906 -0.4425
	-0.4969 0.1035 0.3997	1.0214 0.2479 1.2759
	-0.1703 0.4910 -0.3387	0.9781 0.9578 -0.8803
SC_ppt_doc_xls	0.3643 0.3483 0.2874	1.0822 0.8648 0.7927
	0.4301 0.0112 -0.5588	1.0348 0.0225 -1.3107
	0.2950 -0.4877 0.2173	0.8536 -1.1236 0.5840

KLT can also be presented as the luma-chroma mode. Luma Y is approximated by the average of R, G and B. The definitions of two chroma components differ in each video sequence. From the results obtained in *Table 3.1* it can be seen that for some sequences the transform matrices are similar. In order to derive several new transform matrices for coding tests, approximate matrices are generated in *Table 3.2*.

*Table 3.2 KLT approximate matrices.*

Sequences	Transform matrix	Approximate transform matrix
Kimono	0.3374 0.3277 0.3349	$\begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/4 & 1/4 & -1/2 \\ 1/2 & -1/2 & 0 \end{matrix}$
	0.2369 0.2651 -0.4980	
	0.4930 -0.4840 -0.0231	
Park Scene	0.3563 0.3344 0.3092	(KLT-A)
	0.2387 0.2336 -0.5277	
	0.4839 -0.5095 -0.0066	
Ducks and Legs	0.3469 0.3395 0.3137	$\begin{matrix} 1/3 & 1/3 & 1/3 \\ 3/8 & 1/8 & -1/2 \\ -3/8 & 1/2 & -1/8 \end{matrix}$
	0.3555 0.1207 -0.5238	
	0.3670 -0.4989 0.1341	
SC_Map	0.3295 0.3179 0.3525	(KLT-B)
	0.3732 0.1462 -0.4807	
	0.3618 -0.5134 0.1248	
SC_pcb_layout	0.3333 0.3406 0.3261	No approximation
	-0.4969 0.1035 0.3997	
	-0.1703 0.4910 -0.3387	
SC_ppt_doc_xls	0.3643 0.3483 0.2874	$\begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{matrix}$
	0.4301 0.0112 -0.5588	
	0.2950 -0.4877 0.2173	
		(KLT-C)

Approximate matrices and corresponding inverse transform matrices are derived in the following equations:

- KLT-A

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/4 & 1/4 & -1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.19)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 2/3 & 1 \\ 1 & 2/3 & -1 \\ 1 & -3/4 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} \quad (3.20)$$

- KLT-B

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.21)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 8/9 & 8/9 \\ 1 & 16/45 & -56/45 \\ 1 & -56/45 & 16/45 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} \quad (3.22)$$

- KLT-C

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 3/8 & 1/8 & -1/2 \\ -3/8 & 1/2 & -1/8 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.23)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2/3 \\ 1 & 0 & -4/3 \\ 1 & -1 & 2/3 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix}. \quad (3.24)$$

In all three KLT approximations, luma component Y is derived by averaging R, G and B components, which is the same as that in P-DCT. Chroma components are derived using low complexity integer multiplications and divisions by  $2^n$ . In the computational matrix of KLT-A, one of the chroma components  $C_2$  is proportional to R-G, which is the difference of red and blue components. The other one is proportional to  $(R + G)/2 - B$ , which is the difference between B and the average of R and G. The approximate KLT-B transform matrix in equation (3.21) was also verified experimentally by other authors on still images [36] [46]. The transform matrix of KLT-C is the same as that in 3-point P-DCT.

KLT and approximations KLT-A (or B or C) are all included in the experiments in this Chapter. These transform matrices are pre-computed and saved in the configuration file for each sequence.

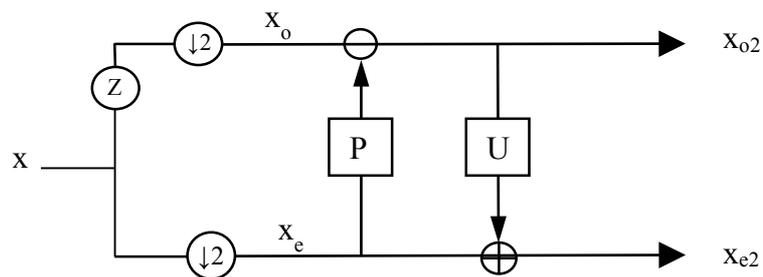
### 3.4 Lifting-based design of new colour transforms

The lifting scheme was designed for wavelet transform by Sweldens [47]. The wavelet transform function involves Fourier Transform which limits the utility of multidimensional bounded domains. The lifting scheme avoids Fourier analysis and enables infinite number of wavelets from the initial one. The lifting scheme is also used in other applications, such as butterfly structure for Fast Fourier transform [48], image compression in JPEG200 [49], YCoCg colour transform [38] and pattern matching [50].

This section reviews the lifting scheme and introduces a lifting-based design of new colour transforms.

#### 3.4.1 Two channel lifting scheme

The lift scheme of wavelet transform is illustrated in *Figure 3.3*.



*Figure 3.3 Lifting scheme of the wavelet transform.*

There are three operations defined in lifting scheme:

- Step 1: split

The signal  $x$  has to be split into two disjoint components:  $x_e$  and  $x_o$  which are the even and odd components, respectively. The two components are represented as following equations:

$$\begin{aligned}x_e(n) &= x(2n), \\x_o(n) &= x(2n + 1).\end{aligned}\tag{3.25}$$

- Step 2: predict

The odd component is based on a linear combination of the the even component. The odd component  $x_{o2}$  is then generated by using the prediction operator,  $P$ , and the difference of two components, shown in the equation as:

$$\begin{aligned}x_{o2}(n) &= x_o(n) - P(x_e(n)), \\&= x(2n + 1) - P(x(2n)).\end{aligned}\tag{3.26}$$

- Step 3: update

The even component is generated by using the updating operator,  $U$ , on odd component and adding the results to  $x_e$ . The updated even component is represented as:

$$x_{e2}(n) = x_e(n) + U(x_{o2}(n)).\tag{3.27}$$

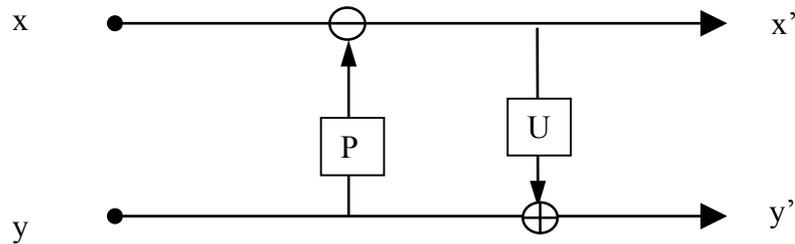
By substituting the equation (3.26) in equation (3.27), it yields as:

$$\begin{aligned}x_{e2}(n) &= x_e(n) + U(x_o(n) - P(x_e(n))), \\&= x_e(n) + U(x_o(n)) - UP(x_e(n)).\end{aligned}\tag{3.28}$$

Similarly, for a test image  $x$ , it is split into two disjoint subsets which represent detail component and coarser component, respectively [50]. The detailed component is then

replaced by the difference of two components along with the prediction operation  $P$ . The coarser is then generated by applying the updating operation  $U$ . The lifting-based parameters in  $P$  and  $U$  operations can be designed to achieve the optimal results.

The wavelet transform lifting scheme can be developed into a two channel lifting scheme, as shown in *Figure 3.4*. Two input signals are in  $x$  and  $y$  channels which represent detail and coarser component, respectively.



*Figure 3.4* Flow graph of 2-channel lifting scheme.

The two components after prediction and updating are derived as:

$$x'(n) = x(n) - P(y(n)), \quad (3.29)$$

$$y'(n) = y(n) + U(x(n)) - UP(y(n)). \quad (3.30)$$

### 3.4.2 Three channel lifting scheme

The two channel lifting scheme can be expanded into three channels by adding a third channel  $z$ . In this case, the lifting scheme includes a detail component  $y$  and two coarser components  $x$  and  $z$ , as shown in *Figure 3.5*.

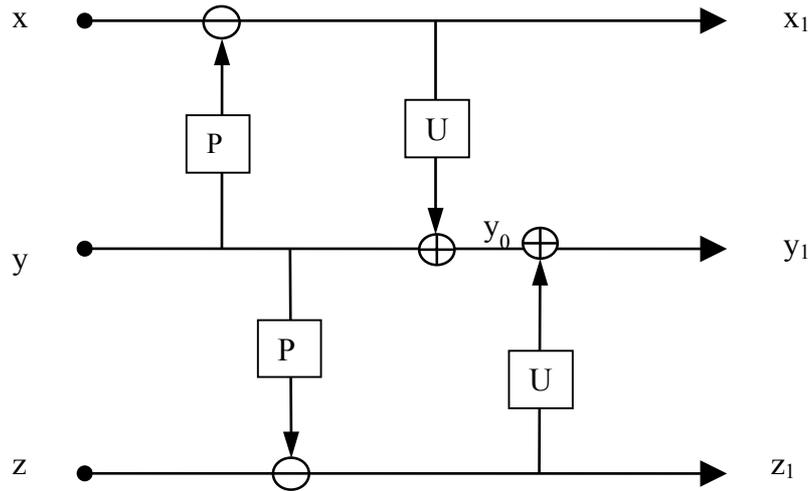


Figure 3.5 Flow graph of 3-channel lifting scheme.

- Predict

The two coarser components are generated by using the prediction operation:

$$x_1 = x - P(y), \tag{3.31}$$

$$z_1 = z - P(y). \tag{3.32}$$

- Update

The detail channel y is generated after two levels of updating, as illustrated in Figure 3.5. The derivation equations are described as:

$$y_0 = y + U(x_1), \tag{3.33}$$

$$y_1 = y_0 + U(z_1). \tag{3.34}$$

By substituting the equation equation (3.33) in equation (3.34), it yields:

$$\begin{aligned} y_1 &= y + U(x_1) + U(z_1). \\ &= y + U(x_1) + U(z_1). \end{aligned}$$

(3.35)

By substituting the equation (3.31) and (3.32) in equation (3.35), the final result of  $y$  channel,  $y_1$ , is derived as:

$$\begin{aligned} y_1 &= y + U(x - P(y)) + U(z - P(y)). \\ &= y + U(x) + U(z) - 2UP(y). \end{aligned}$$

(3.36)

This three channel lifting scheme can be applied in colour transforms where  $x$ ,  $y$  and  $z$  are replaced by R, G and B, respectively.

### 3.4.3 Proposed three channel lifting-based colour transforms

Based on the three channel lifting scheme introduced in Section 3.4.2, a lifting-based colour transform, named 3-channel PPU is proposed in this thesis. P stands for prediction operator and U stands for updating operator. In the proposed method, two chroma components are firstly predicted by using colour differences: one is proportional to  $(B-G)$ , and the other is proportional to  $(R-G)$ . The simply use of colour differences is inspired from the transform matrices of YCoCg and 3-point P-DCT. Each of them has one chroma component represented as  $(R-G)$ .

Luma Y is computed as a updating result from the two chroma components and green colour channel, shown in the equation:  $Y = G + \alpha C_{p1} + \beta C_{p2}$ , where  $\alpha$  and  $\beta$  are updating parameters in lifting scheme. The sum of two parameters is ranging from 0 to 1. The values of these two factors indicate the weights of the two chroma components in luma. The forward transform is given by the equations below:

$$\begin{aligned} C_{p1} &= R - G; \\ C_{p2} &= B - G; \\ Y &= G + \alpha C_{p1} + \beta C_{p2} \\ &= \alpha R + (1 - \alpha - \beta)G + \beta B. \end{aligned}$$

(3.37)

Note that  $(0 < \alpha + \beta < 1)$ . The corresponding inverse transform is described by the following equations:

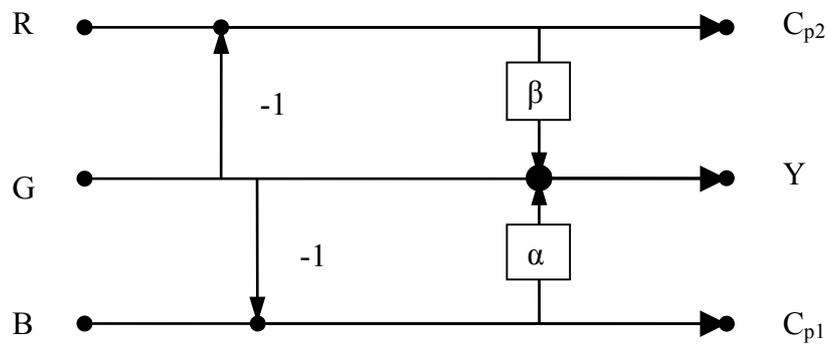
$$G = Y - \alpha C_{P1} - \beta C_{P2};$$

$$R = C_{P1} + G;$$

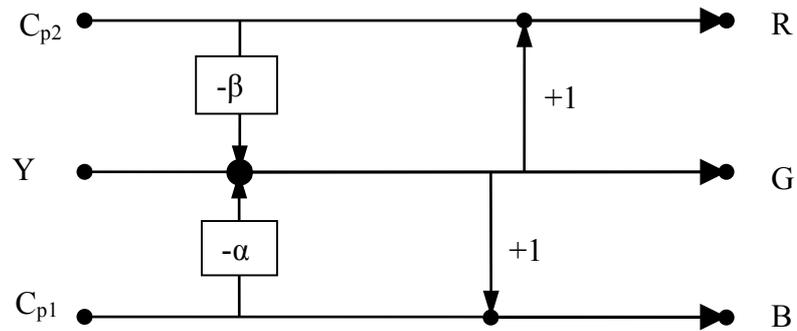
$$B = C_{P2} + G.$$

(3.38)

The flow graphs of forward and inverse transforms are depicted in Figure 3.6.



a) Forward transform



b) Reverse transform

Figure 3.6 Flow graph of the RGB ↔ 3-channel PPU lifting colour conversion.

The luma component Y is computed by using the scaling factors  $\alpha$  and  $\beta$  as well as the two chroma components  $C_{p1}$  and  $C_{p2}$ . The transform matrix is given by the equation below:

$$\begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix} = \begin{bmatrix} \alpha & (1 - \alpha - \beta) & \beta \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.39)$$

Different values of  $\alpha$  and  $\beta$  result in varied equations for computing Y. For example, with  $\alpha = \beta = 1/3$ , Y is computed as simply the average of R, G and B pixel values. Two updating parameters  $\alpha$  and  $\beta$  are set to generate series of PPU transform matrices. The structure is symmetrically designed in this chapter, which means the two scale factors are the same. Experiments with asymmetric designs of PPU extension are suggested to be conducted in the future. In the experiments we used four pairs of parameters setting, as shown in *Table 3.3*. The values of parameters  $\alpha$  and  $\beta$  are inspired from the other previously used colour transform matrices, such as PPU-1 and 2. PPU-3 is a special case designed from signal processing aspect. PPU-4 is developed from PPU-3.

*Table 3.3 Parameter setting of four PPU transforms.*

PPU type	$\alpha$	$\beta$	Transform matrix		
PPU-1	1/3	1/3	1/3	1/3	1/3
			0	1/2	-1/2
			1/2	-1/2	0
PPU-2	1/4	1/4	1/4	1/2	1/4
			0	-1/2	1/2
			1/2	-1/2	0
PPU-3	0	0	0	1	0
			0	-1/2	1/2
			1/2	-1/2	0
PPU-4	3/8	3/8	3/8	1/4	3/8
			1/2	-1/2	0
			0	-1/2	1/2

### 3.4.3.1 PPU-1

When  $\alpha = \beta = 1/3$ , Y is computed simply as the average of R, G and B pixel values, which is the same as in 3-point P-DCT and KLT. The transform matrix is given by the equations below:

$$\begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.40)$$

And the inverse matrix is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -2/3 & 4/3 \\ 1 & -2/3 & -2/3 \\ 1 & 4/3 & -2/3 \end{bmatrix} \begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix}. \quad (3.41)$$

The forward transform matrix is similar to the approximate KLT matrices. The only difference is that the description of one chroma channel is  $(B - G)$ , the difference between B and G, rather than  $G - (R + B)/2$ , the difference between G and the average of R and B.

### 3.4.3.2 PPU-2

When  $\alpha = \beta = 1/4$  (PPU-2), the equation of computing Y is the same as that in YCoCg. The transform matrix is given by the equations below:

$$\begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.42)$$

And the inverse matrix is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & 3/2 \\ 1 & -1/2 & -1/2 \\ 1 & 3/2 & -1/2 \end{bmatrix} \begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix}. \quad (3.43)$$

### 3.4.3.3 PPU-3

No-update scenario (PPU-3) is also considered as Y simply equals to G when  $\alpha = \beta = 0$ . This is a special case designed inspired from signal processing. The transform matrix is given by the equations below:

$$\begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.44)$$

And the inverse matrix is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix}. \quad (3.45)$$

### 3.4.3.4 PPU-4

PPU-4 is an extension version of PPU-1 to avoid using 1/3 (3/8 can be implemented by shifting). It uses larger scale factors which results in less weight of G component in computation of Y. It is also The transform matrix is given by the equations below:

$$\begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix} = \begin{bmatrix} 3/8 & 1/4 & 3/8 \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/2 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (3.46)$$

And the inverse matrix is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -3/4 & 5/4 \\ 1 & -3/4 & -3/4 \\ 1 & 5/4 & -3/4 \end{bmatrix} \begin{bmatrix} Y \\ C_{p1} \\ C_{p2} \end{bmatrix}. \quad (3.47)$$

We can also convert RGB into PPU by using lifting. For example, the PPU-4 lifting is represented as:

$$C_{p1} = R - G;$$

$$\begin{aligned}
C_{P2} &= B - G; \\
t &= (C_{P1} + C_{P2}) * 3 \gg 3; \\
Y &= G + t.
\end{aligned}
\tag{3.48}$$

The PPU-4 lifting, along with introduced PPU-1~ 4 are included in the experiments in this chapter.

### 3.5 Performance evaluation

This section introduces test configurations and performance evaluation metrics. The JCT-VC common test conditions for HEVC Range Extensions [51] define a set of configurations and test sequences used in experiments during the development of support for several sampling formats as well as increased bit depths. The common test conditions are desirable to conduct experiments in a well-defined environment and ease the comparison of the outcome of experiments. It defines three test conditions, three encoding modes and two QP ranges that reflect a variety of interesting application areas.

In objective evaluations, the *Bjontegaard Delta rate* (BD-rate) is used to calculate the bit rate reduction and the *Bjontegaard Delta PSNR* (BD-PSNR) is employed to measure the distortion. The performances of each colour component as well as average RGB are evaluated using both methods.

#### 3.5.1 Test sequences

JCT-VC group provided a list of sequences used for HEVC range extension. The dataset in the experiments in this chapter is composed of six sequences chosen from the list in [45], three of natural content and three of screen content, with different characteristics, resolutions and frame rates. The configurations of six test sequences are as given in *Table 3.4*. The first frame of each sequence is shown in *Figure 3.7*.

Table 3.4 Test sequences.

Sequences type	Sequences name	Resolution	Frames rate (Hz)	Bit depth
Natural content (NC)	Kimono	1920 × 1080	24	10
	ParkScene	1920 × 1080	24	10
	Ducks&Legs	1920 × 1080	30	10
Screen content (SC)	Sc_map	1280 × 720	60	8
	Sc_pcb_layout	1920 × 1080	20	8
	Sc_ppt_doc_xls	1920 × 1080	20	8



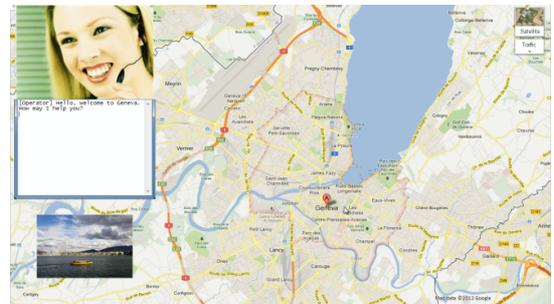
Ducks and legs



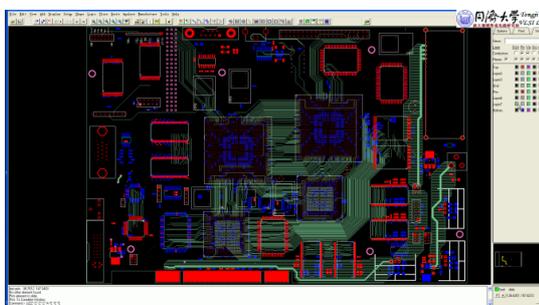
Kimono



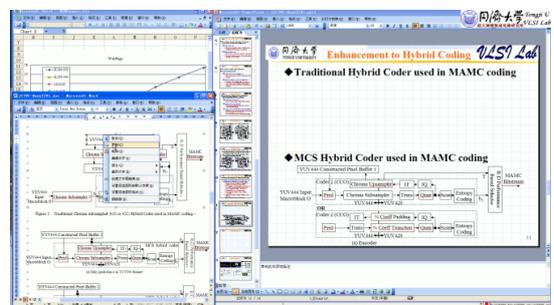
Park scene



SC\_map



SC\_pcb layout



SC\_ppt\_doc\_xls

Figure 3.7 First frames of each test sequence.

Natural content test sequences are camera-captured natural videos with continuous movement. The rest three are screen content sequences which are screen-prints of computer desktops. The experiments evaluate the performance of different colour conversion used in HEVC, and also compares the performances between natural content and screen content.

### 3.5.2 Software reference configurations

The software used in the experiments is HEVC range extension test model version 10.0 (HM10.0\_RExt2.0). There are three test configurations introduced in common test conditions [51] : All Intra (AI), Random Access (RA) and Low Delay (LD). The test condition used in our experiments is AI, where each picture in a video sequence is encoded as intra frame. No temporal reference picture is used. The parameters are set the same as the default in AI-main encoder configuration file except the *InternalBitDepth*, which is the codec operating bit depth, set as 14. This is because the colour conversion process scales all pixels into 14 bits to reduce the colour transform distortion. In that way, the corresponding video compression should also be in 14 bits. For each video sequence four quantisation parameter values are to be used depending upon the configuration. These values define the QP values used for the I frames in a sequence, and QP values for other frames are further defined in encoder configuration files. Since the experiments are in AI test condition, all frames are quantised under the same step size. In our experiments the quantisation parameter values are set as 22, 27, 32 and 37. Another aspect we consider is chroma formats. HEVC range extension supports three modes: RGB 4:4:4, YCbCr 4:2:2 and YCbCr 4:2:2. Experiments investigating YCbCr coding tools should also support other colour spaces. Note that all tests in this chapter are based on 4:4:4 chroma format.

The configuration file of test sequence gives information of sequence size and other features. An example is shown in Figure 3.8. Input file is the pre colour transformed 14 bit 4:4:4 sequence. Only first 20 frames are coded in the experiments.

```

#===== File I/O =====
InputFile           : kimono_ycbcr.yuv
InputBitDepth       : 14           # Input bitdepth
InputChromaFormat   : 444         # Ratio of luminance to chrominance samples
FrameRate           : 24          # Frame Rate per second
FrameSkip           : 0           # Number of frames to be skipped in input
SourceWidth         : 1920        # Input frame width
SourceHeight        : 1080       # Input frame height
FramesToBeEncoded   : 20         # Number of frames to be coded

```

Figure 3.8 An example of test sequence configuration (Kimono).

### 3.5.3 Evaluation metrics

For evaluation purpose, coding performance for YCbCr 4:4:4 sequences and original colour format of RGB 4:4:4 are evaluated. This section includes implementation of the above mentioned colour transforms. The transforms are referred to YCbCr, YCoCg, 3-point P-DCT, KLT and 3-channel PPU. The objective video distortion metrics, PSNR, was used to evaluate the various transforms:

$$\text{PSNR}_{\text{Total}} = 10 \log_{10} \frac{\text{MAX}^2}{(\text{MSE}(\text{R}) + \text{MSE}(\text{G}) + \text{MSE}(\text{B}))/3} \text{ dB}, \quad (3.49)$$

where MAX is the maximum pixel value. For 14-bit pixel, this is 16383 ( $2^{14}-1$ ).

The aim of the objective tests was to compare the colour transform in coding performance. For each sequence, the experiments results are four pairs of bitrate and PSNR, which can be shown in the RD-curve as four points. From the RD-curve, the bit rate reduction can be computed using BD-rate and the distortion can be measured by using BD-PSNR [52]. The function add-in excels tables in VCEG-AE07 [53] compute the BD-rate and BD-PSNR automatically. During years the interpolation used to compute BD-rate has changed. The JCT-VC document [51] provides the latest version, in which BD-rate is computed using piece-wise cubic interpolation.

### 3.6 Experiments and results

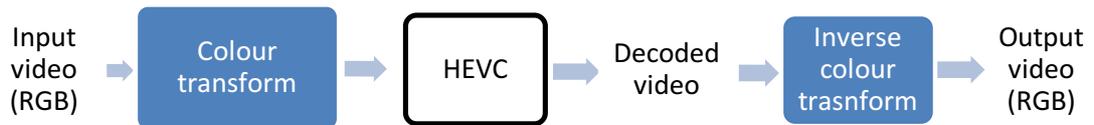
There are two experiments in this section:

1. Colour transform error analysis

It aims to compute the distortion from the colour transforms by simply conducting the colour transform and inverse transform.

2. Colour transform for HEVC.

The block diagram of the experiment is shown in *Figure 3.9*. The original RGB 4:4:4 sequences were converted to a luma-chroma mode colour space which represented as YCxCz in this chapter. YCxCz include YCbCr, YCoCg, 3-point DCT, KLT and 3-channel PPU. After HEVC codec, the decoded sequences were converted back to RGB colour space by using the corresponding inverse transform.



*Figure 3.9* Block diagram of experiments.

The objective evaluation metric BD-rate is used to compute the average bit rate saving. We also compared the performance of screen content and non-screen content.

#### 3.6.1 Colour transform error analysis

For colour transform used in video compression, colour transform itself need to be lossless or near lossless. The experiments in this section aims to compare the transform errors. The experiments include a forward transform from RGB colour space into YCxCz colour and a corresponding inverse transform. The results for each sequence are shown in the *Table 3.5- Table 3.10*. Total PSNR is computed from equation 3.37.

Table 3.5 Results of colour transform comparison (kimono).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	54.70	61.92	54.10	55.78
YCoCg	66.24	61.46	66.22	64.01
KLT	54.59	60.27	64.17	57.96
3-channel PPU	59.63	64.17	58.20	60.02
3-point P-DCT	56.57	64.58	59.98	59.26

Table 3.6 Results of colour transform comparison (Park Scene).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	54.72	61.93	54.16	55.83
YCoCg	66.32	61.52	66.18	64.05
KLT	54.66	60.61	64.29	58.09
3-channel PPU	59.86	63.69	58.21	60.04
3-point P-DCT	56.82	64.29	60.29	59.47

Table 3.7 Results of colour transform comparison (Ducks&Legs).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	54.69	61.92	54.04	55.75
YCoCg	66.22	61.45	66.22	64
KLT	54.25	62.62	62.88	57.94
3-channel PPU	59.89	63.77	58.15	60.03
3-point P-DCT	56.76	64.35	60.23	59.42

Table 3.8 Results of colour transform comparison (SC\_map).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	44.16	50.26	44.15	45.41
YCoCg	57.18	50.26	54.09	52.94
KLT	42.52	52.66	50.76	46.33
3-channel PPU	47.07	51.13	45.95	47.55
3-point P-DCT	45.16	51.08	48.41	47.55

Table 3.9 Results of colour transform comparison (SC\_pcb\_layout).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	46.85	55.86	47.62	48.69
YCoCg	67.78	55.71	59.95	58.90
KLT	57.35	47.63	52.58	50.86
3-channel PPU	53.03	57.38	54.25	54.53
3-point P-DCT	52.58	57.72	54.39	54.55

Table 3.10 Results of colour transform comparison (SC\_ppt\_doc\_xls).

	R PSNR (dB)	G PSNR (dB)	B PSNR (dB)	Total PSNR (dB)
YCbCr	42.75	48.88	42.47	43.88
YCoCg	60.94	48.72	60.54	52.98
KLT	45.62	54.52	48.73	48.31
3-channel PPU	47.84	55.20	48.24	49.40
3-point P-DCT	47.64	57.93	48.57	49.62

The results of five colour conversions of both natural content and screen content sequences are shown in *Table 3.5* to *Table 3.10*. (*Table 3.5- Table 3.7* for natural content; *Table 3.8 - Table 3.10* for screen content). The colour conversion distortion comes from the computation when rounding decimals into integers. For each test sequence, YCoCg colour conversion achieves the highest PSNR which means the YCoCg colour transform has the smallest transform error. The colour transform from RGB to YCoCg only generates errors in forward transform by bit shifting. The inverse transform of YCoCg is lossless since it only involves adding. Other colour conversions generated rounding errors from both direct and inverse transforms.

