Algorithms for Compression of High Dynamic Range Images and Video

by

Vladimir Dolzhenko

A Doctoral Thesis

Submitted in partial fulfilment of the requirements
for the award of

Doctor of Engineering

of Loughborough University

31.05.2015

Abstract

The recent advances in sensor and display technologies have brought upon the High Dynamic Range (HDR) imaging capability. The modern multiple exposure HDR sensors can achieve the dynamic range of 100-120 dB and LED and OLED display devices have contrast ratios of 10^5:1 to 10^6:1.

Despite the above advances in technology the image/video compression algorithms and associated hardware are yet based on Standard Dynamic Range (SDR) technology, i.e. they operate within an effective dynamic range of up to 70 dB for 8 bit gamma corrected images. Further the existing infrastructure for content distribution is also designed for SDR, which creates interoperability problems with true HDR capture and display equipment.

The current solutions for the above problem include tone mapping the HDR content to fit SDR. However this approach leads to image quality associated problems, when strong dynamic range compression is applied. Even though some HDR-only solutions have been proposed in literature, they are not interoperable with current SDR infrastructure and are thus typically used in closed systems.

Given the above observations a research gap was identified in the need for efficient algorithms for the compression of still images and video, which are capable of storing full dynamic range and colour gamut of HDR images and at the same time backward compatible with existing SDR infrastructure. To improve the usability of SDR content it is vital that any such algorithms should accommodate different tone mapping operators, including those that are spatially non-uniform.

In the course of the research presented in this thesis a novel two layer CODEC architecture is introduced for both HDR image and video coding. Further a universal and computationally efficient approximation of the tone mapping operator is developed and presented. It is shown that the use of perceptually uniform colourspaces for internal representation of pixel data enables improved compression efficiency of the algorithms. Further proposed novel approaches to the compression of metadata for the tone mapping operator is shown to improve compression performance for low bitrate video content. Multiple compression algorithms are designed, implemented and compared and quality-complexity trade-offs are identified. Finally practical aspects of implementing the developed algorithms are explored

by automating the design space exploration flow and integrating the high level systems design framework with domain specific tools for synthesis and simulation of multiprocessor systems. The directions for further work are also presented.

Vladimir Dolzhenko, 31st May 2015

Acknowledgements

I would like to express my special appreciation to my advisor Dr Eran Edirisinghe who has been tremendous mentor to me over all these years. Without his help, advice and encouragement completing this work would not be possible. I would like to thank Dr Helmut Bez for his course and guidance during annual reviews.

I would like to thank my sponsoring company Apical Ltd, its CEO Dr Michael Tusch and the industrial supervisor Dr Slava Chesnokov for all the help with tuition fees and for providing priceless experience of understanding and designing imaging algorithms and complex systems. Thanks to their understanding at the times when the University assignments clashed with the hard deadlines.

I would like to thank all the staff of Loughborough University Systems Engineering Doctorate Centre for brilliant management of the programme and for providing world class training in systems engineering. Specific thanks to Dr Stuart Burdges for excellent introduction to Systems Engineering. I am thankful to all my fellow students, it was fun time together and I have learned a lot from you.

A special thanks to my family. I am very grateful to my mother-in-law, father-in-law, my mother and father for all their help, encouragement and understanding.

I would like to specifically thank my beloved wife Anastasia Lazareva for all her help, encouragement and belief in me. It is because of you I did not give up on this work. After all the sacrifices you did on my behalf it is your thesis as well.

Vladimir Dolzhenko, 31st May 2015

Table of contents

Abbreviations

AVC – Advanced Video Coding, see H.264

CABAC – Context Adaptive Binary Arithmetic Coding, entropy coding method used in H.264 and H.265 CODEC

CAVLC – context Adaptive Variable Length Coding, a variant of low complexity entropy coding for H.264 video compression standard

CCTV – Closed Circuit Television, typically local surveillance system as opposed to broadcast television.

CF - Colour Filter, a bandpass optical filter.

CFA – Colour Filter Array. Method of capturing colour images using a single monochrome sensor. One of the widely used patterns is the Bayer pattern when odd lines have alternating red and green filters and even lines have alternating green and blue filters.

CODEC – COder-DECoder – device or algorithm for compression and decompression of the data

CPU – Central Processing Unit, the device to fetch program and data from the memory, or peripheral devices, execute the commands and store the result back.

CTB – Coding Tree Block, processing unit in H.265 encoding standard.

CU - Coding Unit, processing unit in H.265 encoding standard.

DCT – Discrete Cosine Transform

DSP – Digital Signal Processor – highly embedded processor highly optimized for signal processing tasks, for example, filtering or Fourier transforms.

DST – Discrete Sine Transform

FIR – Finite Impulse Response (a type of digital filter)

H.264 – ITU-T and MPEG standard for video compression. Also known as AVC and MPEG4 part 10. This is an advanced block based video coding algorithm.

H.265 - ITU-T and MPEG standard for next generation video compression. Also known as HEVC. Typically the encoder uses 10x processing power of H.264 and achieves half the bit rate with the same visual quality. The decoding complexity is comparable with H.264 and extra tools allow parallel processing to increase the throughput.

HDL - Hardware Description Language, a machine language used to define the structure and behaviour of the logical devices. Notable examples of such languages include VHDL, Verilog, System Verilog and System C.

HDR – High Dynamic Range (more than 70-80 dB).

HDTV – High Definition Television, common resolutions 1920x1080 and 1280x720 pixels

HEVC – High Efficiency Video Coding, see H.265

ILM – Industrial Light and Magic, a visual effect company founded by George Lucas. Promotes a family of image compression standards for high dynamic range images, e.g. OpenEXR.

INCOSE - International Council on Systems Engineering

ISO – International Standard Organization

ISO/IEC - International Standard Organization/International Electrotechnical Committee

ITU-T – International Telecommunication Unit

JPEG – Joint Photographer Expert Group – committee that introduced block based algorithm for lossy compression of the photographic images. The algorithm is based on 8x8 block DCT transform followed by quantization and entropy coding of the coefficients.

JPEG2000 – wavelet-based algorithm for photographic image compression.

JPEG-XR – recent (20091X) standard for high fidelity image compression. Based on lapped orthogonal transform. The developers claim JPEG2000 performance at baseline JPEG cost, but current codec implementations do not achieve promised quality.

LCD – Liquid Crystal Display, a flat display technology that uses liquid crystals for light modulation.

LDR – Low Dynamic Range (up to 60-70dB)

LED – Light Emitting Diode, a semiconductor device for transforming electricity to light.

MARTE – “Modelling and Analysis of Real-Time and Embedded Systems”, a UML profile for real-time system design

MB -– MacroBlock, a rectangular coding unit for H.264 standard

MPEG – Motion Picture Experts Group, introduced standards for video and audio compression and retrieval.

NAL – Network Abstraction Layer, part of the standard focusing on packet representation of the encoded data to facilitate transmission over networks.

NALU – Network Abstraction Layer Unit

OLED – Organic Light Emitting Diode, the electroluminescent device based on organic compound. Used to create high contrast emissive displays.

OMG – Object Management Group, a standardization body

PIZ – a compression algorithm used in OpenEXR file format, combination of wavelet transform and Huffman encoding.

PU - Prediction Unit, processing unit in H.265 encoding standard.

SAO - Sample Adaptive Offset, visual image enhancement algorithm in H.265 encoding standard.

SoC – System on Chip, an integrated circuit containing multiple interconnected blocks, for example, several processors, imaging pipeline, graphics accelerator, video encoder and decoder accelerators, all communicating via high speed interconnect.

SPIHT - Set Partitioning in Hierarchical Trees, a wavelet based algorithm for image and video compression

SVC – Scalable Video Coding, an extension for H.264 video compression allowing to transmit efficiently several versions of the same content with different resolutions, frame rate or image quality.

SysML – “Systems Modelling Language”, graphical language for representation of the system architecture.

TFT – Thin Film Transistor, a technology for improving the image quality and addressability of the LCD displays by deposition a polysilicon transistor to each pixel.

TIFF – Tagged Image File Format – Adobe standard for compressed and uncompressed image container.

TU - Transform Unit, processing unit in H.265 encoding standard.

UHDTV – Ultra High Definition Television, usually 3840x2160 pixels and above

UML – Universal Modelling Language, a graphical language for representation of the software architecture. Also used to model non-software systems.

VCEG – Video Coding Expert Group

VDP – Visual Difference Predictor, an algorithm for estimation of the probability of detection the differences between two images. Can be used for automated image quality optimization.

WDR – Wide Dynamic Range (this acronym is mostly used by CCTV camera manufacturers). Synonym for HDR in this thesis.

WZ -– Wiener- Ziv, algorithm for independent entropy coding of correlated information sources. The Winner-Ziv theorem states that if the correlation is known it is possible to encode the sources separately with the same compression efficiency as if they were encoded combined.

List of figures

List of tables

Chapter 1

Introduction

The recent advances in sensor technology have resulted in the introduction of digital cameras with High Dynamic Range (HDR) imaging capabilities. At a high level, HDR imaging enables the representation of pixel colour by a greater bit depth as compared to the traditional 24 bit colour depth representation used by Standard Dynamic Range (SDR) imaging technologies. At present cameras with 100 dB dynamic range (16 bit effective bit depth) are produced commercially and ones with even higher ranges are about to move into mass production. This advancement is widely accepted to resolve many outstanding issues in the perceived quality of natural scenes in the digital entertainment industry (e.g. TV, DVD, games etc.). HDR imaging is also expected to play an important role in the surveillance industry, especially in outdoor and air-born surveillance camera systems, where the cameras have a high probability to be exposed to high dynamic range images (e.g., due to bright sunlight, fire, bright reflections, shadows).

In parallel with the advances of the high dynamic range support capability of sensors, the digital display technology has also progressed. At present dynamic range of displays can easily cross 8bit (~48dB). Current TFT displays boast 1000:1 or better contrast ratio. Local backlight dimming schemes or OLED displays can increase this ratio several times. In other words at present the technology exists for the enjoyment of high dynamic range videos on screen, in entertainment applications or to produce these images for either better visual or machine vision based inspection in surveillance applications.

1.1 Research problem

Despite the abovementioned technological advances a large bottleneck exists in the capabilities of compressing high dynamic range images/video for efficient transmission and storage. Transmission/storage is the most important stage between the effective capture and display of digital content. Conventional 8 bit compression standards such as JPEG, MPEG are unable to support high dynamic range images. Even the state-of-the-art BluRay and HD DVDs are using 8 bit MPEG2/MPEG4 compression. JPEG2000 image standard can support up to 16 bit (98dB) images, but it is not currently in widespread use and cannot support very high dynamic ranges.

The task of creating the compression algorithm cannot be successful without taking into account the properties of the entire image distribution systems. Different areas of application have very different requirements for the desired algorithm. It is very important to adopt a systemic approach in the algorithm design and optimize the whole end-to-end system performance, starting from the scene illumination and input optical system to the display device and viewing conditions. Furthermore, the design and architecture of distributed surveillance systems dictate the requirements of the camera and, as one further components, the compression subsystem.

1.2 Sponsoring company

As the research project commenced as an EngD project it was planned that Apical Ltd (2015) would be involved from the early stages for providing directions and focus for research The ultimate aim of this role being played by Apical was that the research results will contribute to the company's product range. To facilitate this an industry supervisor was appointed; who positively contributed to the discussions regarding the research.

During the course of the project the key focus of this research was shifted at a number of occasions to align it with the company’s Research and Development roadmap.

1.3 Research aim and objectives

The aim of current research is to explore novel high dynamic range image and video compression algorithms and carry out system designs for their implementation.

The research was organized to meet the final agreed objectives:

1. Review literature of the subject are of High Dynamic Range imaging find the main existing approaches to acquisition, storage and consumption of high dynamic range content. Carry out a further literature review in related subject areas to especially include quality metrics and tone mapping operators.

2. Explore the design, implementation and testing of novel algorithms and approaches for the compression of high dynamic range images. The main focus is on making the proposed schemes backwards compatible with as little overhead as possible.

3. Investigate the possible use of method within a wider range of standard image compression algorithms and widely used colour spaces. The objective is to optimize the performance of systems by varying the compression algorithms adopted and to study the importance of using perceptually uniform colour space representations.

4. Apply the algorithms and optimizations proposed in 3 above to video compression. As state of art video compression algorithms are more efficient than still image compression algorithms due to the presence of temporal redundancy and reduced eye sensitivity to temporal noise, the metadata compression algorithm may need updating.

5. Study the current trends in complex System-on-Chip designs and investigate the application of these methods for practical HDR video CODEC implementations.

6. Review the results and formalize the objectives for further study.

1.4 Chapter description

In order to facilitate the presentation of the report and clarity of presentation, it is divided into the following chapters:

Chapter 1 is an introduction to the research field of High Dynamic Range imaging, which explains the importance of the subject area and outstanding research problems. Further the aim and objectives of the research to be conducted within the context of this thesis are stated.

Chapter 2 contains background information on video encoding standards used within the paper, the mature H.264 standard, the Scalable Video Coding (SVC) standard based on it and the next generation H.265 standard which aims for high resolution and high fidelity applications providing 2x reduction of the bit stream size over H.264 delivering the same visual quality. The chapter ends with short description of high dynamic range imaging and explanation of algorithm complexity.

Chapter 3 is dedicated to literature review. The areas of lossless and lossy HDR image and video compression are extensively covered. Further alternative methods - using less conventional image data representation or compression schemes are reviewed. The chapter ends with a section on system design where system integration of the proposed algorithms and approaches to its implementation in silicon, are discussed.

Chapter 4 is dedicated to introduction of the proposed novel scheme for backward compatible HDR image compression. A two layer compression scheme is introduced and a method for inverse tone mapping is discussed. Image compression scheme optimizations are also discussed, especially ones studying the effects of perceptually uniform colour spaces and using different image compression standards.

Chapter 5 is dedicated to high dynamic range video compression. Novel and efficient methods for metadata compression are introduced and tested.

Chapter 6 describes research in the area of system level design automation. The trends in complex systems-on-chip are studied and the solution for design space exploration on multiprocessor systems is introduced.

Chapter 7 concludes the dissertation and presents potential areas for further research and improvement.

1.5 Research contribution

Over the course of the research presented in this thesis the following contributions were made:

- A novel two layer backward compatible HDR compression algorithm was proposed:
    - Fully backward compatible SDR layer
    - High quality local tone mapping algorithms utilised
    - Moderate overhead (10-30 %)
    - Flexible within usage scenarios
- The framework for formal comparison of image and video compression algorithms was introduced. The results were obtained for the following algorithms:
    - JPEG2000 vs JPEG-XR vs JPEG
    - H.264 vs H.265
    - HDR encodings: floating point, luv, Bef
- .

- An algorithm for creating an inverse tone mapping model for local and global tone mapping algorithms was proposed.
- Two variants of meta data compression approaches were proposed that demonstrated:
    - Low complexity (for images)
    - Low bitrate (for video)
- A bitstream format for metadata encodings was proposed.
- Based on analysis of the design trends and challenges in high performance embedded systems, a framework for the implementation of the HDR compression pipeline proposed in this thesis on system-on-chip, was introduced.

The original research presented in this thesis has led to the publication of a number of peer reviewed conference papers. The relevant list of publications is presented in Appendix 4.

Chapter 2

Background

2.1 Introduction

This chapter presents the research background on which the work presented in this thesis has been based on. In particular it introduces the reader to the H.264 (AVC & SVC), HEVC standards and the fundamentals of High Dynamic Range (HDR) video coding.

2.2 H.264/AVC Video coding standard

In early 1998, the Video Coding Experts Group (VCEG) ITU-T SG16 Q.6 issued a call for proposals on a project named H.26L, with the target to double the coding efficiency of video (which means halving the bit rate necessary for a given level of fidelity) in comparison to any other existing video coding standards, for a broad variety of applications. The first draft design for that new standard was adopted in October of 1999. In December of 2001, VCEG and the Moving Picture Experts Group (MPEG) ISO/IEC JTC 1/SC 29/WG 11 formed a Joint Video Team (JVT), with the charter to finalize the draft of the new video coding standard for formal approval as H.264/AVC in March 2003. (Wiegand *et al.*, 2003)

As with preceding standards the H.264/AVC document only describes the bit stream syntax elements and a hypothetical reference decoder. This approach opens more flexibility for the design and optimization of the encoder and guarantees that any compliant bitstream will be correctly decoded by any compliant decoder (within the level and profile constraints).

The H.264 encoder belongs to a family of hybrid coders where both spatial and temporal redundancies are exploited to reduce the bit rate for transmission and storage. The image is split into rectangular macroblocks before encoding.

The four main elements of the coding pipeline used in H.264 standard (Sullivan and Wiegand, 2005) are as follows:

Prediction – the process of inferring samples of the current block based on the previously processed samples. The process of inferring the samples from the current frame is called intra-prediction or, if it is a previously processed frame, inter-prediction. Intra-prediction does not rely on history and should be used in key frames – for example first frame in the video sequence or the first frame of the group of pictures. The inter prediction is typically more efficient – the static parts of the video sequence can be encoded with a minimal amount of bits, but relies on encoding/decoding history, reduces transmission error resilience and support of non-linear playback modes.

Transform – the difference between the actual image and its prediction is still highly correlated spatially, the linear transform de-correlates the data and allows more bits on the components with more energy and more visual quality impact. The ideal de-correlation is achieved with Karhunen-Loewe transform (KLT), but for practical images discrete cosine transform (DCT) achieves close results with less operations and without need to transmit the basis vectors. The H.264 standard uses approximation of 4x4 or 8x8 DCT transform with integer coefficients. This approach allows encoder and decoder to work in synchronisation and eliminate drift between them caused by round off errors and differences in implementation.

Quantization – the only step (apart from deblocking filter which will be described later) where actual loss of the precision is done – other steps are reversible. It is also one of the points for tradeoffs between quality and bitrate. The quantizer in H.264 is uniform with dead zone. The effective quantization step is different for different positions of transformed data.

Entropy coding – the process of transforming all the syntax elements – prediction modes, motion vectors, residual data - into bits of the result bitstream. More frequent symbols are transmitted by shorter bit strings. Two methods are used – context adaptive variable length coding (CAVLC) where fixed bit strings are used for syntax elements, but the tables are switched depending on context. The context adaptive binary arithmetic coding (CABAC) is more complex computationally but achieves 5-15% better performance (Wiegand *et al.*, 2003) because fractional number of bits is used for syntax elements and also because it adapts the probability model to the actual statistics of the encoded data.

The hybrid coding can often lead to objectionable blocking artefacts especially at low bit rates. The de-blocking filter (List *et al.*, 2003) estimates the smoothness area across the block edges and filters the samples if the difference can be attributed to quantization of the transform coefficient. The filter allows to achieve better visual quality for the same bit rate or save 5-15% of bit rate with the same visual quality (Wiegand *et al.*, 2003). The filter algorithm is standardized and enables the use of inter prediction on the filtered frames without the decoder drift.

The H.264 standard also has provisions for transmitting the data - the network abstraction layer (NAL). The image data and parameter sets are encapsulated into NAL units (NALU). The units can be either transmitted as a bitstream and specific bit patterns are inserted to mark the boundaries of each unit or framing can be done by the underlying mechanism, e.g. RTP packets. The NALUs carry parameter sets for the whole sequence and for individual pictures. The image data is sent in video coding layer (VCL) NALUs. The data can be flexibly formatted and less frequent, but more important data can be sent over error resilient channels and video data can be delivered by less robust means. There are also mechanisms for reducing the impact of packet loss:

- slices - the parts of the image are encoded independently. The slices can be processed in parallel with little or no dependency between them. Slices can be used to restrict the error propagation and to limit the size of NALUs to for the maximum packed size.
- constrained intra prediction where only intra coded samples are used for prediction. The inter coded samples are marked as unavailable as they can be corrupted.
- data partitioning - the macroblock header data and residuals can be transmitted in separate NALUs.
- redundant slices - approximation of the frame regions, typically with lower quality (to be used when the primary representation is lost).

2.3 Scalable Video Coding.

Scalable video coding (SVC) was standardized in 2005 by MPEG and VCEG as an Amendment G of H.246 standard (Schwarz, Marpe and Wiegand, 2007) (Figure 2-1).The standard's initial target was to enable transmitting several representations of the same video content, which can be decoded at different frame rates, resolutions and fidelity, depending on the equipment capabilities and network throughput. The essential requirements were:

- Coding efficiency similar to single layer coding - for each subset of the bit-stream. The result is more efficient than simulcast, but there is still a overhead of 10-50% for the same fidelity.
- Little increase in decoding complexity compared to the single layer approach, with linear scaling for spatio-temporal resolution.
- Support for temporal, spatial and quality scalability. The sub-streams can have different frame rate (e.g. 10/15/30 fps), resolution (QVGA/VGA/HD), parts of the frame, or fidelity.
- Support for a backward compatible base layer. The SVC stream can be decoded by a standard H.264 decoder for base layer presentation.
- Support for simple bit stream adaptations after encoding. For example, network node can strip unnecessary layers by dropping NALUs without decoding the underlying video.

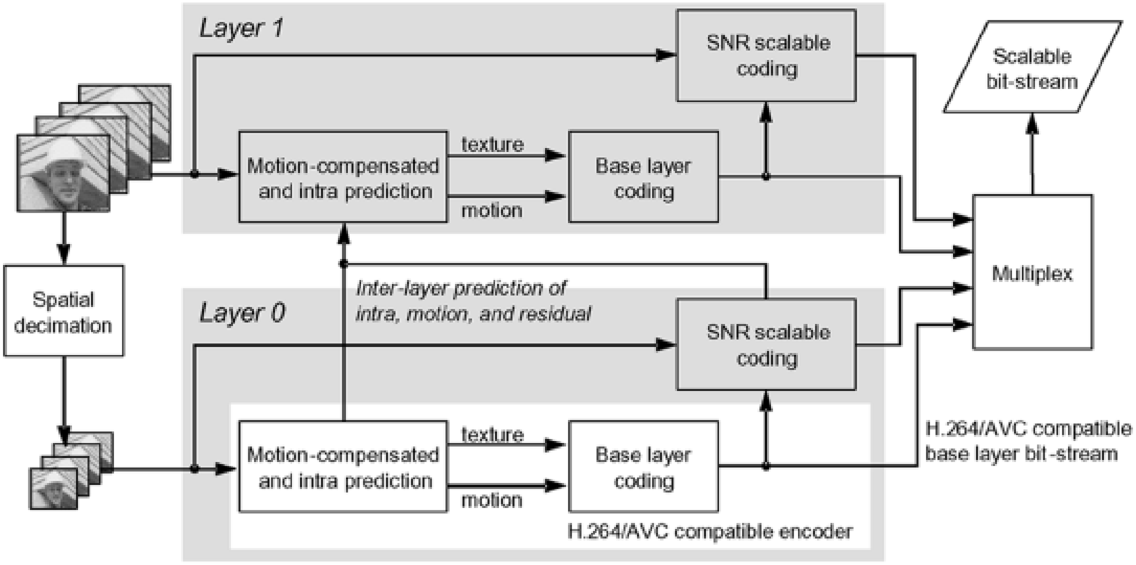

Layer 1
SNR scalable coding
Scalable bit-stream
Motion-compensated and intra prediction
texture
motion
Base layer coding
Multiplex
Spatial decimation
Inter-layer prediction of intra, motion, and residual
Layer 0
SNR scalable coding
Motion-compensated and intra prediction
texture
motion
Base layer coding
H.264/AVC compatible base layer bit-stream
H.264/AVC compatible encoder

Figure 2-1 Scalable Video Coding (Schwarz, Marpe and Wiegand, 2007)

The SVC standard is based on corresponding H.264 syntax elements. Extra coding tools include:

- temporal identifiers in NALU to handle the temporal scalability. The NALUs for redundant temporal layers can be dropped after header decoding. The encoder guarantees that lower temporal layers do not refer the pictures of higher temporal layer.

- spatial scalability is achieved by upscaling the base layers. Arbitrary crops and resolution ratios are supported. Extra macroblock syntax elements allow inferring the prediction modes, motion vectors, samples and residuals from co-located base layer macroblocks. The constraints on inter layer intra prediction allows single motion compensation loop decoding: inter coded samples of base layer are never used for higher layers, so motion compensation only within the decoded layer is used. This arrangement greatly reduces frame memory bandwidth requirements for the decoder.
- quality scalability is implemented as a limiting case of spatial scalability with 1:1 layer ratios. If the layers have matching macroblocks then higher layers can use the de-quantized transform coefficients of the base layer as a predictor and only refinement bits (with smaller QP) are transmitted. Furthermore, the enhancement layer transform coefficients can be distributed among several NALUs, the header signals the first and the last coefficient position in scan order.
- NALUs are inherited from H.264/AVC and an extra 3 byte prefix is added to signal temporal quality and spatial layers. SVC bit-stream also contains a H.264 compliant sub-stream. In order to add SVC layer information to the sub-stream prefix NALUs were introduced - they precede non-SVC units and only contain SVC header information.

There is a number of tradeoffs between the number of encoded bitrate points in the stream and the coding efficiency. On one hand having multiple quality points allows better adaptation for the network conditions. On the other hand, smaller steps between layers decrease the coding efficiency and lead to encoding similar information twice. Furthermore, switching between layers generally can be done only on key frames. To address this issue medium grain scalability was added to SVC design (Schwarz, Marpe and Wiegand, 2007). The decoder or the network can drop a subset of NALUs for the enhancement layer to trade off between bitrate and quality. However this would lead to drift between the encoder and the decoder as the decoder would reconstruct slightly different frame content compared to the encoder and use it for motion compensation. To address this issue the key pictures were introduced in the SVC design. The coarsest temporal layer frames are transmitted as key pictures and motion compensation is done within the base layer. These pictures are used as resynchronization points between the encoder and decoder. The coding drift is limited to the temporal refinement pictures. This approach allows the use of motion compensation, on the highest quality representation layer and thus to increase the coding efficiency.

Careful design of the SVC encoder allows to achieve temporal scalability without loss of rate-distortion performance, spatial scalability at typically 10% loss of compression efficiency (for dyadic spatial scalability) and 10% loss for compression for quality scalability with bit rate spanning the ratio of 2-3 between the highest and the lowest rate points (Schwarz, Marpe and Wiegand, 2007).

It should be noted that the SVC design does not support different bit per pixel setting in the base and enhancement layers. For example it is not possible to encode the base layer as 8 bits Main profile of H.264 and add 10 bit enhancement layer for high fidelity applications. Potentially the decision of not supporting such use cases was made because 10 bit display equipment was not widely available at the date of standardization.

2.4 H.265 High efficiency video coding

High efficiency video coding (HEVC) is the successor of H.264/AVC. Again, it is a joint effort of MPEG and ISO/IEC. The study began by VCEG in 2004 and in April 2013 the H.265 standard was ratified by ISO/IEC.

The high level HEVC encoder block diagram is presented in Figure 2-2. (Sullivan *et al.*, 2012)

The new standard achieves up to 50% reduction of the bit rate compared to H.264 with the same visual quality with similar decoder complexity and approximately 10 times more operations on the encoder side (Murakami *et al.*, 2010).

As with H.264 the performance increase was not due to one particular tool, but by incremental improvements over all aspects of the codec design (Sullivan *et al.*, 2012):

- Generalizing the macroblock approach to utilize a quad-tree structure. The HEVC can use coding blocks of up to 64x64 size which can be subdivided into block size down to 4x4 pixels. Such flexibility allows to code efficiently both large areas with smooth gradients (e.g. sky) and small details.

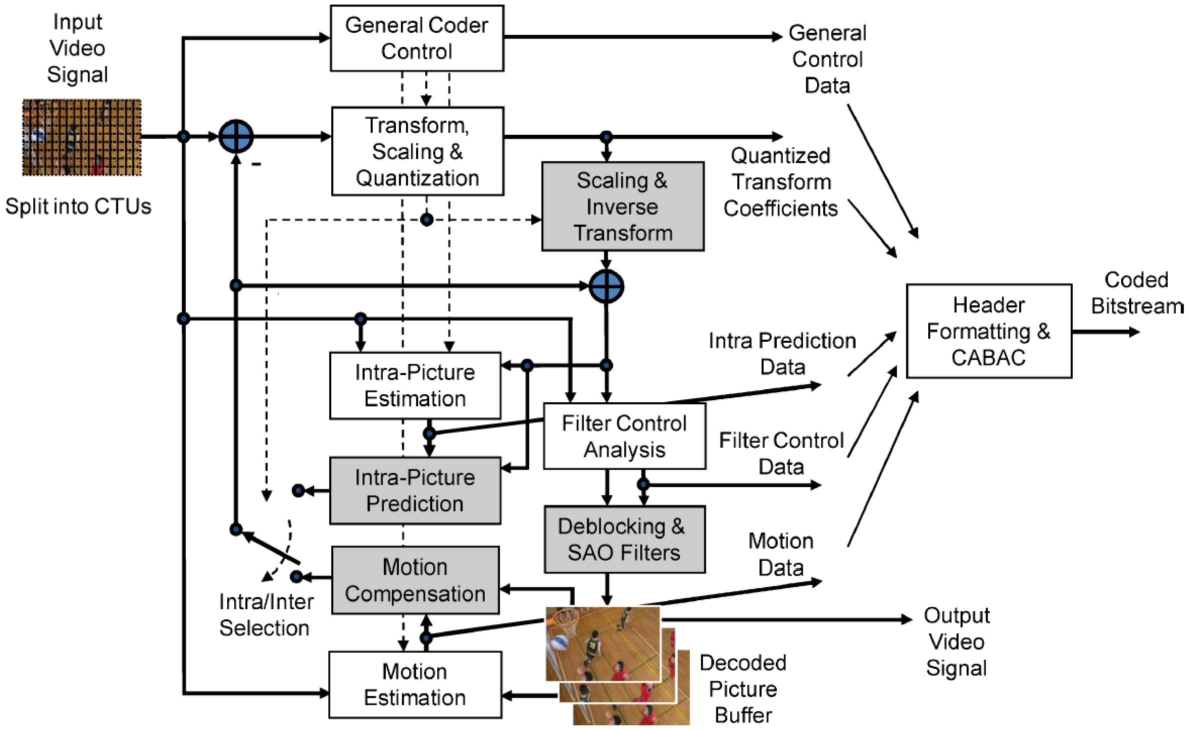


Figure 2-2 High Efficiency Video Coding (Sullivan *et al.*, 2012)

- Prediction can be intra frame and inter-frame. Both variants can use a variable block size from size 32x32 to 8x8. Intra prediction uses more directions than in AVC (32 directional plus 3 planar modes vs 6 directional and 3 planar). The directional modes use a unified algorithm suitable for different block sizes. Filtering near edges aims to reduce blocking artefacts. The Inter frame prediction supports asymmetric partitioning (where square block is divided in proportion 3:1 or 1:1) with variable block sizes. Motion vectors have 1/4 luma sample precision (same as AVC), but interpolation at half sample and quarter sample positions is done with 8 tap and 7 tap filters respectively. Wider filters allow better reconstruction and using linear filters for quarter sample positions (compared to 2 stage approximation in AVC) reduce number of rounding operations and allow more flexibility in calculations and better precision. As with AVC the encoder and decoder can use several reference frames, but in HEVC the signalling of used frames is explicit rather than incremental. This improves error resilience and simplifies stream editing operations. The signalling of the prediction modes was improved - the encoder and decoder predict three most probable intra modes or up to 5 motion vectors and only the index is transmitted if the prediction is correct, otherwise the index of the remaining 32 intra modes or motion vector difference is sent.

- The transform used is generalized DCT, with a basis matrix of 32x32 defined in the standard. Smaller size transforms (16x16 to 4x4) are inferred by skipping rows and columns within this matrix. Although all the components are integers the matrix has symmetry properties to allow fast DCT-line implementations. For 4x4 intra predicted blocks alternative 4x4 DST-like transform is specified. It improves coding efficiency by 1% by better matching the statistical properties of the residual amplitudes (Sullivan *et al.*, 2012).

- Entropy coding is CABAC (support of CAVLC was dropped). The main improvements were targeting more parallelized processing: more bits are coded in the bypass mode, less contexts, less dependency of the next context on previously coded bits.

- In loop filters were improved - the de-blocking filter was simplified (less number of boundary strength settings) and dependency on scan order was reduced (all vertical and all horizontal edge filtering can be done in parallel). Further a sample adaptive offset (SAO) filter was added. The filter has 3 modes of operation and can improve monotonic areas by reducing banding artefact in mode 1 or refine the reconstructed edges in mode 2. The filter parameters can be inherited from the block to the left or above by using the merge flag. The higher level syntax was extended to allow better error resilience, network encapsulation, editing operations, temporal scalability and efficient parallel processing.

- Tiles are independently decodable rectangular regions. The main purpose is to enable parallel decoding. They can also be used for decoding only a ROI of the image.

- Dependent slices improve the granularity of the packetization without loss of efficiency because of inter-slice prediction.

- Wavefront parallel processing (WPP) allows parallel decoding of the rows of CTB, but as each row starts 2 CTBs later than the row above the already decoded blocks are used to adapt the contexts of the arithmetic coder and use pixels and metadata for prediction.

- The frame types and decoding dependencies are signalled in NALU headers. This allows better flexibility in bitstream splicing operations while keeping the compression efficiency with complex frame dependency schemes. Further the temporal sub layer and the temporal switch pictures are signalled in the NAL header which allows achieving temporal scalability without parsing the NALUs.

2.5 High dynamic range imaging

The recent advances in sensor technology have resulted in the introduction of digital cameras with high dynamic range imaging capabilities. At present cameras with 100 dB dynamic range are produced commercially in a large scale and ones with even higher ranges have commenced commercial production. This advancement is widely accepted to be resolving many outstanding issues in the perceived quality of natural scenes in entertainment industry (TV, DVD, games etc.). These are also expected to play an important role in the surveillance industry, especially in outdoor and air-born surveillance camera systems, where the cameras have a high probability to be exposed to high dynamic range images (e.g., due to bright sunlight, fire, bright reflections, shadows).

There are several methods for creating HDR content. Some use computer graphics (CG) methods (Ward, 2001) to artificially create HDR content, whilst others use specifically built sensors and sensing technologies to capture natural scenes with high dynamic variation of pixel intensities. The dynamic range of an image sensor is determined by the photodiode used. The dynamic range is limited by the well capacity for the highlights and by the noise level in the dark areas. Some preliminary tests with modern sensors (from OmniVision, Panasonic, Altasens, Sony and other sensor vendors) show that the noise level is proportional to square root of the pixel intensity. This implies that the main noise source is apparently shot noise (if the photodiode accumulated N electrons during the frame it is expected that this figure will vary by √N counts). The sensor typically provides 50-70 dB dynamic range, before any image processing is performed.

There are several known methods for increasing the dynamic range of images:

- Improving sensor technology and the use of image processing algorithms. There are high quality CMOS sensors with 14 bit analog-to-digital converter (ADC) (Sony, 2013) which when combined with state of the art noise reduction algorithms (Apical, 2014) can achieve 75 dB of raw sensor data plus 6-12 dB gain in noise reduction, a 80-90dB range in total. This method does not have limitations and artefacts typically created by other approaches, but is expensive both in terms of sensor cost and operations required for the noise reduction.

- Multiple sensitivity sensors, for example Fujifilm Super CCD SR (dpreview staff, 2003). The sensor contains photodiodes of both a smaller size (to capture the highlights) and a larger size (to capture shadows). The digital processing combines the data from the sensors to form the final HDR image. The technology has the advantage of capturing the images with the same exposure time (reducing motion and flicker artefacts), but the imbalance of the sensor sites and charge bleeding from the oversaturated high sensitivity photocells, may reduce image quality.

- Multiple threshold sensors (Hertel *et al.*, 2008) - The saturation threshold is changed over pixel integration time allowing bright parts of the image to accumulate electrons over a shorter period of time. This approach uses standard CMOS sensor technology, and can benefit from all process improvements for SDR sensors. The main drawback is pixel-to-pixel variation of the effective threshold which is amplified by exposure ratio. This typically leads to strong fixed pattern noise which reduces the usable dynamic range.

- Multiple exposure operation - this is the most widely used method for HDR imaging. This method can be used for both still image capture and for video. The benefits of the method include the usage of the standard sensor technology and ability to precisely control exposure times. The main drawback of this method is that the exposures are taken in different instants in time, so moving objects may change their position between the frames. The post processing algorithms typically make motion detection or motion compensation before blending the exposures, but there are still fundamental problems with gap filling - some of the areas are either over exposed in the long exposure frame or occluded in short exposure frame. This may lead to 'ghosting' or to 'noise trails' after moving objects. Another significant problem for such sensors is mains flickering when artificial light is used. As the artificial light flickers with two times the mains frequency, the actual scene intensity changes with the period of 10ms or 6.7ms. As the long and short exposures are integrated over different amount of time the measured intensity ratio is no longer proportional to the exposure ratio and varies over time. This is hard problem to solve and typically the surveillance cameras turn off the HDR mode when flicker is detected.