To compare the efficiency of colour transform in video coding, it is necessary to minimize the colour transform error. One of the ways is to increase the bit depth in transform to increase the precision. Our experiments show that all colour transform methods have infinite PSNR when the colour transform is processed in 14 bits. This is because the value of the MAX in PSNR computational equation becomes much larger ( $2^{14}-1=16383$ ). The colour transform distortion is approximately unnoticeable in high bit depth.

### **3.6.2 Colour transform for HEVC**

The experiments evaluate the performance of different colour transforms which applied for HEVC. The test sequences and test conditions are introduced in Section 3.5. The total bitrate and PSNR give results of overall performance, the BD-rate and BD-PSNR of three colour components and total RGB are computed of each sequence.

#### **3.6.2.1 The results of each colour transform**

The results of each colour conversion method compared with YCbCr are shown in *Table 3.11* to *Table 3.18*. The comparisons between different colour transforms are summarised in Section 3.6.2.2.

Table 3.11 BD-rate of RGB components for HEVC with YCoCg compared to HEVC with YCbCr.

YCoCg	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	-6.0%	30.3%	-8.5%	-2.74%
	NC2-ParkScene	-8.4%	30.1%	-20.0%	-5.97%
	NC3-DucksLegs	-2.4%	33.6%	-4.8%	1.76%
Screen Content	SC1-map	-1.5%	16.5%	-8.8%	-0.46%
	SC2-pcb_layout	5.7%	13.5%	-0.8%	4.16%
	SC3-ppt_doc_xls	-4.5%	14.0%	-12.9%	-4.84%

Table 3.12 BD-PSNR of RGB components for HEVC with YCoCg compared to HEVC with YCbCr.

YCoCg	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	0.12	-0.64	0.16	0.05
	NC2-ParkScene	0.25	-0.81	0.49	0.15
	NC3-DucksLegs	0.03	-1.09	0.22	-0.07
Screen Content	SC1-map	0.11	-1.21	0.67	0.03
	SC2-pcb_layout	-1.30	-3.19	0.30	-0.91
	SC3-ppt_doc_xls	0.57	-2.07	2.08	0.68

Table 3.13 BD-rate of RGB components for HEVC with KLT compared to HEVC with YCbCr.

KLT	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	-17.8%	44.4%	-14.6%	-7.03%
	NC2-ParkScene	-18.8%	40.6%	-26.3%	-10.81%
	NC3-DucksLegs	-6.1%	64.9%	-15.8%	-1.43%
Screen Content	SC1-map	-6.4%	25.5%	-16.5%	-3.23%
	SC2-pcb_layout	1.8%	17.9%	-5.8%	2.06%
	SC3-ppt_doc_xls	-0.3%	24.2%	-7.8%	1.02%

Table 3.14 BD-PSNR of RGB components for HEVC with KLT compared to HEVC with YCbCr.

KLT	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	0.39	-0.89	0.29	0.13
	NC2-ParkScene	0.59	-1.06	0.70	0.29
	NC3-DucksLegs	0.19	-1.72	0.68	0.07
Screen Content	SC1-map	0.47	-1.69	1.30	0.24
	SC2-pcb_layout	-0.40	-4.33	1.60	-0.42
	SC3-ppt_doc_xls	-0.09	-3.23	1.27	-0.17

Table 3.15 BD-rate of RGB components for HEVC with 3-channel PPU-1 compared to HEVC with YCbCr.

PPU-1	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	-5.1%	37.2%	-10.3%	-2.17%
	NC2-ParkScene	-6.1%	31.0%	-22.3%	-6.32%
	NC3-DucksLegs	7.0%	37.5%	-10.6%	1.81%
Screen Content	SC1-map	-3.5%	10.6%	-14.7%	-5.19%
	SC2-pcb_layout	-7.7%	1.5%	-13.4%	-8.15%
	SC3-ppt_doc_xls	-1.4%	5.7%	-11.3%	-4.53%

Table 3.16 BD-PSNR of RGB components for HEVC with 3-channel PPU compared to HEVC with YCbCr.

PPU	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	0.06	-0.68	0.21	0.04
	NC2-ParkScene	0.17	-0.84	0.57	0.16
	NC3-DucksLegs	-0.21	-1.18	0.43	-0.05
Screen Content	SC1-map	0.25	-0.78	1.13	0.38
	SC2-pcb_layout	1.92	-0.63	3.91	2.34
	SC3-ppt_doc_xls	0.12	-0.87	1.79	0.69

Table 3.17 BD-rate of RGB components for HEVC with 3-point P-DCT compared to HEVC with YCbCr.

3-point P-DCT	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	-15.1%	57.6%	-12.5%	-4.12%
	NC2-ParkScene	-17.5%	56.7%	-27.5%	-8.01%
	NC3-DucksLegs	-13.2%	78.0%	-10.5%	1.58%
Screen Content	SC1-map	-5.7%	26.7%	-14.4%	-1.61%
	SC2-pcb_layout	0.6%	19.7%	-5.5%	2.03%
	SC3-ppt_doc_xls	-7.0%	24.2%	-17.1%	-5.17%

Table 3.18 BD-PSNR of RGB components for HEVC with 3-point P-DCT compared to HEVC with YCbCr.

3-point P-DCT	Sequences	R	G	B	Total RGB
Natural Content	NC1-Kimono	0.34	-0.99	0.24	0.08
	NC2-ParkScene	0.54	-1.37	0.70	0.21
	NC3-DucksLegs	0.46	-1.94	0.45	-0.03
Screen Content	SC1-map	0.42	-1.82	1.13	0.12
	SC2-pcb_layout	-0.16	-4.41	1.59	-0.39
	SC3-ppt_doc_xls	1.02	-3.20	2.87	0.75

R-D curves for each test sequence are shown in *Figure 3.12* to *Figure 3.15*.

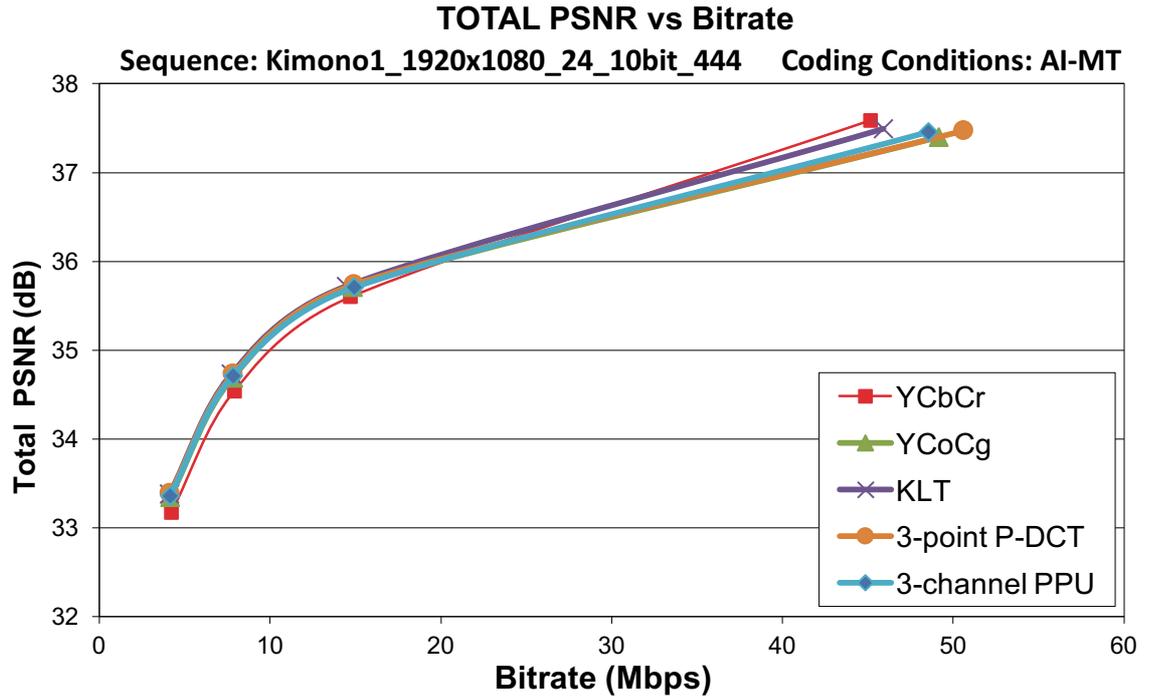


Figure 3.10 R-D performance for “Kimono”.

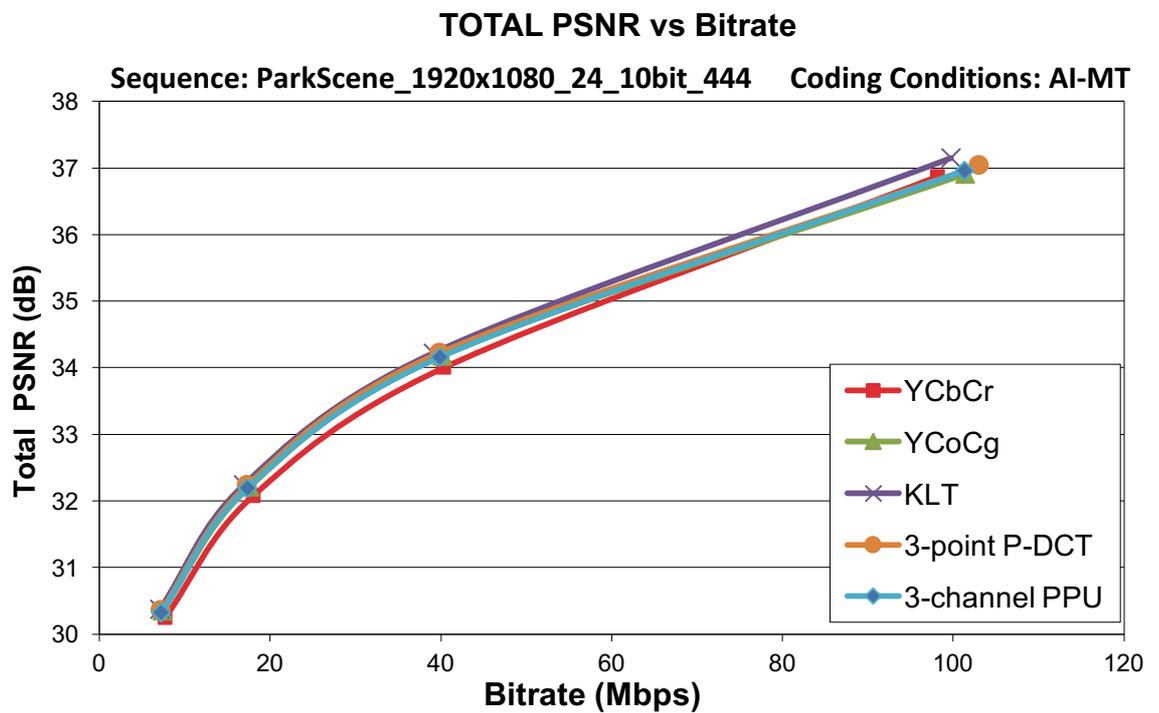


Figure 3.11 R-D performance for “Park Scene”.

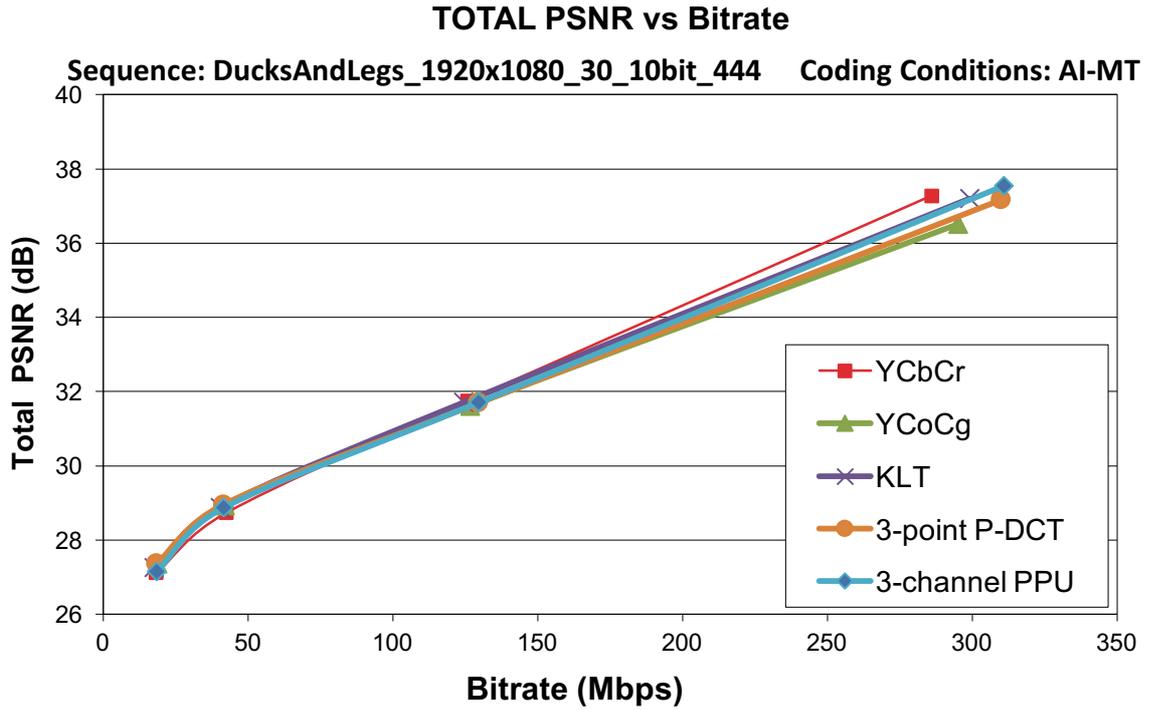


Figure 3.12 R-D performance for “Ducks&Legs”.

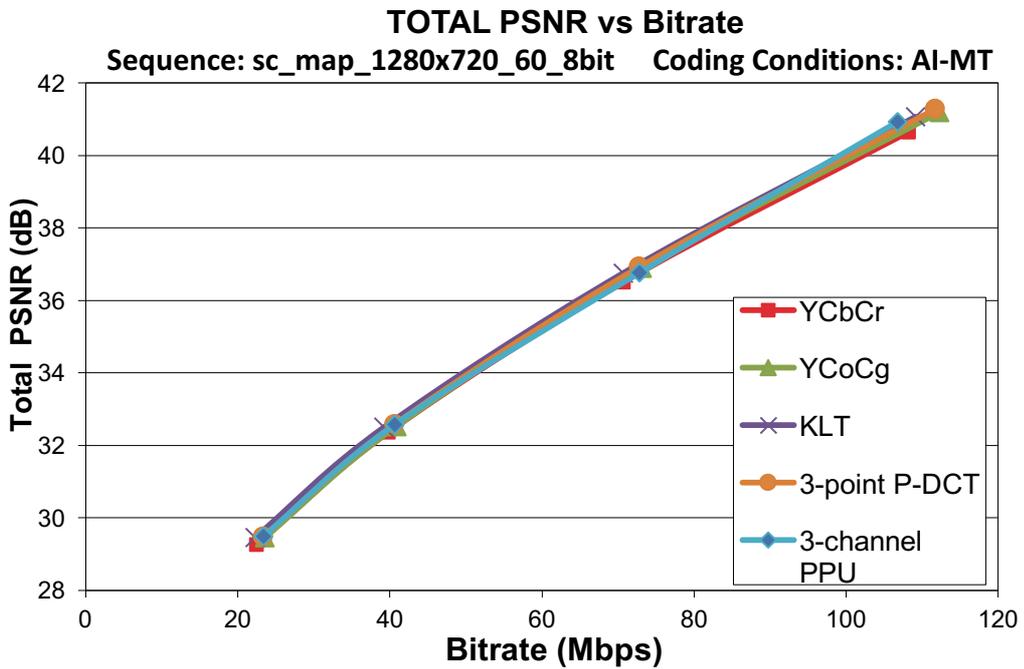


Figure 3.13 R-D performance for “SC\_map”.

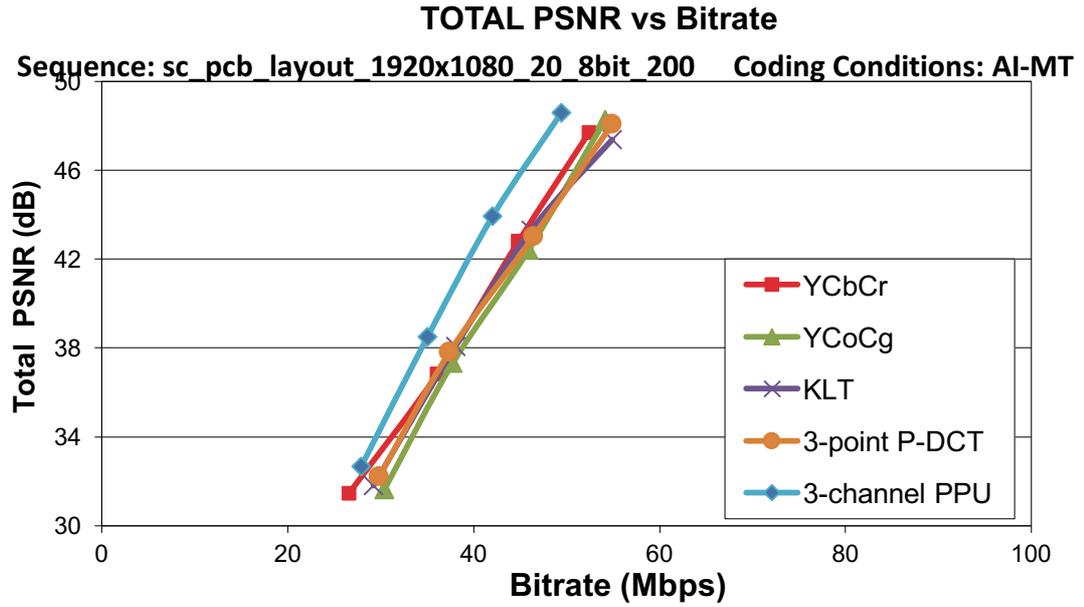


Figure 3.14 R-D performance for “SC\_pab layout”.

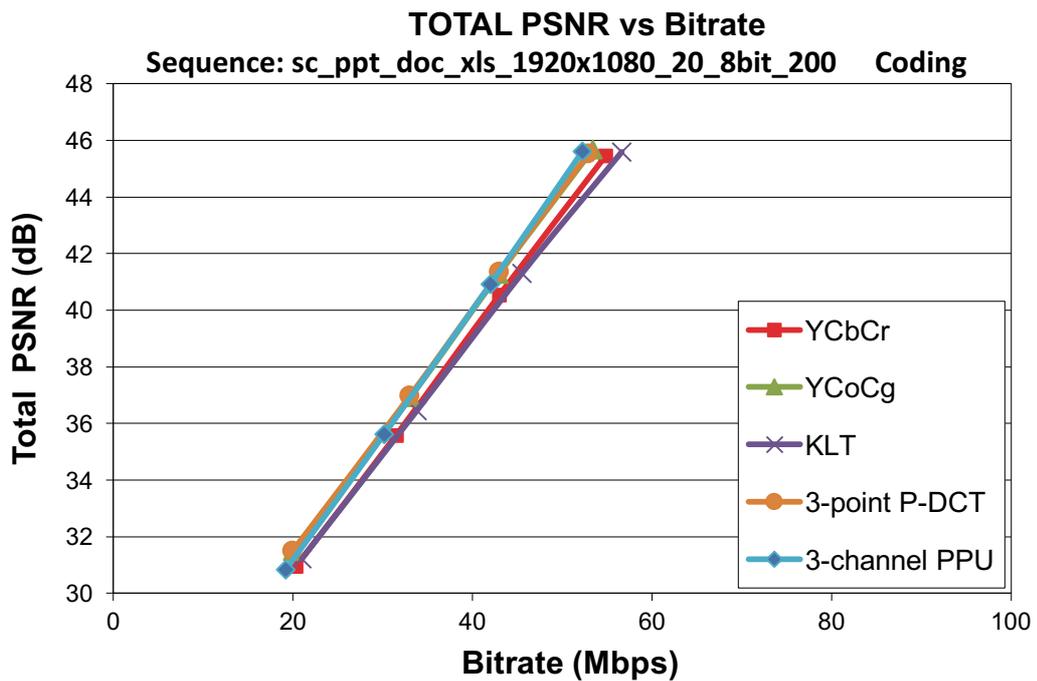


Figure 3.15 R-D performance for “SC\_ppt\_doc\_xls”.

The results show R-D curves of five colour conversions for each sequence, including both natural content and screen content. The curves in the figures are very close although slight difference indicates coding performance improvement.

### 3.6.2.2 The results comparisons

The results of comparisons are summarised in *Table 3.19*. The related graph comparisons are shown in *Figure 3.16*.

*Table 3.19* compares the RD performance by using BD-rate to compute the average bitrate saving between two rate-distortion curves. Set the RD performance of YCbCr as the reference as it is the most commonly used in video compression. Negative value means reduced bitrate percentage. The distortion is measured by using BD-PSNR, shown in *Table 3.20*. YCbCr colour space is set as the reference. The values in table with the unit dB indicate the average PSNR increases compared with YCbCr.

*Table 3.19 Comparisons of BD-rate of different colour transforms in HEVC (YCbCr is the reference).*

	Test Sequences	YCoCg	KLT	KLT-A	KLT-B	P-DCT (KLT-C)	PPU-1
Natural Content	NC1-Kimono	-2.74%	-7.03%	-6.93%	-6.74%	-4.12%	-2.17%
	NC2-ParkScene	-5.97%	-10.81%	-10.13%	-8.97%	-8.01%	-6.32%
	NC3-DucksLegs	1.76%	-1.43%	-1.59%	-1.1%	1.58%	1.81%
	Average	-2.32%	-6.42%	-6.22%	-5.60%	-3.52%	-2.23%
Screen Content	SC1-map	-0.46%	-3.23%	-6.47%	-4.43%	-1.61%	-5.19%
	SC2-pcb_layout	4.16%	2.06%	-6.76%	-3.7%	2.03%	-8.15%
	SC3-ppt_doc_xls	-4.84%	1.02%	-2.89%	-5.64%	-5.17%	-4.53%
	Average	-0.38%	-0.05%	-5.37%	-4.59%	-1.58%	-5.96%
Average		-1.35%	-3.24%	-5.80%	-5.10%	-2.55%	-4.09%

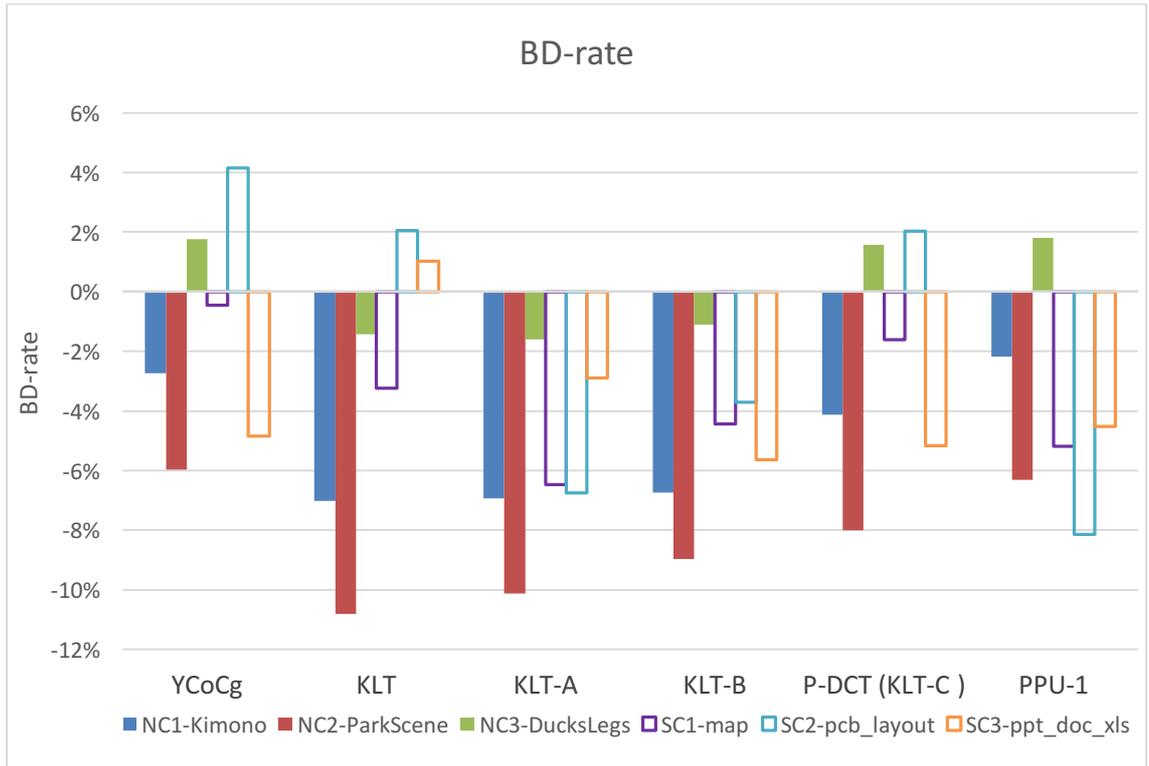


Figure 3.16 BD-rate comparisons of colour transforms.

Table 3.20 Comparisons of BD-PSNR(dB) of different colour transforms in HEVC (YCbCr is the reference).

	Test Sequences	YCoCg	KLT	KLT-A	KLT-B	P-DCT (KLT-C)	PPU-1
Natural Content	NC1-Kimono	0.05	0.13	0.13	0.12	0.08	0.08
	NC2-ParkScene	0.15	0.29	0.27	0.24	0.21	0.21
	NC3-DucksLegs	-0.07	0.07	0.08	0.06	-0.03	-0.03
Screen Content	SC1-map	0.03	0.24	0.49	0.33	0.12	0.12
	SC2-pcb_layout	-0.91	-0.42	1.83	0.99	-0.39	-0.39
	SC3-ppt_doc_xls	0.68	-0.17	0.37	0.83	0.75	0.75

KLT is the most efficient method for natural content but not for screen content. This is because the KLT matrix is computed from R, G and B components of 20 frames. *Table 3.21* shows the KLT matrix MSE by computing the difference of KLT matrix of 20 frames and KLT matrix of each frame. In screen content, the frame difference activities in screen content are more likely to have dramatically change (as discussed in Section 2.3.2). Screen content gained more bitrate saving from 3-channel PPU and 3-point P-DCT. A common feature of KLT, 3-channel PPU and 3-point P-DCT is that the Luma Y is the average of R, G and B components. Chroma matrix in KLT differs from each sequence depending on the content. 3-channel PPU and 3-point P-DCT have one chroma component represented the same.

*Table 3.21 KLT matrix MSE*

Sequence	NC1	NC2	NC3	SC1	SC2	SC3
Matrix MSE ( $10^{-7}$ )	1.4	7.9	1.23	30.6	21.9	14.2

For natural content, KLT achieves highest average PSNR difference in dB over the whole range of bitrates. For screen content, 3-channel PPU-1 and 3-point P-DCT gain higher average PSNR. Both BD-rate and BD-PSNR evaluation methods give the same comparison result. Based on the results that natural content and screen content present different trends on colour transform, it is necessary to implement an improved colour transforms for screen content. The experimental results of different colour transforms for screen content are shown in Section 3.6.3.

### 3.6.3 Colour transforms for screen content

In this section, the performances of seven different colour transforms for screen content are evaluated. The original test sequences are in RGB colour space with 8 bits per component. The dataset used in this chapter includes five screen content sequences from the common test conditions. The first frame of each sequence is shown in *Figure 3.17*.

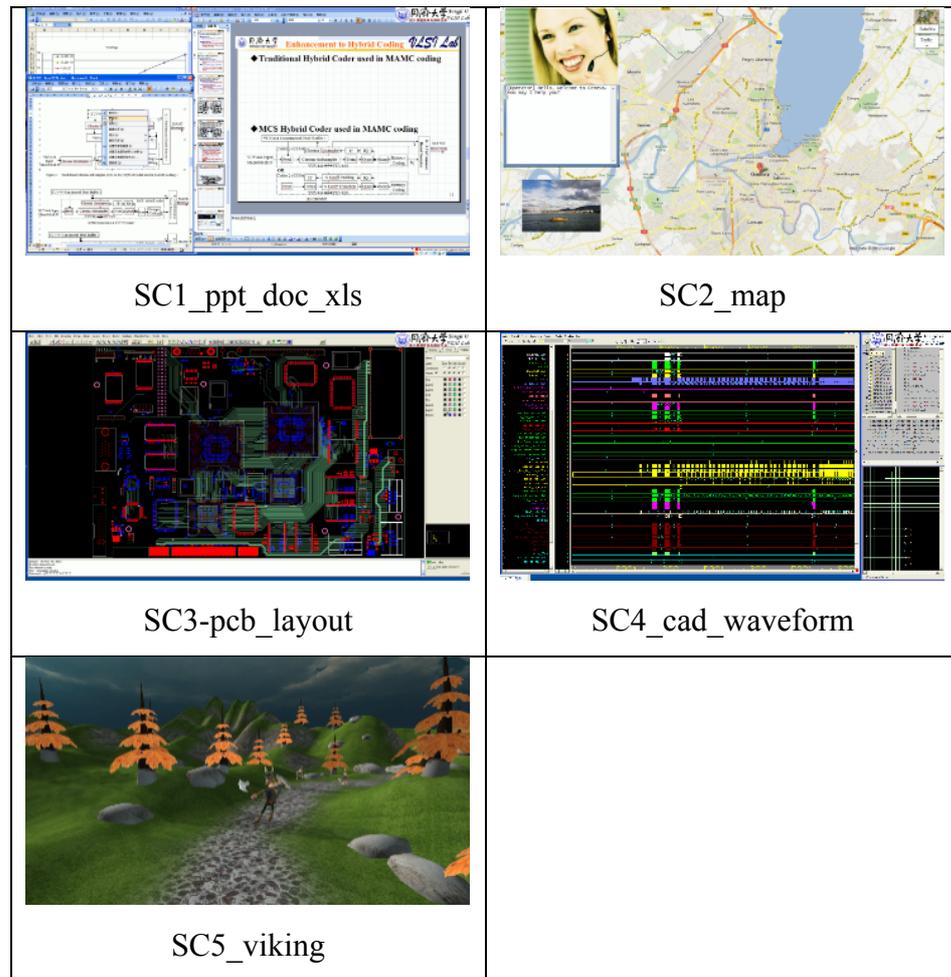


Figure 3.17 First frame of each screen content test sequence.