Other methods for HDR image acquisition include: colour filter array modulation (Nayar and Mitsunaga, 2000), adaptive integration method (Punchihewa, Hamamoto and Kojima, 2011) and lin-log sensors (Chou *et al.*, 2014).

2.6 Complexity of the compression algorithms

In this thesis the term 'complexity' will be used within a limited scope as the more established asymptotical big O estimations are not very informative in the context of the research conducted. Firstly, all of the compression algorithms used in this thesis have O(N) complexity with respect to image size. As the algorithm needs to at least access each image pixel before compression, no improvement on this estimate is expected. The main difference between algorithm performance lies in the constant hidden in the O() notation. It can vary several orders of magnitude for different algorithms (Murakami *et al.*, 2010). As one of the areas of the research is video compression for surveillance applications it means that the frame encoding should be completed in a predefined amount of time, typically 33 milliseconds or 16 milliseconds on average. Although the encoding time of individual frame may take slightly more than this average, it means that other frames should either be completed in shorter time or dropped from being encoded altogether.

The factors that determine the complexity of an algorithm include:

- Number of processor operations (e.g. additions/multiplications) per frame
- Number of conditional branches in the algorithm (non-linear control flow affects the pipelining of the operations)
- Amount of storage memory required (e.g. for all the reference frames for video encoding)
- The size of the local working set, typically proportional to the macroblock size (the required processor cache size)
- Number of independent processing threads (potential speedup when multiple cores are available)
- Number of bit manipulation operations (typically very inefficient on general purpose microprocessors)
- Length of non-parallelizable dependence loops (e.g. the next bit of CABAC encoding can not be output before the previous bit is processed)

These factors should be taken into account during algorithm development and system implementation.

Chapter 3

Literature Review

3.1 Introduction

This chapter provides an overview of the prior art in HDR image and video compression. Initially it reviews the algorithms for lossy and lossless image compression and subsequently presents existing algorithms for HDR video compression. Further, alternative compression schemes that uses less widespread algorithms or data representation schemes are also reviewed. Finally the chapter concludes with a review of the literature on system design - both from a perspective of using the algorithm and implementing it in silicon.

The HDR image and video compression algorithms proposed in literature can be classified in a number of different groups. The taxonomy is illustrated in Figure 3-1.

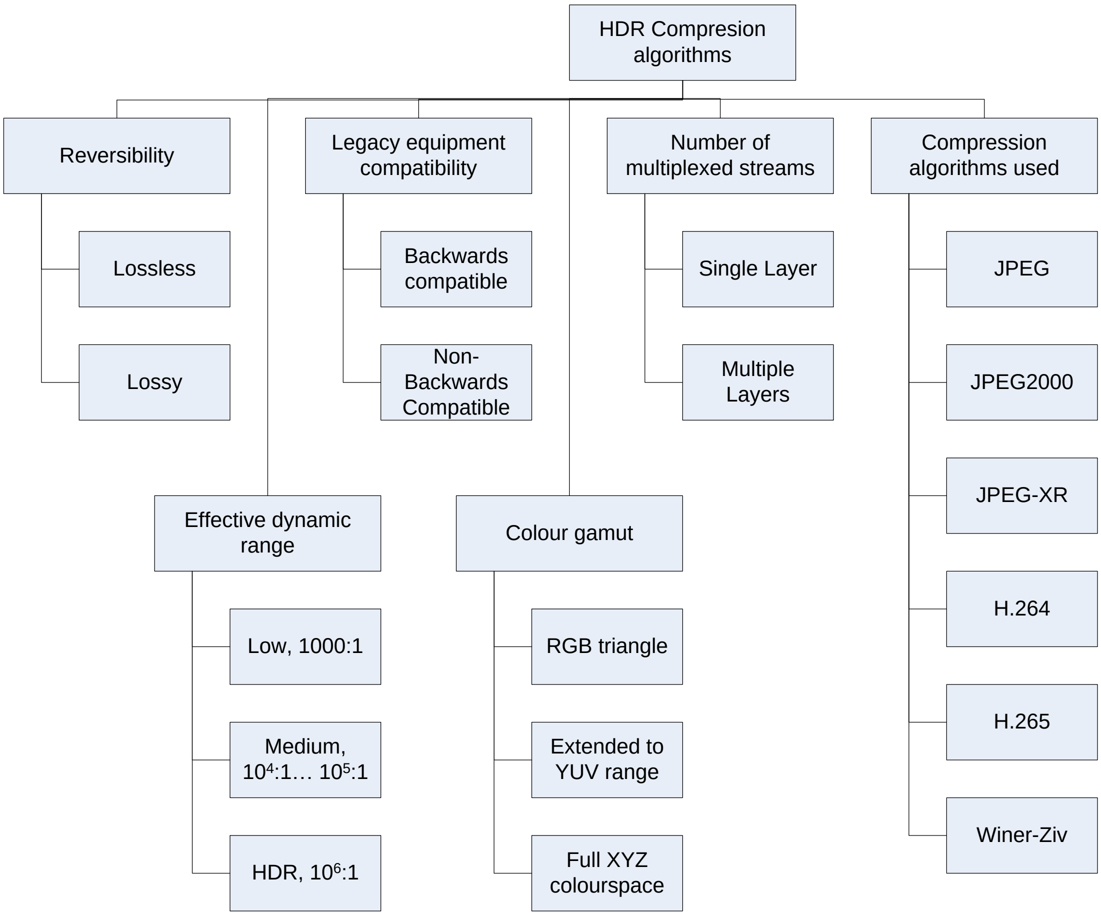


Figure 3-1 Taxonomy of the HDR compression algorithms

3.2 HDR image compression

There are some well known lossless formats for storing HDR image content. For example, TIFF (Adobe, 1992) supports a single precision floating point representation. Although this format is virtually lossless, most of the bits are wasted in storing an image with much higher precision than the human eye can ever determine.

Radiance RGBE uses three 8 bit R, G, B channels and a group exponent E as the fourth channel. The output value is calculated as $R_{out} = R \cdot 2^{E-128}$. Although this method is easy to encode and decode, the colour gamut is limited by primaries triangle. The perceptual distance between codes is far from uniform, making gradations visible at saturated blue and magenta (Ward, 2005) colours.

The Industrial Light and Magic company (ILM) introduced OpenEXR standard uses half float representation of the colours. To improve compression efficiency wavelet transform and PIZ compression are utilized.

Ward Larson (1998) proposed LogLuv encoding. The main compression gain is due to better quantization of the visible colourspace. Lightness L is log encoded, u', v' are normalized and do not depend on brightness. This format is implemented in TIFF library using 24 and 32 bits per pixel.

Mantiuk et al (2006) introduced a perceptually uniform colourspace based on the integral of detection threshold versus intensity functions.

The existing La*b* and Lu*v* colourspace are only approximately uniform. The just noticeable distance can vary from 1 to 4 in different places. Bezryadin et al (2008) introduced the Bef colourspace, based on Cohen’s metric, which is much more uniform over the colour gamut.

The division of HDR compression algorithms into lossy and lossless categories is rather artificial. Lossless compression for LDR images implies that the decoder will produce the same bit pattern as was at the input of encoder. In HDR domain the input of encoder is typically data in IEEE floating point format. Lossless HDR compression means that the quantization errors will never exceed detectable threshold. Lossy HDR formats allow significantly lower bit rates, but some areas of the image can be distorted.

There are different approaches for 'lossy' compression of HDR images. From the point of backward compatibility they can be broadly grouped into three categories: non-backwards compatible, reusing the existing algorithms for new content and backward compatible with the existing LDR compression algorithms.

One example of non-backward compatible algorithms was introduced by Xu et al (2005) who addressed the high dynamic range compression problem by logarithmic encoding the R, G, B components. Although this did not give advantage in lossless modes over the OpenEXR format, the lossy performance was better than JPEG-based HDR methods, such as that of Ward and Simmons (2004).

Okuda and Adami (2007a) improved this approach by using the LogLuv colourspace (Reinhard *et al.*, 2005). This technique enabled coverage of the full gamut of perceivable colours and more perceptually uniform quantization. De-correlation of the channels also served to improve the coding efficiency.

Kaida and Okuda (2008) used a hill function to precondition data before compressing by SPIHT wavelet transform. This function enables to allocate more bits for visually more important intensities.

Springer and Kaup (2009) further improved Okuda and Adami's method by utilizing contrast weighting functions to improve on rate distortion optimization.

Microsoft proposed HD Photo standard for storing HDR images (Srinivasan *et al.*, 2007). This standard was adopted by as ISO/IEC standard 29199-2. Another name for this standard is JPEG XR. Even though JPEG2000 - based methods outperform JPEG XR in quality for a given bit rate (see, for example, (Springer and Kaup, 2009)), the compression complexity is generally less.

Another approach to building the HDR compression algorithm is using existing compression engines (JPEG, MPEG etc) and perceptual colourspaces, such as Lu*v* or LogLuv (see Appendix 1).

Mantiuk et al (2006) first converted the image to luv colourspace similar to LogLuv. Chromacity components u',v' are calculated using the same formula as LogLuv, but the luminance is transformed through the companding curve consisting of linear, power and logarithmic pieces. The quantization step after such transformation closely matched the threshold-versus-intensity curves. This enables to encode full dynamic range in 12 bits. JPEG encoding was extended to 12 bits to store the data with better precision.

Full LDR stream compatibility has the important advantage that full interoperation with legacy software and hardware remains in tact. Most algorithms reuse as much information as possible from the LDR layer and add an enhancement layer for the values which cannot be reproduced in LDR due to clipping in bright areas, insufficient quantization in dark areas and gamut limitations.

Ward and Simmons (2004) store tone mapped images as a standard JPEG stream. The ratio of HDR and corresponding LDR images is encoded and sent as subband data. The gamut is extended by saturation companding prior compression and applying reverse transform during decompression. Further the full range of the YUV colourspace was used. This approach enabled better colour representation near the saturated areas.

Mantiuk et al (2006) used an inverse tone mapping function to de-correlate the information in LDR and HDR layers. The enhancement layer contains differences between results of the inverse tone mapping converted to the luv colourspace. To improve compression further and reduce artefacts, the enhancement layer is filtered to mask invisible differences.

Okuda and Adami (2007b) also use a 2 layer encoding approach in which the inverse tone mapping is approximated by the Hill function and the residual ratio image is compressed using wavelet transforms.

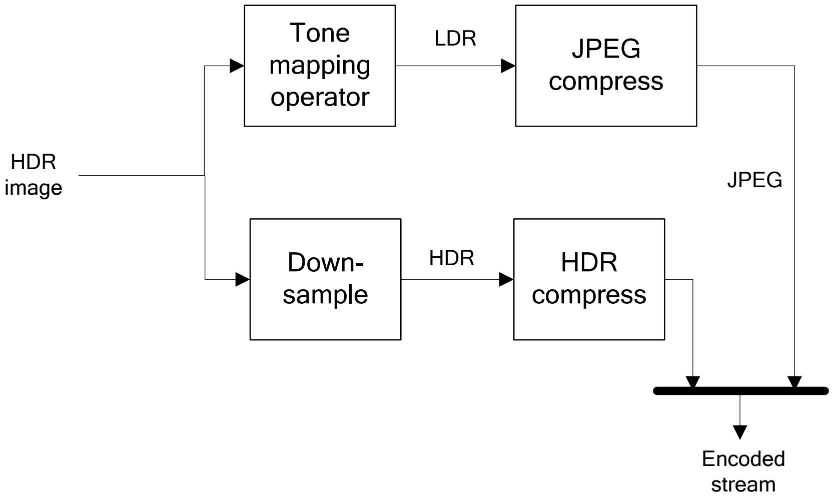


Figure 3-2 HDR compression algorithm (Ward, 2006)

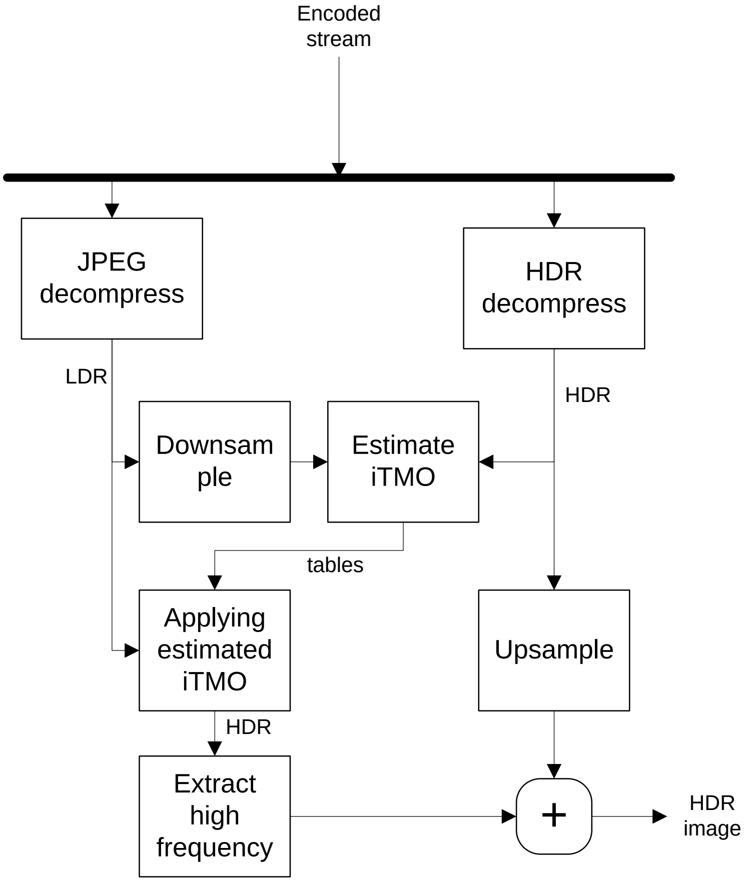


Figure 3-3 HDR decompression algorithm (Ward, 2006)

Ward (2006) introduced a general approach to the distribution of HDR content. During compression the tone mapped result is stored as the LDR layer and a downsampled HDR image is stored as enhancement layer (Figure 3-2). No restrictions are imposed on compression and tone mapping algorithms. When the HDR image is upsampled during decompression the higher frequencies are taken from the LDR image (Figure 3-3). For local tone mapping operators the technique for restoring impulse response function is introduced.

3.3 HDR video compression

Similar approaches can be used for HDR video compression.

Mantiuk et al (2006) introduced a perceptually uniform colourspace based on the integral of detection threshold versus intensity functions. The image data is then compressed using a conventional MPEG encoder. The usage of a perceptually uniform colourspace allows exploiting standard rate-distortion optimization techniques for bitrate control.

Zhang et al (2011) implement a non-backward compatible scheme. The HDR image is LogLuv transformed and then undergoes non-linear Lloyd-Max quantization followed by wavelet-based process of detail masking according to contrast sensitivity curve. The filtered result is encoded with 14 bit H.264 encoder.

Scalable Video Coding (SVC) extension for H.264 was ratified as annex G in 2007 (*H. 264*, 2013) It defines the way of multiplexing the base and enhancement layers for video coding and a framework for spatial, temporal and quality scalability. However it is difficult to adapt SVC directly for high dynamic range video encoding. The standard does not allow different bit depth for the base and enhancement layers, and bit depths above 8 are not supported by most profiles and the JSVT reference model. The bit depth 8 is not sufficient for the enhancement layer. Furthermore, in case of quality scalability there is no provision for a non-linear relationship between layers and the transform coefficients of residual are predicted from the base layer instead of decoded image data.

Segall (2006) proposed a backward - compatible two layer CODEC. The base layer is tone mapped LDR content and the enhancement layer is encoded similarly to quality enhancement for Scalable Video Coding. The inter-layer intra prediction uses inverse tone mapping of the base layer to predict the samples of HDR stream. The inverse tone mapping is modelled as a linear relationship with linear transform parameters transmitted for each macroblock.

Winken et al (2007) proposed an enhancement to the current SVC standard, where base layer is tone mapped LDR content, and the refinement layer is the application of inverse tome mapping operator to the base layer. The inverse tone mapping operator is global and can be chosen between a linear relation, piecewise-linear relation and full lookup table.

Mantiuk et al (2007) use MPEG encoding of the tone mapped LDR frame as a base layer and globally - optimized inverse tone mapping curve for prediction of HDR content from LDR. The difference between predicted HDR and actual HDR content is filtered and invisible noise is masked using contrast sensitivity functions and encoded as a separate MPEG stream.

Ward (2006) has introduced a general approach to distribution of HDR content. The tone mapped result is stored as the LDR layer and a downsampled HDR image is stored as the enhancement layer. No restrictions are imposed on compression and tone mapping algorithms. When HDR image is up-sampled, the higher frequencies are taken from LDR image. For local tone mapping operators, a technique for restoring impulse response function is introduced.

Koz & Dufaux (2012) use perceptually uniform colour transform to find the optimal global piecewise linear curve to decorrelate the LDR and HDR layers. Use of a perceptually uniform colourspace also enhances the tone mapping operator's visual appearance.

Liu et al (2008) use a two layer codec structure where motion compensation is done on base LDR layer only to reduce computational complexity. The inverse tone mapping is modelled as a linear relationship between LDR and HDR content. The linear transform parameters (per each macroblock) are predicted from neighbours and entropy encoded.

New compression standards, for example H.264 and H.265 have provisions for higher bit depth and multi-layer coding which facilitate design of HDR compression algorithms, In this work we explore the methods of optimization of two and one layer scheme image quality and computation time by adopting perceptually uniform colourspaces and residual encoding algorithms.

The key to effective video compression are effective entropy coding methods, efficient decorrelation of the data (to avoid encoding same or similar information twice) end efficient bit allocation – using more bits for visually important parts of the image.

Pioppo et al (2006) discusses the usage of foveation to improve the bitrate with minimal impact on visual image quality. The algorithm finds the fixation point on the image and uses finer quantization in the area of interest and coarser quantization in the remaining areas.

Segall et al (2009) discusses a low complexity compression scheme for multiple exposure data. The sensor emits alternating frames with different exposure times. The subsequence of frames with one of the exposures is encoded using standard methods and forms the base layer. The alternative exposure information is predicted from the base layer (using information about exposure differences) and the residual is encoded as an enhancement layer. The computational burden of registering and fusing the images with different exposures is moved to the decoder side.

Chen et al (2010) provides in-depth overview of the existing and emerging approaches to encoding video using perceptual metric. The review includes ROI-based and multiview methods.

Barkowsky and Callet (2010) explored the correlation between different visual difference prediction operators and visual rankings of the human observer. The research confirmed the relevance of Mantiuk's VDP approach.

Velastin and Remagnino (2006) give the overview of the challenges and applications within wide-area surveillance systems.

3.4 Alternative compression schemes

The literature review highlighted growing interest in alternative compression schemes for LDR and HDR video. Classic compression algorithms are asymmetric and efficient encoding is significantly more computationally intensive compared to decoding. Although this is not a limitation for a broadcasting scenario where one encoder feeds multiple decoders, it is preferable to reduce the encoder complexity in the CCTV use case where encoder is in an embedded device and the data stream may be just recorded and never seen until some incident happens.

One of such approaches is compression of CFA data from sensors (Koh and Mitra, 2003), (Lukac and Plataniotis, 2006), (Lee and Cho, 2010). The sponsoring company experience suggests that the data should still be processed before compression (e.g. noise reduction) and benefits of such scheme are questionable. The amount of the data before compression is similar in CFA and classic YUV scenarios (typically 12 bits/pixel), but in order to preserve enough data for efficient noise reduction and colour interpolation algorithms, the compression efficiency should be increased. Furthermore, the CF image consists of 4 colour planes which are on one hand strongly correlated because of content and the overlap of filter spectral characteristics, but on the other hand the deviations are used for recovering of fine details of the image. The correlation parameters are not constant through the image and the change is dependent on content. All these factors are obstacles to efficient compression, as without de-correlation, the coding efficiency will drop significantly and compression artefacts will lead to noticeable colour interpolation errors.

Another approach is compression of alternating short and long exposure frames (Segall, Zhao and Rubinstein, 2009), (Mangiat and Gibson, 2011). This approach is backwards compatible, but still does not exploit fully the redundancy between exposures. The benefits include minimal modifications to existing camera and moving process of registering the exposures to the decoder side. One of the challenges of this approach is non-linear dependency between exposure time and pixel value – because of non-linearity of pixel level processing and a global tone mapping curve. Although this is not an unavoidable obstacle (Mitsunaga and

Nayar, 1999) it requires a close integration of the image processing front end and the encoder.

The distributed video coding (Pereira, 2009) can be used to encode intermediate frames. The application typically include intra-coding of the reference frames and WZ coding of the intermediate frames. The main obstacle of WZ coding is correct approximation of the correlation parameters of the source. Estimation on the transmitting side diminishes the benefits of distribution coding approach, for example by making computationally intensive motion search. Some schemes use estimation on the receiving side and a low bandwidth back channel to transmit just enough bits to decode, but this approach requires low network latency (or large amount of memory for frame buffers on the encoder and decoder side) and not applicable for video recording scenarios.

3.5 System design

A model of a CCTV installation was chosen as the target for practical implementation of the proposed compression algorithms. The CCTV analytic field is growing rapidly and as a result it is more likely to adopt any new technology compared, for example, to cameras used within broadcast and media distribution. Usually the CCTV camera equipment is chosen by the installers, not the end users and high dynamic range capabilities are already deemed to be required in this market sector.

The systems aspect of the CCTV installations has different levels – from wide area sensor networks to installation units (cameras, wiring, and observation rooms) and individual camera designs. The question of the purpose of camera installations was raised by (Walters, 1995). The conflicts of interests of the parties involved in CCTV system installation were identified and quality metrics were indicated. The wide area networks provide further challenges in processing and storing the incoming information as the number of cameras increases and the installation is often on an ad hoc basis (when new equipment is gradually added to the system without careful planning, but only to close the holes in the area subject to constraints on camera placement). Even tracking of a suspected person across different cameras can be challenging. Understanding the performance requirements for the whole CCTV system and flowing them down to camera design and compression engine design is a complex research project (see also (Leszczuk, Stange and Ford, 2011), (Alston and Campbell, 2010) and (Denman *et al.*, 2011)) .

Another area for systematic design is the camera itself. The optimal design requires matching the performance of the all the camera components – optics, sensor, imaging pipeline, compression engine, signal processor – to achieve image quality and other design constraints (cost, power dissipation, processing delay etc). Other CCTV system elements – transmission network, storage database and operator room ergonomics – also need careful consideration to achieve required system goals (e.g. security at reasonable cost).

Another line of enquiry that emerges is the High Efficiency Video Coding (HEVC) algorithm. The goal of the HEVC project was achieving next generation of video compression, i.e. processing high resolution video (e.g. 4k and 8k UHDTV) at increased compression rate. The algorithm achieves 3.5-4 times smaller bitrate than MPEG-2 or 1.7-2 times better than H.264 (Ohm *et al.*, 2012). The decoding complexity is comparable with H.264 and specific measures were taken to facilitate parallel decoding. The encoding complexity is 5-10 times larger than H.264 (Murakami *et al.*, 2010). As with H.264 standard the coding gain was achieved by incremental improvements of individual parts of algorithm. The approach is to use wider palette of the coding tools and allow the encoder to choose best ones for a given frame or frame area and signal the choice to the decoder. For example, for intra-frame prediction there are now 35 modes compared to 9 in H.264. The ‘macroblocks’ can be up to 32x32 in size (compared to 16x16 in H.264) and can be divided into smaller units down to 4x4 using hierarchical quadtree. The entropy coding uses a binary arithmetic codec with improved context adaptivity. The Draft International Standard was approved in July 2012 (Ericsson, 2012). The reference implementation was made available though Fraunhofer Institute and is currently being integrated into framework for encoding experiments. The proposed backwards compatible HDR video compression algorithm in this thesis (see Chapter 6) can gain from HEVC coding tools, e.g., decreased bitrate of the base layer due to mode efficient coding, a increased precision of the data due to the flexible macroblock division structure suitable for encoding HDR residual. Here, large areas of small variations are combined with individual pixels or clusters with large discrepancy, where the HDR scene exceeds gamut or dynamic range of the base layer.

The increased computational complexity of modern of video codecs and requirements for low power consumption led to the growth of system-on-chip devices. The modern SoCs consist of multiple processor cores – specialized and general purpose, hardware accelerators for specific algorithms, graphics accelerators, peripherals and memory controllers on a single die. Single chip solution reduces cost and size of the mobile devices. The energy consumption is one of the main constraints in SoC design because of several factors – slow progress in portable power supplies, market demand for increased operational time, heat dissipation (hot device in inconvenient to hold and heating sensor in camera increases noise), safety considerations (battery with high energy density can be dangerous). The requirements for exponential growth of the operations per second (Berkel, 2009) to be done within the system lead to a highly integrated system and as the design cycle for new IC remains the same or shrinks, the automated tools for designing and modelling such complex SoCs are vital. One of the research directions was the investigation of interoperability of general purpose system level design tool (Topcased/SysML/MARTE) and SoC design tool (Daedalus, 2012). The aim was to create an adaptation layer and allow automatic generation and simulation of the multi-core SoC from the SysML/MARTE model. The scripting language (Python) was used to translate the model from standard OMG XMI format to domain-specific format of system description in Daedalus (Daedalus, 2012)

The current trend for mixing and matching the CCTV equipment imposes a strong requirement on interoperability with non-HDR equipment.

Backwards compatibility requires a multi-layer stream format, similar to scalable video coding (SVC). The SVC standard (Schwarz, Marpe and Wiegand, 2007) allows spatial, temporal and quality scalability within the same bitstream. Although the bitrate is typically greater than the bitrate of the non-scalable stream by 10%, the approach is more efficient than simulcast. Another trait of the SVC standard is that the bistream can either be decoded by a conventional H.264 decoder (which omits SVC-specific data) and can be rewritten to an AVC compliant stream, without full decoding, but just parsing the packet headers and discarding extension packets. The proposed algorithm uses a similar framework of extra NAL units, which reference the related units in the base video stream. The video compression algorithm also uses single loop decoding to reduce the decoder and encoder computation complexity. According to (Schwarz, Marpe and Wiegand, 2007) single loop increases bitrate by 5-7%.

In the contributory chapters of this thesis presented next, we use the research background knowledge and literature review related findings summarised above for the purpose of proposing novel approaches to HDR image and video representation and coding.

Chapter 4

High Dynamic Range Still Image Compression

4.1 Introduction.

In the previous chapter several algorithms for compression of HDR images were explained. One of the general approaches for backwards compatible encoding was presented by Ward (2006) In this scheme the tone mapped result is stored as the LDR layer and a downsampled HDR image is stored as enhancement layer. No restrictions are imposed on compression and tone mapping algorithms. When the HDR image is upsampled the higher frequencies are taken from the LDR image. One of the disadvantages of this algorithm is the increased complexity of the decoder. Not only it has to decompress both streams, but in case of local tone mapping operator, complex calculations are required to restore the impulse response function. Global tone mapping operators are easier to handle, but they have worse performance.

Local tone mapping operators model the adaptivity of the human eye. As Johnson and Fairchild (2006) mentioned, the radius of spatial adaptation corresponds to 1/4 to 1/2 of image size. In that case the calculation of local inverse tone mapping function can be done on a very coarse grid (10x10 to 20x20 gave good results during experiments). Each function can be represented as piecewise-linear function with limited amount of nodes. This approach enables to shift the calculation burden towards the encoder and enables very efficient GPU acceleration of this operation - just one 3D texture lookup per pixel.

The calculation of the inverse tone mapping table can be performed by either sorting the pixel intensities in the pixel neighbourhood as proposed by Ward (2006) or by using weighted least squares fitting where the weights used decreases with the distance from the node position. The least squares method requires just one pass over the image to accumulate the necessary sums. It can be accelerated using GPU hardware.

The inverse tone mapping allows predicting of the HDR content from LDR images with higher accuracy and enables to store the difference information as the HDR layer. When the

LDR content is no longer required the HDR image can just omit LDR layer and inverse tone mapping table and transmit HDR data as single layer.

Compact representation of tone mapping operator can have other advantages: first of all, the typical usage of HDR image assumes tone mapping to device with dynamic range, in between LDR in HDR content. The tone mapping table can greatly accelerate the tone mapping algorithm. For some applications there will be no need for decoding HDR layer at all. Secondly when HDR only content is distributed, the relatively small tone mapping table will allow speeding up the image adaptation for the particular device.

4.2 Overall codec structure

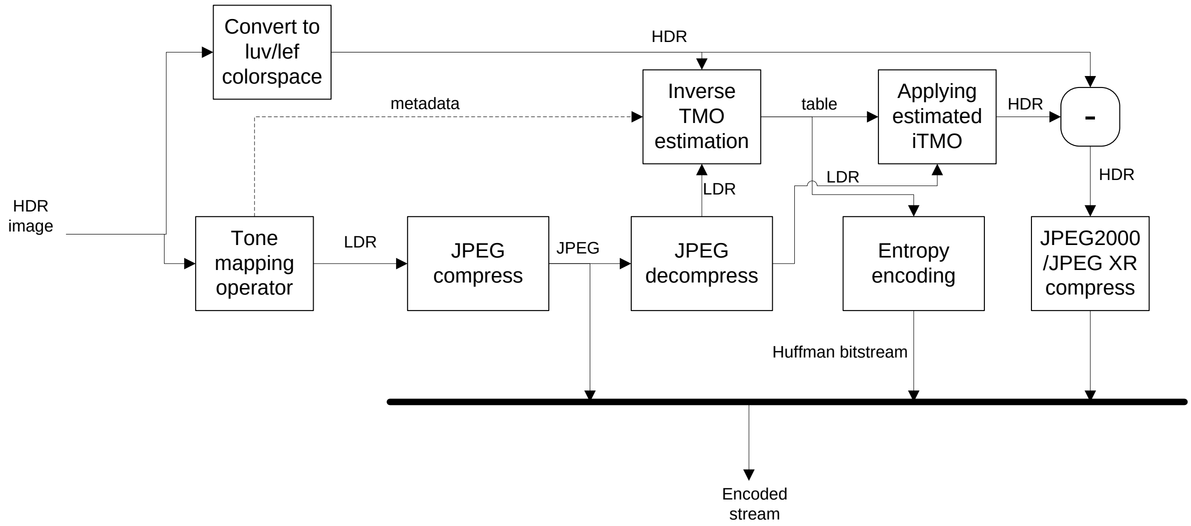


Figure 4-1 Proposed HDR encoder structure

The encoding algorithm (see figure 4.1) starts from the tone mapping of the HDR image in CIE XYZ colour coordinates to create the base layer with LDR content, which will be available for the users with legacy decoders. The LDR content undergoes JPEG compression. At the same time the HDR content is subjected to colour conversion transform, for example to luv or to the lef colourspace. These colourspaces are more perceptually uniform than the original XYZ colourspace. The transformed HDR content along with the decoded LDR content is used to estimate the inverse tone mapping operator. The metadata from the forward tone mapping operator can be used in this step if available. The estimated inverse tone mapping operator is applied to the decoded LDR content and the result is subtracted from the HDR content. The residual is encoded using an image compression algorithm with 14 or more bits of precision – JPEG2000 or JPEG-XR. The inverse tone mapping operator table is compressed (by the combination of a linear predictor and Huffman encoder) and multiplexed into the output bitstream.

The decoding algorithm reassembles the HDR image in the following order: first the LDR image is decompressed, then the inverse tone mapping operator is decompressed and applied to this image, At the next step the HDR residual is decompressed and added to the tone mapped result. Finally the colourspace is converted back to CIE X,Y,Z to obtain the final image.

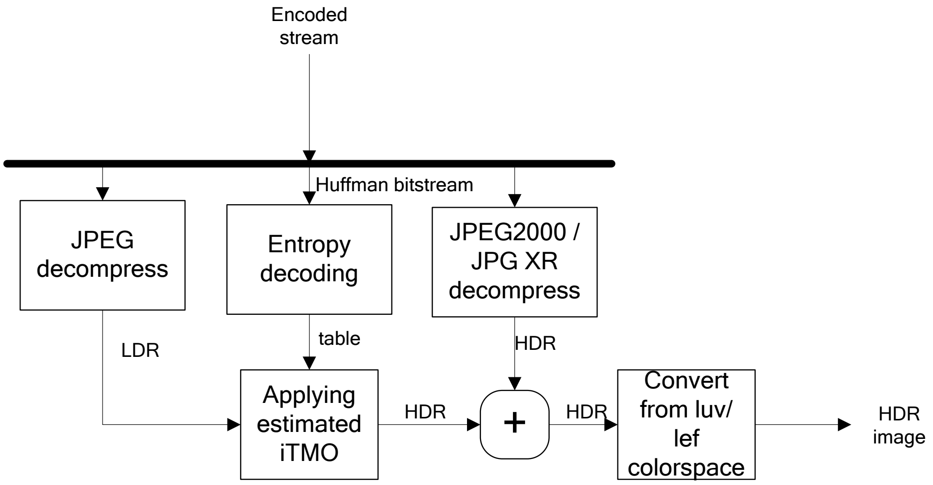


Figure 4-2 Proposed HDR decoder structure

The decoder first decompresses the LDR content, and then the inverse tone mapping operator parameters are decompressed and inverse TMO is applied to the LDR content to create the predictor. Then the residual is decompressed and added back to the predictor. The last step is conversion to the target colourspace (Figure 4.2).

Unlike Ward's algorithm (2006) (Figure 3-2 and Figure 3-3) the compression-decompression technique in encoder allows correcting compression artefacts of LDR layer. The calculation of the inverse tone mapping operator can be either 'blind' - the only assumption of spatial smoothness on TMO. Sometimes additional information can be used to speed up the calculations. For example, in Rahman's Retinex (1996) the impulse response curve of the operator is linear or logarithmic function at any given point of the image. In either case the algorithm runs on the encoder side where full information about LDR and HDR content is available.

The details of each block will be explained in the following sections

4.2.1 Calculation of inverse tone mapping tables

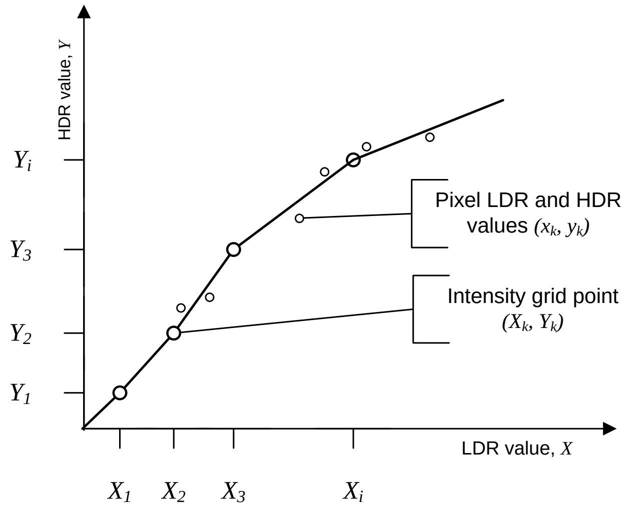

HDR value, Y
Y_i
Y_3
Y_2
Y_1
Pixel LDR and HDR values (x_k, y_k)
Intensity grid point (X_k, Y_k)
LDR value, X
X_1
X_2
X_3
X_i

Figure 4-3 Mapping in the intensity domain

The tone mapping tables can be calculated using the least squares method. Let x_k be LDR pixel intensities and y_k be HDR pixel intensities in a given neighbourhood Ω of a given grid node. The positions X_i of the nodes of piecewise linear approximation are fixed, typically they are uniformly spaced (Figure 4-3). The node values Y_i should minimize the cost function:

$$E = \sum_{i=1}^{N} \sum_{k \in K_i} w_k \left[\frac{Y_i - Y_{i-1}}{X_i - X_{i-1}} (x_k - X_{i-1}) + Y_{i-1} - y_k \right]^2 \tag{4.1}$$

where

$$K_i := \{k : x_k \in [X_{i-1}, X_i]\} \tag{4.2}$$

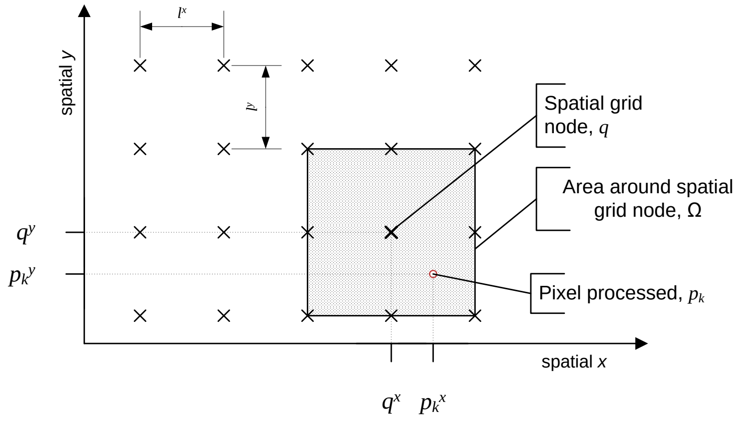


Figure 4-4 Mapping in the spatial domain.