The software used in the experiments was HEVC Range Extension test model version HM15.0-RExt8+SCM2.0 [54]. The other parameters were set the same as the same as listed in Section 3.5.2.

BD rate results are shown in *Table 3.22*. The performance of YCbCr is set as the reference as it is the most commonly used colour space in video compression. Negative values in represent bit rate savings. The related graph comparisons are shown in *Figure 3.18*.

Table 3.22 Comparisons of BD-rate of different colour transforms for screen content in HEVC (YCbCr is the reference).

Sequences	YCoCg	P-DCT	PPU-1	PPU-2	PPU-3	PPU-4	PPU-4 lifting
SC1	-9.73%	-8.45%	-8.33%	-8.18%	-7.28%	-8.25%	-9.86%
SC2	-7.18%	-4.79%	-6.28%	2.6%	-2.87%	-7.18%	-8.89%
SC3	-2.57%	-3.52%	-4.75%	-4.29%	28.09%	-5.01%	-5.22%
SC4	-4.85%	-3.89%	-3.7%	-4.24%	1.04%	-3.36%	-3.71%
SC5	-16.34%	-14.05%	-13.19%	-13.62%	-9.75%	-13.18%	-16.6%
Average	-8.13%	-6.94%	-7.25%	-5.55%	1.85%	-7.40%	-8.86%

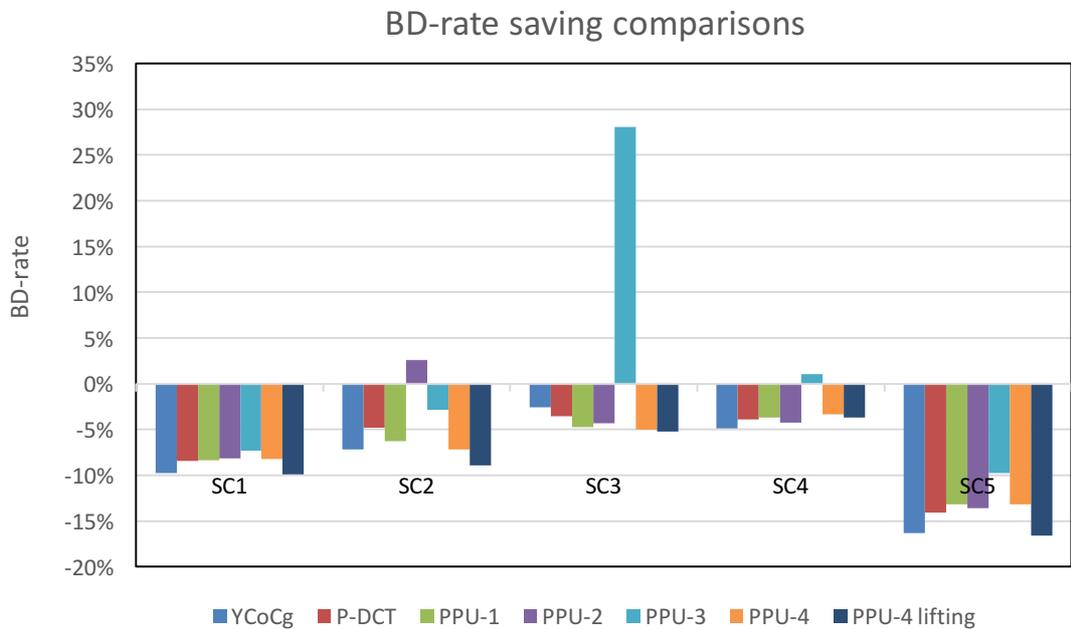


Figure 3.18 BD-rate comparisons of colour transforms for screen content (negative value indicate BD-rate saving).

From the overall average results, the PPU-4 lifting colour transform method achieves the highest BD-rate saving. The difference of PPU-4 and PPU-4 lifting is that the former one is implemented by using the transform matrix and the latter one is implemented using the lifting equations. The lifting approach has improved the pixel accuracy in the colour transform. For each test sequence, all proposed colour transform approaches achieved improvement compared with YCbCr, except PPU-3 which is designed for a no-update scenario. In PPU-3 colour space domain, the chroma components are highly correlated. It does not remove the colour inter-spectral redundancy in RGB components. However, there is still BD-rate saving for test sequences SC5-viking, not as much as other colour transforms, though.

### 3.7 Discussion

KLT is the best colour transform method for natural content, which can achieve up to 10.81% BD-rate saving (or up to 1.83dB BD-PSNR improvement) comparing to YCbCr. For screen content, on the contrary, 3-channel PPU achieved better performance than other methods. Although KLT had the similar improvement for natural content (up to 10.81% BD-rate saving), it is not an efficient method for screen content, particularly considering the computation complexity. Note that the KLT transform matrix is computed individually for each test sequence. It increases the computational complexity for the evaluation and implementations, which is not optimal for real-time video compression but can be considered in other applications such as DVD.

In 3-point P-DCT conversion, the bit rate increased more than 50% for natural content sequences, especially for test sequence “*Ducks&legs*”, which gained 78% bitrate increases compared to YCbCr. By comparing the BD-PSNR results with YCbCr, all other colour conversions generated more distortion on green component but gained higher PSNR of blue and red components, although the overall distortion was reduced.

The results show that the proposed PPU-4 lifting is the most efficient approach over all test screen content sequences (up to 16.6% BD-rate saving). However, the selection of two parameters in lifting scheme needed to be improved. It is noted that PPU-3 which

remains large redundancy in three channels also improved the coding efficiency for some screen content. More investigations are needed on colour component analysis for screen content to conduct the content-adaptive colour transforms.

Other colour conversion methods either reduced the computational complexity or achieved higher de-correlation. The objective evaluation metrics showed that our proposed methods achieved bitrate saving and reduced the distortion compared with YCbCr. However, on aspect of visual experience, it concerns about the *Human Visual System* (HVS) sensitivity to colours. Considering visual quality of sequences, the subjective metrics are necessary as part of the performance evaluation metrics. In future work, subjective evaluation is another relevant factor should be considered in dealing with colour transform design.

### 3.8 Summary

There are two colour transforms proposed in this chapter: KLT approximations and 3-channel PPU. Several commonly used colour space conversion methods are also included in the tests. Results showed that all other colour conversion methods achieved improvement on average compared to YCbCr.

Screen content which explored in this chapter is one of the next generation content. Next two chapters develop the coding solutions for another type of next generation content, UHD content.

## Chapter 4 Contouring artefacts removal for HEVC compressed UHD content

While Chapter 3 explored the colour transform for one of the next generation content, screen content, this chapter develops the coding tools for another type of next generation content: UHD content. False contouring is one of the visual artefacts in compressed UHD video sequences which affect the visual experience. This chapter proposes a contouring detection method and a dithering contouring removal method. The contouring detection aims to separate the contouring area and non-contouring area, so that dithering method can be used to remove the visual contouring while keeping artefact-free regions unaffected.

### 4.1 The overview of contouring artefacts

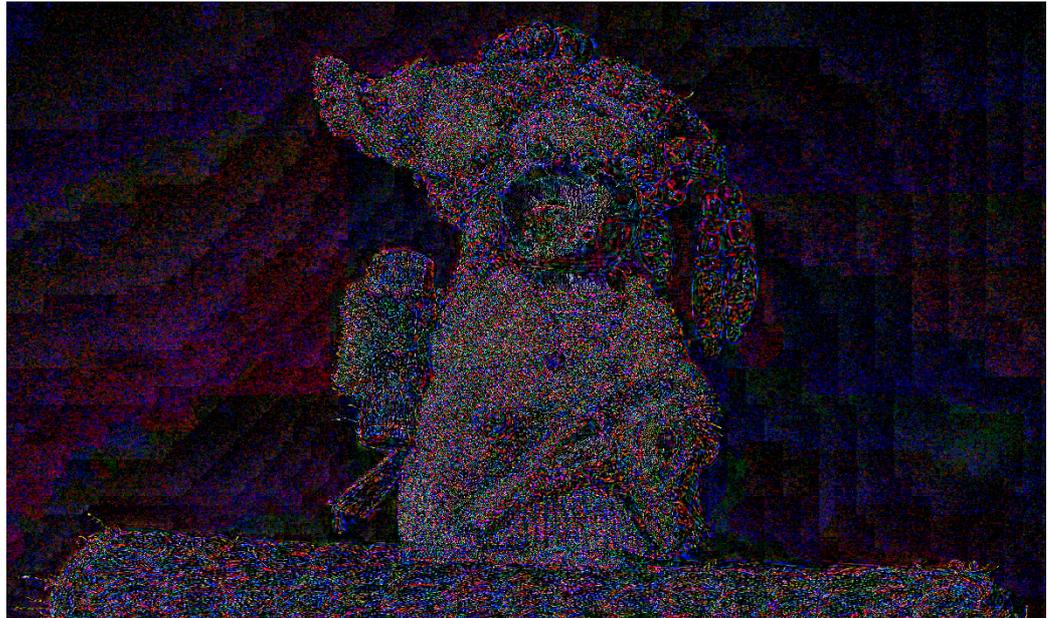
As television display have become larger and the resolutions of video content have become higher, some artefacts, which are not noticeable on small displays, have now become visible. One such type of artefact is the false contouring which shows as banding in images. It commonly occurs in smooth gradient areas (*e.g.* sky, plain wall) where neighbouring pixels only differ slightly [55]. After compression, smooth gradient areas break down and appear as several contouring bands. Compared to HD content, the contouring artefacts are amplified and become more noticeable in higher resolution content, *e.g.* UHD video sequences. An example of contouring artefacts in UHD content is shown in *Figure 4.1*. The frame is from a test sequence named *Ningyo* (source from BBC R&D). The background on both left and right sides of the doll were smoothly changed due the luminance background light. These smooth gradient area resulted in contouring banding lines in the compressed frame in *Figure 4.1* (b). Figure (c) shows the scaled pixel difference between original and compressed pixels. In smooth gradient regions, which are the background area in *Ningyo* sequence, the gradient details are lost during compression.



(a) Original frame is artefacts free



(b) Compressed frame has contouring artefacts at the background



(c) Pixel difference (scaled) between (a) and (b)

Figure 4.1 Example of contouring artefacts in Ningyo sequence.

Due to the increased artefact visibility in UHD content, the appearance of contouring is much less tolerable, which has a negative effect on the visual experience. Therefore, it is essential to detect the contouring regions and remove the contouring artefacts from UHD content.

During the process of transform and quantisation in image compression, the coding error results in visual artefacts. Contouring is one of the artefacts caused by quantisation. In the smooth gradient regions, the DCT high frequency coefficients have low amplitudes [56] making them prone to rounding to zero in quantisation. Since the high frequency coefficients contain less-important image information, i.e. small details, the loss of high frequency components in smooth gradient area results in contouring bands in flat regions. Table 4.1 shows pixel values of  $4 \times 4$  blocks from luminance component of both original sequence and decoded *Ningyo* sequence. The pixel values in the original block are very similar. After HEVC, the decoded block contains two pixel values: 41 on the left side and 40 on the right side, which visually is shown as a contouring line in-between. The two tables on the right side in Table 4.1 show the corresponding DCT coefficients of the two blocks. From the coefficients analysis we can see that there are several small

*CHAPTER 4. CONTOURING ARTEFACTS REMOVAL FOR HEVC COMPRESSED UHD CONTENT*

coefficients quantised and rounded to zero. These small coefficients are high frequency coefficients. The contouring artefacts are due to the lack of these high frequency coefficients.

*Table 4.1 Contouring block analysis (luma component).*

Block pixel values				DCT coefficients			
41	41	40	40	160.5000	0.4619	0	-0.1913
39	39	40	40	-0.2706	-0.2500	0	-0.1036
40	40	40	40	1.5000	1.3858	0	-0.5740
41	41	40	40	0.6533	-0.6036	0	-0.2500
Original block				Original block			
41	41	40	40	162.0000	1.8478	0	-0.7654
41	41	40	40	0	0	0	0
41	41	40	40	0	0	0	0
41	41	40	40	0	0	0	0
Decoded block with contouring				Decoded block with contouring			

Most previous studies of contouring artefacts that are reported within the open literature focus on general video content. To the best of the author’s knowledge, the case of contouring artefacts occurring in UHD content has not been given great attention by researchers in the past and this motivated the present study.

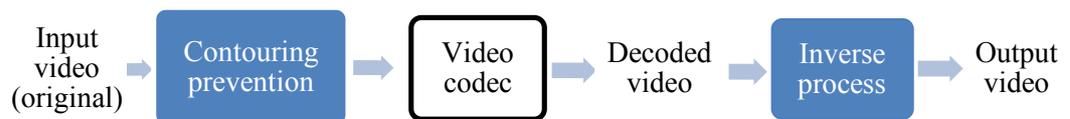
The contouring artefacts removal work can be classified into three approaches, depending on whether the approach is before, after or within video codec: pre-codec approach, post-processing approach and in-codec approach. The flowchart of the three approaches are shown in *Figure 4.2*. The blue blocks represent the adding processing. The proposed method in this chapter is a post-processing approach which removes the contouring from decoded frames. The pre-codec approaches and in-codec approaches are introduced in Chapter 5. It should be noted the contouring detection and contouring removal methods proposed in this thesis are derived on luminance component as an instance. This is because human eyes are more sensitive to brightness difference than colour difference. However, all of the models and the proposed methods can be extended to other two

chrominance components, which can be implemented on luminance and chrominance components separately.

1) Post-processing approach



2) Pre-codec approach



3) In-codec approach



Figure 4.2 Three contouring reduction approaches.

The proposed method includes two parts: contouring detection and contouring removal. The contouring detection is used to separate contouring area from non-contouring area. The contouring removal is to remove the contouring artefacts from contouring area. The proposed method has the following features:

- Firstly, the contouring detection and contouring removal are applied at block-level, which is more computational efficient compared to pixel-based methods, especially for UHD content. The contouring detection is applied using the local contrast. Several block sizes are used to compute local contrast to adapt different sizes of contouring regions. The corresponding contouring removal is applied using the same block size as that in the contouring detection process.
- Secondly, the contouring removal method is only applied in contouring area while keeping the artefact-free regions unchanged. This can be achieved by using the contouring map. The binary contouring map is generated after contouring

detection. It contains two colours: black area (values equal to zero) are the contouring regions, and white colour (values equal to one) represent non-contouring area.

- Thirdly, it enhances gradient smoothness and block boundary smoothness in the contouring regions, which improves the visual quality. It aims to remove the contouring bands/lines by smoothing each block boundary in contouring and also keeping the gradient smoothness.
- Also, it has low computational complexity since the dithering is implemented in the spatial domain by adding a noise map and adjusting block average values, as opposed to the transform domain dithering stated in [55].
- Moreover, the proposed method is effective in removing different scales of contouring bands.

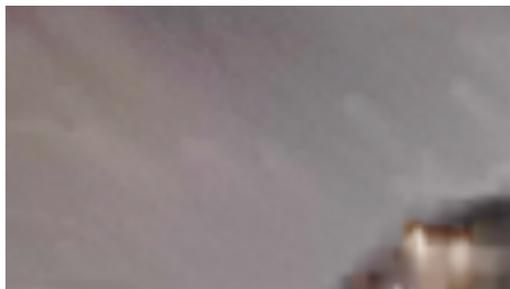
The rest of this chapter is organised as follows: Section 4.2 introduces related work. The proposed contouring detection method and contouring removal method are described in Section 4.3 and 4.4, respectively. Section 4.5 presents performance evaluation followed by experimental results in Section 4.6. Section 4.7 discusses the data analysis. Finally, the summary is drawn in Section 4.8.

## **4.2 Related work**

There have been several studies on reducing contouring artefacts. Some contouring preventing approaches dither pixels before encoding to stop the loss of high frequency component during quantisation [57] [58]. One of the dithering methods is adding noise which was originally introduced in [57]. The other contouring removal methods are post-processing which applies dithering or filters on compressed sequences which already have contouring artefacts [55] [56] [59].

As motioned earlier, there are three approaches for contouring artefacts removal. The previous related work is mainly focused on post-processing approaches. Ahn et al. [59] proposed a three-step contouring removal algorithm: random shuffler followed by low-

pass filtering and dithering. Random shuffler swaps pixel randomly in a local block, which dithers pixels but generates noise. A low-pass filter is then applied to remove the noise. However, it produces other false contours. The dithering step aims to remove false contouring from low-pass filter. The shortcoming of this method is that it generates more noise from the shuffler, and the filter could blur the image. It also caused diffusion differ error between output of low-pass filter and quantized image. Lee et al. [60] employed a one-dimensional variable-size filter on the detected contouring pixels to reduce false contouring. It firstly detects the contouring regions by using bit-depth reduction. The second stage is to separate false contours and edges or textures using directional contrast features. The advantage of this method is that the proposed filter is only applied at detected contour pixels. But the smooth filter could also blur the real edges. Yoo et al. [61] applied a  $16 \times 16$  block based in-loop de-contouring algorithm by adding pseudorandom noise masks to the blocks to dither the pixels. The flat regions were detected via smooth gradient count. This, however, makes the picture look noisy and is not successful in removing contouring bands in UHD video sequences according to our experiments. Jin et al. [55] used the dithering method by adjusting the average value of each block to improve the gradient smoothness with the trade-off in the block boundary smoothness. Additionally, some other de-contouring methods used probabilistic dithering [62] or generated noise according to temporal correlation and brightness information [63]. Bhagavathy et al. [62] applied a multi-scale contouring detection by using multi-scale analysis of the neighbourhood of each pixel. It computed the expected mean colour value in the local blocks. The proposed probabilistic dithering method in [62] is based on the distribution of colours in the neighbourhood. Due to the availability of the source code, we only tested the Jin's method. However, since the adjustment is predicted from blocks above and to the left of the current block, it is found to generate a new artefact of diagonally oriented texture, as shown in *Figure 4.3*.



*Figure 4.3* Experiment result by using method proposed in [55].

There are also some methods following the pre-codec approach or in-codec approach. Roberts [57] suggested to use pseudo-random noise in picture coding, which applied as a pre-processing approach. It simply added some noise to the signal before quantisation and subtracted the same noise at the reviver. It removed the banding lines from the low definition resolution ( $340 \times 340$ ) pictures in [57]. From the initial test results we noticed adding noise on UHD content can not prevent the false contouring artefacts, as shown in *Figure 4.4*. Matsuo *et al.* [64] proposed a bit-depth reduction contouring prevention method for UHD content. It applied bit-depth reduction before encoding and bit-depth reconstruction after decoding. The bit depth reduction was performed by converting 12-bit pixel data into 8-bit using Lloyd-Max quantisation. The bit depth reconstruction was produced by using the side information generated from encoder side. The disadvantage of this method is that it generated additional side information which need to be sent to the decoder. It increased the bitrate significantly when the contouring area is large, for example the *Ningyo* sequence as shown in *Figure 4.1*.



*Figure 4.4* Test result of applying method proposed in [57].

Another two in-codec contouring prevention methods are introduced in [65] and [66] by using the adaptive quantisation in HEVC. Dias *et al.* [65] proposed a fine quantisation algorithm by reducing the quantisation parameters in areas which are prone to contouring artefacts. The quantisation parameters selection is based on a pre-generated coding flag map. This method is designed for the higher bitrate scenario. The results showed the bitrate was increased up to 51%. Another disadvantage is that the parameters can not be selected automatically.

### 4.3 Contouring detection

This section analyses the pixels in contouring area and proposes a block based contouring detection method.

#### 4.3.1 Contouring artefacts analysis

The luminance components of the original frame and the HEVC compressed frame are shown in *Figure 4.5*. The contouring bands in UHD content usually have small step-changes in pixel values. Based on our experiments using 8-bit sequences, the difference in the pixel value between two adjacent contouring bands is  $\pm 1$ . *Figure 4.6* compares the luminance signal of one line in the same position from the original frame and the compressed frame which suffers from contouring. The original frame has a small gradient across a large area and the compressed frame has several noticeable contouring bands. The pixel value graph of the compressed frame shows a “staircase” effect.

The aim of the proposed method is to apply dithering only in contouring area and leave the artefact-free areas (e.g. object edges) unaffected. Thus, it is necessary to detect the regions with contouring and generate a contouring map.



(a) Original frame luma component



(b) Compressed frame luma component

Figure 4.5 Original and compressed frames in Ningyo sequence (luma component).

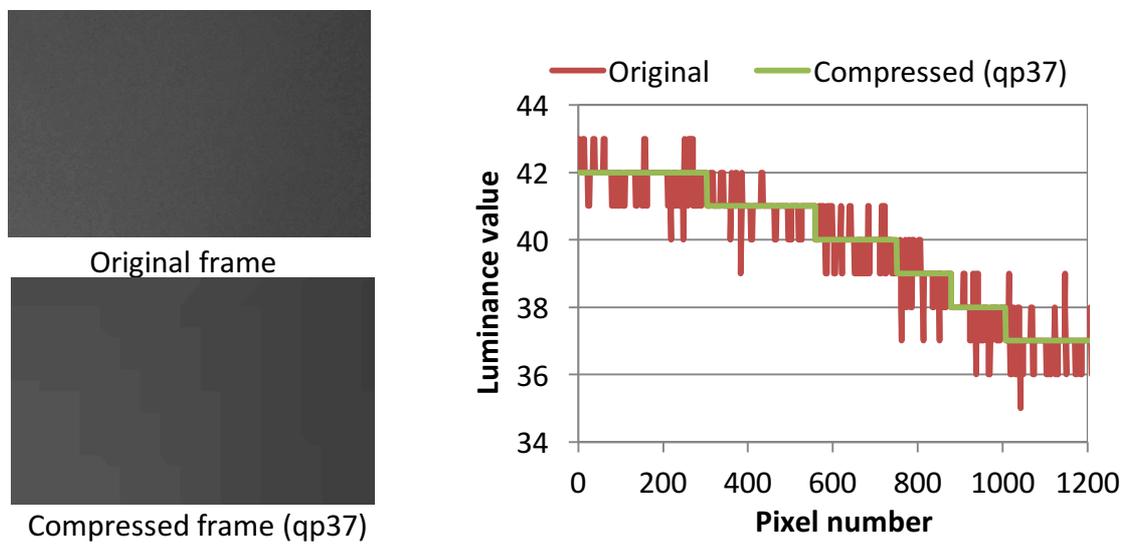


Figure 4.6. Luminance signal from original frame and compressed frame.

### 4.3.2 Proposed contouring detection method

Previously used contouring detection methods include variance, Weber ratio, entropy and contrast [60]. Local derivation (a measure of contrast) is commonly used in contouring detection since it can easily separate the false contouring and real edges [59]. Some narrow contouring can be mistaken for blocking artefacts. Blocking and contouring artefacts are both caused by coarse quantisation. Blocking is likely to happen in over-compressed images where brightness discontinuity is seen across the block boundaries. The way to distinguish contouring from other blocking artefacts is to see whether it occurs in a smooth gradient area or not. Therefore, setting a proper threshold of local contrast can successfully detect the contouring regions.

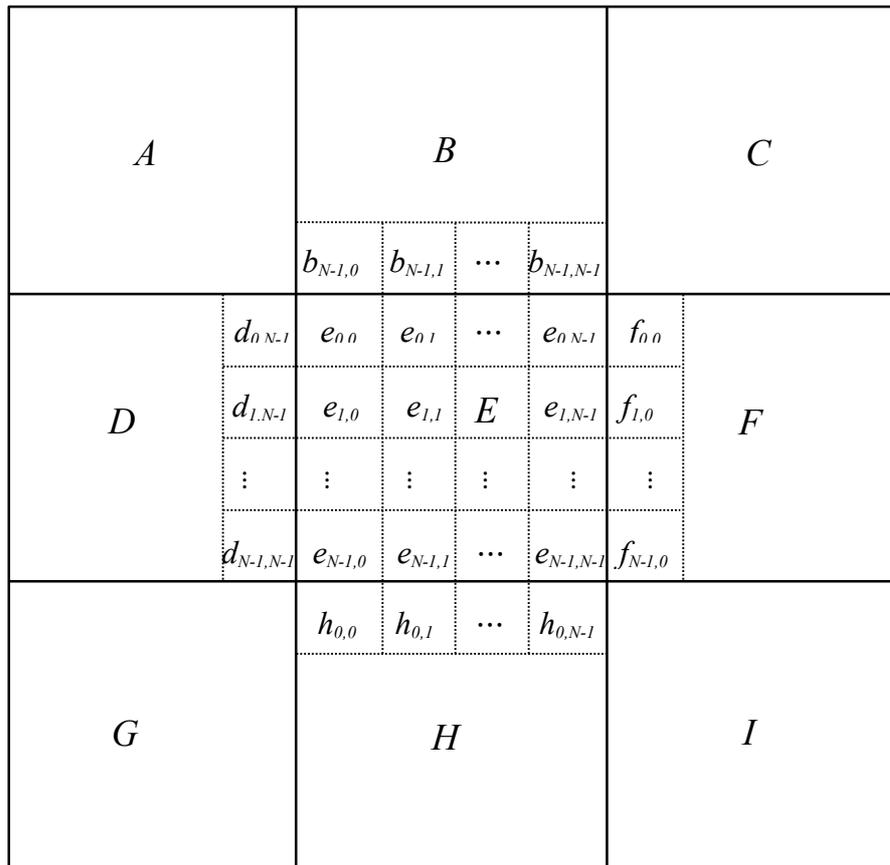


Figure 4.7 Relative locations of current block  $E$  and its neighbouring blocks.

The proposed contouring detection is applied block-by-block and is based on the local contrast. *Figure 4.7* shows the relative locations of the current block  $E$  and its neighbouring blocks. Let  $\mu_x$  ( $x = A, \dots, I$ ) denote the average pixel value of each block in *Figure 4.7* and  $\bar{\mu}$  is the average of  $\mu_x$ . The contrast  $c$  of the current block is computed as:

$$c = \sqrt{\frac{\sum(\mu_x - \bar{\mu})^2}{9}}. \quad (4.1)$$

Let  $\lambda_x$  represent the difference between each block mean  $\mu_x$  and the overall average value  $\bar{\mu}$  as:

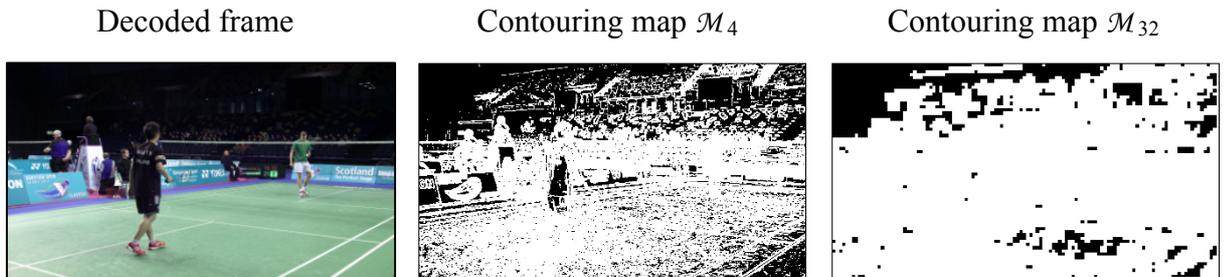
$$\lambda_x = \mu_x - \bar{\mu}, \quad (4.2)$$

which is in the range of  $[-1, 1]$  if the current block is in the smooth gradient area which is typical for contouring regions for 8-bit content. Now given that  $\lambda_x^2 \leq 1$ ,  $\sum \lambda_x^2 \leq 9$ . The range of result of local contrast for pixels in the contouring region is derived as:

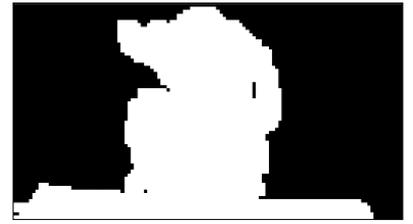
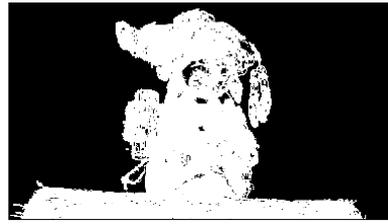
$$c = \sqrt{\frac{\sum(\mu_x - \bar{\mu})^2}{9}} = \sqrt{\frac{\sum \lambda_x^2}{9}} \leq \sqrt{\frac{9}{9}} = 1. \quad (4.3)$$

The threshold of contrast is set as 1 and the blocks that have low contrast ( $c \leq 1$ ) are included in the contouring map.