If p_k is the spatial position of pixel k and q is spatial position of given grid node on the image, then

$$w_k = (1 - \frac{|p_k^x - q^x|}{l^x})(1 - \frac{|p_k^y - q^y|}{l^y}) \qquad (4.3)$$

where superscript denotes the coordinate component and l^x, l^y are distances between grid nodes (Figure 4-4).

The formula (4.1) is a quadratic form with regard to Y_i, which can be minimized by setting derivatives to zero:

$$\frac{\partial E}{\partial Y_i} = 0 \qquad (4.4)$$

This leads to the system of the linear equations:

$$A_m Y_{m+1} + B_m Y_m + C_m Y_{m-1} = D_m , \qquad (4.5)$$

$$m = 1..N-1$$

with boundary conditions:

$$Y_o = 0 \;\; , Y_N = \max Y \qquad (4.6)$$

and linear equation coefficients:

$$A_m = -\alpha_m^2 S_m^{xx} + \alpha_m S_m^x$$

$$B_m = \alpha_{m-1}^2 S_{m-1}^{xx} + \alpha_m^2 S_m^{xx} - \alpha_m S_m^x + S_m^0 \qquad (4.7)$$

$$C_m = -\alpha_{m-1}^2 S_{m-1}^{xx} + \alpha_{m-1} S_{m-1}^x$$

$$D_m = \alpha_{m-1} S_{m-1}^{xy} - \alpha_m S_m^{xy} + S_m^y$$

where

$$\alpha_m = \frac{1}{X_m - X_{m-1}}$$

$$S_m^0 = \sum_{k \in K_m} w_k$$

$$S_m^y = \sum_{k \in K_m} w_k y_k \qquad (4.8)$$

$$S_m^{xy} = \sum_{k \in K_m} w_k y_k (x_k - X_m)$$

$$S_m^{xx} = \sum_{k \in K_m} w_k (x_k - X_m)^2$$

The system of equations can be easily solved by Gaussian elimination in $\Theta(N)$ operations as the matrix is very sparse and only has three adjacent non-zero diagonals.

4.2.2 Compression of the tone mapping tables

Although the structure of the tables is not very large, further reduction can be achieved by quantizing coefficients to 12 bits and using lossless compression methods for each of intensity planes. If $\omega_{x,y,i}$ is quantized value of Y_i for position (x, y) and intensity i, then the predictor is defined as following:

$$\begin{aligned} \overline{\omega}_{x,y,i} &= \omega_{x-1,y,i} + \omega_{x,y-1,i} - \omega_{x-1,y-1,i}, && y > 1, x > 1 \\ \overline{\omega}_{x,1,i} &= \omega_{x-1,1,i}, && x > 1 \\ \overline{\omega}_{1,y,i} &= \omega_{1,y-1,i}, && y > 1 \\ \overline{\omega}_{1,1,i} &= \omega_{1,1,i-1}, && i > 1 \end{aligned} \tag{4.9}$$

The difference $\omega_{x,y,i} - \overline{\omega}_{x,y,i}$ is Huffman encoded. To save the alphabet and decrease the size of the table an approach similar to encoding of DC coefficients of JPEG is utilized (Wallace, 1992). The entropy encoded code word is a prefix representing the length of the binary string which immediately follows. Such compression enables to reduce the size of the table typically by 40%.

4.2.3 Encoding of the HDR content

The high dynamic range image is encoded using *luv* colourspace (Mantiuk, Myszkowski and Seidel, 2006):

$$l = \begin{cases} a \cdot y, & y \le y_l \\ b \cdot y^c + d, & y_l < y \le y_h \\ e \cdot \log(y) + f, & y > y_h \end{cases} \tag{4.10}$$

$$u = \frac{410 \cdot 4X}{X + 15Y + 3Z} \tag{4.11}$$

$$v = \frac{410 \cdot 9Y}{X + 15Y + 3Z} \tag{4.12}$$

The values for parameters $a \ldots f$ correspond to VDP contrast sensitivity function (Table 4.1).

Values for l are quantized to the nearest integer and stored as a 12 bit channel. Values for u, v are also quantized to the nearest integer, but 8 bit is sufficient for storage. The difference image has one extra bit for the sign.

Table 4.1 The *luv* colourspace parameters

a	769.18	e	181.7
b	449.12	f	-90.160
c	0.16999	y_l	0.061843
d	-232.25	y_h	164.10

The u,v colour coordinates do not produce perceptually uniform encoding. The Bef2 colourspace introduced by Bezryadin et al (2007) provides better uniformity across RGB gamut (Bezryadin, Burov and Tryndin, 2008). The Bef2 colourspace is defined as:

$$\begin{pmatrix} D \\ E \\ F \end{pmatrix} = \begin{pmatrix} 0.2053 & 0.7125 & 0.467 - \\ 1.8537 & -1.2797 & -0.4429 \\ -0.3655 & 1.0120 & -0.6104 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \qquad (4.13)$$

$$\begin{aligned} & B = \sqrt{D^2 + E^2 + F^2} \\ & e = E / B \\ & f = F / B \end{aligned} \qquad (4.14)$$

where X,Y,Z are CIE colour coordinates.

The definition for brightness gives correspondence $Y = B / \sqrt{2}$ for D65 illuminant, so this factor was introduced into formula (4.16) when calculating the perceptually uniform 'luminance' component for lef colourspace.

In our experiments with the lef colourspace the e, f coefficients were multiplied by 472 and quantized to 9 bits with sign (eq. 4.15).

$$\begin{aligned} & B = \sqrt{D^2 + E^2 + F^2} \\ & e' = 472 \cdot E / B \\ & f' = 472 \cdot F / B \end{aligned} \qquad (4.15)$$

$$l' = \begin{cases} a \cdot B / \sqrt{2}, & B / \sqrt{2} \le y_l \\ b \cdot (B / \sqrt{2})^c + d, & y_l < B / \sqrt{2} \le y_h \\ e \cdot \log(B / \sqrt{2}) + f, & B / \sqrt{2} > y_h \end{cases} \qquad (4.16)$$

4.3 Experimental setup

In the first round of the experiments three different variants of the HDR CODEC were analyzed for compression ratio versus image distortions:

- The first approach uses Ward's scheme (2006) where a full resolution LDR image is transmitted along with downscaled HDR content and a lookup table for the inverse tone mapping operator.
- The second approach stores residual HDR image at full resolution as proposed above. On one hand, this approach does not replicate low frequency information which is already encoded in LDR image, but on the other hand the residual image is mostly noise-like, with sharp edges corresponding to gamut clipping.
- The third approach uses non-backward compatible HDR encoding, similar to Okuda & Adami (2007b).

The Table 4.2 summarizes implementation advantages and disadvantages of each method.

Table 4.2 Comparison of the compression methods

	Advantages	Disadvantages
Ward (2006), downscaled HDR image with full scale LDR	-Faster HDR decompress -Not tied to HDR compression standard	-Higher frequencies are taken from the LDR image. JPEG artifacts reintroduced to HDR image. -Lower frequencies are already encoded in LDR image.
Inverse tone mapping (iTMO) and residual in JPEG2000	-Avoiding compression of low frequencies twice.	-Larger JPEG2000 image, longer recompression. -Handling of extended gamut may require attention.
luv in JPEG2000	-Most efficient of three.	-Not backwards compatible

Comparison of the performance of the first two approaches would explore the importance of the de-correlation of the HDR and LDR layers and transmitting the high spatial frequency components of the HDR content.

Comparison of the performance of the first two algorithms with the third one would highlight the overhead of implementing backward compatibility requirement in the CODEC.

The image set used in experiments is presented in Figure 4-5. The tone mapping operator used is Pattanaic's (1998) multiscale observer model as a local tone mapping operator providing naturally looking images.

The quality of compressed images was assessed by HDR Visual Difference Predictor (Mantiuk *et al.*, 2005). As HDR VDP metric predicts only luminance difference, mean distance in CIE L*a*b* colourspace was used to estimate colour fidelity (see also (Okuda and Adami, 2007b)) as following:

$$MDLab = \frac{1}{S}\sum\sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \tag{4.17}$$

where S is image area and sum is calculated over all image pixels.

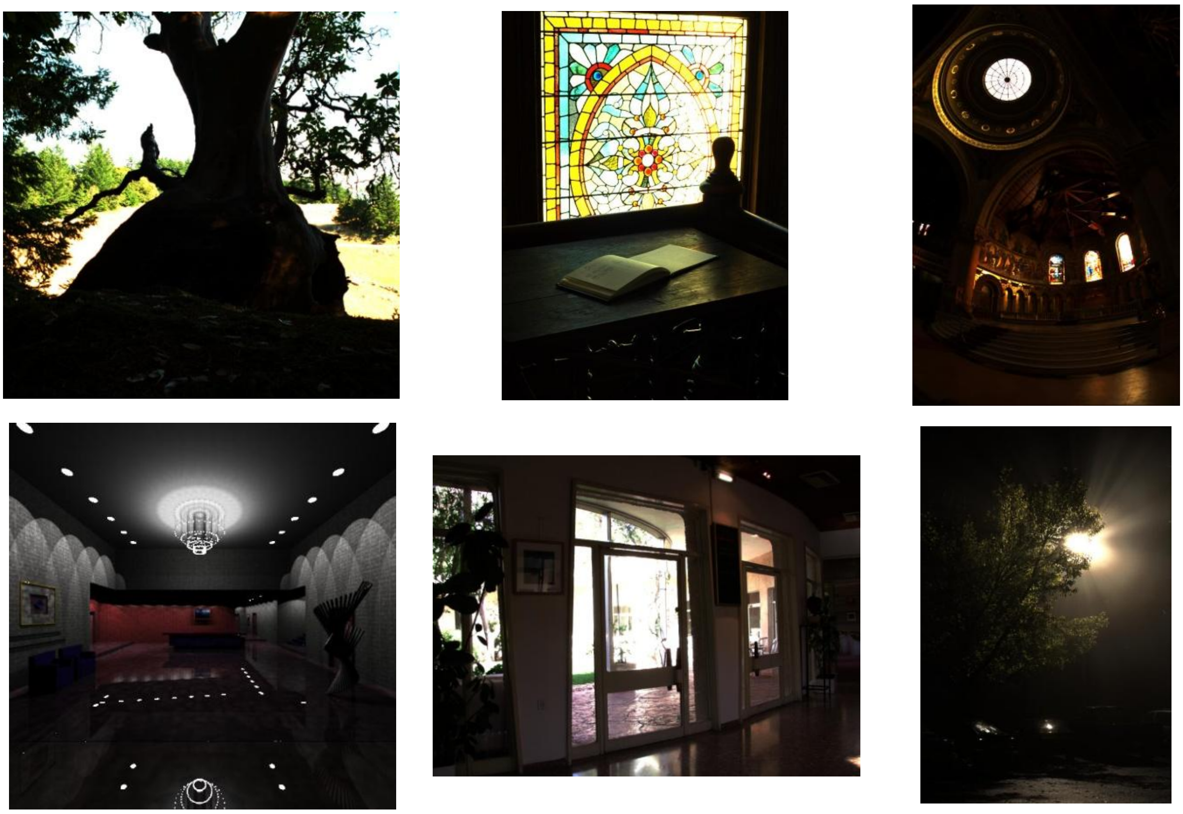


Figure 4-5 Examples of HDR images used in the experiments (Reinhard *et al.*, 2005)

The second round of the experiments targeted optimizing the CODEC structure by comparing the performance of different tone mapping operators and different compression algorithms.

Three different variations of the proposed CODEC were analysed:

The first series of experiments stores residual HDR. On one hand, this approach does not replicate low frequency information, which is already encoded in the LDR image, but on the other hand the residual image is mostly noise-like, with sharp edges corresponding to gamut clipping. JPEG 2000 and JPEG XR were tested as the residual codec, for both luv and lef colourspaces.

The second series of experiments uses non-backward compatible HDR encoding as in Okuda and Adami (2007a). Different colourspace encodings were utilized – Bef, luv, lef and floating point RGB. The experiment shows the 'cost' of backwards compatibility in terms of bitstream overhead and explores the importance of perceptually uniform coding.

The third series uses JPEG2000 and JPEG XR for storing floating point data (Srinivasan *et al.*, 2008). The encoding resembles a logarithmic encoding. This encoding has being standardized in JPEG XR.

Table 4.3 summarises implementation advantages and disadvantages of each methods.

Table 4.3 Advantages and disadvantages of the compression methods

	Advantages	**Disadvantages**
Residual in JPEG2000	- Avoids compression of low frequencies twice.	- Larger JPEG2000 image, computationally intensive codec - Handling of the extended gamut may require attention.
luv/lef in JPEG2000	- More efficient.	- Not backwards compatible
Floating point in JPEG2000, JPEG XR	- Simple encoding. - Standard for JPEG XR	- Less compression efficiency because encoding is not perceptual.

The tone mapping operator used in the experiments is Pattanaic's (1998) multiscale observer model.

The quality of compressed images was assessed by HDR Visual Difference Predictor (Mantiuk *et al.*, 2005). As HDR VDP metric predicts only the luminance difference, the mean distance in CIE L*a*b* colourspace (4.17) was used to estimate colour fidelity (see also (Okuda and Adami, 2007b))

Another metrics for colour fidelity was MDef – mean distance on ef plane of the Bef colourspace.

$$MDef = \frac{1}{S}\sqrt{(e_1 - e_2)^2 + (f_1 - f_2)^2} \qquad (4.18)$$

4.4 Experimental results

The first round experiments showed that the table representation of the inverse tone mapping operator is relatively compact (4-8 kilobytes for 20x20x20 table). As the Figure 4-6 shows the precision of the inverse tone mapping is important for correct de-correlation of LDR and HDR layers and improves the coding efficiency. The correct choice of forward and backward tone mapping operators can reduce the 'compatibility overhead' in VDP metrics to 10-30% of the stream size.

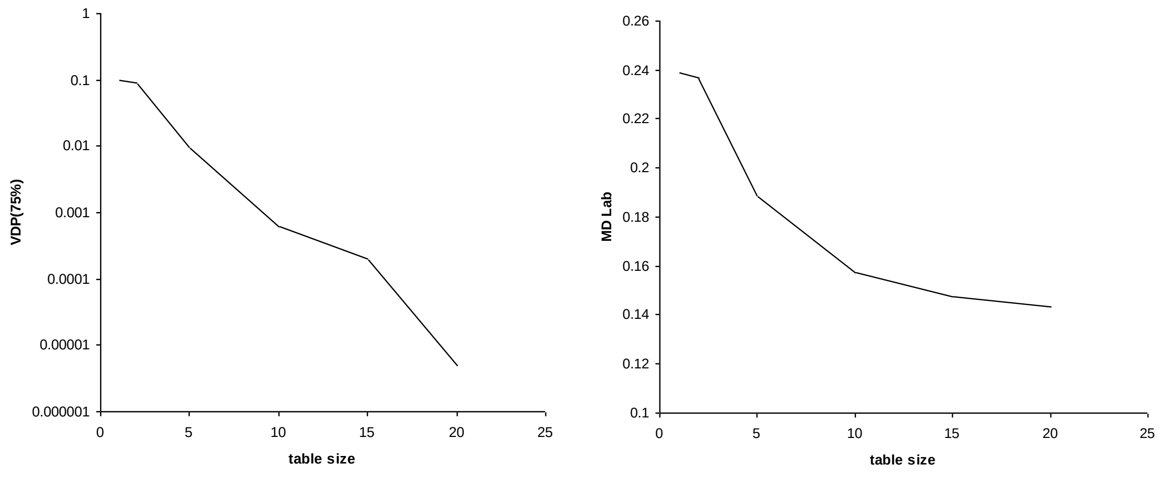


Figure 4-6 VDP and Mean distance in L*a*b* for different inverse tone mapping table sizes

The experiments also show that the coding efficiency of the approach with residual enhancement layer is better than that of downscaled HDR image (Figure 4-7Figure 4-8). The former method avoids replicating the same information in LDR and HDR layers. The positive effect of de-correlation outweighs the problems with sharp edges and noise of the residual image. Experiments also suggest that the more bits are allocated to the HDR enhancement layer the better overall performance of the codec is.

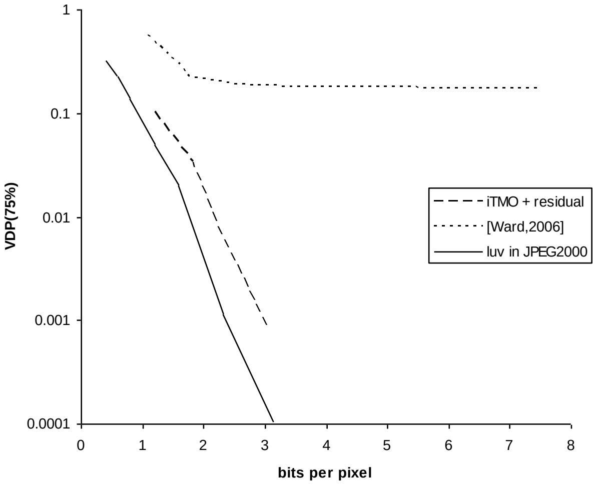


Figure 4-7 VDP metrics of the tree methods

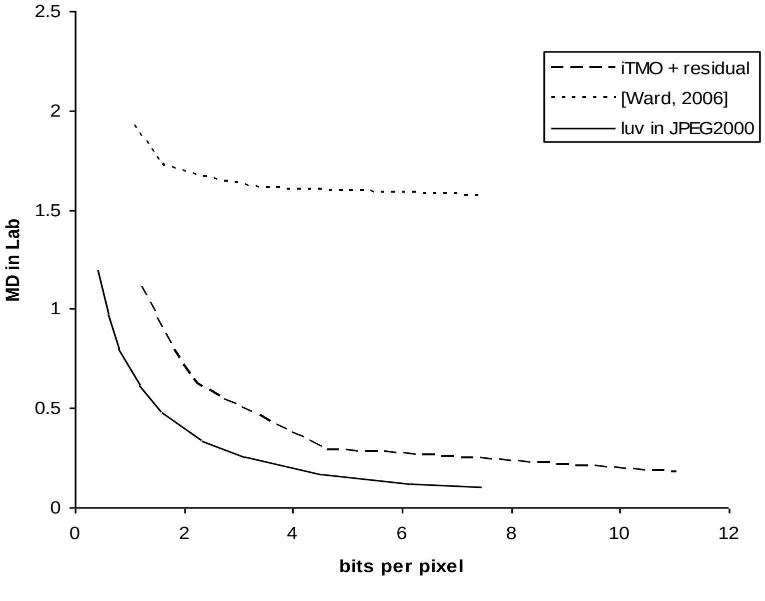


Figure 4-8 Mean distance in L*a*b*

The experiments revealed that the HDR layer in many cases consists of the noise, which is invisible on the final image. The coding efficiency of and the visual quality the scheme can be improved by masking of the invisible noise proposed in (Mantiuk, Myszkowski and Seidel, 2006) and contrast sensitivity weighting introduced in (Springer and Kaup, 2009).