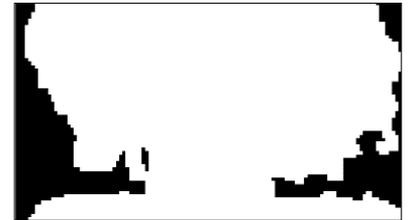
The frames (a)–(g) in *Figure 4.8* are seven 8-bit HEVC compressed UHD ( $3840 \times 2160$ ) sequences with the source information shown in *Table 4.2*.



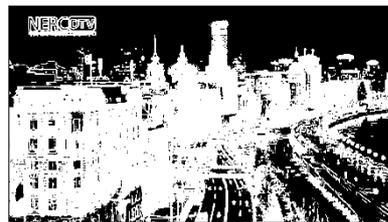
(a) *Badminton*



(b) *Ningyo*



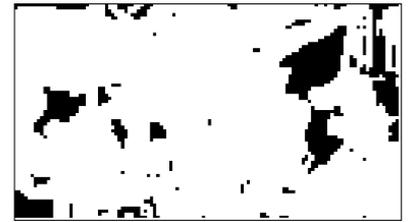
(c) *NingyoPompoms*



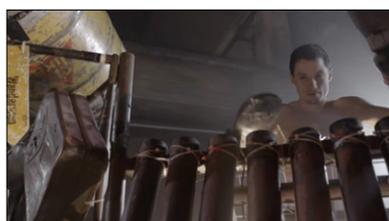
(d) *Bund Nightscape*



(e) *Young Dancer*



(f) *Candle Smoke*



(g) *No Sleep*

Figure 4.8 Decoded frames and contouring maps.

Table 4.2 The specifications of test UHD sequences.

Sequences name	Source	Resolution	Bit depth	Framerate (fps)
<i>Badminton</i>	<i>BBC R&amp;D</i>	3840 × 2160	8	50
<i>Ningyo</i>	<i>BBC R&amp;D</i>	3840 × 2160	8	50
<i>Ningyo Pompoms</i>	<i>BBC R&amp;D</i>	3840 × 2160	8	50
<i>Bund Nightscape</i>	<i>SJTU</i>	3840 × 2160	8	30
<i>Young Dancer</i>	<i>EUB</i>	3840 × 2160	8	50
<i>Candle Smoke</i>	<i>EUB</i>	3840 × 2160	8	50
<i>No Sleep</i>	<i>Kamerawerk</i>	3840 × 2160	8	60

All four sequences are in YCbCr colour space with chroma subsampling 4:2:0. Some sequences show wide contouring bands in large areas, such as *Ningyo* which was generated by *British Broadcasting Corporation, Research and development* (BBC R&D). In these sequences, the contouring bands exist in compressed video even with lower quantisation parameter (QP) values, e.g. QP = 22. The contouring appears more noticeably with a larger QP values. The other sequences present narrow contouring in several small areas, such as in the compressed *Badminton* sequence (produced by BBC R&D), and the compressed Eurovision *Young Dancers* sequence (produced by the *European Broadcasting Union* (EBU) [67]). The contouring artefacts in the compressed *Badminton* and *Young Dancers* sequences are only noticeable with higher QP values. The other sequence from EBU is *Candle Smoke* [68] where some contouring artefacts occur during compression on the smooth area in the background. Another sequence included in the test is *Bund Nightscape* (available in the 4K video sequence dataset from SJTU [69]) where some contouring artefacts occur during compression on the top smooth area in the sky. The Sequence *No Sleep* is from *Kamerawerk*. There is no substantial contouring in this sequence when it is highly compressed, except in the brightness gradient area caused by the bright light source at the rear.

The contouring maps  $\mathcal{M}_{32}$  and  $\mathcal{M}_4$  of each UHD sequence in *Figure 4.8* are generated by computing the contrast with block size  $N$  equal to 32 and 4, respectively. The black area indicates the regions with contouring artefacts and the white area is artefact-free. It is noted that particularly with  $N = 32$ , this block level contouring detection method can yield false positives in some rare cases of repeated patterns smaller than the block size. This problem is practically unlikely in the case of  $N = 4$ . Thus any detected areas in  $\mathcal{M}_{32}$  outside  $\mathcal{M}_4$  are excluded. Another reason of choosing  $N = 4$  and 32 is that the minimum and maximum transform block size in HEVC is  $4 \times 4$  and  $32 \times 32$ , respectively.

## 4.4 The proposed contouring removal method

This section introduces a multi-scale dithering method for contouring artefacts removal in compressed UHD video sequences. The overall procedure of the proposed contouring removal algorithm is shown in Section 4.4.1, followed by the high frequency component dithering and block average dithering in Section 4.4.2 and Section 4.4.3, respectively.

### 4.4.1 Overall procedure

According to our experiments, the wide contouring bands cannot be significantly removed when the dithering is performed in a small block size alone, e.g.  $4 \times 4$ . An example of dithering by  $4 \times 4$  blocks is shown in *Figure 4.9*. The processed frame still has contouring lines.



*Figure 4.9 Test result of pixel dithering on  $4 \times 4$  blocks only*

The large-block dithering, on the other hand, is able to fade away contouring lines and make contouring bands less noticeable. However, it is not sufficient to remove all contouring bands (further discussed in Section 4.7). Thus, the proposed method uses a multi-scale de-contouring algorithm with dithering on two different block sizes. The larger-block dithering step is expected to reduce wider contouring bands while smaller-block dithering is aimed at correcting smaller contouring areas in the vicinity of real image edges and also breaking down any remaining contouring in the large contouring areas. The overall procedure of the proposed method is shown in *Figure 4.10*. The two dithering block sizes used in experiments are  $32 \times 32$  and  $4 \times 4$  as the transform block size in HEVC is ranging from  $32 \times 32$  to  $4 \times 4$ . The test UHD video sequences with contouring artefacts used in this chapter are compressed with the H.265/HEVC coding standard. The following sub-sections introduce more details of each step.

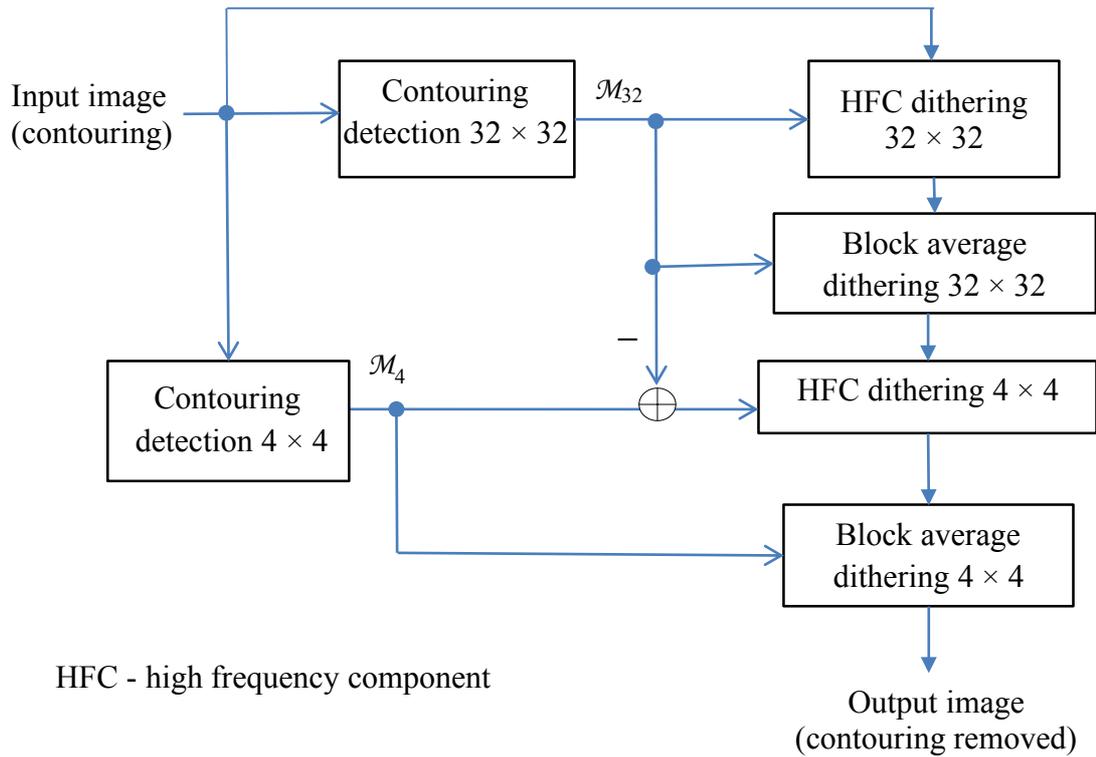


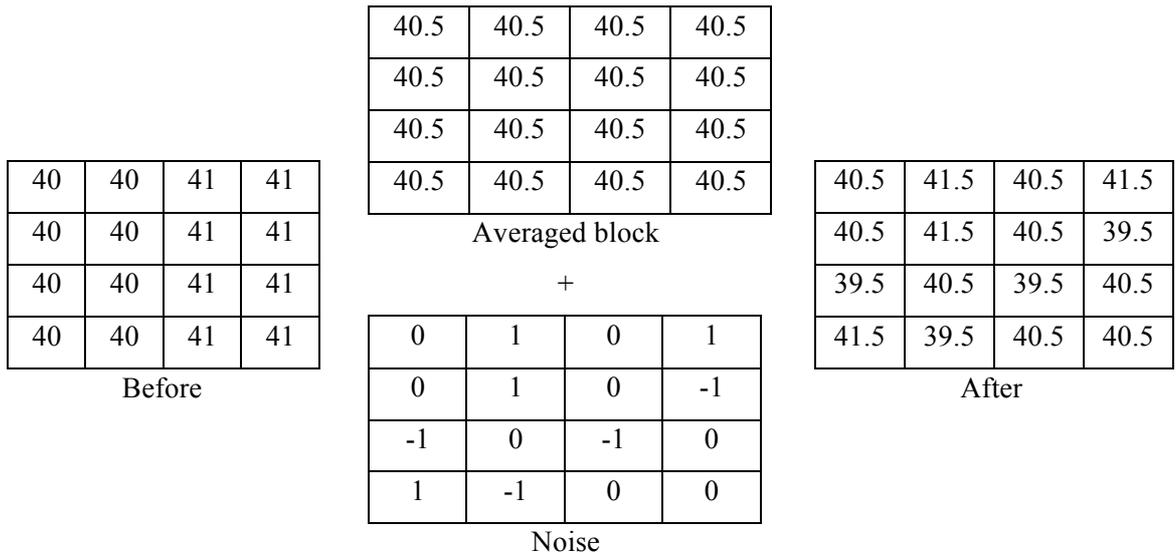
Figure 4.10 Overall procedure of the proposed contouring removal algorithm.

#### 4.4.2 High frequency component dithering

Since the average values of neighbouring blocks differ by a small amount ( $-1 \leq \lambda_x \leq 1$ ), a noise map consisting of integer values -1, 0 and 1 is generated. We start with a uniform noise distribution and round the values to the nearest integer that resulting in -1, 0 and 1, with the probabilities of 0.25, 0.5 and 0.25, respectively. The high frequency component dithering is applied only once for each block within the contouring map, either  $32 \times 32$  dithering for larger contouring regions or  $4 \times 4$  dithering for smaller areas. The contouring map  $\mathcal{M}_{32}$  is included in  $\mathcal{M}_4$ , where  $\mathcal{M}_4$  typically reaches closer to object edges, i.e. areas without contouring. Thus, the  $4 \times 4$  high frequency component dithering is only applied in the  $\mathcal{M}_4$  area that does not overlap with  $\mathcal{M}_{32}$  ( $\mathcal{M}_4 \setminus \mathcal{M}_{32}$ ), i.e. in the contouring area where high frequency component dithering has not been applied yet. An example of high frequency component dithering on a  $4 \times 4$  block is shown in Figure 4.11. It first computes the average value of all pixels in the current block and forms an averaged block. A noise map is then generated and added to the averaged block. The pixel values are processed in floating point during the de-contouring and rounded to the nearest integer

*CHAPTER 4. CONTOURING ARTEFACTS REMOVAL FOR HEVC COMPRESSED UHD CONTENT*

afterwards. The high frequency component dithering can move the diagonal contouring lines to the block boundaries. An example is shown in *Figure 4.12*. Frame in (a) is a decoded frame which has diagonal contouring lines. After applying the high frequency component dithering, the contouring lines move to block boundaries in the processed frame (b). Such an artefact is then effectively removed in the block average dithering process when the block boundary smoothness is taken into account.



*Figure 4.11 An example of high frequency component dithering (4 × 4).*



*Figure 4.12 Decoded frame (a) and processed frame after dithering (b).*

### 4.4.3 Block average dithering

Block average dithering adjusts the average value of each block to achieve gradient smoothness and block boundary smoothness. The adjustment  $\delta$  is predicted from neighbouring blocks. The  $32 \times 32$  block average dithering is firstly applied in contouring map  $\mathcal{M}_{32}$ . It can make most contouring lines fade away and become less noticeable. But it is not sufficient to remove all contouring lines, especially the horizontal and vertical lines. Thus, smaller blocks have to be additionally processed to remove the remaining contouring bands. Therefore, the  $4 \times 4$  block average dithering is applied in contouring map  $\mathcal{M}_4$  which includes the small contouring area as well as the regions in  $\mathcal{M}_{32}$ .

The gradient smoothness of the current block is computed before dithering as:

$$G_0 = (\mu_E - \mu_A)^2 + (\mu_B - \mu_D)^2 + (\mu_E - \mu_I)^2 + (\mu_H - \mu_F)^2, \quad (4.4)$$

where  $\mu_x$  represents the average value of each block. In block average dithering, a small adjustment value  $\delta$  is added to the current block's average value. The gradient variation after adjustment is then derived as:

$$\begin{aligned} G(\delta) &= ((\mu_E + \delta) - \mu_A)^2 + (\mu_B - \mu_D)^2 + ((\mu_E + \delta) - \mu_I)^2 + (\mu_H - \mu_F)^2, \\ &= 2\delta^2 + 2\delta(m + k) + G_0, \end{aligned} \quad (4.5)$$

where  $m = \mu_E - \mu_A$  and  $k = \mu_E - \mu_I$ .

The block boundary smoothness measures pixel differences across block boundaries. Unlike the dithering method in [55] where only above and left block boundaries are included in the smoothness function, all four block boundaries are taken into account in this chapter. This is to avoid generating diagonal texture artefacts. Let **H1** and **H2** represent the horizontal pixel difference vector across left and right block boundaries, respectively. **V1** and **V2** are the vertical pixel difference vectors across the top and bottom block boundaries, respectively. Pixel differences are represented as:

$$\mathbf{V}_1 = \begin{bmatrix} e_{0,0} - b_{N-1,0} \\ e_{0,1} - b_{N-1,1} \\ \vdots \\ e_{0,N-1} - b_{N-1,N-1} \end{bmatrix}, \mathbf{V}_2 = \begin{bmatrix} e_{N-1,0} - h_{0,0} \\ e_{N-1,1} - h_{0,1} \\ \vdots \\ e_{N-1,N-1} - h_{0,N-1} \end{bmatrix},$$

$$\mathbf{H}_1 = \begin{bmatrix} e_{0,0} - d_{0,N-1} \\ e_{1,0} - d_{1,N-1} \\ \vdots \\ e_{N-1,0} - d_{N-1,N-1} \end{bmatrix}, \mathbf{H}_2 = \begin{bmatrix} e_{0,N-1} - f_{0,0} \\ e_{1,N-1} - f_{1,0} \\ \vdots \\ e_{N-1,N-1} - f_{N-1,0} \end{bmatrix}. \quad (4.6)$$

The block boundary smoothness is represented as:

$$E_0 = \|\mathbf{H}_1\|^2 + \|\mathbf{H}_2\|^2 + \|\mathbf{V}_1\|^2 + \|\mathbf{V}_2\|^2, \quad (4.7)$$

where  $\|\mathbf{■}\|$  is the Euclidean norm of a vector. To adjust the average value of a block, an offset of  $\delta$  is added to each of its pixel values. The block boundary smoothness after block average dithering is:

$$E(\delta) = 4N\delta^2 + 2\delta(h + v) + E_0, \quad (4.8)$$

$$h = \sum_{j=0}^{N-1} (e_{j,0} - d_{j,N-1} + e_{N-1,j} - f_{j,0}), \quad (4.9)$$

$$v = \sum_{j=0}^{N-1} (e_{0,j} - b_{j,0} + e_{j,0} - h_{0,j}), \quad (4.10)$$

where  $h$  and  $v$  are the sum of pixel differences in horizontal and vertical directions, respectively.

The aim is to find a suitable  $\delta$  that can improve gradient smoothness and also the block boundary smoothness. For each function, lower values lead to stronger smoothness. However, the two functions have different minimum points which means  $G(\delta)$  and  $E(\delta)$  cannot be minimised simultaneously. Denote  $Y$  as the cost function of  $\delta$ , which consists of the normalised  $G(\delta)$  and  $E(\delta)$  as:

$$Y = \frac{G(\delta)}{\alpha} + \frac{E(\delta)}{\beta}, \quad (4.11)$$

where factors  $\alpha$  and  $\beta$  are the local maximum values of each function. The aim is to get the suitable  $\delta$  which can minimise the cost function defined as:

$$\delta = \operatorname{argmin} \left( \frac{G(\delta)}{\alpha} + \frac{E(\delta)}{\beta} \right). \quad (4.12)$$

Since the graph of quadratic function  $Y$  is an upward parabola (the coefficient of square term is positive), the minimum point of function  $Y$  is achieved when the differential of the cost function equals zero. The first derivative of function  $Y$  is computed as:

$$Y' = \frac{4}{\alpha} \delta + \frac{2(m+k)}{\alpha} + \frac{8N}{\beta} \delta + \frac{2}{\beta} (h + v). \quad (4.13)$$

By setting the derivative of function  $Y$  to zero,  $\delta$  is derived as:

$$\delta = \frac{-(m+k)\beta - \alpha(h+v)}{2(\beta + 2N\alpha)}. \quad (4.14)$$

The adjustment  $\delta$  takes values from a narrow range since the pixel values on two neighbouring contouring bands are very close.

## 4.5 Performance evaluation

In this section, the proposed method is evaluated by using both objective and subjective methods. It also describes the methodology for carrying out visual quality evaluation and the experiment setup.

The experiments in this chapter are applied to seven 8-bit HEVC (HM-12.0 and HM-16.0, Random Access, Main profile) compressed UHD ( $3840 \times 2160$ ) sequences listed in *Table 4.2*. The first frame of each test sequences is shown in *Figure 4.8*. Each video sequence is tested with four different fixed quantisation parameter values: 22, 27, 32, 37. The HEVC test conditions are listed in *Table 4.3*. Both objective and subjective evaluation methods are used in this chapter.

Table 4.3 Test conditions.

HEVC test model	HM-12.0, HM 16.0
Configuration	Random Access
QPs	22, 27, 32, 37
Intra period	32 for 30 fps* (Bund Nightscape); 48 for 50 fps (e.g. Badminton); 64 for 60 fps (No Sleep)

\*fps= frames per second

### 4.5.1 Objective evaluation

As introduced in Section 2.3, MSE and PSNR are typically used in objective quality evaluation. HEVC uses BD-PSNR to evaluate bit-rate reduction, and BD-rate to evaluate distortion. However, PSNR is not accurate for video quality evaluation since it is based on average of squared error. SSIM is another objective evaluation metric used in video quality evaluation [27]. Instead of using mean square errors, it measures the similarity of local pixel information between two images. The the similarity of two images is computed as:

$$SSIM(x, y) = \frac{2(\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}, \quad (4.15)$$

where  $\mu_x$  and  $\mu_y$  are the averages of  $x$  and  $y$ , respectively;  $\sigma_x^2$  and  $\sigma_y^2$  are variances of  $x$  and  $y$ , respectively; and  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ . Two variables  $c_1$  and  $c_2$  are computed as:

$$\begin{aligned} c_1 &= (k_1L)^2, \\ c_2 &= (k_2L)^2. \end{aligned} \quad (4.16)$$

where  $L$  is the dynamic range of the pixel-values (255 for 8-bit pixel),  $k_1$  and  $k_2$  are set as default values with 0.01 and 0.03, respectively.

## 4.5.2 Subjective evaluation

It is also necessary to apply subjective test to evaluate the video quality, especially for visual artefacts. This section includes the evaluation method and the room setup.

### 4.5.2.1 Room setup

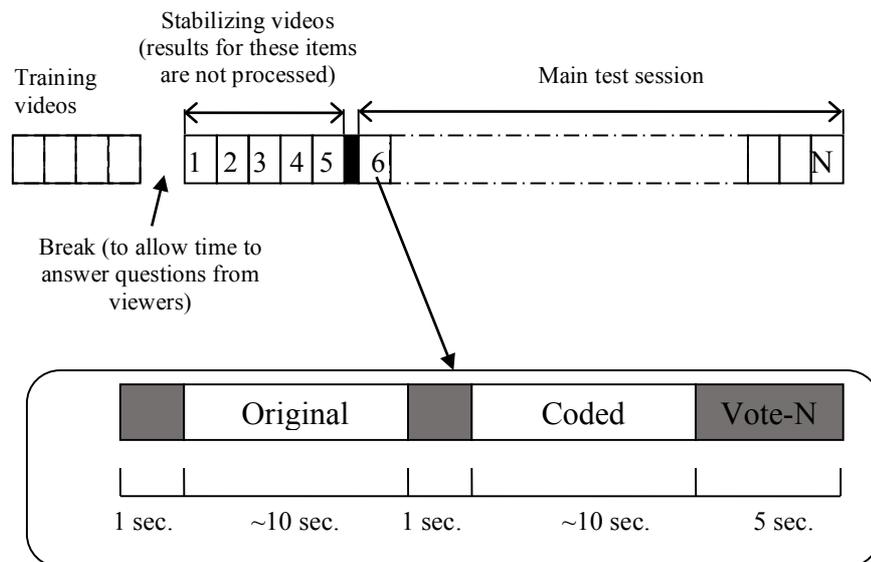
The room is set up following the guidelines from International Telecommunication Union (ITU) [71] [72] on viewing distance and room lighting. The recommend viewing distance is  $1.5H$  [73] ( $H$  is the height of the picture). The monitor used in the tests is a Panasonic TX-50AX802B Smart 3D Ultra HD 50" LED TV. There are up to five viewers in one session sitting in two rows: 3 viewers in front and 2 viewers in back, as shown in *Figure 4.13*. During a test session, the room is illuminated by lights fitted at the back of the TV display.



*Figure 4.13 Test room setup.*

### 4.5.2.2 Evaluation Method

The evaluation method used in our experiment is Degradation Category Rating (DCR). The formal methods and guidelines DCR test method are described in ITU Recommendations ITU-T P.910 [73]. The structure of the DCR method is shown in *Figure 4.14*. A video clip pair includes displaying of original sequence (10 seconds), gap (1 second), coded sequence (10 seconds) and vote (5 seconds). Before the main test session, there are 4 - 5 “dummy videos” to stabilize the viewers’ option. The data issued from these videos are not taken into account in the results of the test.



*Figure 4.14* Presentation structure of test session and each video clip pair.

## 4.6 Experiments and results

The results of the de-contouring experiments are shown in *Figure 4.15*. The pictures have been cropped to show the key contouring areas. The left column shows compressed frames (QP = 37) with contouring, and the right column shows the de-contoured frames by performing the proposed dithering method. The contouring artefacts are mostly removed from the test sequences. Note that the contouring artefacts can be hard to verify on the printed paper, the coloured images are available in [74], or at [75]. *Figure 4.16* compares the de-contouring results of the *No Sleep* sequence by using de-contouring method in [55] and the proposed method in this chapter. The de-contouring method in

[55] could lead to discontinuity at the block boundaries and generate diagonal texture, which is visibly avoided in the proposed method.

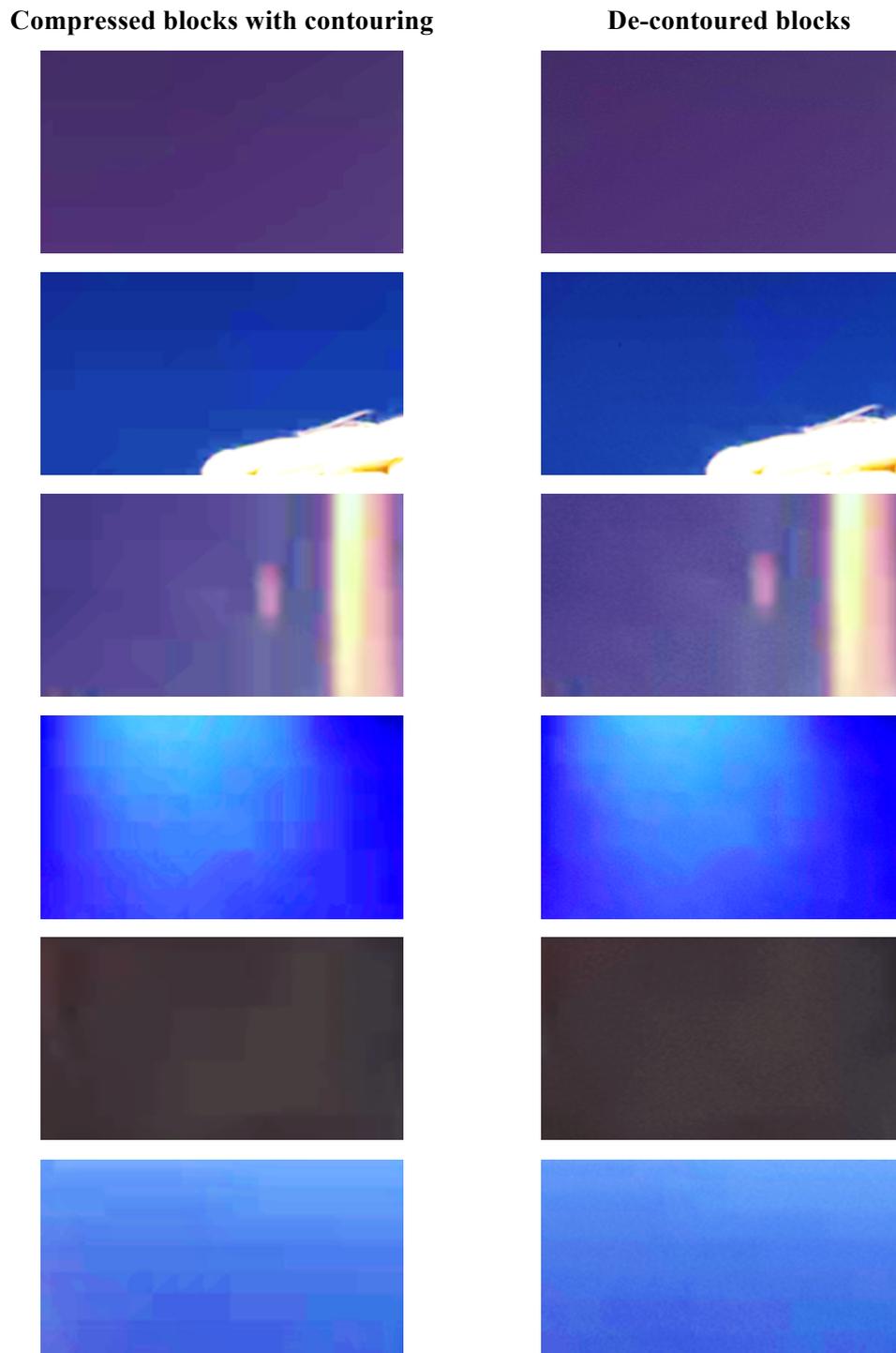


Figure 4.15 Frames before (left) and after dithering (right).



Figure 4.16 The results of Jin's method (left) and proposed method (right) of No Sleep sequence (cropped frame).

The rest of this section shows both objective and subjective test results for three sequences: *Badminton*, *Bund Nightscape* and *Ningyo*.

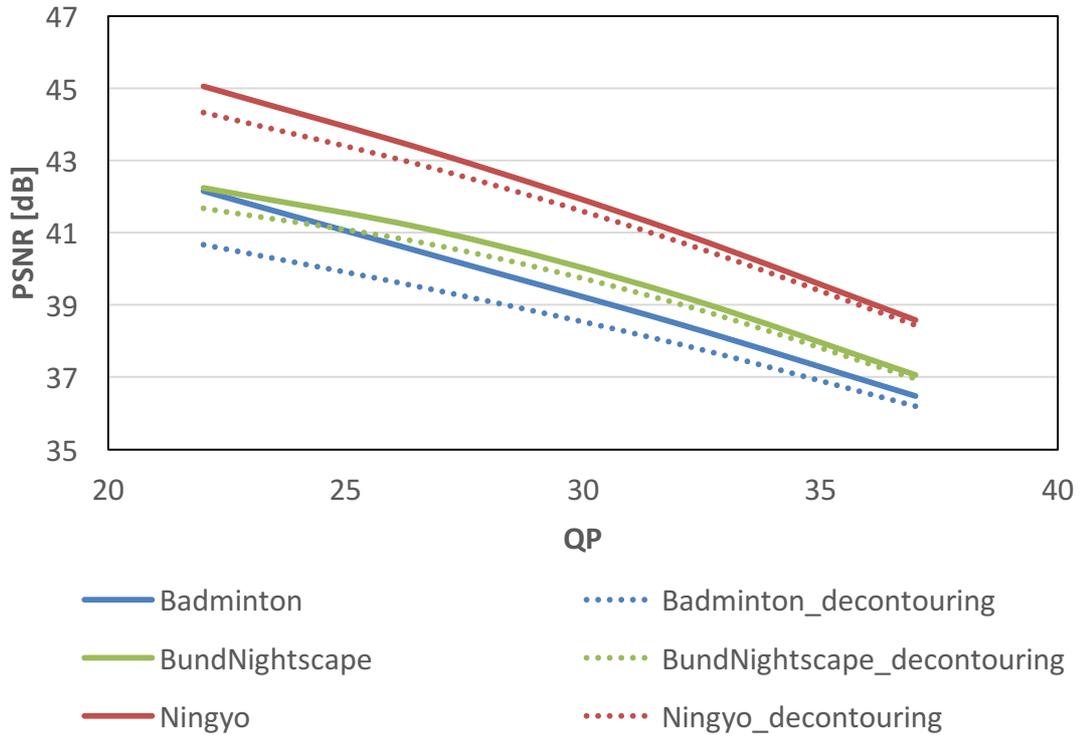
#### 4.6.1 Objective evaluation results

The results of objective evaluation methods PSNR and SSIM are shown in *Table 4.4*. The smaller value of PSNR and SSIM indicate the larger average pixel difference between the processed images and the original images. Although the contouring artefacts has been removed after applying proposed method, both PSNR and SSIM are reduced.

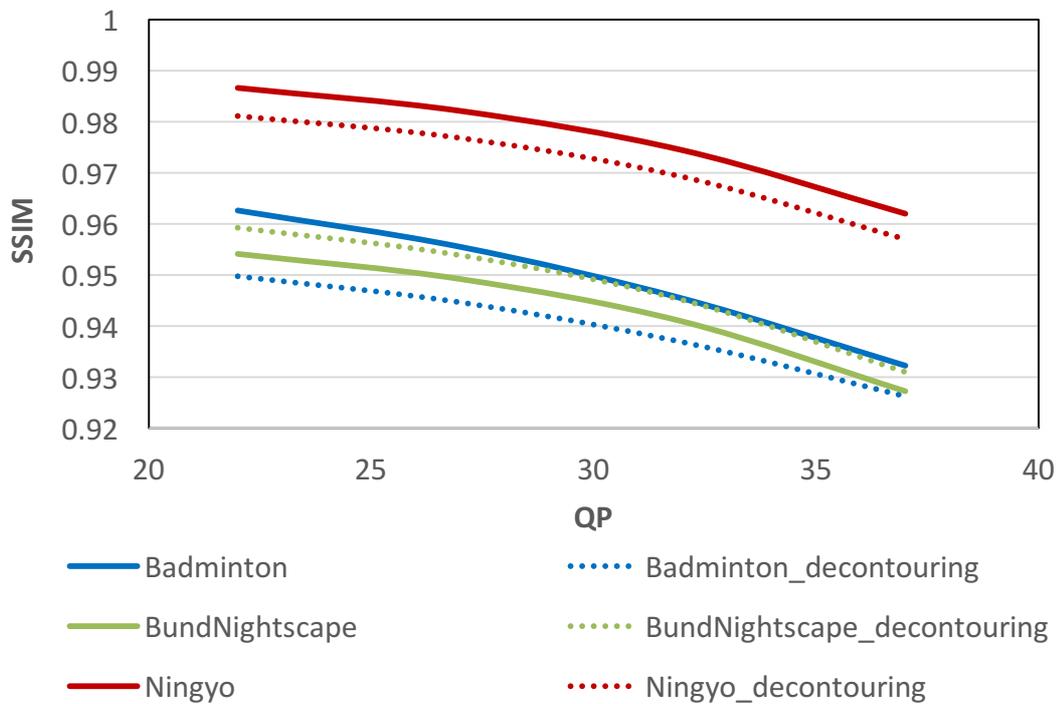
Table 4.4 Objective evaluation results.

Sequence	QP	PSNR (dB)		SSIM	
		HEVC	De-contouring	HEVC	De-contouring
Badminton	22	42.15	40.67	0.9626	0.9498
	27	40.31	39.38	0.9555	0.9447
	32	38.47	37.92	0.9454	0.9369
	37	36.47	36.19	0.9323	0.9263
Bund Nightscape	22	42.24	41.68	0.9541	0.9593
	27	41.02	40.63	0.9492	0.9539
	32	39.27	39.04	0.9409	0.9451
	37	37.07	36.96	0.9273	0.9310
Ningyo	22	45.06	44.33	0.9866	0.9811
	27	43.17	42.73	0.9821	0.9768
	32	41.02	40.76	0.9744	0.9692
	37	38.58	38.44	0.9620	0.9570

The corresponding graphs are shown in *Figure 4.17* and *Figure 4.18*.



*Figure 4.17 Objective metric PSNR results.*



*Figure 4.18 Objective metric SSIM results.*

According to the results listed in *Figure 4.15* we can see, the proposed method in this chapter visually improves the quality by removing the contouring artefacts. However, from the PSNR and SSIM results, the de-contoured pictures get larger average pixel differences, as shown in *Figure 4.17* and *Figure 4.18*. The objective evaluation methods are sometimes unreliable due to the fact that they use global or local block average information. It is necessary to conduct both objective and subjective evaluations to measure the video quality.

#### 4.6.2 Subjective results

The three sequences we used in the subjective experiments were selected so that they represent different features. The contouring features of three sequences are listed in *Table 4.5*. The contouring level rank is according to the subjects' feedback.

*Table 4.5 Test sequence contouring features.*

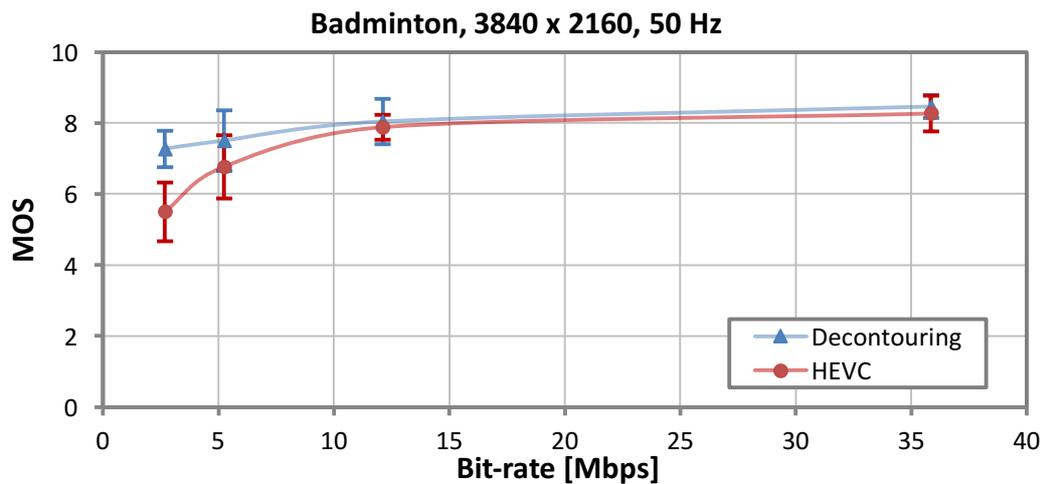
Sequences	Contouring Features	Contouring level
<i>Badminton</i>	Small area, moving camera	Low
<i>Bund Nightscape</i>	Small area, still camera	Low
<i>Ningyo</i>	Large area, still camera	High

The *Badminton* sequence has multiple small contouring regions. The camera is moving and there are fast motion objects (badminton players). According to the subjects' feedback, it is difficult to observe the contouring artefacts in small areas. More than 70% subjects believed the contouring artefacts in *Badminton* and *Bund Nightscape* sequences are harder to recognise compared to *Ningyo* sequences. There is relatively large area of contouring regions in *Ningyo* sequence (more than half area of each frame). The contouring artefact occurs even with lower QP values (i.e. QP = 22).

There were 35 subjects participated in the tests. The subjective evaluation results of the experiments are shown in *Table 4.6*. The MOS vs. Bit rate plots of three test sequences are shown in *Figure 4.19*. The points and triangles represent the actual results at each QP test point. The error bar indicates the confidence intervals of each MOS test point.

Table 4.6 Subjective evaluation results of the experiments.

Sequence	QPs	Bit-rate [Mbps]	MOS		
			HEVC	De-contouring	MOS difference
Badminton	22	35.8507	8.269	8.462	2%
	27	12.1076	7.885	8.038	2%
	32	5.2414	6.769	7.500	10%
	37	2.6834	5.500	7.269	24%
Bund Nightscape	22	9.1646	8.038	8.385	4%
	27	4.3163	6.615	8.423	21%
	32	2.3053	6.500	8.308	22%
	37	1.3198	5.846	7.962	27%
Ningyo	22	20.6505	5.577	8.846	37%
	27	7.9112	4.500	7.769	42%
	32	3.8982	4.269	7.731	45%
	37	2.0727	2.769	7.346	62%



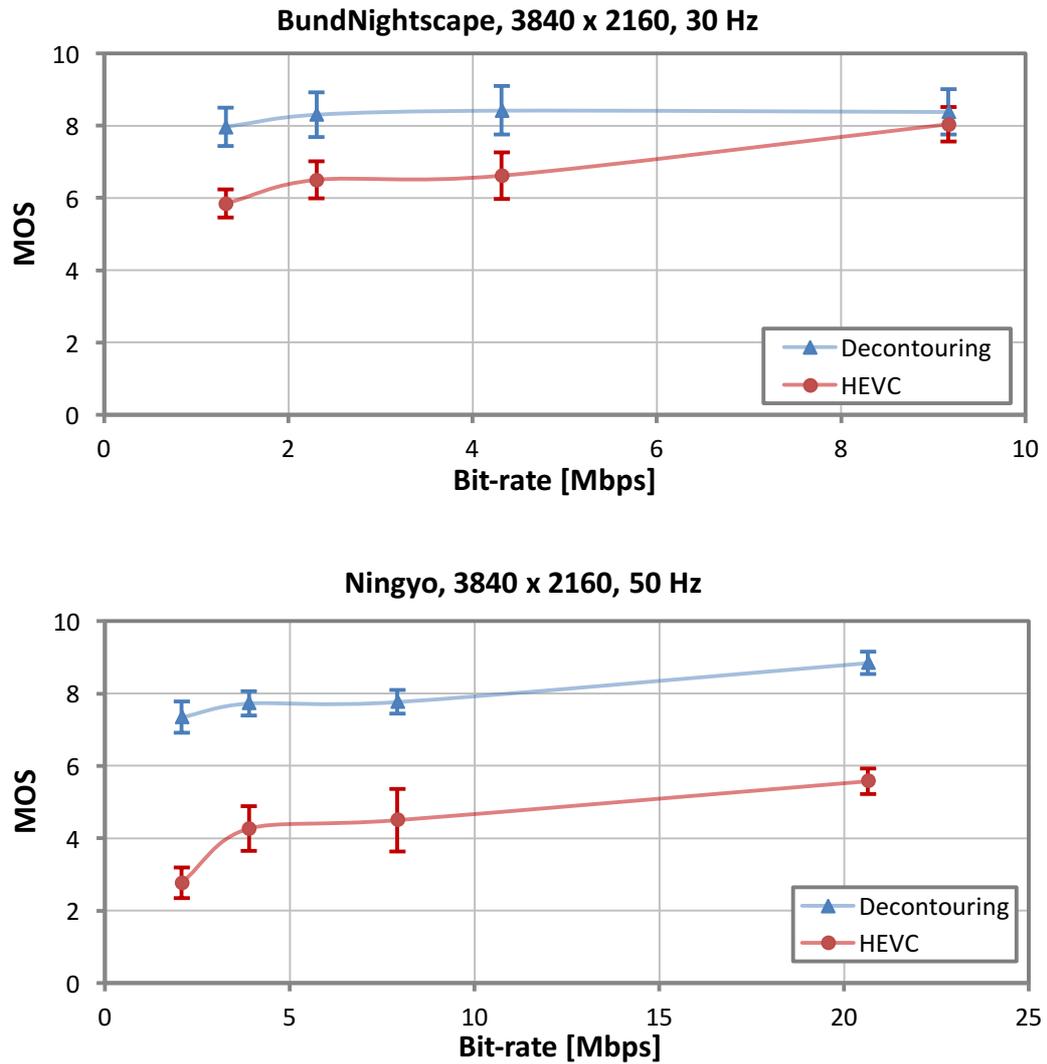


Figure 4.19 Subjective evaluation results.

The results show the proposed dithering method has higher MOS values at each test point. The improvement is larger at lower bit-rate, for example, from 2.77 to 7.25 for *Ningyo* sequence. For Badminton and Bund Nightscape sequences, the contouring artefacts are only found with higher QP values. It can also be seen from MOS difference in Table 4.6. The MOS difference at various bit rates is shown in Figure 4.20. For each test sequence, MOS difference is larger at lower bit rate point (or higher QP point). For example, the MOS difference is up to 62% for *Ningyo* sequence (QP=37).

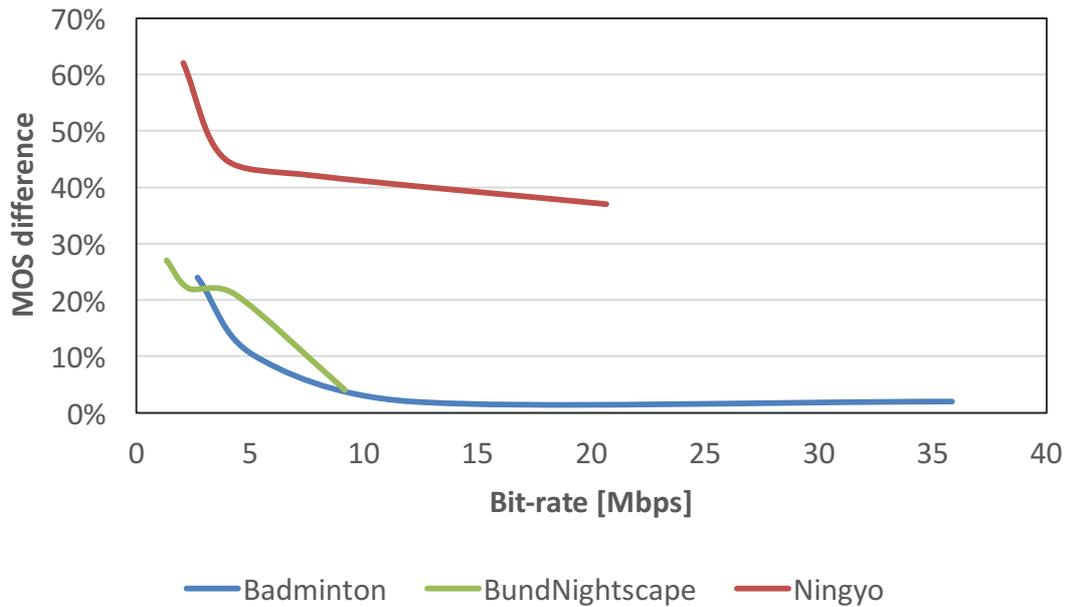


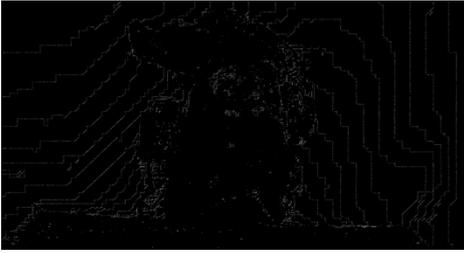
Figure 4.20 MOS difference at various bit rates.

## 4.7 Discussion

The dithering is a process of adding noise to adjust the pixels. The processed result of each step is listed in *Table 4.7*. The second column shows the corresponding noise map in each step. The white spots indicate noise. From the allocation of the noise map we can see that:

- The  $32 \times 32$  high frequency component dithering adds noise on over 80% pixels in contouring map. It is able to fade away contouring lines and make contouring bands less noticeable. However, it is not sufficient to remove all contouring bands.
- The  $4 \times 4$  dithering aims at dithering pixels in smaller contouring areas in the vicinity of real image edges and also breaking down any remaining contouring in the large contouring areas.
- Block average dithering adds noise on blocks near contouring lines. The “stair-case” indicates that the diagonal contouring lines are moved to the block boundaries after high frequency component dithering process.

Table 4.7 The processed result of each step and the corresponding noise map.

Processed frame	Noise map
<p>1. HFC dithering (N=32)</p> 	
<p>2. Block average dithering (N=32)</p> 	
<p>3. HFC dithering (N=4)</p> 	
<p>4. Block average dithering (N=4)</p> 	

The pixel profile of a part of a selected horizontal line in the contouring area in the original frame, compressed frame and de-contoured frame of the *Ningyo* sequence are drawn in *Figure 4.21*. It is noted that the proposed dithering lessens the “staircase” effect and thus reduces the contouring artefacts effectively.

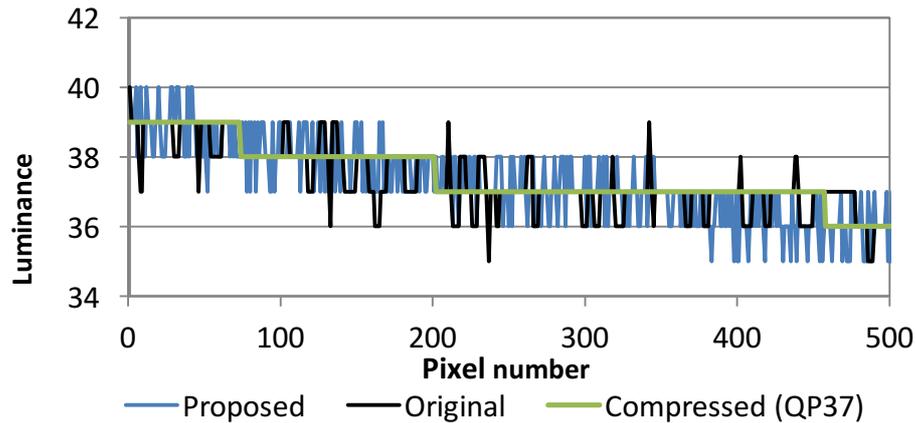


Figure 4.21 The horizontal profile (part) of the luminance signal for the selected line draws from Ningyo sequence.

The post-processing contouring artefacts removal methods can successfully improve the compressed UHD content. The application after HEVC codec means the multimedia display devices need the additional function in the decoder. However, it is also important to prevent artefacts during video compression. A novel design of contouring prevention application is needed to apply at HEVC encoder while keep the decoder unchanged, which motivated the study in the next chapter (Chapter 5).

## 4.8 Summary

This chapter proposed a contouring detection method and a contouring removal dithering method for UHD content. It was applied as a post-processing method on HEVC compressed UHD video sequences which already have contouring artefacts. Both objective and subjective evaluation methods were applied in the experiments. The results showed the proposed method successfully removed the contouring artefacts and improved the overall visual quality. The next chapter proposes contouring prevention methods applied as pre-codec and in-codec approaches.

## Chapter 5 Contouring artefacts prevention for UHD content in HEVC

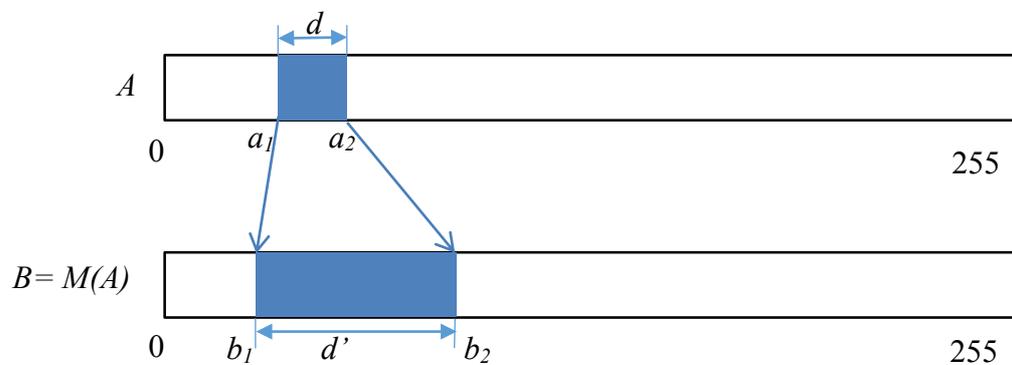
Chapter 4 detailed the proposed pixel dithering method for contouring removal in compressed UHD sequences. This chapter proposes pixel mapping methods to reduce contouring artefacts for UHD content in HEVC. The proposed contouring reduction methods use content adaptive pixel mapping algorithm. In pixel mapping, some pixels were expanded while others were compressed. The proposed algorithm can be implemented as a pre-codec (Approach 2 in *Figure 4.2*) or in-codec (Approach 3 in *Figure 4.2*). Both approaches were implemented in the experiments.

### 5.1 Pixel mapping algorithm

This section includes pixel mapping, inverse pixel mapping and related parameters setup.

#### 5.1.1 Pixel mapping

The structure of pixel mapping algorithm is shown in *Figure 5.1*.



*Figure 5.1 The structure of pixel mapping.*

For 8-bit signals, pixel values range from 0 to 255. In *Figure 5.1*,  $A$  represents the original pixels,  $B$  represents the according pixels after mapping. The mapped pixels values range is the same as that in the original pixels, which is from 0 to 255. The darker interval indicates the contouring range pixel values are expanded into larger range. The parameter of expanded scale is *multifactor* ( $n$ ), which indicates how many times the contouring pixel range has been expanded. All the symbols used in this chapter are described in *Table 5.1*.

*Table 5.1 Symbols and descriptions.*

<b>Symbol</b>	<b>Description</b>
$A$ :	<i>original pixel value</i>
$B$ :	<i>mapped pixel value</i>
$a_1$ :	<i>threshold values (minimum)</i>
$a_2$ :	<i>threshold values (maximum)</i>
$b_1$ :	<i>mapped value of <math>a_1</math></i>
$b_2$ :	<i>mapped value of <math>a_2</math></i>
$d$ :	<i>width of pixel range (<math>a_2 - a_1</math>)</i>
$d'$ :	<i>width of pixel range after Pixel Mapping (<math>b_2 - b_1</math>)</i>
$P$ :	<i>peak value of histogram of contouring area</i>
$M ()$ :	<i>pixel mapping function</i>
$M^{-1} ()$ :	<i>inverse pixel mapping function</i>

Two threshold values  $a_1$  and  $a_2$  separate all pixel values into three intervals:  $0 \sim a_1$ ,  $a_1 \sim a_2$ ,  $a_2 \sim 255$ . The pixels ranging from  $a_1$  and  $a_2$  are selected as the contouring pixels. The width of this range is  $d$  ( $d = a_2 - a_1$ ). In most cases, the values of pixels in contouring areas are in a relatively small range. This is because the contouring artefact occurs in smooth gradient areas where the neighbouring pixels differ slightly. It can be seen from the histogram of the contouring pixels of each test sequence in *Figure 5.2*. In the histogram, x-axis represents the pixel values from 0 to 255 (for 8 bits signal), with the frequency counts of each pixel value showing in the y-axis.

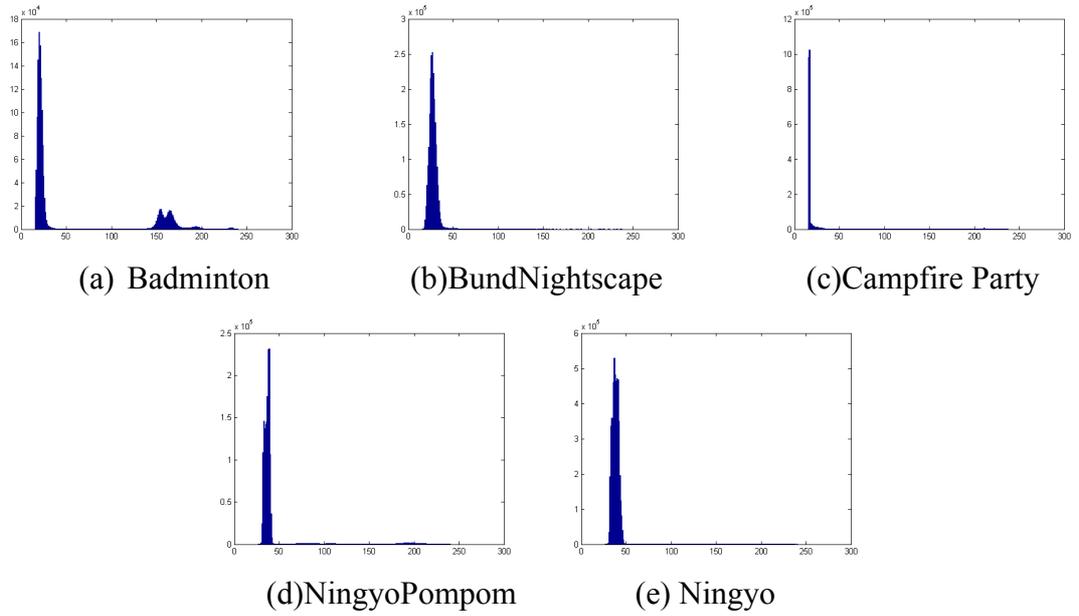


Figure 5.2 Histogram of the pixels in the contouring regions (X axis: pixel value; Y axis: number of pixels).

The threshold values  $a_1$  and  $a_2$  are computed as:

$$a_1 = P - \frac{d}{2}; \quad (5.1)$$

$$a_2 = P + \frac{d}{2}. \quad (5.2)$$

$P$  is the peak value of histogram of contouring area, and  $d$  is width of the contouring pixel range. The peak value of the four test sequences is shown in *Table 5.2*. It indicated that the contouring artefacts are more likely to occur in darker area (small luma pixel values). Width  $d$  is a parameter which can be set in the experiments.

Table 5.2 Peak pixel value of histogram in contouring area.

Sequence	Peak value
Badminton	20
Bund Nightscape	27
Campfire Party	17
Ningyo Pompoms	39
Ningyo	37

The contouring pixel value range width  $d = a_2 - a_1$ , which is then expanded after pixel mapping:

$$d' = n \times d = b_2 - b_1, \quad (5.3)$$

where  $n$  ( $n \geq 1$ ) is the multifactor,  $b_2$  and  $b_1$  represents the minimum and maximum values of contouring regions after pixel mapping, respectively. Pixels, which equal to  $a_1$  in  $A$ , are converted to  $b_1$  in  $B$  after non-uniform pixel mapping. The pixel mapping algorithm for each pixel value range is listed in Table 5.3.

Table 5.3 The pixel mapping algorithm.

$A$	$B = M(A)$
$< a_1$	$A \times \frac{255 - d'}{255 - d}$
$a_1 \sim a_2$	$b_1 + (A - a_1) \times n$
$> a_2$	$255 - \left( (255 - A) \times \frac{255 - d'}{255 - d} \right)$

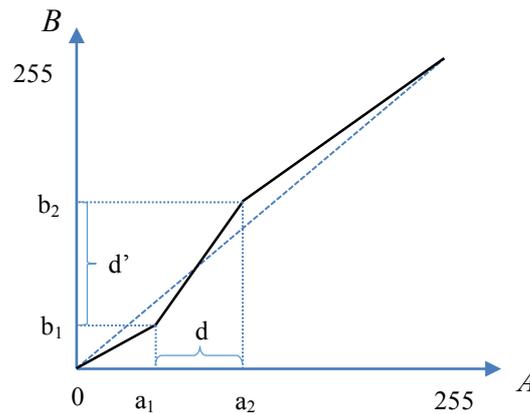
The minimum value of contouring value range after pixel mapping is computed as:

$$b_1 = (a_1 - 1) \times \frac{255 - d'}{255 - d} + 1. \quad (5.4)$$

And the related maximum value is derived as:

$$b_2 = b_1 + n \times d. \quad (5.5)$$

The corresponding relation of pixel values before and after mapping is plotted in *Figure 5.3*. The dash line indicates the same value as the original pixel and the solid line represents the proposed pixel mapping algorithm.



*Figure 5.3 The correlation of pixel values before and after reconstruction.*

The solid line represents the proposed pixel mapping algorithm in this chapter. The dash line ( $B=A$ ) is used as reference with all pixels unchanged.