In the second round of the experiments the measurements and visual comparisons consistently favoured the use of the JPEG 2000 algorithm over JPEG XR, which enabled achieving 30-50% smaller bitrate in both backward compatible and HDR-only scenarios. The results when using JPEG XR may be improved in future if the new revisions of the encoder would use better rate distortion optimization.

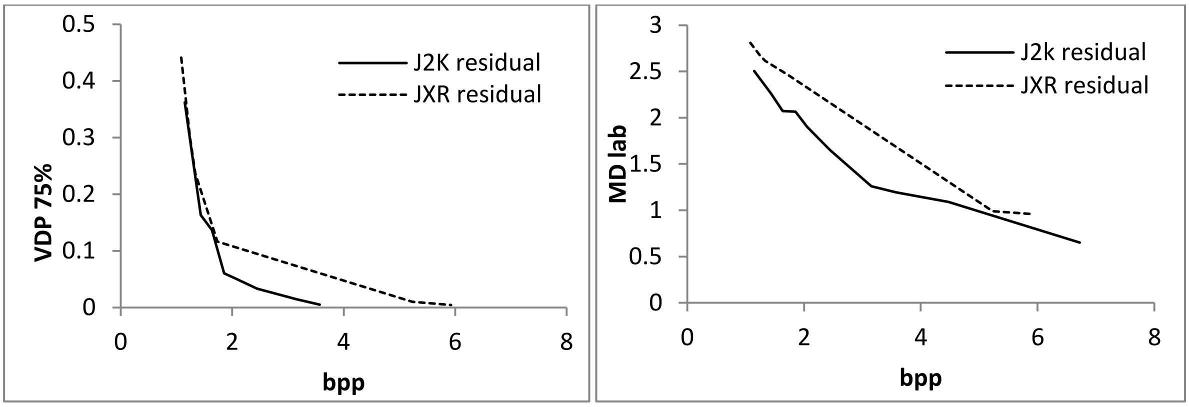


Figure 4-9 VDP and mean distance in L*a*b* for two layer codec: JPEG2000 and JPEG XR

In the first series of the second round of experiments the performance of two high bit depth CODECs was compared: JPEG2000 vs JPEG-XR. These CODECs were used for compression of the HDR residual using the two-layer scheme. The results are illustrated in Figure 4-9. The results show that JPEG2000 consistently outperformed the JPEG-XR for medium bit rates (2-4) bpp, although performance for lower bitrates was comparable.

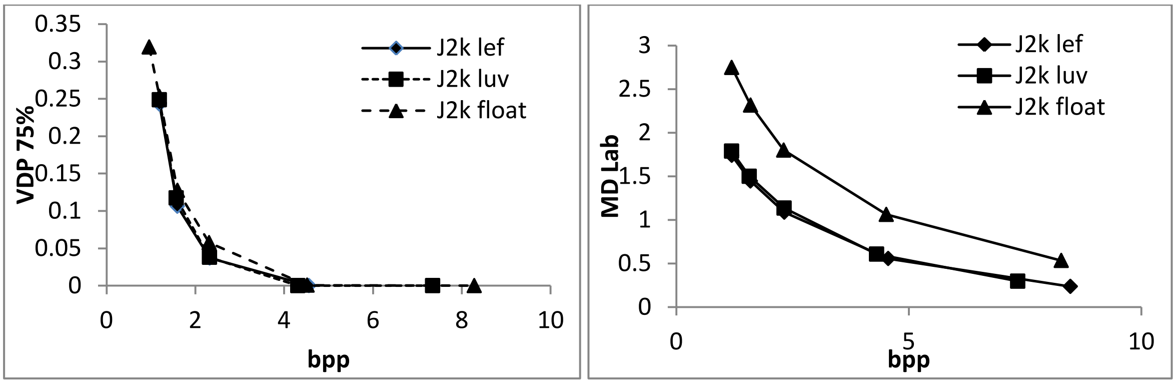


Figure 4-10 VDP and mean distance in L*a*b* for different colourspaces

In the second series of the second round of experiments, different colourspaces were used for the HDR datapath of the two layer CODEC. The compression algorithm for the enhancement layer was fixed to JPEG2000. Figure 4-10 illustrates the results.

The comparison of different colour conversions for HDR-only compression revealed the importance of perceptually uniform colourspaces. While the difference between luv and Bef colourspaces was insignificant, they consistently outperformed floating point formats.

The experiments revealed that the HDR layer in many cases consists of the noise, which is invisible on the final image. The coding efficiency and the visual quality of the scheme can be improved by masking of the invisible noise after Mantiuk et al (2006) and contrast sensitivity weighting after Springer and Kaup (2009).

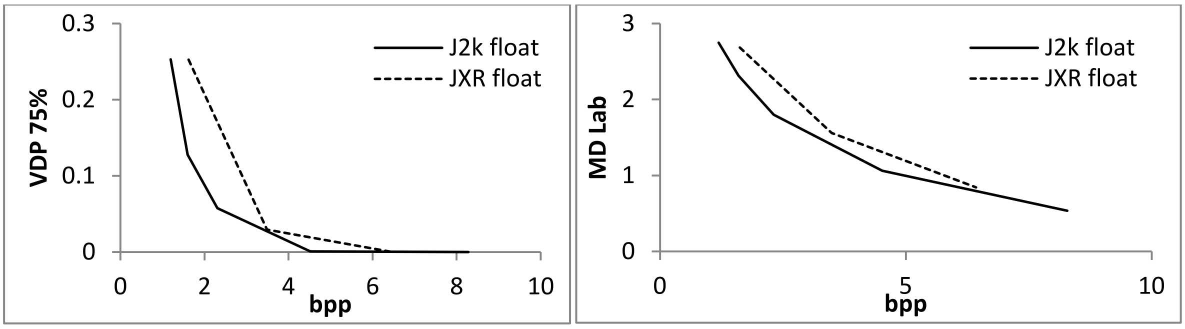


Figure 4-11 VDP and mean distance in L*a*b*for JPEG 2000 and JPEG XR

The third series of the second round of experiments used the non-backward compatible single layer operation. The Figure 4-11 illustrates the results. The HDR data was converted to float by proposed JPEG-XR extension (Srinivasan *et al.*, 2008). It the data was compressed either by JPEG-XR or JPEG2000. The results showed 20% bitrate saving when using JPEG2000 CODEC

4.5 Conclusion

The HDR image CODEC structure with a non-uniform inverse tone mapping operator was introduced. The representation of the inverse tone mapping function does not depend on the particular structure of TMO and allows very compact representation in the coded stream. We have shown that the increased precision of the de-correlation of LDR and HDR encoded layers and using perceptually uniform colour spaces increased the coding efficiency.

It was shown that the use of perceptually uniform colour spaces increased the coding efficiency. Using floating point encoding reduces coding efficiency (by 10-50% in file size). The JPEG XR algorithm in HDR layer offers worse performance than JPEG 2000 and its use can be justified only when computational resources are limited.

The format introduced not only provides backward compatibility with the existing LDR equipment, but also enables support of HDR-only content with even better coding efficiency using the same structure of HDR decoder.

Chapter 5

High Dynamic Range Video Compression

5.1 Introduction

In the previous chapters a novel method for compressing high dynamic range images was proposed. One of the main goals in its design was to propose a backward compatible algorithm which allowed to encode, decode and present the HDR images using conventional SDR equipment. The solution proposed used a two layer encoding approach where the base layer is the tone mapped SDR image and enhancement layer is the difference between the source HDR content and one predicted from the SDR layer that is transmitted. Further a method for compact and versatile representation on the inverse tone mapping operation was introduced. The representation exploited the characteristics of adaptation of the human visual system and was shown to be suitable for a wide range of tone mapping operators. Furthermore, the research of colour spaces for the representation of the HDR layer showed a strong advantage of using perceptually uniform spaces rather than ad hoc colour spaces based on power law or logarithmic encoding. In this chapter the methods proposed for HDR image compression in Chapter 4 will be extended to compression of HDR video.

5.2 Overview of the proposed method

Most video compression techniques use a global tone mapping operator model. While this approach greatly reduces the computational effort for encoding and decoding, it leads to reduced compression efficiency of the HDR content, i.e. results in either lower visual quality or an increased bitrate. Other algorithms (Ward, 2006) require complex calculations on the decoding side to recover the impulse response curve.

In contrast local tone mapping operators model the adaptivity of the human eye. As Johnson and Fairchild (2006) mentioned, the radius of spatial adaptation corresponds to 1/4 to 1/2 of image size. In that case the calculation of local inverse tone mapping function can be done on a very coarse grid (10x10 to 20x20 gave good results during experiments). Each function can be represented as a piecewise-linear function with a limited amount of nodes. This approach enables to shift the calculation burden towards the encoder and enables very efficient GPU acceleration of this operation - just one 3D texture lookup per pixel.

The calculation of the inverse tone mapping table can be performed by either sorting the pixel intensities in the pixel neighbourhood (Ward, 2006) or by using weighted least squares fitting where the weights used decreases with the distance from the node position (Dolzhenko, Chesnokov and Edirisinghe, 2010). The least squares method requires just one pass over the image to accumulate the necessary sums. It can be accelerated using GPU hardware (for example for GPU-assisted histogram collection (Shams and Kennedy, 2007)). Current implementation also uses temporal smoothing of the sums to reduce variations of the tone mapping table and increase its compression ratio.

The inverse tone mapping allows the prediction of HDR content from LDR images with higher accuracy and enables the storage of the difference information as the HDR layer. When the LDR content is no longer required the HDR image can just omit the LDR layer and the inverse tone mapping table and transmit HDR data as single layer.

Compact representation of the tone mapping operator can have other advantages: first of all, the typical usage of HDR content assumes tone mapping to the dynamic range of the display device, which is in between LDR and HDR content. The table can greatly accelerate the tone mapping algorithm. For some applications there will be no need for decoding the HDR layer at all. Secondly when HDR only content is distributed, the relatively small tone mapping table will allow speeding up the image adaptation for the particular device. Furthermore, the actual dynamic range of the mobile displays can change significantly with changes of ambient light and backlight (Chesnokov, 2011). The tone mapping metadata allows the image to be dynamically adjusted to current viewing conditions.

However, the relative size of the tone mapping lookup table becomes significantly large in case of video compression: most frames in a video are B or P frames with very low bitrate. Transmitting the lookup table every frame without exploiting temporal redundancy may significantly reduce the compression efficiency. Given these observations in section 6.2 we proposed a novel HDR video compression CODEC that is capable of achieving significant gains in compression efficiency.

5.3 The proposed CODEC

Figure 5-1 and Figure 5-2 respectively illustrate the block diagrams of the structure of the proposed encoder and decoder.

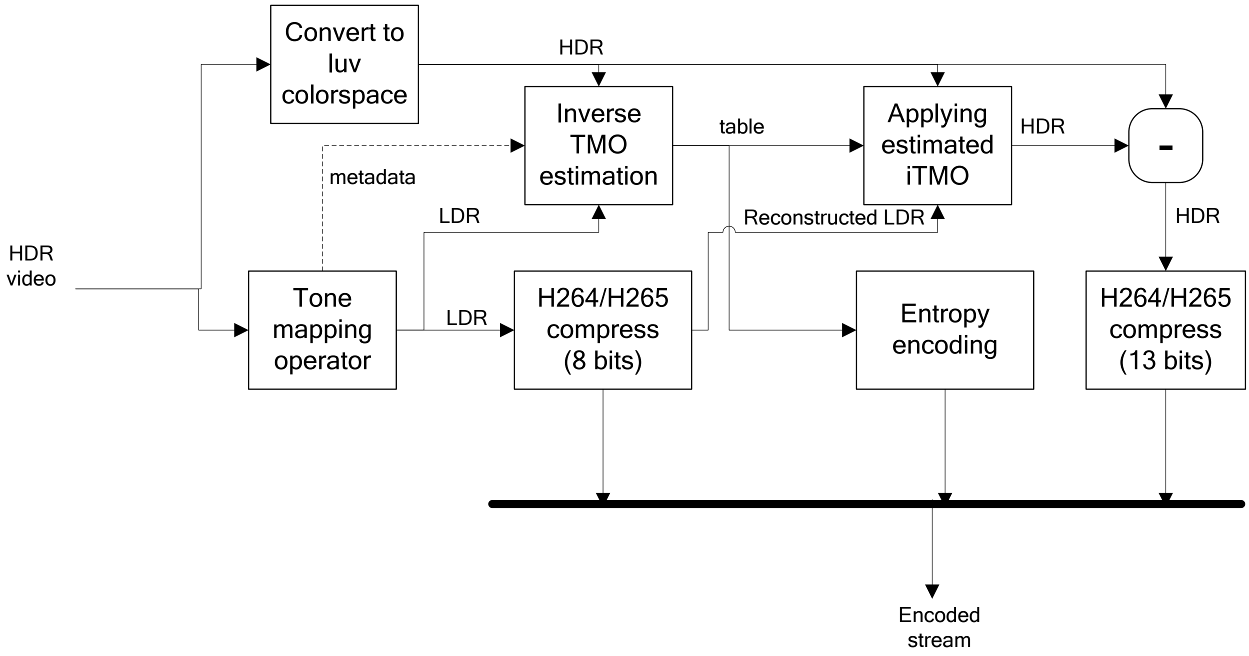


Figure 5-1 Proposed two layer HDR video encoder

Figure 5-1 illustrates the proposed encoder structure which is common to all two layer codecs. The LDR video is created by using a tone mapping operator and compressed for backward compatibility of the base layer video stream using standard 8 bit encoding (e.g. H264/H265). As a result of the encoder prediction loop operation the exact representation of the decompressed frames at the decoder end is available. At the same time the image data is converted to a perceptually uniform colourspace (luv) for encoding with higher bit width. The inverse tone mapping is estimated from the HDR and LDR images, using additional data from the forward tone mapping operator if available. The goal of the inverse tone mapping estimation is to achieve maximal de-correlation of the LDR and HDR content. The inverse tone mapping is applied to the aforementioned decompressed frame data to predict the HDR content. The residual between predicted and actual HDR content is encoded using the enhancement layer video CODEC.

The inverse tone mapping estimation algorithm uses the *source frame* LDR data to calculate the transform parameters. At the same time the *decoded* LDR frames are used to predict the HDR content. The quantization-dequantization loop prevents from round-off error accumulation.

As the tone mapping operator works on larger scale the difference between using decoded LDR image versus original LDR image for parameter estimation is negligible, but this arrangement allows decoupling of image pre-processing (forward and inverse tone mapping) with block based encoding and leads to reduction of the coding delay. The decoded LDR image is a by-product of the H264/H265 encoding loop and the HDR enhancement layer can be encoded in parallel with LDR layer.

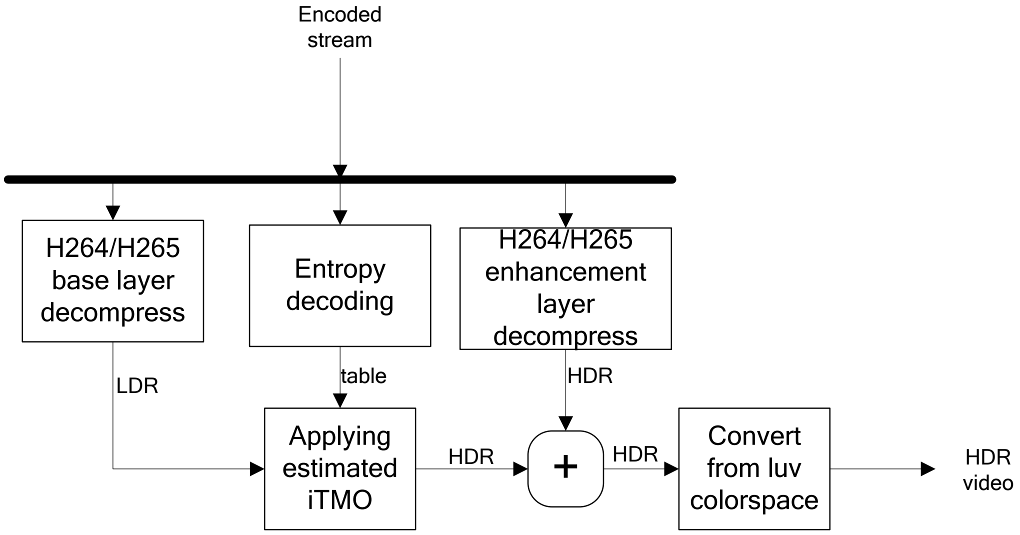

Encoded stream
H264/H265 base layer decompress
Entropy decoding
H264/H265 enhancement layer decompress
LDR
table
HDR
Applying estimated iTMO
HDR
+
HDR
Convert from luv colorspace
HDR video

Figure 5-2 Proposed HDR video decoder

Figure 5-2 illustrates the video decoder that decodes the two video layers, i.e. base and enhancement layers and predicts the HDR data from the LDR data using the transmitted tone mapping operator. The decoding delay can be minimized as the decoding can be done on a block-by-block basis.

The following sections present the operational details of some of the more important processing blocks that have been presented above.

5.3.1 Calculation of the inverse tone mapping tables

The tone mapping tables can be calculated using the least squares method. Let x_k be the LDR pixel intensities and y_k be the HDR pixel intensities in a given neighbourhood Ω of a given grid node. The positions X_i of the nodes of piecewise linear approximation are fixed, typically they are uniformly spaced (Figures Figure 4-3, Figure 4-4). The node values Y_i should minimize the cost function:

$$E = \sum_{i=1}^{N} \sum_{k \in K_i} w_k \left[\frac{Y_i - Y_{i-1}}{X_i - X_{i-1}} (x_k - X_{i-1}) + Y_{i-1} - y_k \right]^2 \tag{5.1}$$

where

$$K_i := \{k : x_k \in [X_{i-1}, X_i]\} \tag{5.2}$$

and w_k decreases with the increase of distance from mesh node.