The plot is continual and not overlapped, so the conversion is reversible. However, the pixel mapping process has slightly loss due to the rounding errors.

### 5.1.2 Inverse pixel mapping

The inverse pixel mapping is applied on the output sequences after the decoder procedure in some methods, e.g. pixel mapping pre-codec which detailed in Section 5.2.1. Since the pixel mapping is reversible and the reconstructed pixel ranges are not overlapped, the inverse pixel mapping after decoder does not need the contouring detection map. The conversion of the inverse reconstruction algorithm is derived as in *Table 5.4*.

Table 5.4 The pixel inverse mapping.

$Y'$	$Y$
$< b_1$	$Y' \times \frac{255 - d}{255 - d'}$
$b_1 \sim b_2$	$a_1 + (Y' - b_1)/n$
$> b_2$	$255 - ((255 - Y') \times \frac{255 - d}{255 - d'})$

The inverse pixel mapping is applied for all pixels in the frame. Two threshold values  $b_1$  and  $b_2$  divided the pixel range into three parts, as shown in *Figure 5.1*, where each pixel is inversely converted according to the catalogue range in *Table 5.4*.

### 5.1.3 Parameters setup

There are two parameters to be set in the above pixel mapping and inverse pixel mapping processes: width  $d$  and multifactor  $n$ .

The difference values of parameter width  $d$  have difference pixel mapping range. The larger pixel mapping range can coverage more pixels for mapping process to avoid occurring contouring. However, it also increases the bitrate according to our test results in *Table 5.5*. Positive values indicate BD-rate increase. Also, from the histogram of each test video sequences we can see that the contouring pixels are allocated in a relatively small range.

The multifactor  $n$  is another parameter set in the experiments. It is a trade-off between the frame quality improvement and the overall bit-rate increase. The larger multifactor increases the pixel difference. Thus more high frequency information are added in gradient smooth area, which is more likely to retain after quantization. However, this also forces the further block partitioning in encoder, which results in the significantly bitrate increase, as shown in *Table 5.5*. The challenge is to set the suitable value of multifactor which can maximumly avoid the contouring artefacts but not raising the bit rate too much.

Table 5.5. BD-rate results of Badminton sequence with different parameter setups.

width	Multifactor					
	1.5	2	2.5	3	3.5	4
3	0.08%	0.65%	1.12%	4.27%	6.9%	9.01%
5	0.23%	3.44%	6.33%	11.48%	19.97%	29.90%
7	0.51%	5.03%	10.14%	19.66%	34.02%	53.06%
9	1.70%	5.58%	13.76%	28.11%	47.08%	72.46%

## 5.2 Implementation

The previous proposed de-contouring dithering algorithm is applied on the compressed frame as a post processing approach. The compressed frames usually suffer contouring artefacts in the smooth gradient area. The contouring regions can be easily detected and the contouring lines can be removed. However, the non-uniform pixel mapping we discussed in this chapter is an approach to apply on the input original video sequences which not yet present the contouring bands. The contouring detection in this chapter is detecting the regions which are highly likely to occur the contouring artefacts. The contouring detection algorithm was introduced in Section 4.3.

From the results shown in *Figure 5.4* we can see, the contouring maps generated from the original sequences and the decoded sequences are very similar. As the threshold value of contrast set as 1, the two contouring maps result in the same.

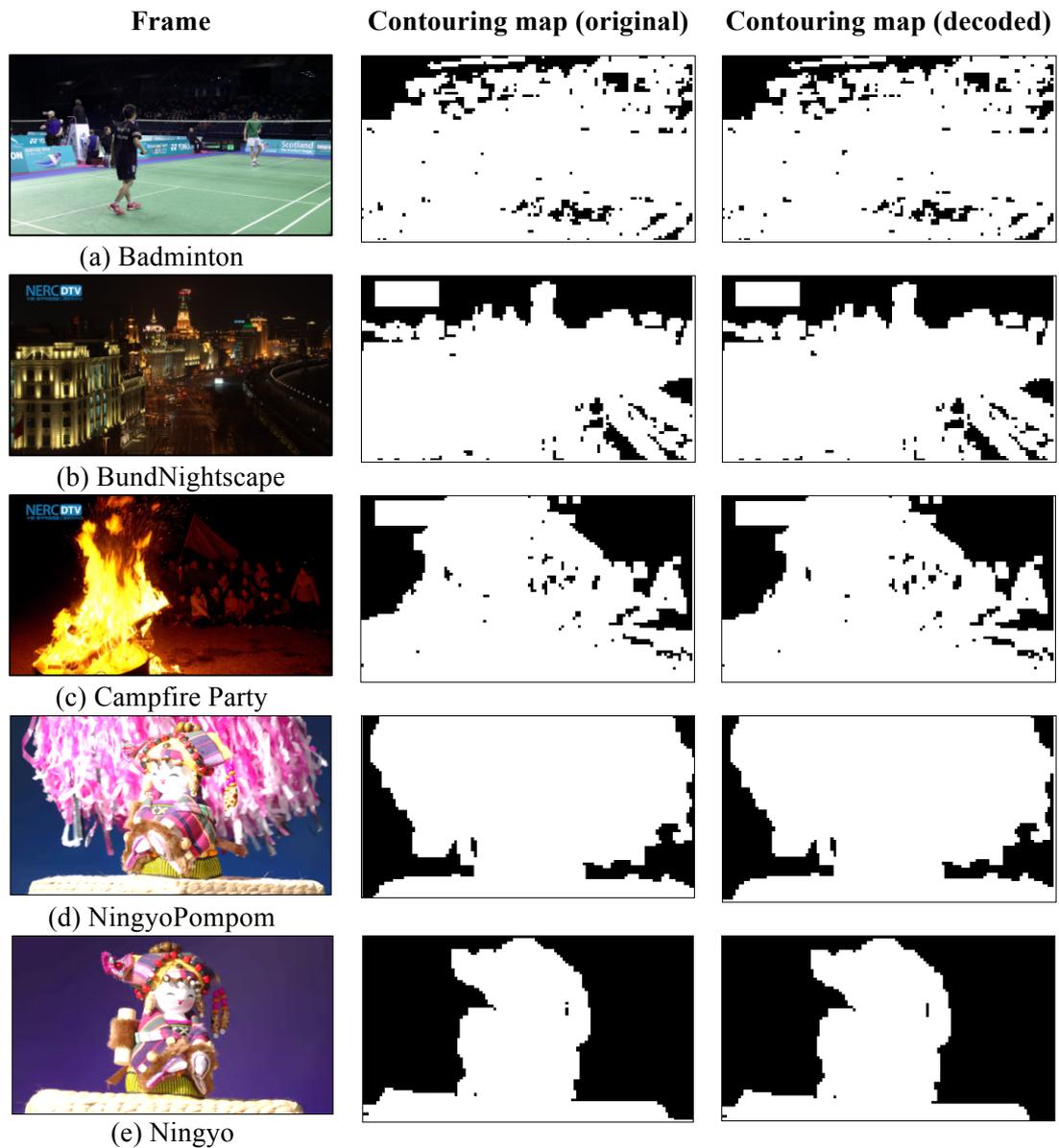


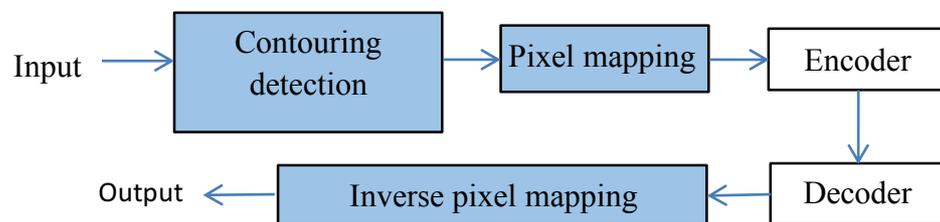
Figure 5.4 Contouring maps of original and decoded frames.

Another challenge is the segmentation of contouring and non-contouring regions in HEVC generates new artefacts. In the previous de-contouring dithering algorithm, contouring area are separated from the rest by using contouring detection, which also shows in a different colour in the contouring map. The de-contouring dithering algorithm only applied on the contouring area while the non-contouring area are kept unchanged. However, non-uniform pixel mapping cannot be applied only in contouring area. Firstly, it generated noticeable discontinue lines at the contouring region boundaries. Secondly, it interrupted intra and inter prediction which increases the bit-rate, especially the intra

prediction for I frames which raises the bit-rate relatively high. Thirdly, the same contouring map needed to be sent to the decoder for inverse pixel mapping. By considering all of these disadvantages, contouring region segmentation is discarded but the contouring pixel value range segmentation is applied in this algorithm. It means the whole process of pixel reconstruction depends on the original value of the pixel rather the location of it. All pixel values from 0~255 are separated into three intervals where each has its own direct and inverse pixel mapping formula.

### 5.2.1 Pixel mapping pre-codec

The overview procedure of the proposed pre-codec pixel mapping method is shown in *Figure 5.5*. Before the encoder, input video sequences were transformed by reconstructing the pixel values. The compressed sequences were then converted by using the inverse pixel mapping after decoder. Based on our experiments, the contouring artefacts mainly come from luminance component. Thus the experiments in this chapter are implemented on the luminance component only, although this algorithm can be expanded into two chrominance components as well.



*Figure 5.5 Overview of proposed pre-codec method.*

### 5.2.2 Pixel mapping in-codec

The proposed method can also be implemented as an in-codec process integrated in HEVC codec. There are two methods detailed in this section which implement proposed pixel mapping algorithm in HEVC. *Figure 5.6* shows the block diagram of HEVC. The pixel mapping in-codec process add proposed algorithm within HEVC codec.

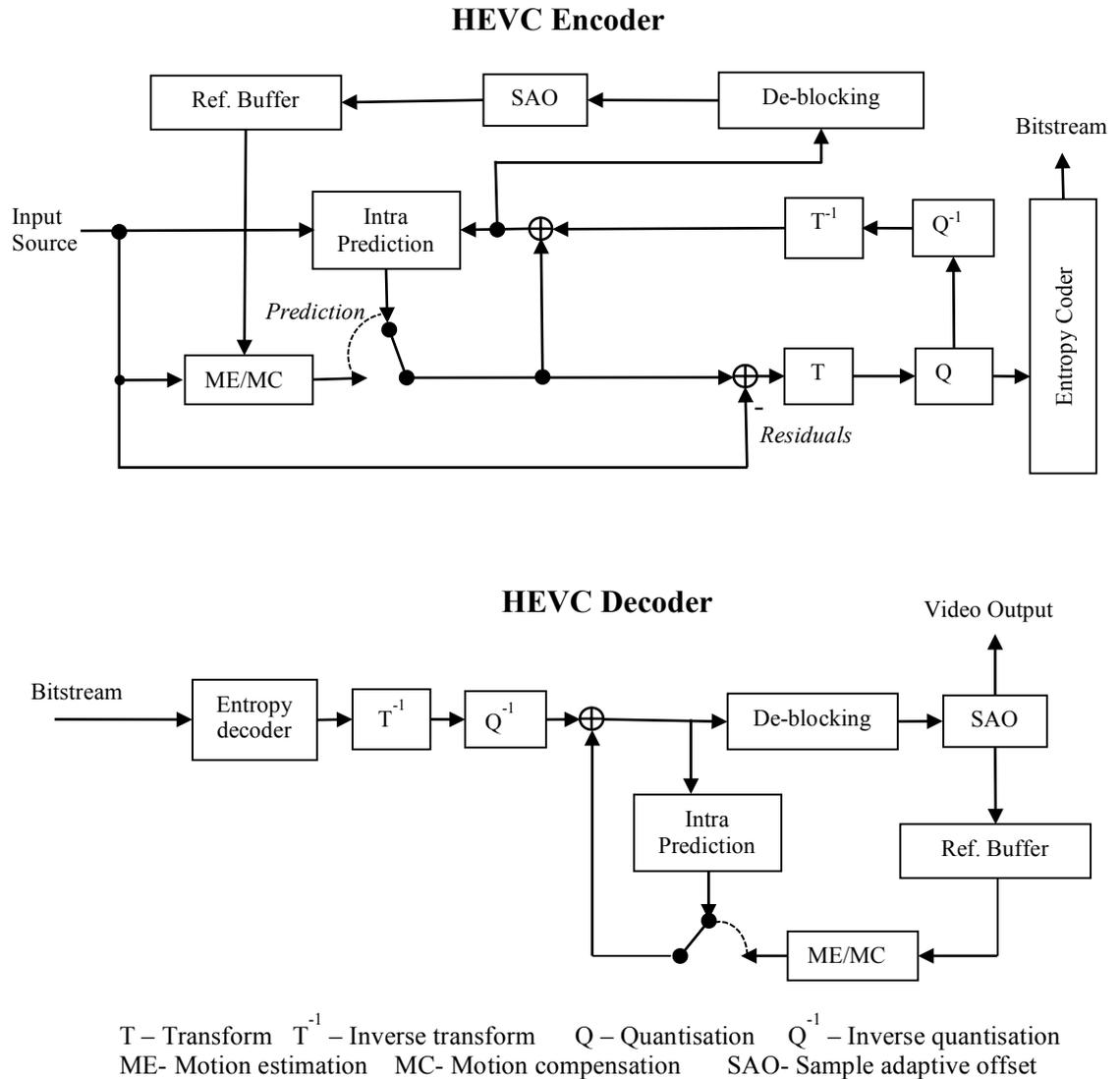


Figure 5.6 Block diagram of HEVC.

One method applies pixel mapping before transform, and inverse pixel mapping after inverse transform. Both original and prediction pixels are converted in a mapped domain where more contouring pixels are expanded into larger value range. In this way, the details in smooth gradient area are more like to remain after quantisation, which prevents the contouring artefacts. This method is named *Pixel Mapping In-codec –A* (PMIC- A) in this thesis.

The other method is also applied an in-codec approach. It computes residual by using the mapped original and predicted pixels. At the decoder, the residual is added to prediction

to reconstruct the pixel. The advantage of using this method is that the HEVC decoder is unchanged. This method is named *Pixel Mapping In-codec -B* (PMIC- B) in this thesis.

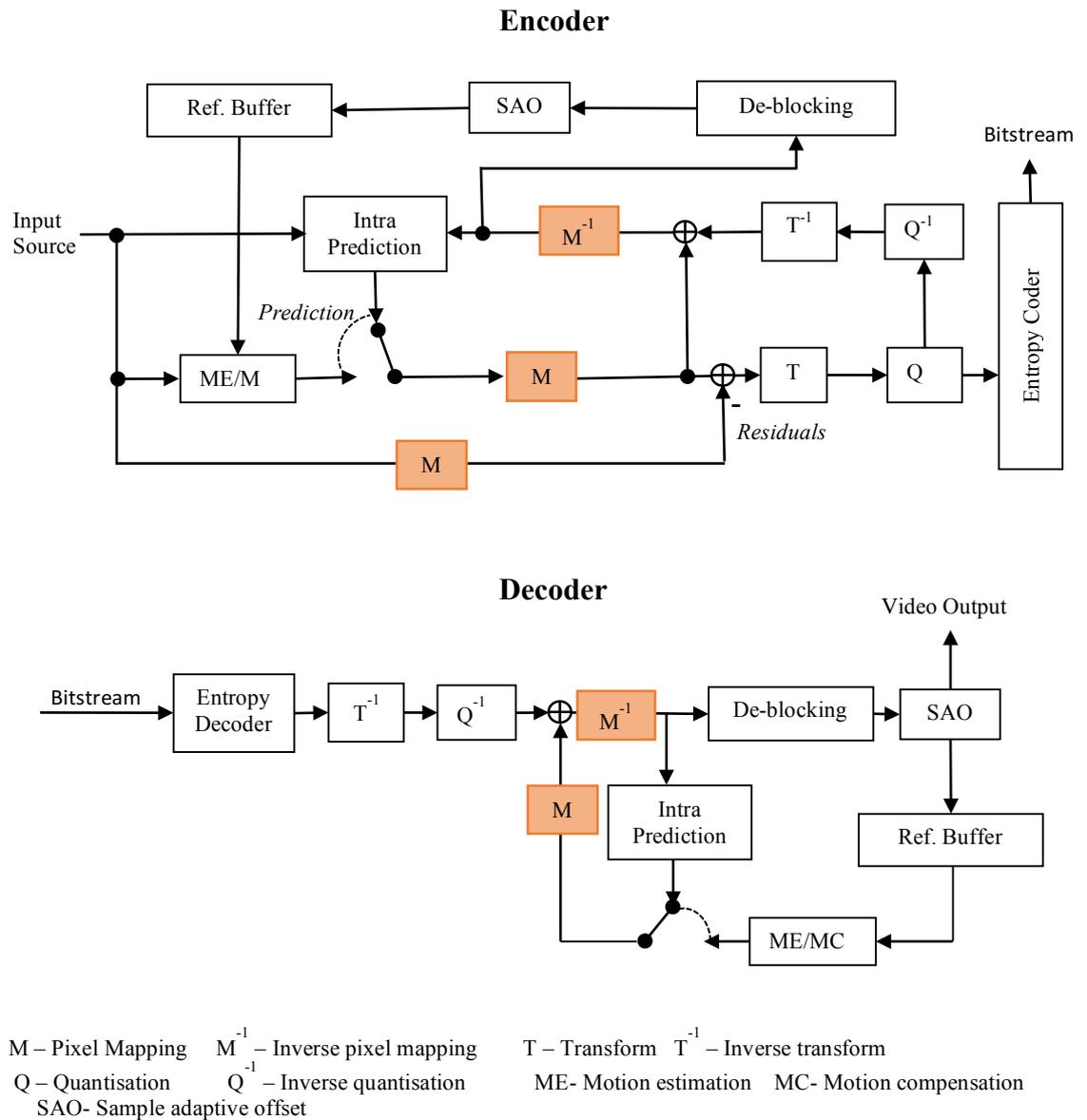


Figure 5.7 Block diagram of proposed PMIC- A method.

### 5.2.2.1 Pixel Mapping In-Codec -A (PMIC-A)

The block diagram of the proposed method PMIC-A is shown in Figure 5.7. There are three blocks added to the proposed method: pixel mapping on original pixels, pixel mapping on the predicted pixels and inverse pixel mapping at reconstructed pixels.

After intra or inter prediction, the pixel mapping algorithm is applied on both original and the predicted pixels. The residual  $R$  is then computed as:

$$R = M(I_{org}) - M(I_{pred}), \quad (5.6)$$

where  $M$  is the function of proposed pixel mapping algorithm,  $I_{org}$  is the original pixel, and  $I_{pred}$  represents the predicted pixel.

After transform, quantisation, inverse quantisation and inverse transform, the reconstructed residual is represented as  $R'$ .

At the decoder, the reconstructed pixel  $I'$  is computed as:

$$I' = M^{-1}(R' + M(I_{pred})), \quad (5.7)$$

Where  $M^{-1}$  is the function of inverse pixel mapping algorithm. The  $M$  and  $M^{-1}$  functions are using the equations listed in *Table 5.3* and *Table 5.4*, respectively.

### 5.2.2.2 Pixel Mapping In-Codec – B (PMIC-B)

The encoder and decoder block diagram of the proposed PMIC-B method are shown in *Figure 5.8*. The residual is computed the same as that in PMIC-A. The reconstructed pixel at decoder is computed as:

$$I' = R' + I_{pred}, \quad (5.8)$$

PMIC-B only modifies the HEVC encoder while the decoder is the same as that in the HEVC decoder.

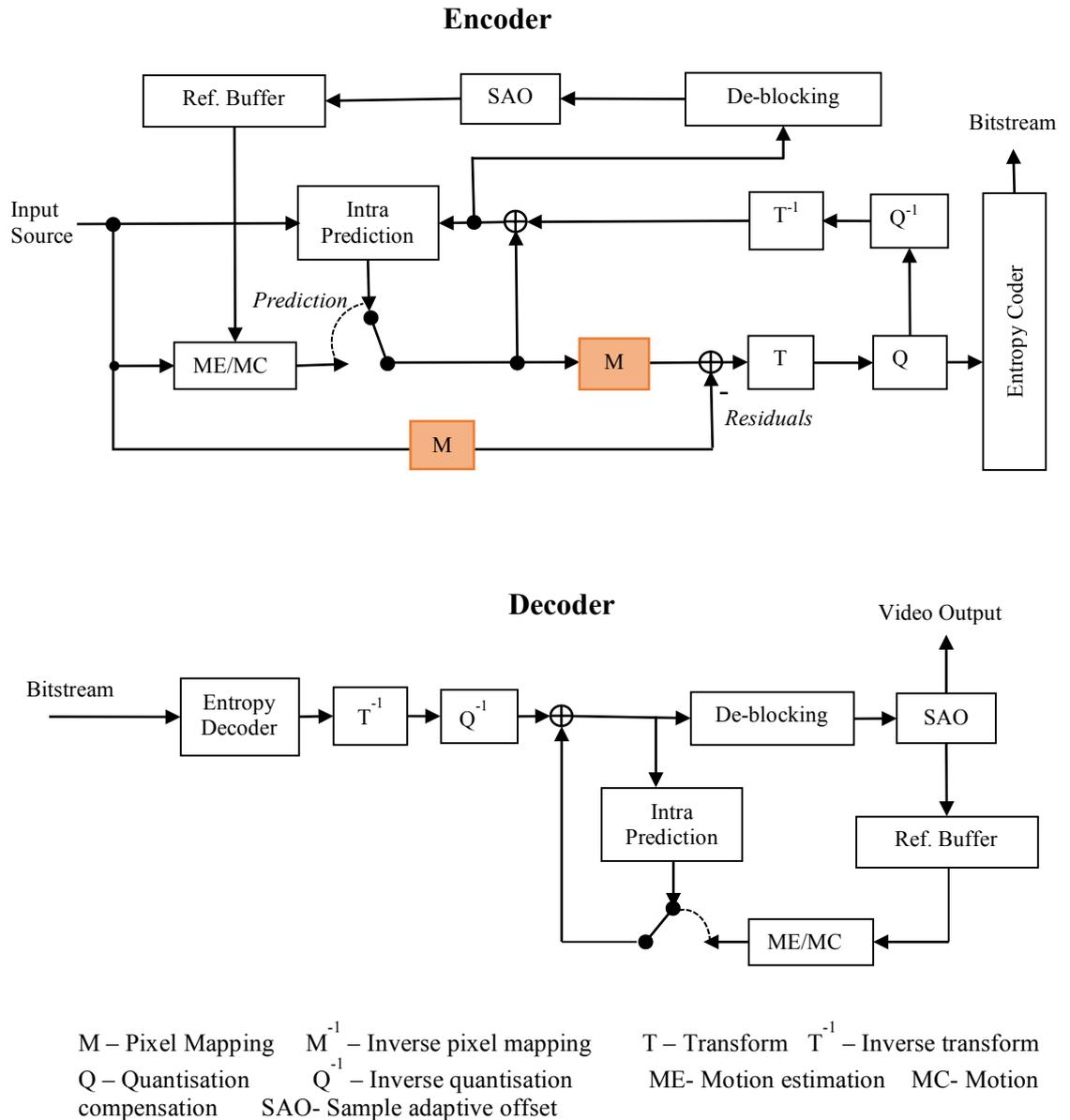


Figure 5.8 Block diagram of proposed PMIC-B method.

### 5.3 Performance evaluation

In this section, the proposed methods are evaluated by using the subjective evaluation method. The methodology for carrying out visual quality evaluation and the experiment setup are described in Section 4.6.2. The HEVC test conditions and parameters for pixel mapping algorithm are listed in *Figure 5.5*.

Table 5.6 Test conditions and parameter setup.

HEVC test model	HM 16.0
Configuration	Random access
QPs	22, 27, 32, 37
Intra period	32 for 30 fps *(Bund Nightscape); 48 for 50 fps (Badminton and Ningyo);
Multifactor	1.5
Width	7

\*fps= frames per second

## 5.4 Experimental results and analysis

This section includes the subjective evaluation results. The results are compared with objective evaluation PSNR results. It also analyses the MOS BD-rate and PSNR BD-rate.

### 5.4.1 Subjective test results

In total, there were 35 subjects participated in the experiments within seven groups. The subjective test results MOS vs. Bit rate plot of all three test sequences are shown in *Figure 5.9* and *Figure 5.10*. The points and triangles represents the actual results at each QP test point. The error bar indicates the confidence intervals of each MOS test point. The subjective evaluation results MOS BD-rate of proposed pixel mapping methods compared to decontouring method are shown in *Figure 5.11* and *Table 5.7* by using Bjøntegaard-Delta bit rate (BD-rate) computation [53].