The formula (5.1) is a quadratic form with regard to Y_i which can be minimized by setting derivatives to zero:

$$\frac{\partial E}{\partial Y_i} = 0 \tag{5.3}$$

This leads to the system of linear equations:

$$A_m Y_{m+1} + B_m Y_m + C_m Y_{m-1} = D_m, \; m = 1..N-1 \tag{5.4}$$

Boundary conditions:

$$Y_o = 0 \;, Y_N = \max Y \tag{5.5}$$

The system of equations can be efficiently solved by Gaussian elimination as the matrix is very sparse and only has three adjacent non-zero diagonals.

5.3.2 Compression of the tone mapping tables

The compression of the tone mapping tables require improvements over simple techniques (Dolzhenko, Chesnokov and Edirisinghe, 2010). The H.265 bitrate of 5Mbits/s corresponds to average of 170kbits/frame, so the compressed table size of 17 kbits means that an overhead of 10% is used for representing the tone mapping data alone. Thus the inherent inefficiency of the two-layer approach increases the overhead compared with the possible alternative, i.e. the direct one layer coding.

After experiments with different predictors the following compression scheme for the tone mapping tables was decided:

1. The 3D lookup table undergoes Karhunen-Loeve transform along the intensity axis to de-correlate the coefficients. The basis vectors are transmitted as the Supplementary Enhancement Information (SEI) message, typically per sequence. The SEI message format is described in more detail in Appendix 2.The table can be estimated in the encoder from the analysis of the incoming video or one of the pre-calculated tables can be chosen.

2. Each column corresponding to the same spatial grid location is predicted from a linear combination of up to four adjacent vectors chosen from already decoded spatial neighbours for current frame and 9 spatial neighbours in the already decoded reference frames (if available). The algorithm uses a combination of linear equation solving to narrow the subset to 8 candidates and then brute-force search for refinement of the choice of 4 vectors constrained by quantization granularity. The quality is estimated by substituting values to equation (5.1) and monitoring the change of the cost function.

3. The indexes of the basis vectors, linear coefficients and residuals are entropy encoded. The basis vector indexes are either copied from 3 node neighbours or transmitted as a new set. The algorithm attempts to encode using all 4 variants and chooses the one with the best cost function. The linear coefficients and residuals are quantized and arithmetically encoded before transmission.

5.3.3 Encoding of the HDR content

The high dynamic range video is encoded using the luv colourspace defined as (Mantiuk, Myszkowski and Seidel, 2006):

$$l = \begin{cases} a \cdot Y, & Y \le y_l \\ b \cdot Y^c + d, & y_l < Y \le y_h \\ e \cdot \log(Y) + f, & Y > y_h \end{cases} \qquad (5.6)$$

$$u = \frac{410 \cdot 4X}{X + 15Y + 3Z} \qquad (5.7)$$

$$v = \frac{410 \cdot 9Y}{X + 15Y + 3Z} \qquad (5.8)$$

where X , Y and Z are CIE colour coordinates.

The values for parameters $a...f$ corresponding to the VDP contrast sensitivity function are given in Table 5.1:

Table 5.1 The *luv* transform parameters

a	769.18	e	181.7
b	449.12	f	-90.160
c	0.16999	y_l	0.061843
d	-232.25	y_h	164.10

Values for l are quantized to the nearest integer and stored as a 12 bit channel. Values for u,v are also quantized to the nearest integer, but 8 bits are sufficient for their storage.

5.4 Experimental setup

The above CODEC structure for HDR video encoding and decoding was implemented and used in conducting a number of experiments. Section 5.4 presents the details of the experimental setup used. Section 5.5 presents the experimental results and a detailed analysis of the CODEC's performance.

The source video sequences (see Figure 5-3) were taken from VQEG video database(VQEG, 2013). The data was logarithmically encoded for 16 bit dynamic range.

Figure 5-3 Examples of the test video sequences.

The tone mapping operator used in the experiments is Pattanaic's (1998) multiscale observer model. The source videos were taken from LIVE VQEG database (Seshadrinathan, Soundararajan, Alan C Bovik, *et al.*, 2010), (Seshadrinathan, Soundararajan, Alan Conrad Bovik, *et al.*, 2010).

The H.264 encoding was done based on the JM18.5 software (Sühring, 2013). The H.265 encoding was done using HM11.0 model (*High Efficiency Video Coding (HEVC)*, 2013). The QP parameters were adjusted to balance the bitrates of the base and enhancement layers.

The first series of experiments compared HDR only compression (compression of luv data with 12 bit luma and 8 bit chroma settings for the compressor) using H.264 and H.265 encoders.

The second series used the proposed backwards compatible method with H.264 and H.265 encoders. 8 bit mode Main Profile was used for the base layer and 13 bit luma and 9 bit chroma modes were used for the enhancement layer.

The quality of compressed images was assessed by HDR Visual Difference Predictor 2 (Mantiuk *et al.*, 2011). As HDR VDP2 metric predicts only luminance difference, Mean Distance in CIE L*a*b* colourspace was used to estimate colour fidelity (Okuda and Adami, 2007b) as follows:

$$MDLab = \frac{1}{S}\sum\sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \qquad (5.9)$$

where S is the image area and the sum is calculated over all image pixels.

The speed of the processing was not measured or recorded as the algorithms for the estimation and application of tone mapping operators and algorirhms for colourspace conversion executed significantly faster than the actual video compression algorithm. It is also noted that the video compression libraries used were not speed optimized.

5.5 Experimental results

Figure 5-4 illustrates the HDR video quality metric VDP2 vs bit rate and colour fidelity metric Mean Distance (MD) vs the bit rate graphs for coding with H.265. It can be seen that the difference between HDR-only and proposed backwards compatible algorithm correspond to 10-20% for bitrates of 0.5 bpp with the difference becoming negligible for bitrates above 1.5 bpp. The MDLab per pixel error is larger as the CODEC uses coarser quantization for the chromacity data and exploit the effects of contrast sensitivity masking of the human visual system.

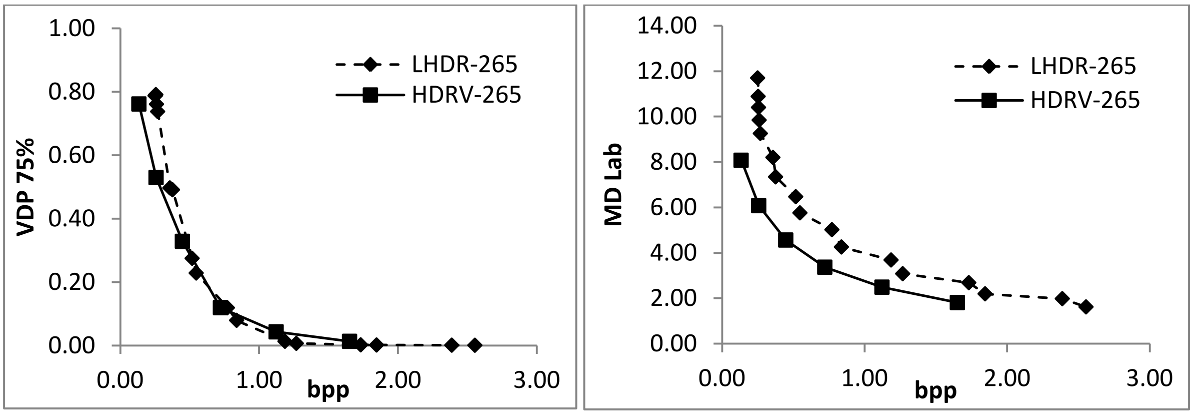


Figure 5-4 VDP and Mean Distance in L*a*b* for single layer and two layer codec using H.265 compression

The corresponding graphs for coding with H.264 are illustrated in Figure 5-5. The results for relative performance of the HDR and backwards compatible algorithm are similar, but the bitrate is up to 50% larger than with H.264 algorithm.

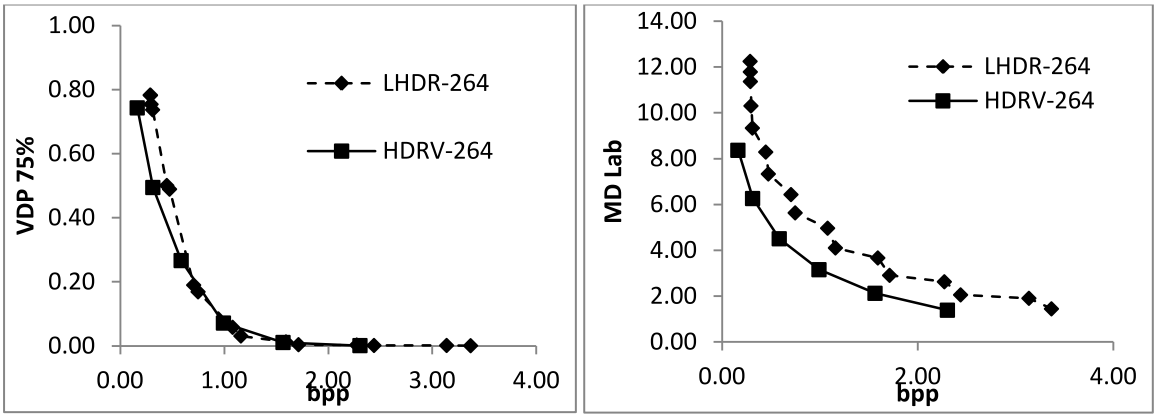


Figure 5-5 VDP and mean distance in L*a*b* for single layer and two layer codec using H.264 compression

The experiments conducted above revealed improved performance of tonemap curve compression algorithm (average savings of 25% of metadata bitrate) which gave an overall encoding gain of 3% as compared to previous implementations (Dolzhenko, Chesnokov and Edirisinghe, 2010).

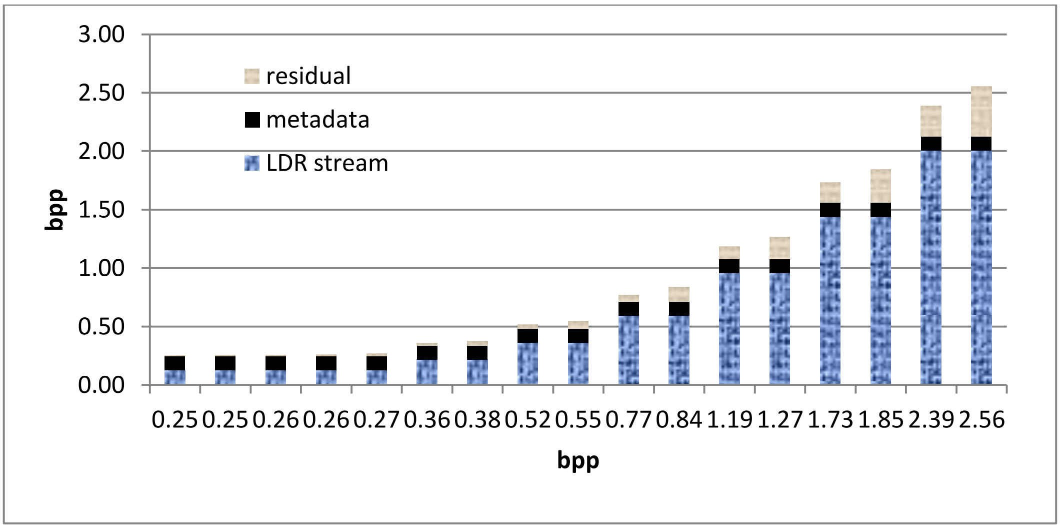


Figure 5-6 Bit distribution for two layer codec based on H.265 compression

The bitrate overhead for backward compatible coding with reference to HDR-only compression was measured to be between 10% and 35% depending on quality metrics and the sequence for the rates above 0.7 bpp (Figure 5-6). The LDR stream occupies 80% of the data, so the proposed method saves 35% of bitrate compared to a simulcast approach with the same image quality.

5.6 Conclusion

In this chapter a two layer HDR video CODEC structure with a non-uniform inverse tone mapping operator was introduced. It was shown that the representation of the inverse tone mapping function does not depend on the particular structure of the tone mapping operator and allows very compact representation in the coded stream, while providing good de-correlation of the base and enhancement layers.

The proposed CODEC not only provides backward compatibility with the existing LDR equipment, but also enables support of HDR-only content with even better coding efficiency using the same structure of HDR decoder. The detailed experiments conducted indicated that the enhancement layer contains a high level of noise which is invisible. Therefore the use of a noise masking filter (Mantiuk *et al.*, 2007) can further improve compression efficiency.

Further improvements of the compression efficiency can be achieved by exploiting the correlation of the H.264/H.265 prediction modes and motion vectors between the base and enhancement layer. The enhancement layer codec will have additional syntax elements (similar to Scalable Video Coding (Schwarz, Marpe and Wiegand, 2007)) that direct the decoder use the corresponding prediction data from the base layer.

The scope of the tone mapping operator compression scheme is not limited to video compression use case. The similar algorithm can potentially be used for still image compression, although high computational complexity and smaller overall compression gain would limit the practical applications. As the relative proportion of the tone mapping operator metadata payload is smaller in the file, improving its compression ratio brings marginal improvements to overall compression efficiency.

Chapter 6

Integration of the SysML Design Framework with a Domain-Specific Modelling System

6.1 Introduction

In the previous chapters algorithms for the compression of high dynamic range images and video were discussed. A two layer compression scheme was introduced and improved upon. Design decisions were explored for trade-offs between the algorithmic complexity and compression efficiency. The algorithms proposed required high processing bandwidth: for the two layer encoder the CODEC was required to run twice as well as making inter-layer prediction. In this chapter the aspects of practical implementation of the CODECs will be discussed.

6.2 Current trends in System-on-Chip design process

The progress of the semiconductor technology has enabled creating complex, yet compact System-on-Chip (SoC) devices, including dozens of CPU and signal processing cores. The increasing performance of the integrated circuits has allowed increasing the Utility Function of the modern consumer devices. However, while the number of operations per second or number of transistors utilised grows exponentially (Moore’s law (2006)), the power consumption is limited by available battery energy and results in heat dissipation. Furthermore, the design cycle duration for new generations of the SoC platforms has stayed the same or is shrinking at present. Therefore within this condition the increase of the system designer’s productivity, effective modelling and the system performance estimation at early stages of development are important areas of research and development.

The Table 6.1 tabulates the recent historical trends in mobile SoC platform evolution (Berkel, 2009). It is evident that the system’s numeric performance requirements have grown exponentially over time, but the availability of power available is growing in a linear manner. Furthermore, a typical consumer product’s life cycle has either remained the same or was reduced over time. The factors that result in this trend include increasing global competition and development of new standards and technologies. For example new generation image encoding algorithms has enabled achieving 30-50% reduction of the compressed bitrate at the cost of 2x-7x increase in computational complexity. (Murakami *et al.*, 2010)

Table 6.1 Trends in mobile SoC platform evolution (Berkel, 2009)

Year	1995	2000	2005	2010	2015
cellular generation	2G	2.5-3G	3.5G	pre-4G	4G
cellular standards	GSM	GPRS/ UMTS	HSPA	HSPA/LTE	LTE/LTE-A
downlink bitrate[Mb/s]	0.01	0.1	1	10	100
display pixels[×1000]	4	16	64	256	1024
battery energy[Wh]	1	2	3	4	5
CMOS[ITRS, nm]	350	180	90	50	25
PC CPU clock[MHz]	100	1000	3000	6000	8500
PC CPU Power[W]	5	20	100	200	200
PC CPU MHz/W	20	50	30	30	42
phone CPU clock[MHz]	20	100	200	500	1000
phone CPU power[W]	0.05	0.05	0.1	0.2	0.3
phone CPU MHz/W	400	2000	2000	2500	3000
workload [GOPS]	0.1	1	10	100	1000
software[MB]	0.1	1	10	100	1000
#programmable cores	1	2	4	8	16

There is also a recent trend towards incorporating more complex algorithms into embedded hardware. For example the first generation IP CCTV cameras supported compression and transmission only. However the latter generations added motion detection and the recently emerged products have supported object and scene recognition (e.g. people, cars, license plates). Further directions of extended functionality include more complex types of scene comprehension such as identifying loitering, stationary objects, and crowd behaviour patterns. The resolution of cameras has also grown with time for example the first generation IP cameras supported VGA resolutions and the current generation of devices typically has resolutions of 2-5 megapixels and above.

The current market of SoC platforms has experienced a constant or shrinking design cycle for each generation. For example Apple has delivered a new generation of an iPhone annually, since 2007. The platforms for the first two generations were designed within 3 years, the platform for the subsequent generation was designed within 2 years, and the platform for the last generation was designed over the span of 11 months. (Apple, 2007)(Apple, 2009)(Apple, 2012)

Within the context of the research presented in this thesis the experience of interaction with SoC producers for the consumer market highlighted several trends. Firstly, the time to market is considered as one of the most important drivers. Functional product with minor defects but meeting the design goals (e.g. hardware deficiencies that can be corrected or concealed by firmware) is preferred to a defect-free product that can be delivered three months later. Secondly, the expected product life time is limited. The design cycle for a new generation of a product commences every 6 or 12 months. In fact a typical design is expected to be obsolete in few years' time. Within this context the division between software and hardware becomes less obvious – the software and hardware platforms are co-designed and have similar life cycles. For example one part of a system is described in HDL and is implemented as gates and another is implemented in C and is written to the accompanying flash memory.

The growing performance of the SoC designs is often 'hidden' from the consumer by the growing complexity of the software algorithms. To this effect several factors need to be considered. The raw performance requirements grow, for example, the camera or screen resolution increases with each generation. More complex algorithms are utilized and migrated from cutting edge technology to a commodity, for example, most mid-range compact digital cameras and mobile phones offer face/smile tracking features. The pixel interpolation and noise reduction algorithms become more efficient and more complex, for example, the latest CCTV cameras show resolution beyond the Nyquist limit. There is also a software bloat factor that initiates as more complex algorithms are developed at shrinking design time, the code optimization moves down in list of priorities. The limiting factor for current SoC designs is not the area of the silicon used but power consumption – both in terms of battery life and heat dissipation. Complex power management schemes are now a characteristic of a modern SoC, for example, voltage regulation and clock gating are crucial for keeping the thermal envelope within safe limits (Yang *et al.*, 2012). Some companies use an aggressive clock control logic, which supplies absolute minimal number of clock cycles that are required to complete the processing for each operating scenario.