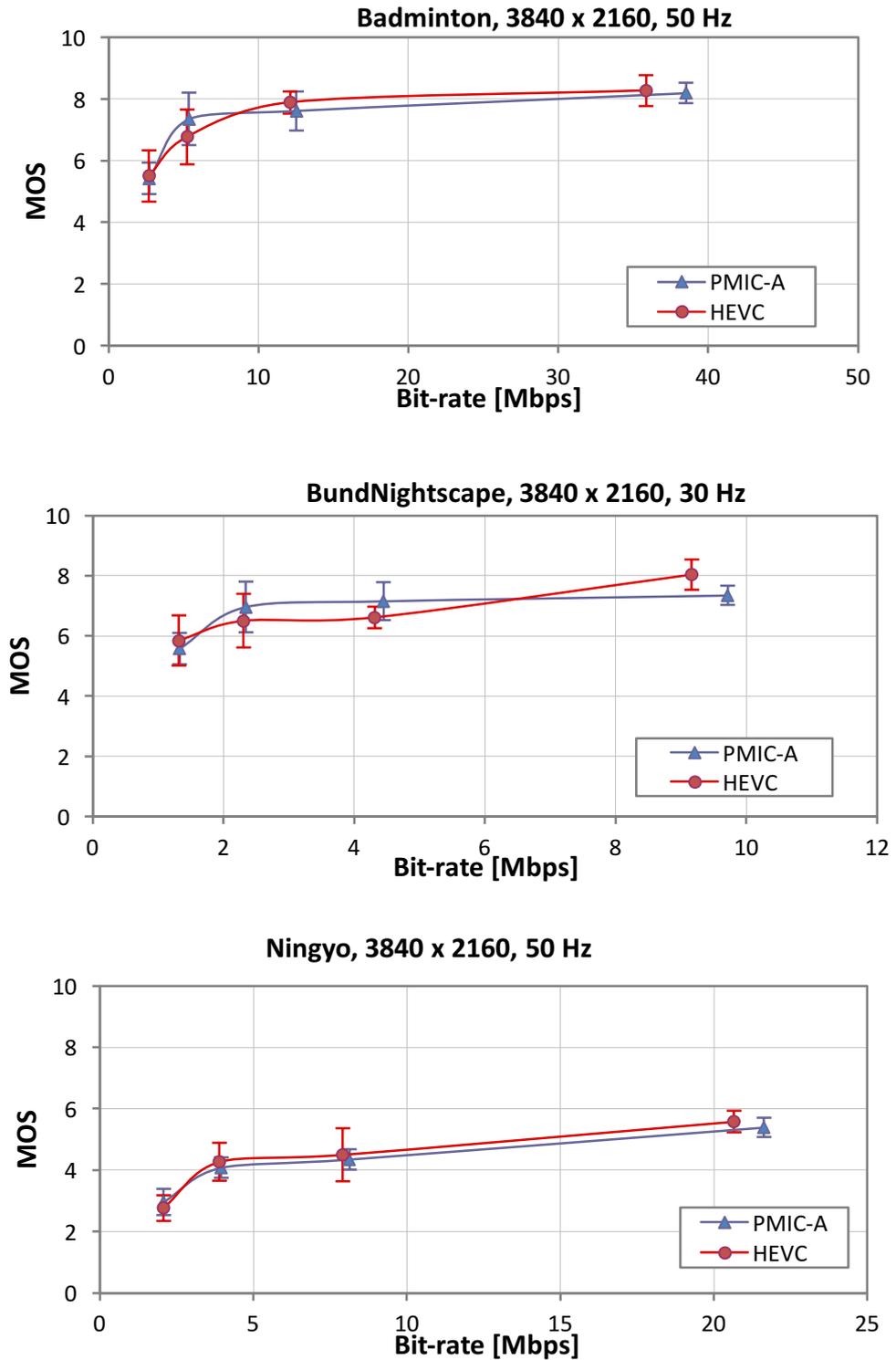


Figure 5.9 Subjective test results of proposed PMIC-A method

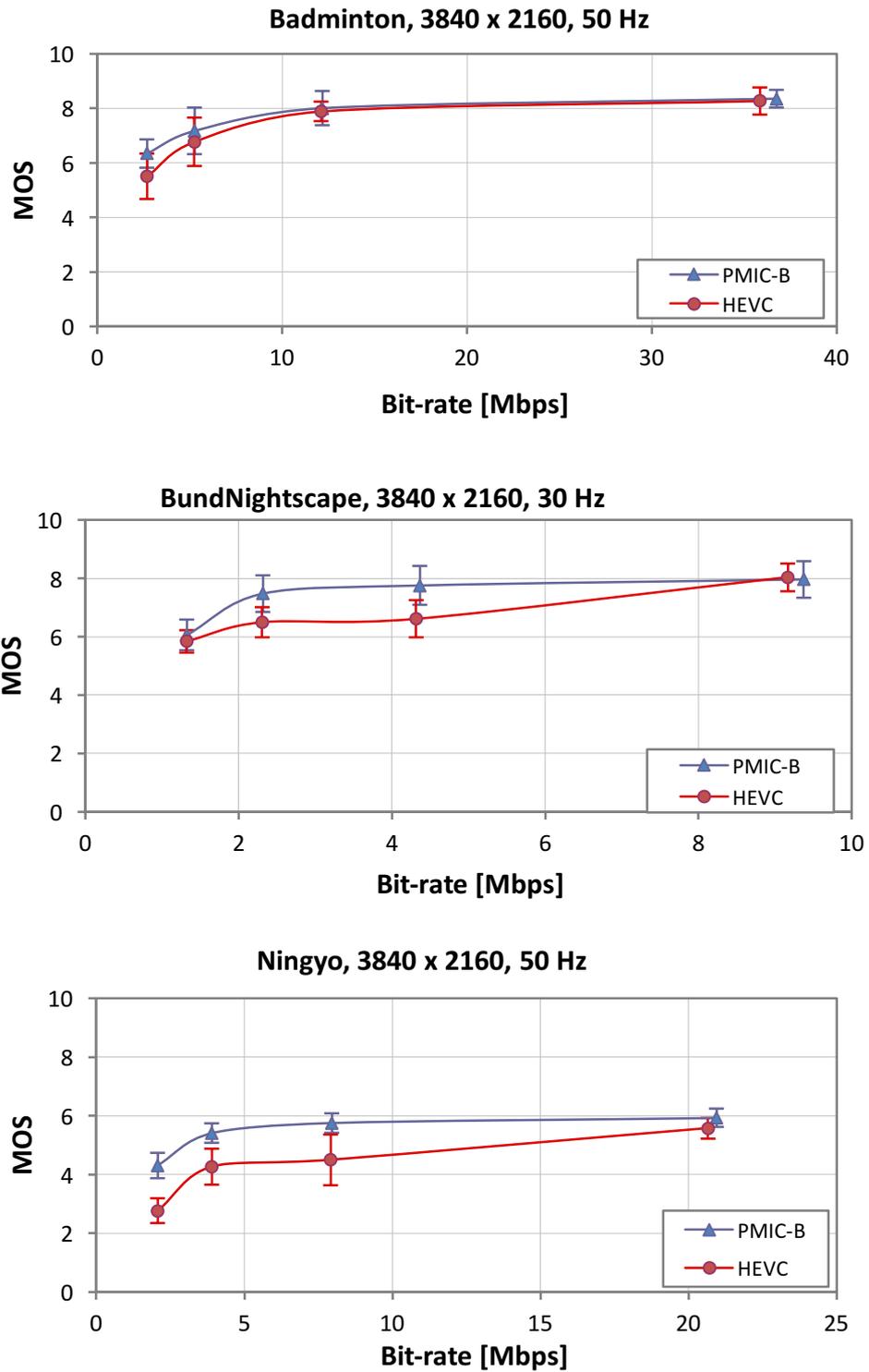


Figure 5.10 Subjective tests result of proposed PMIC-B method

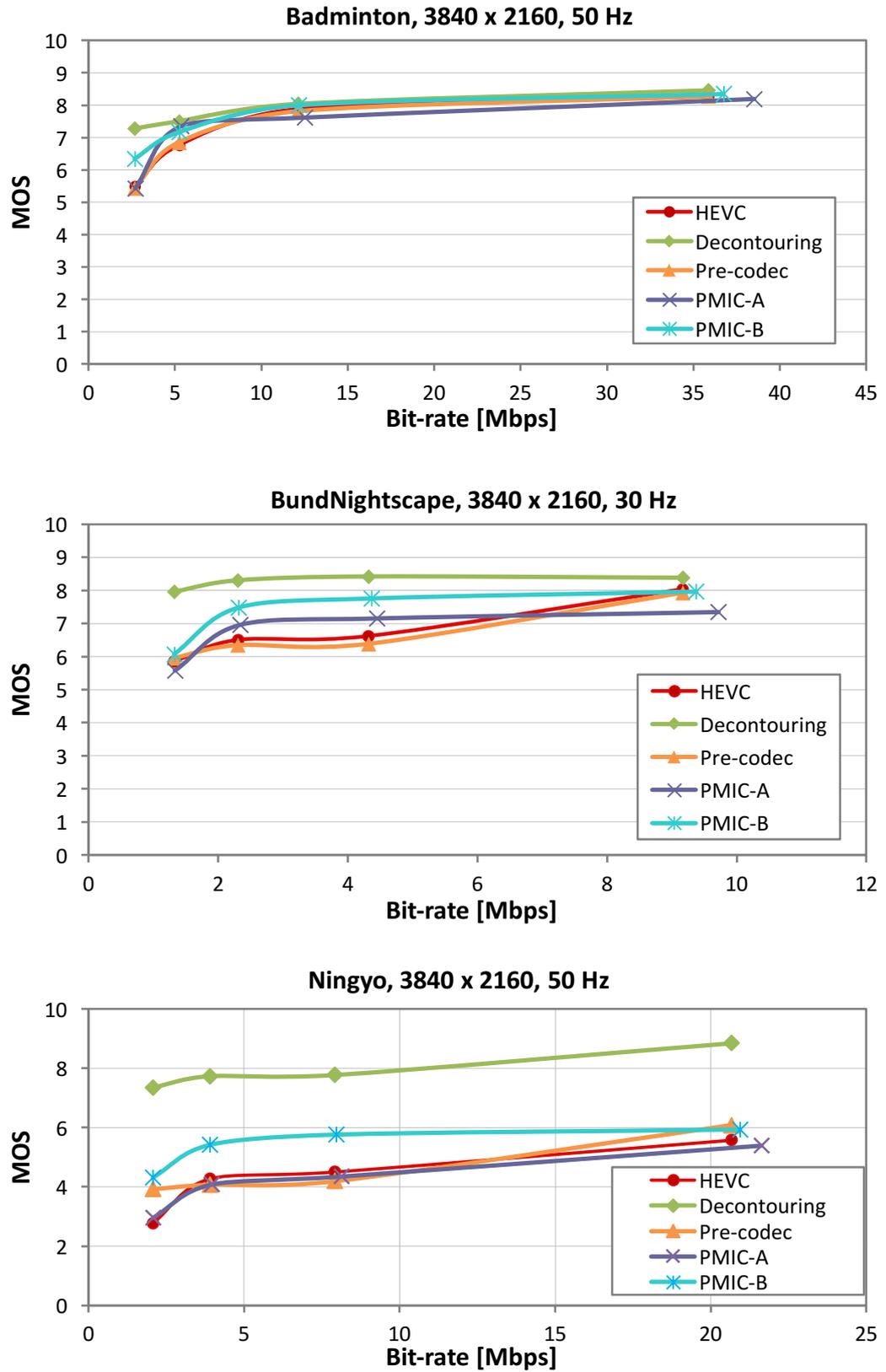


Figure 5.11 Subjective evaluation results.

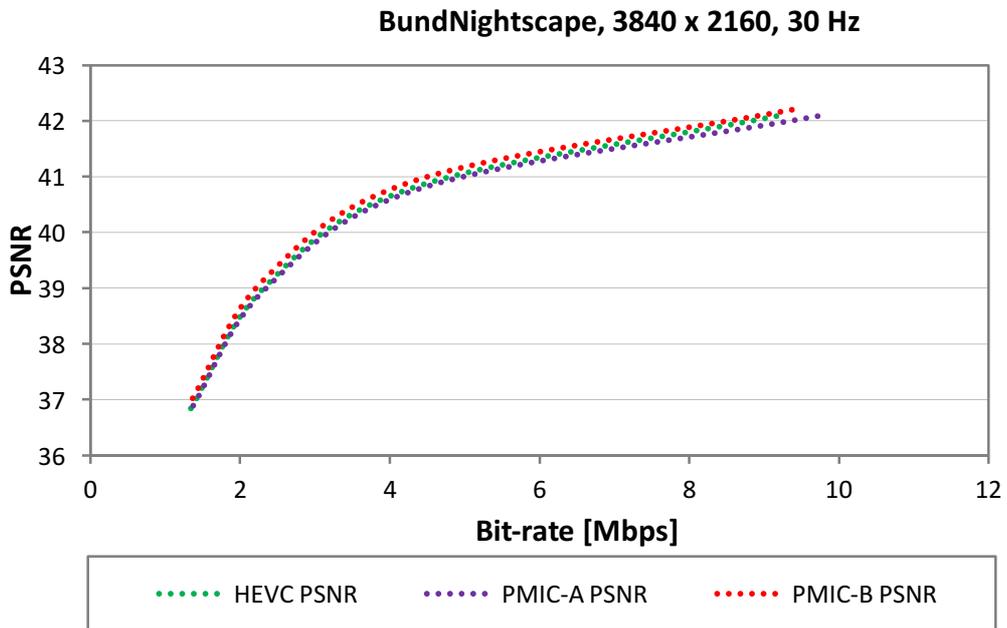
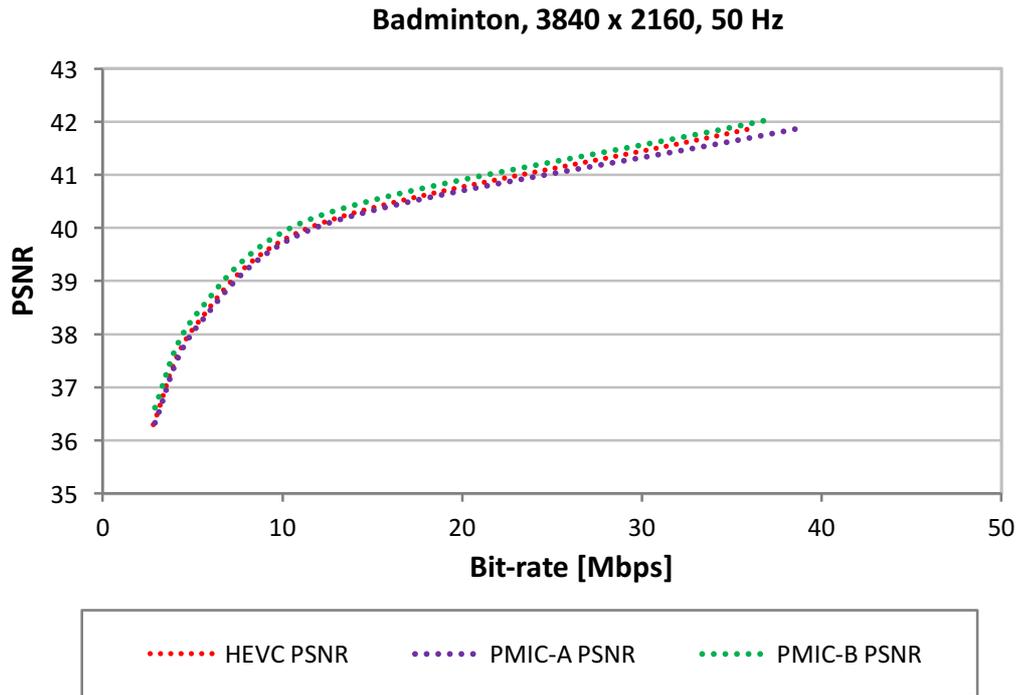
Table 5.7 Subjective test results MOS BD-rate.

	Sequences		
	Badminton	Bund Nightscape	Ningyo
De-contouring	-34%	-85%	N/A
Pre-codec	1%	19%	21%
PMIC- A	-5%	-34%	17%
PMIC- B	-23%	-59%	-78%

Both MOS vs. Bit rate curves and MOS BD-rate results show the video quality improvement by using the proposed pixel mapping methods. Unlike de-contouring method which remove most of the contouring lines, pixel mapping method aims to prevent and reduce the contouring artefacts during compression. Pixel mapping applied within HEVC has better performance than applying it as pre-codec processing. In Pre-codec processing approach, the pixels are mapped at frame level. Each pixel is rounded into nearest integers. Pixel mapping in-codec methods are applied at block level. The adjusted block can be used as reference to predict other blocks. In this way, the adaptive parameters are more accurate.

#### 5.4.2 Objective and subjective metrics comparisons

This section compares the objective and subjective test results of each test sequence. The comparisons of RD-curves are shown in *Figure 5.12* and *Figure 5.13*.



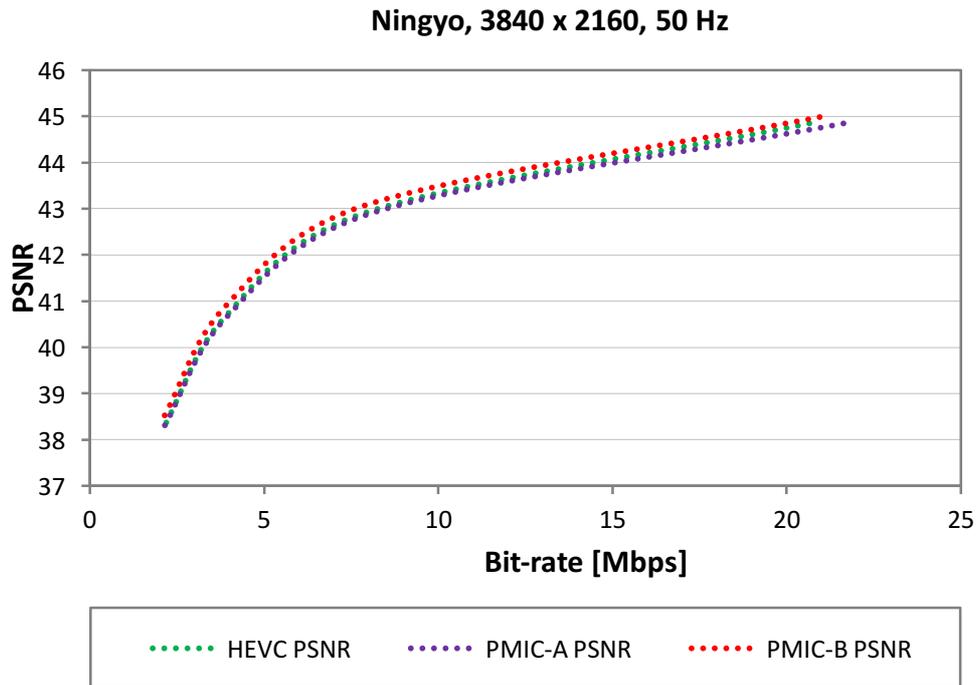
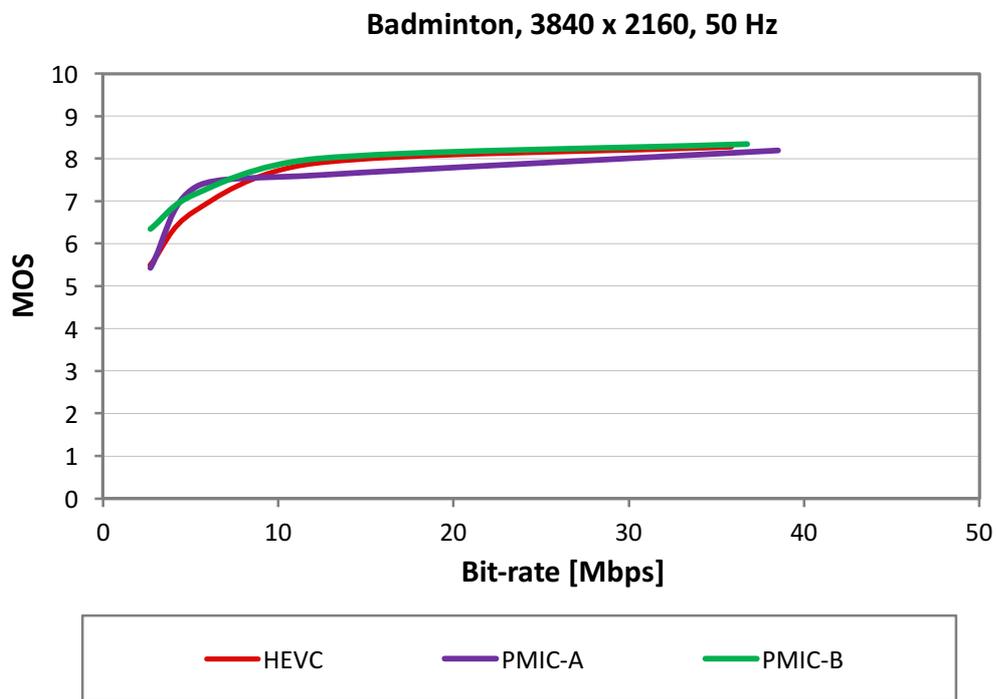


Figure 5.12 Objective test results.



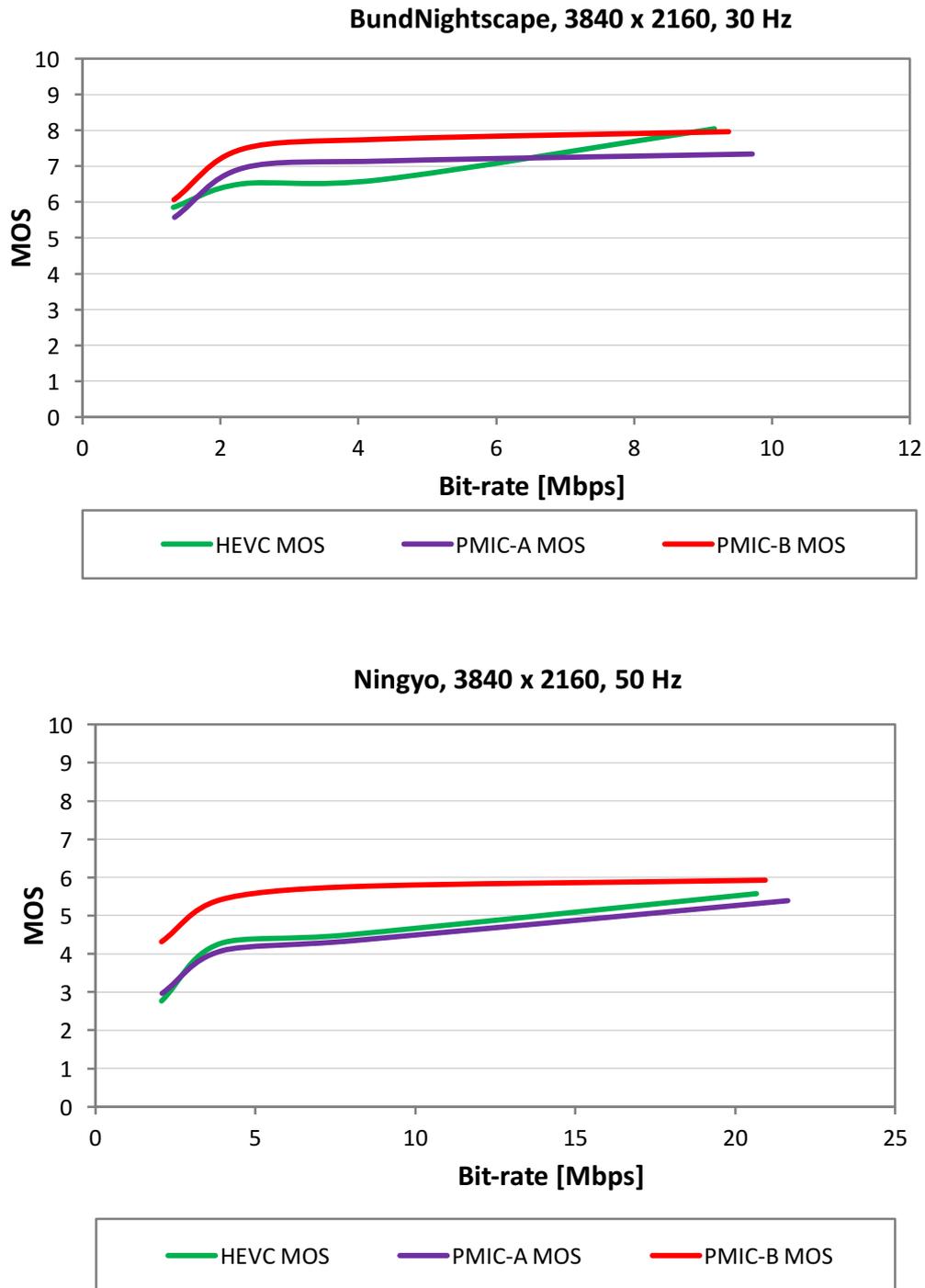


Figure 5.13 Subjective test results.

There are some findings from the results:

- The dash lines represent the objective PSNR results on aspect of varies bit rates. The plots of PSNR curves are very close. In objective results, the bitrate increase

of proposed PMIC methods slightly at certain PSNR level. For example, PSNR BD-rate results is ranging from 1.1% to 1.2% for PMIC- B method.

- The subjective results show BD-rate increase while objective results show BD-rate saving. That is to say, the commonly used PSNR is not suitable to be used for evaluating the contouring artefacts related video quality. For instance, by using the proposed PMIC methods, the PSNR BD-rate increases while MOS BD-rate gains saving.

### 5.4.3 Pixel mapping in-codec applied on intra blocks only

It is noted that the majority bit rate are consumed by intra frames/blocks from the results in Section 5.4.2. This section includes results of both PMIC methods applied on intra blocks only and compares with PMIC rest results. The PSNR BD-rate increase results are computed and listed in *Table 5.8* and *Figure 5.14* compared to the subjective MOS BD-rate results.

*Table 5.8* BD-rate results of PMIC.

	PMIC-A		PMIC-B		PMIC-A_intra		PMIC-B_intra	
	MOS BD-rate	PSNR BD-rate	MOS BD-rate	PSNR BD-rate	MOS BD-rate	PSNR BD-rate	MOS BD-rate	PSNR BD-rate
Badminton	-5%	12.8%	-23%	1.1%	-20%	1%	-1%	0.5%
Bund Nightscape	-34%	8.2%	-59%	1.2%	-35%	1.2%	-43%	1%
Ningyo	17%	10.6%	-78%	1.4%	-25%	0.8%	-31%	0.8%

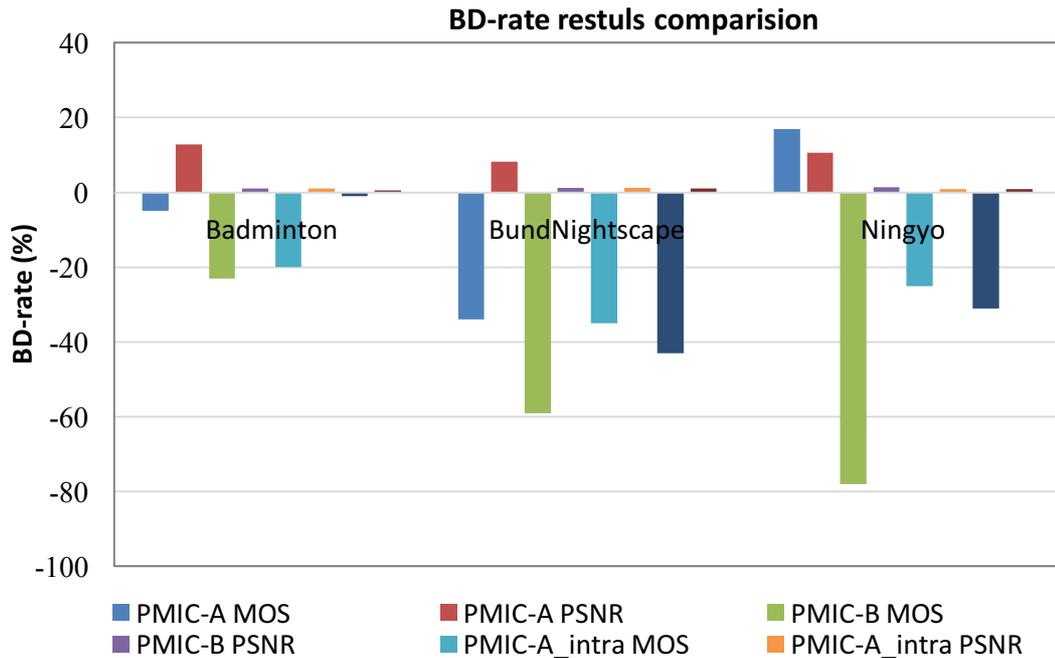


Figure 5.14 PSNR and MOS BD-rate results comparison.

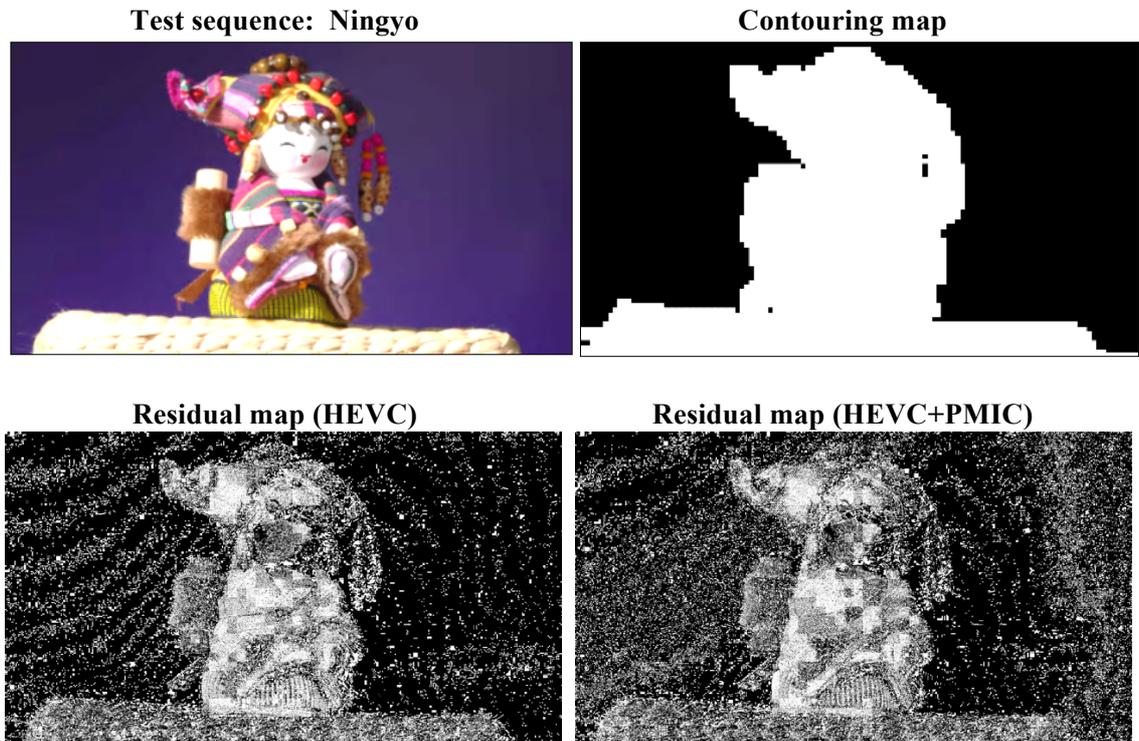
The PMIC methods applied on only intra blocks means the residual computed from intra prediction are computed by using the mapped original and predicted pixel values. PMIC adjusts residual in contouring area, which reduces contouring artefacts in intra predicted blocks. However, for inter frames/blocks which are predicted from blocks from neighbouring frames, the contouring artefacts are still existed which can affect the viewing experience. It can be seen from the PMIC MOS results that both PMIC– A and B have higher MOS BD-rate than that in related intra only results. The PSNR BD-rate is slightly changed since the pixel mapping algorithm is adjusting pixel values in a relatively small range.

## 5.5 Discussion

The pixel mapping approaches reduce the contouring artefacts for UHD content in HEVC. This section includes the discussion on aspects of residual analysis of pixel mapping algorithm, complexity comparisons and sensitivity of defined parameters.

### 5.5.1 Residual analysis of pixel mapping algorithm

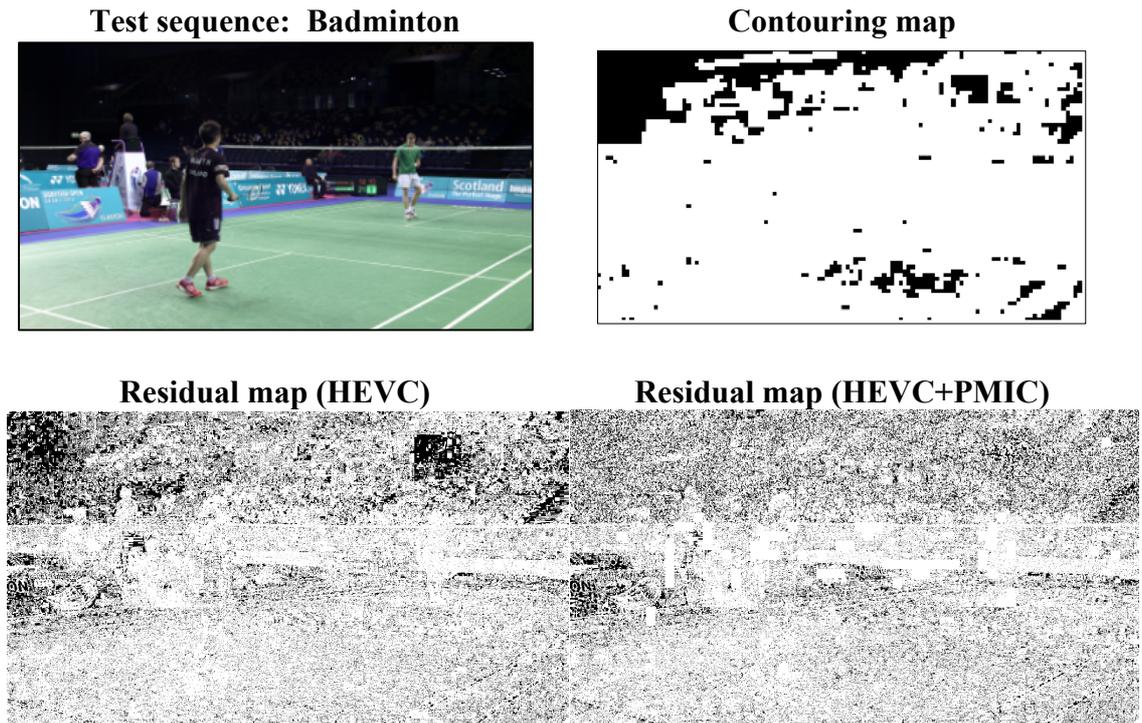
The residuals are increased in contouring area since they are computed by using the mapped original and predicted pixel values. This is proved in the residual map analysis for *Ningyo* sequence which shown in *Figure 5.15*.



*Figure 5.15 Residual map analysis (Ningyo).*

In the residual map, black area indicates no residual. The “Residual map (HEVC)” shows the residual computed from HEVC codec sequence. The “Residual map (HEVC+ PMIC)” shows the map computed from our proposed PMIC codec sequence. The white area in the second residual map which indicates the adjusted residual increases residuals in contouring area.

The residual map analysis results of *Badminton* sequence are shown in *Figure 5.16*.



*Figure 5.16 Residual map analysis (Badminton).*

For *Badminton* sequence, it is difficult to tell the differences from two residual maps. *Figure 5.17* lists residual values at each pixel in the local  $8 \times 8$  block at location (360, 410) from the contouring area. The pixels have zero residual then generate residual ranging from -2 to 2.

1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

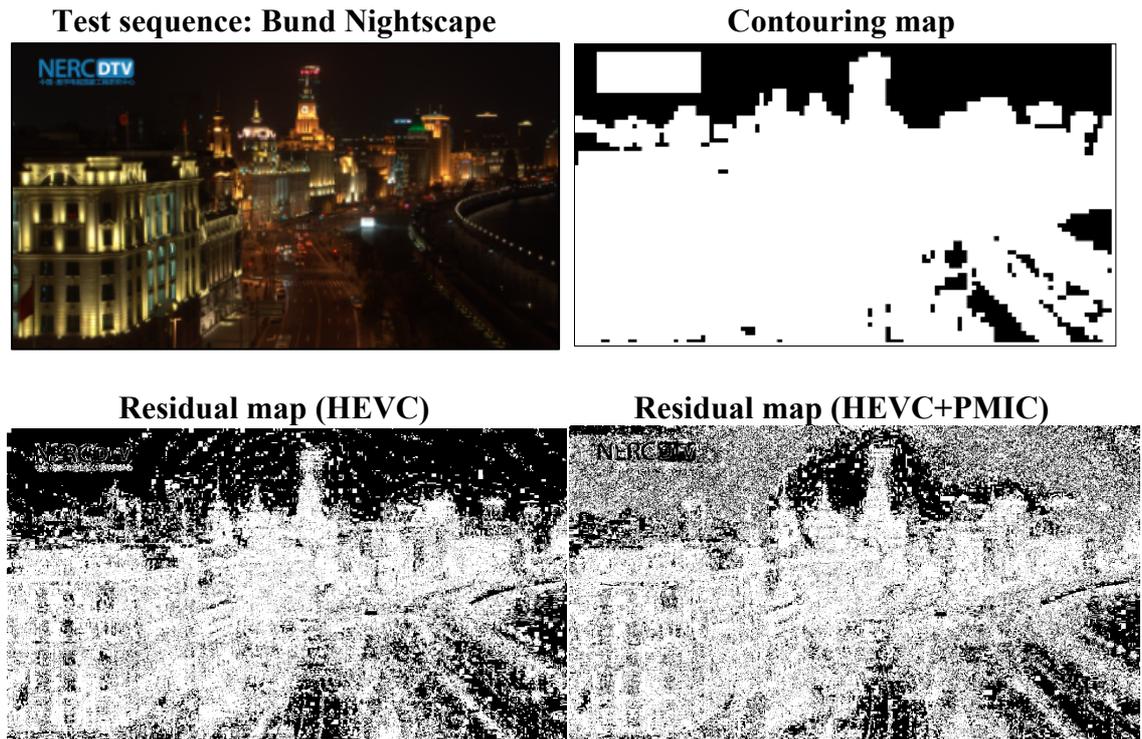
1) Residuals from HEVC coded block.

1	1	1	1	1	1	1	2
1	1	1	1	2	0	1	1
2	2	1	2	0	1	1	2
2	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1
1	1	1	2	1	1	1	1
1	1	2	3	2	1	2	2
1	2	3	2	8	4	2	2

2) Residuals from HEVC+ PMIC codec block.

Figure 5.17 Residual values analysis from  $8 \times 8$  blocks (Badminton).

The residual map analysis results of *Bund Nightscape* sequence are shown in *Figure 5.18*. In this sequence, the contouring area is dark sky (top area in the image), where more residuals are added by using the pixel mapping method.



*Figure 5.18 Residual map analysis (Bund Nightscape).*

### 5.5.2 Complexity comparisons

The proposed methods have low computation complexity. The related cluster configurations are listed in *Table 5.9*. Compared with HEVC (HM 16.0), the encoding time of pixel mapping method is reduced to 79% (PMIC-A) and 76% (PMIC-B); the decoding time is reduced to 72% (PMIC-A) or remain similar as 109% (PMIC-B).

Table 5.9 Cluster configurations.

Operation System	Debian - 3.2.0-24-generic
Compiler	gcc 4.8.2
CPU	Intel Xeon E3-1230@3.30Ghz
RAM	16GB - no swap
HD	fast disk array
Processes	1 process per core

### 5.5.3 Sensitivity of defined parameters

There are two parameters defined in pixel mapping algorithm: width  $d$  and multifactor  $n$ . The BD-rate results of PMIC- B with various parameters are listed in *Table 5.10*. The positive value indicates BD-rate increase. From the results we can that the parameter multifactor  $n$  is more sensitive than width  $d$ . The average BD-rate results increased from 1.2% to 3.3% with multifactor  $n$  raising from 1.5 to 2. The other parameter width  $d$  has less sensitivity in different settings. The results are very similar with different width values, 1.2% for  $d = 7$  and 1.6% for  $d = 9$ .

Table 5.10 BD-rate results of different parameters.

Sequences name	$d=7, n=1.5$	$d=7, n=2$	$d=9, n=1.5$
<i>Badminton</i>	1.1%	3.7%	1.4%
<i>Bund Nightscape</i>	1.2%	4.2%	1.7%
<i>Campfire Party</i>	1.1%	1.9%	0.8%
<i>Ningyo Pompoms</i>	1.3%	3.0%	1.8%
<i>Ningyo</i>	1.4%	3.7%	2.1%
Average	1.2%	3.3%	1.6%

In this thesis, only one pair of pixel mapping parameters ( $d = 7$ ,  $n = 1.5$ ) was evaluated in the subjective experiments. The investigate with more pixel mapping parameters are suggested in the future.

## **5.6 Summary**

This chapter designed contouring prevention methods for UHD in HEVC and evaluate the proposed methods. The proposed pixel mapping method can be implemented as pre-codec or in-codec approaches. There are two types of in-codec modes: PMIC-A and B. Both subjective and objective evaluation methods are used in the experiments.

## Chapter 6 Conclusions

This thesis presented the study on developing novel technologies and advanced tools for next generation content in HEVC to meet the new market requirements. The next generation video content types which included in this thesis are screen content and UHD content. Chapter 1 introduced the video coding history and next generation video content. It led to the new market requirements which motivated the rest of the study. The new findings and proposed solutions can be classified into two aspects: Chapter 2 and Chapter 3 developed coding tools for screen content for HEVC, Chapter 4 and Chapter 5 proposed coding solutions for UHD content for HEVC.

### 6.1 HEVC analysis and evaluation

Chapter 2 reviewed HEVC by introducing improved coding tools in HEVC that provide better video coding performance and compared with H.264/AVC. HEVC achieved larger gains on higher resolution videos and smaller gains on lower resolutions. Additionally, the performance of adaptive block partitioning in HEVC was evaluated to compare coding performance for both small blocks and large blocks. It can be seen that the adaptive block partitioning used in HEVC has better performance than partitioning the frames as smaller sized blocks. When usage of larger blocks is not allowed, the bit-rate consumption increases significantly for content of larger resolution, especially when inter prediction is used. Transform skipping generates significant improvements on screen content while other classes of sequences are not influenced by full transform skipping used in HEVC.

HEVC applies a more flexible block structure with CTU sizes as large as  $64 \times 64$  pixels and uses quadtree partitioning to recursively divide these into smaller blocks (up to  $8 \times 8$ ). More intra prediction modes (35 compared to 9 in H.264/AVC) and improved motion estimation are also used in HEVC. In-loop processing not only includes an improved De-

blocking filter but also adds the sample adaptive offset to increase the accuracy of reconstructed images. H.264/AVC uses both the CABAC and the CAVLC as entropy coding methods while HEVC retains enhanced CABAC as the only one.

The experiments in Chapter 2 include various types of content, and also different resolution of the video sequences. The experimental results show that HEVC (HM-8.0) improves RD performance for both camera-captured video and screen content videos compared to those from H.264/AVC (JM-18.4). The bit rate saving is up to 56% (e.g. Steam Locomotive in RA-Main). HEVC achieves larger gains on higher resolution videos and smaller gains on lower resolutions. Additionally, the performance of adaptive block partitioning in HEVC has been evaluated to compare coding performance for both small blocks and large blocks. It can be seen that the adaptive block partitioning used in HEVC has better performance than partitioning the frames as smaller sized blocks. When usage of larger blocks is not allowed, the bit-rate consumption increases significantly for content of larger resolution, especially when inter prediction is used. Transform skipping generates significant improvements on screen content while other classes of sequences are not influenced by full transform skipping used in HEVC. By applying transform skipping, bit-rate savings of 0.4% to 16.2% can be seen for screen content sequences.

## 6.2 Colour transforms for HEVC range extensions

Chapter 3 investigated colour transforms in HEVC and proposed colour space conversions for screen content. It firstly introduced conventional colour space YCbCr, which is the most widely used in video compression. However, it is not always the most efficient method in video compression, especially for screen content. It also reviewed the related colour transforms which are proposed in HEVC. More over, some other colour space transforms, such as YCoCg, 3-point P-DCT and KLT are also included the experiments.

There are two methods colour transforms proposed in Chapter 3:

- 1) KLT approximate transforms, named KLT-A, B and C in this thesis.

- 2) The 3-channel PPU colour space transforms, named PPU-1, 2, 3 and 4 in this thesis.

KLT method provides maximum de-correlation in pixels using Eigenvectors but the transform matrix need to be calculated individually for each single video sequence. The KLT maximum for test sequences are approximated into three transforms matrices. The results showed that one of the KLT approximation KLT-A achieves the highest BD-rate saving (5.8%). However, KLT –A is the best colour transform for natural content but not for screen content.

Based on the test results that screen content shows different trends on colour transforms, a new design of lifting-based colour transforms is proposed for screen content, namely 3-channel PPU. It first reviewed the lifting scheme and extended two-channel lifting into three channels. The different parameters in lifting scheme design results in various transform matrices. In this thesis, four groups of parameters were selected based on the luma computations from other colour transforms. The experimental results show all four PPU transforms gained BD-rate saving on average compared with YCbCr colour space transforms. one of the proposed PPU method (PPU-4) achieved the highest BD-rate saving (8.86%).

### **6.3 Contouring artefacts removal for UHD content in HEVC**

Chapter 4 introduced a proposed pixel dithering method to remove the contouring artefacts in HEVC decoded UHD video sequences. The contouring detection separates contouring regions and real object edges. This method adjusts pixels in each block by minimising gradient smoothness and block boundary smoothness with neighbouring blocks. The proposed multi-scale algorithm applies dithering on two different block sizes, which is efficient in correcting wide contouring bands in large regions and also small contouring areas close to real edges. It removes most false contouring from the test sequences without generating new artefacts. Both objective and subjective evaluation methods are used to compare the video qualities. The objective evaluation metrics PSNR and SSIM are used in this chapter. However, none of them can accurately evaluate the

contouring artefacts work. However, PSNR and SSIM are not suitable to evaluate contouring artefacts. This is because the contouring removal algorithm adjusts pixels in a small range (*i.e.*  $\pm 1$ ). Moreover, the contouring area which is smooth gradient has a high similarity between original frame and decoded frame. The subjective MOS is applied to evaluate the performance based on human eyes system. The results show that the video qualities are all improved by using our proposed contouring removal method. The MOS result is increased up to 62% (*i.e.* *Ningyo* sequence).

Chapter 5 proposed two pixel mapping methods for contouring artefacts removal for HEVC. One is a pre-codec approach, and the other one is HEVC in-codec approach. The pixel mapping algorithm is designed to compute the residuals by using the mapped original and predicted pixel values. Both subjective and objective evaluation methods are used to compare the video qualities in this chapter. Although the objective evaluation results PSNR BD-rate gains increases, the subjective MOS showed BD-rate saving. Experimental results show that one type of the pixel mapping in-codec method (PMIC-B) achieved MOS BD-rate saving for up to 78%.

## 6.4 Future work

The colour transforms proposed in Chapter 3 are designed input RGB sequence with a 4:4:4 chroma format. It is based on the fact that the original screen content sequences are all RGB 4:4:4 format at the time we started the colour transform work. The research can be extended into other chroma subsampling formats as well as different bit depth. Also, the proposed colour transforms are implemented as pre-codec approaches, it is suggested that proposed methods can also be applied as in-codec approaches.

On aspect of contouring detection and contouring removal, the experiments were performed only on luminance component. However, the implementation can be extended to two other chrominance components. Also, the proposed dithering method can also be used in higher bit depth contouring removal applications. In this research, there are two parameters set in the pixel mapping algorithm in Chapter 5: width  $d$  and multifactor  $n$ . In

## *CHAPTER 6. CONCLUSIONS*

this thesis, only one pair of pixel mapping parameters was evaluated in the experiments. The investigate with more pixel mapping parameters are suggested in the future.

Both objective and subjective evaluation are applied for UHD coding solutions. The objective (e.g. PSNR and SSIM) metrics are not suitable to use for contouring artefacts. The new design of objective evaluation methods based on HVS is needed for UHD content.

In this thesis, the HEVC tools for screen content evaluated are transform skipping and colour transforms. More attention should be attracted on other coding solutions for screen content, especially the HEVC screen content extension profiles are added in 2016. For mixed texture or content, some coding tools such as prediction still need to be improved.

Screen content and UHD content are the two types of next generation video content which included in this thesis. The investigation on other types of content, e.g. HDR, high bit depth content and high frame rate content, can also be included in future work.

## Bibliography

- [1] G. J. Sullivan, J. -R. Ohm, W. -J. Han, T. Wiegand, “overview of the High Efficiency Video Coding (HEVC) standard”, *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1649-1668, Dec. 2012.
- [2] M. T. Poirazad, C. Doutre, M. Azimi, P. Nasiopoulos, “HEVC: The New Gold Standard for Video Compression: How Does HEVC Compare with H.264/AVC?” *Consumer Electronics Magazine, IEEE*, vol.1, no.3, pp.36-46, July 2012.
- [3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.
- [4] ITU-T, “Advanced video coding for generic audiovisual services,” ITU-T Recommendation H.264, Feb. 2014.
- [5] ITU-T VCEG and ISO/IEC MPEG, “Joint Call for Proposals on Video Compression Technology,” document VCEG-AM91 and WG11 N11113, Kyoto, Japan, Jan. 2010.
- [6] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan and T. Wiegand, “High efficiency video coding (HEVC) text specifications draft 8,” document JCTVC-J1003 of JCT-VC, Stockholm, Sweden, 2012.
- [7] ITU-R, “Parameter values for ultra-high definition television systems for production and international programme exchange,” ITU-R Rec.BT.2020, Aug. 2012.
- [8] J. -R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, T. Wiegand, “Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC),” *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1669-1684, Dec. 2012.
- [9] ITU-T, “Video CODEC for audio-visual services at  $p \times 64$  kbit/s,” ITU-T Rec. H.261, 1993.
- [10] ITU-T, “Video Coding for Low Bitrate Communication,” ITU-T Rec. H.263, version 1, 1995, version 2, 1998, version 3, 2000.
- [11] ITU-T and ISO/IEC JTC 1, “Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video,” ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), version 1: 1994.
- [12] ISO/IEC JTC 1, “Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual),” version 1: 1999, version 2: 2000, version 3: 2004.
- [13] ITU-R, “Parameter values for high definition television systems for production and international programme exchange,” ITU-R Rec.BT.709-5, April 2002.
- [14] S. Sakaida, N. Nakajima, A. Ichigaya, M. Kurozumi, K. Iguchi, Y. Nishida, E. Nakasu, and S. Gohshi, B, “The Super Hi-Vision codec,” in *Proc. IEEE International Conference on Image*

- Processing*, 2007, pp. I.21–I.24.
- [15] J. Zubrzycki, The Olympics in Super Hi-Vision, 2012, <http://www.bbc.co.uk/blogs/researchanddevelopment/2012/08/the-olympics-in-super-hi-visio.shtml> [Accessed: April, 01, 2016].
- [16] T. Lin, P. Hao, “Compound Image Compression for Real-Time Computer Screen Image Transmission,” *IEEE Trans. on Image Proc.*, Vol. 14, No. 8, pp. 993–1005, Aug. 2005.
- [17] H. Meuel, J. Schmidt, M. Munderloh, J. Ostermann, “Analysis of coding tools and improvement of text readability for screen content,” in *Proc. Picture Coding Symposium (PCS) 2012*, May 2012.
- [18] K. McCann, “Progress Towards High Efficiency Video Coding,” *DVB SCENE*, p. 5, March 2012.
- [19] F. Bossen, B. Bross, K. Suehring and D. Flynn, “HEVC complexity and implementation analysis *Circuits and Systems for Video Technology*, *IEEE Transactions on* , vol.22, no.12, pp.1685,1696, Dec. 2012.
- [20] C. Yeo, Y. H. Tan, Z. Li, “Low-complexity mode-dependent KLT for block-based intra coding,” in *Proc. IEEE International Conference on Image Processing 2011*, Sep. 2011.
- [21] C. Abhayaratne, “Lossless and Nearly Lossless Digital Video Coding”, PhD Thesis, Dept. of Electronica and Electrical Engineering, University of Bath, U.K., 2002.
- [22] C. Abhayaratne and DM. Monro, “Embedded-to-lossless coding of motion-compensated prediction residuals in lossless video coding”, *Photonics West 2001-Electronic Imaging*, pp. 175-185, 2000.
- [23] D. Flynn, J. Sole, T. Suzuki, “High efficiency video coding (HEVC) Range Extension text specification: Draft 4,” document JCTVC-N1005 of JCT-VC, Incheon, KR, April 2013
- [24] P. Hanhart, M. Rerabek, F. De Simone, T. Ebrahimi, “Subjective quality evaluation of the upcoming HEVC video compression standard,” in *Proc SPIE Optics+Photonics 2012 Applications of Digital Image Processing XXXV*, Aug. 2012.
- [25] G. Bjontegaard, "Improvements of the BD-PSNR model," Tech. Rep. VCEG-A111, ITU-T SG16/Q6, Berlin, Germany, July 2008.
- [26] F. Bossen, “Common test conditions and software reference configurations,” document JCTVC-H1100 of JCT-VC, San Jose CA, USA, Feb. 2012.
- [27] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.
- [28] F. Zhang; D. Bull, "A Perception-based Hybrid Model for Video Quality Assessment," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no.99, pp.1-1 , 2015.
- [29] R. Weerakkody, M. Mrak, V. Baroncini, J. R. Ohm, T. K. Tan and G. J. Sullivan, "Verification testing of HEVC compression performance for UHD video," *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, Atlanta, GA, pp. 1083-1087, 2014.
- [30] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan and T. Wiegand, “High efficiency video coding

- (HEVC) text specifications draft 10 (for EDIS & Last Call),” document JCTVC-L1003 of JCT-VC, Geneva, CH, Jan. 2013.
- [31] ITU-T, “High efficiency video coding”, ITU-T Rec. H.265, Version 1, 2013, Version 2, 2014.
- [32] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards, and Practice*. Boston, MA: Kluwer, 2000.
- [33] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.
- [34] H. S. Malvar and G. J. Sullivan, “Transform, scale& colour space impact of professional extensions,” ITU-T/ISO/IEC JVT document JCT-H031, Geneva, May 2003.
- [35] K. S. Thyagarajan, *Still Image and Video Compression with MATLAB*. Canada, Wiley, 2011.
- [36] P. Hao and Q. Shi, “Comparative study of color transforms for image coding and derivation of integer reversible color transform”, *Proc. 15<sup>th</sup> Int. Conf Pattern Recognition*, Barcelona, Spain, vol.3, pp.224-227, Sept. 2000.
- [37] H. Malvar and G. Sullivan, “YCoCg-R: A color space with RGB reversibility and low dynamic range”, document JCT-I1014, Norway, July 2003.
- [38] H. S. Malvar, G. J. Sullivan and S. Srinivasan, “Lifting-based reversible colour transformations for image compression,” in *Proc SPIE, Applications of Digital Image Processing XXXI*, 2008.
- [39] ITU-R, “Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios” ITU-R Rec.BT.601-7, March 2011.
- [40] L. Zhang, J. Chen, J. Sole, M. Karczewicz, X. Xiu and J. Z. Xu, "Adaptive Color-Space Transform for HEVC Screen Content Coding," *Data Compression Conference (DCC), 2015*, Snowbird, UT, 2015, pp. 233-242.
- [41] K. Kawamura, H. Kato and S. Naito, "In-loop colour-space-transform coding based on integer SVD for HEVC range extensions," *Picture Coding Symposium (PCS), 2013*, San Jose, CA, 2013, pp. 241-244.
- [42] T. Strutz, "Adaptive selection of colour transformations for reversible image compression," *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, Bucharest, 2012, pp. 1204-1208.
- [43] R. Weerakkody and M. Mrak, "Adaptive low complexity colour transform for video coding," *Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on*, Jakarta, 2014, pp. 1-6.
- [44] T. Strutz and A. Leipnitz, "Reversible Color Spaces without Increased Bit Depth and Their Adaptive Selection," in *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1269-1273, Sept. 2015.
- [45] JTC-VC (2013), *HEVC test sequence* [online] <ftp://ftp.tnt.uni-hannover.de/testsequences>. [Accessed: May, 01, 2014].
- [46] S. V. Assche, W. Philips, and I. Lemahieu, “Lossless compression of pre-press images using a novel colour de-correlation technique,” *Pattern Recognition*, vol. 32, pp. 435-441, 1999.
- [47] W. Sweldens, “The lifting scheme: a custom-design construction of biorthogonal wavelets,”

- Applied and Computational Harmonic Analysis, vol. 3, pp. 186-200, 1996.
- [48] S. Oraintara, Y. J. Chen and T. Q. Nguyen, "Integer fast Fourier transform," in *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 607-618, Mar 2002.
- [49] Chung-Jr Lian, Kuan-Fu Chen, Hong-Hui Chen and Liang-Gee Chen, "Lifting based discrete wavelet transform architecture for JPEG2000," *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, pp. 445-448 vol. 2, Sydney, NSW, 2001.
- [50] T. S. Sinha, D. Chakravarty, R. Patra and R. Raja. "Modelling and Simulation for the Recognition of Physiological and Behavioural Traits Through Human Gait and Face Images", in *Discrete Wavelet Transforms - A Compendium of New Approaches and Recent Applications*, A. Al-Asmari, eds. 2013.
- [51] D. Flynn, K. Sharman and C. Rosewarne, "Common test conditions and software reference configurations for HEVC range extensions," document JCTVC-N1006 of JCT-VC, Vienna, Aug. 2013.
- [52] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," VCEG Contribution VCEG-M33, Austin, April 2001.
- [53] S. Pateux and J. Jung, "An excel add-in for computing Bjontegaard metric and its evolution," VCEG Contribution VCEG-AE07, MA, Jan. 2007.
- [54] HHI, svn HEVC Software [online]. Available: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-15.0+RExt-8.0+SCM-2.0/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-15.0+RExt-8.0+SCM-2.0/) [Accessed: April, 01, 2016].
- [55] X. Jin, S. Goto and K.-N. Ngan, "Composite model-based DC dithering for suppressing contour artifacts in decompressed video," *IEEE Trans. Image Process.*, vol. 20, no. 8, Aug. 2011.
- [56] S. J. Daly and X. Feng, "Decontouring: prevention and removal of false contour artifacts," in *Proc. SPIE 5292, Human Vision and Electronic Imaging IX*, 130, June, 2014.
- [57] L. G. Roberts, "Picture coding using pseudo-random noise," *IRE Trans. Inf. Theory*, vol. IT-8, no.2, pp. 145-154, Feb. 1962.
- [58] V. Ostromoukhov, "A simple and efficient error-diffusion algorithm," in *Proc. SIGGRAPH ACM Comput. Graph., Annu. Conf. Ser.*, 2001, pp. 567-572.
- [59] W. Ahn and J.-S. Kim, "Flat-region detection and false contour removal in the digital TV display," in *Proc. IEEE Int. Conf. Multimedia Expo*, pp. 1338-1341, Jul. 2005
- [60] J.W Lee, B. R. Lim, R. -H. Park, J.-S. Kim, and W. Ahn, "Two-stage false contour detection using directional contrast and its application to adaptive false contour reduction," *IEEE Trans. Consum. Electron.*, vol. 52, no. 1, pp. 129-188, Feb. 2006.
- [61] K. Yoo, H. Song, and K. Sohn, "In-loop selective processing for contour artifact reduction in video coding," *Electron. Lett.*, vol. 45, pp. 1020-1022, Sept. 2009.
- [62] S. Bhagavathy, J. Llach, and J. Zhai, "Multiscale probabilistic dithering for suppressing contour artifacts in digital images," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1936-1945, Sept. 2009.

- [63] J. M. Boyce and A. M. Tourapis, "Comfort noise for compressed video," in *Proc. ICCE*, pp. 323-324, Jan. 2005.
- [64] Y. Matsuo, T. Misu, S. Iwamura and S. Sakaida, "Ultra high-definition video coding using bit-depth reduction with image noise reduction and pseudo-contour prevention," *Visual Communications and Image Processing (VCIP) 2013*, Kuching, 2013, pp. 1-6.
- [65] A. S. Dias and M. Mrak, "Region-adaptive quantisation for prevention of contouring in coded video," *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*, Xiamen, 2015, pp. 1-6.
- [66] N. Casali, M. Naccari, M. Mrak and Leonardi, R. "Adaptive quantisation in HEVC for contouring artefacts removal in UHD content", *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 2577 – 2581, 2015.
- [67] European Broadcasting Union (EBU), *Eurovision Young Dancers Trailer* [online]. Available: <https://tech.ebu.ch/testsequences/eyd> [Accessed: Oct. 13, 2014].
- [68] European Broadcasting Union (EBU), *UHD-1 Test Sequences* [online]. Available: <https://tech.ebu.ch/testsequences/uhd-1> [Accessed: Oct. 13, 2014].
- [69] L. Song, X. Tang, W. Zhang, X. Yang, P. Xia, *The SJTU 4K Video Sequence Dataset*, the *Fifth International Workshop on Quality of Multimedia Experience (QoMEX2013)*, Klagenfurt, Austria, July, 2013
- [70] Kamerawerk GmbH, *No Sleep 4K: The Shoot* [online]. Available: <http://vimeo.com/54796304> [Accessed: Oct. 13, 2014].
- [71] ITU-R, *Methodology for the Subjective Assessment of the Quality of Television Pictures*, ITU-T Rec. BT.500-13, Jan. 2010
- [72] ITU-T, *Subjective Video Quality Assessment Methods for Multimedia Applications*, ITU-T Rec. P.910, April 2008.
- [73] ITU-R, "The present state of ultra-high definition television," ITU-R Rec.BT.2246-2, Nov. 2012.
- [74] Y. Wang, C. Abhayaratne, R. Weerakkody and M. Mrak, "Multi-scale dithering for contouring artefacts removal in compressed UHD video sequences" in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP 2014)*, December 2014.
- [75] IEEE explore, Multi-scale dithering for contouring artefacts removal in compressed UHD video sequences [online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7032274&isnumber=7032060> [Accessed: April, 01, 2016].