6.3 System design process for SoC

The factors named above render the SoC design a challenging system design task, requiring the use of automated tools and system performance modelling at early design stages. The system design process involves hardware-software co-design: the mapping of the algorithms is typically performed with high granularity between heterogeneous processing elements – CPUs with control - and a data flow oriented architecture, and hardware accelerators of different specialization – from 'hardened' algorithms (e.g. image processing pipeline) to more configurable or reusable blocks (VLIW type digital signal processors or graphics accelerator cores). The interconnect plays important role in overall system performance, for e.g. the CCTV camera SoC platforms of one of the leading chip producers revealed that the bottleneck for image processing algorithms lied in the system bus architecture – although the performance of each block and raw memory bandwidth were adequate for the task, the bus arbitration and off-chip memory access patterns prevented the system from operation at full speed. The current generation SoC may contain dozens of cores which communicate in different patterns depending on the algorithm or data processed. Thus the routing of the data streams becomes another design dimension as full crossbar architectures become impractical.(Yang *et al.*, 2012)

The modelling and design space exploration are important activities at early stages of a design cycle. If the functional requirements are known at an early stage of design, the non-functional performance requirements (such as temperature or power consumption) are much harder to satisfy and verify at this stage (Viehl *et al.*, 2008). The challenge is that modelling should be done at higher levels of abstraction. Another bottleneck is that the algorithms may not be fixed at this stage, i.e. too early in their design cycle or the standard is yet to be published. One of the most used approaches is the use of a Y-diagram and its derivatives, e.g. double roof methodology (Teich, 2012). The Y approach starts from the functional and the architectural definitions. The functions are mapped to the hardware resources and the performance parameters are estimated. One of the approaches is utilised in Daedalus framework (Daedalus, 2012) where the algorithms are mapped to Kahn process network (Kahn, 1974). The characteristic feature of the network is deterministic behaviour independent of process and communication delays. The simulation of the network for given scenarios produces execution traces which can be annotated from the platform definition. The approaches vary from the actual simulation of the algorithm on existing platforms to estimation of the complexity by cross-profiling and code analysis. It is reported that the estimation error can be as low as 2-4. (Shibata *et al.*, 2012).

As mentioned by Densmore and Passerone (2006) different tools for system level design and modelling are designed by industry and academia as there is no 'one-size-fits-all', solution. Thus different companies have adopted different design methodologies. Adopting a formal system engineering process to the design may resolve many issues and is hence the focus of this chapter.

A system engineering process, according to INCOSE (2012), consists of seven tasks: State the problem, Investigate the alternatives, Model the system, Integrate, Launch the system, Assess the performance, Re-evaluate (SIMILAR) (Figure 6-1).

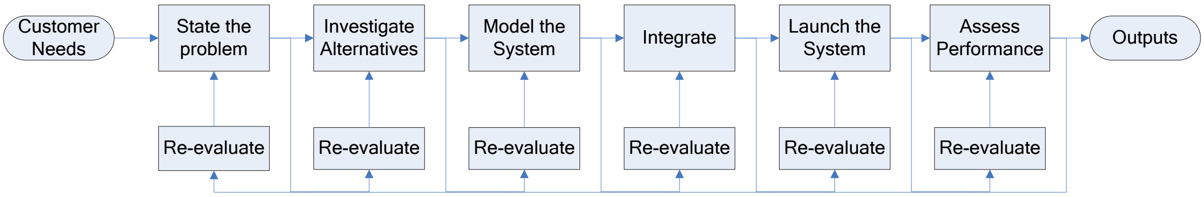


Figure 6-1 The SIMILAR process (Bahill and Gissing, 1998)

The process is interactive and the tasks overlap in time (INCOSE, 2012).

In a SoC design flow, design automation is required to tackle the complexity of the problem. However different parts of the process are addressed by different tools. The tool interoperability is a requirement for successful design flow.

6.4 Integration of SysML/MARTE and DSL framework

The SysML is 'a general-purpose graphical modelling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities (OMG, 2012). The models can be exchanged and transformed using XMI® file format. The SysML facilitates architectural views of complex systems and semantics for the definition of structure, flows of data and their physical quantities, requirements, allocation and traceability. However the profile lacks the tools for analysing timed behaviours.

The MARTE profile adds capabilities to UML for model-driven development of Real Time and Embedded Systems (RTES). This extension, often named as the UML profile for MARTE (in short MARTE), provides support for specification, design, and verification/validation stages (OMG, 2014). The profile provides a common way of modelling hardware and software aspects of the system and enables creation of rich quantitative models. The profile has means for modelling hardware-software interactions, timed behaviours, communications and the allocation of functions to hardware. The profile further adds a NFP (non-functional properties) framework for formalizing NFP requirements (power consumption, bandwidth etc).

The SysML and MARTE profiles complement each other and can be used in parallel (Espinoza *et al.*, 2009), (Mura, Panda and Prevostini, 2008).

The established tools used for system behaviour and modelling include the Daedalus framework for system-level design and exploration of the solution space (see Figure 6-2). In Daedalus the functional specification can be expressed in C or as a polyhedral process network. The platform and the mapping are described in the proprietary XML format.

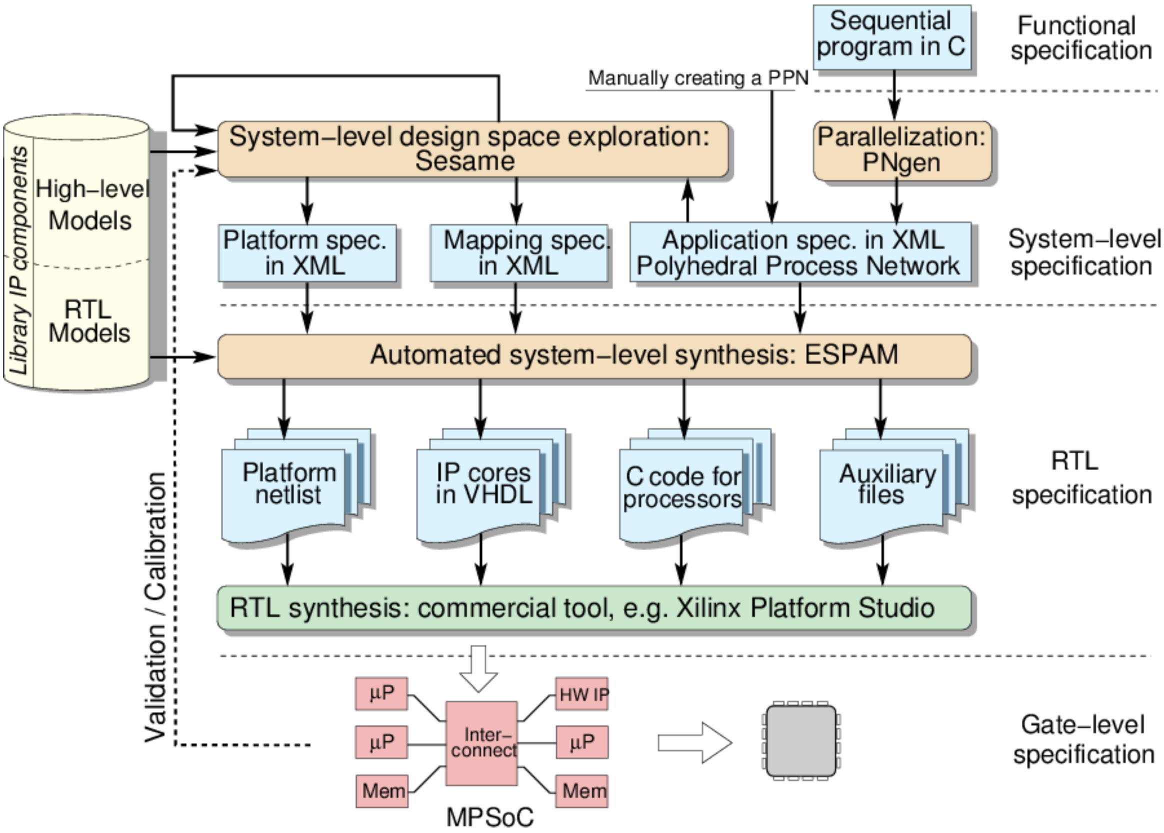


Figure 6-2 The Daedalus Framework (Daedalus, 2012)

The Daedalus framework has been created for high level System on Chip description, rapid prototyping, design space exploration and automatic code generation. The tool follows the Y-chart methodology. The tool flow allows converting C applications into PPN abstraction for performance evaluation of the parallel processes. The platform is modelled by a library of processing elements and interconnects where the application can be mapped. The Sesame component (Pimentel, Erbas and Polstra, 2006) of Daedalus allows rapid design space exploration and the ESPAM framework can convert the solution(s) to a low-level representation (RTL and C code for individual processors). (Daedalus, 2012)

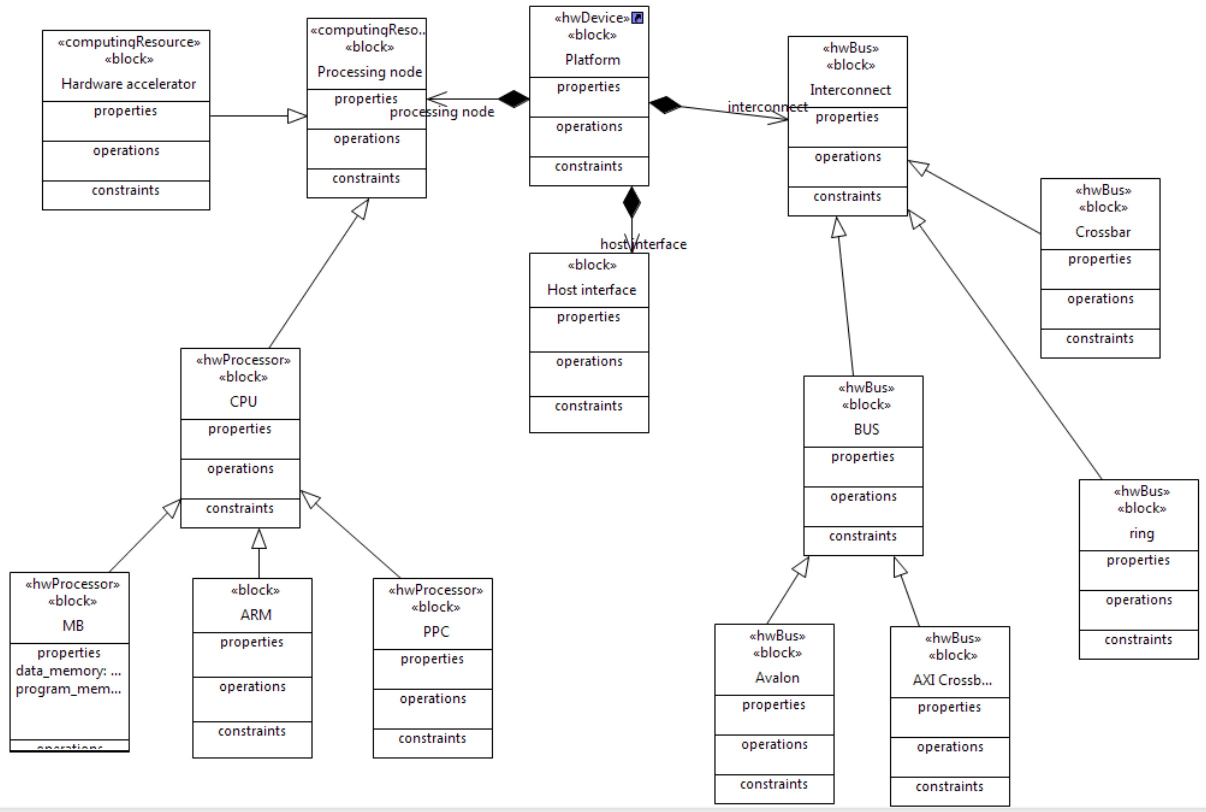


Figure 6-3 Example of Block Definition Diagram for the SoC platform

However the Daedalus framework uses proprietary data interchange formats. The interoperability with SysML would allow to automate the design process and use a unified language for the system definition. An investigation carried out within the context of this research showed that the non-standard Daedalus file format is relatively simple and self-documented to allow automatic transform to and from XMI.

The SysML Block Definition Diagrams (see Figure 6-3) and the Internal Block Diagrams (see Figure 6-4) allow defining the system to necessary detail. The extra properties in the blocks can be used to specify model parameters. The XML exchange tools allow data interchange between frameworks and back-annotation of the SysML model with simulation results.

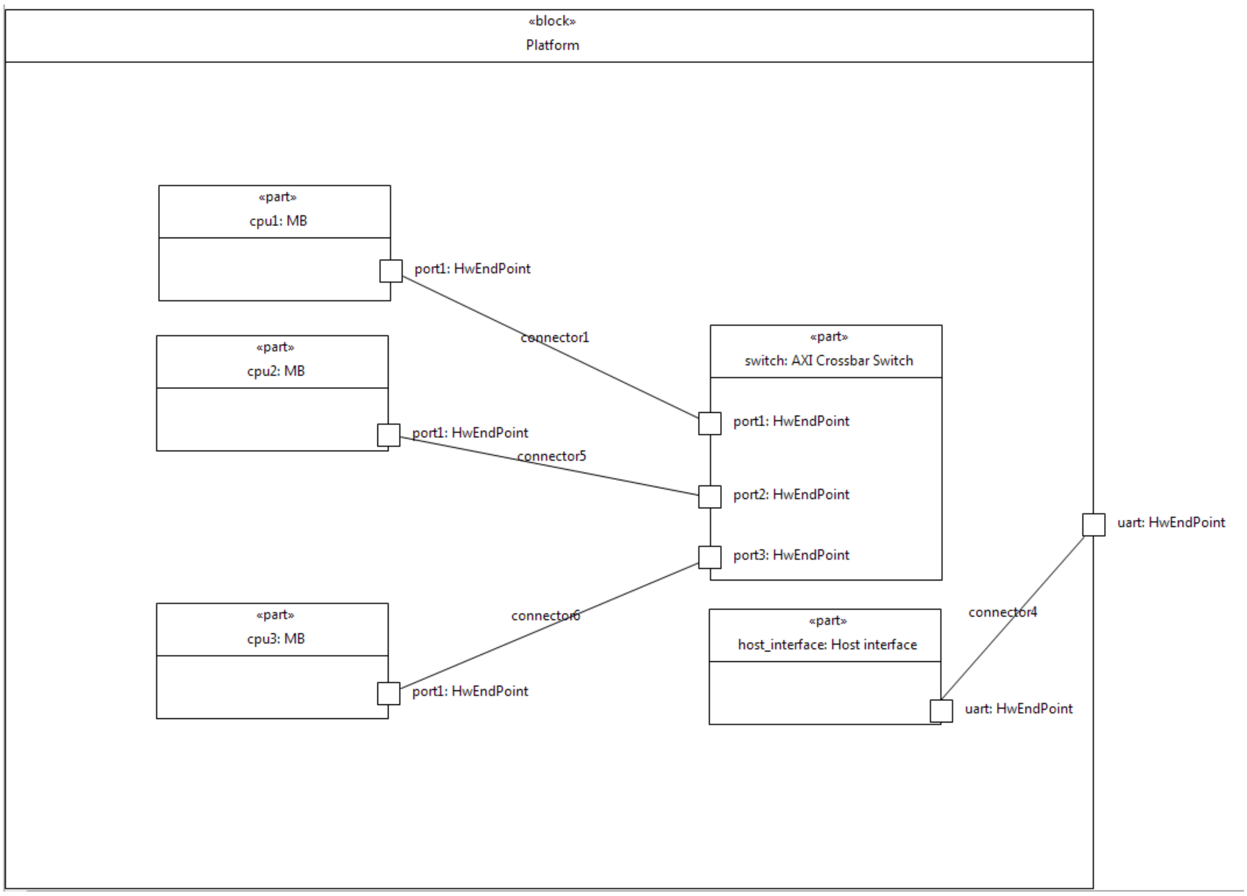


Figure 6-4 Example of Internal Block Diagram for SoC instantiation

6.5 Experimental setup

A proof of concept application was developed to map from the artefacts of the system model to the platform definition of the Daedalus framework. The SysML modelling tool Topcased (2012) was used to create the diagrams. The tool is based on the Eclipse Modelling Platform. Topcased is an open-source project aiming to create an environment for realization of the critical embedded systems including hardware and software. It is based on model-driven engineering methodology. Recent releases that support UML and SysML modelling including SysML 1.2 and use open XMI data interchange formats (TOPCASED, 2012)

The system template was created using the block definition diagram and the modules were instantiated and interconnected using the Internal Block Diagram. The appropriate MARTE profiles were applied to the modules. After the model validation the XMI file (Figure A5.1 in Appendix 5) was parsed and translated to the platform definition file suitable for Daedalus tool flow (Figures A5.2, A5.3 in Appendix 5). The tool was then used to automatically create the executable model of the process and run the simulation.

The execution results can be back annotated into XMI and can be documented and validated against performance requirements.

6.6 Conclusion

The work presented in this chapter demonstrated that it is possible to integrate the design flows and to use the MDE paradigm for automating design of complex SoC systems. SysML modelling framework allows creating the models suitable for automatic performance evaluation. The flexibility and interoperability of XML file format allow using automated design flows involving different vendors and methodologies. The forward and backward annotation allows consistent use of the same SysML model for end-to-end design flow.

The work proposed in this chapter can be improved by creating the profile for the multiprocessor systems with relevant performance properties, which can be filled by automated design flow. Further tighter integration with UML automatic code generation will enable the opportunity to build systems without needing low level programming.

As the goal for this work is to automate and facilitate the design space exploration, scripting or constraining the system parameters and making automatic search for the best architecture is the next step.

Chapter 7

Conclusion and further work

7.1 Problem statement

Current technology allows for capturing and reproducing images with a dynamic range exceeding 100dB (or contrast ratio of 10^5:1). The existing compression algorithms are the main bottleneck in supporting the end-to-end transmission of such high quality imagery. Firstly, such algorithms were developed for the representation and coding of 8 bits per pixel, per colour, images and video with in the gamma corrected domain that was more than sufficient to match the performance of the past generation of cameras and displays. Although the extension of modern image/video coding algorithms (e.g. H.264 and H.265) to cover the compression of images/video with a wider data range is possible, these approaches have several drawbacks. First of all the bitstream that results from an extended CODEC becomes incompatible with that of a standard 8 bit CODEC. This precludes adoption of such schemes in areas outside closed installations provided by a single vendor (e.g. legacy systems). Secondly, the design of such extended CODECs typically use the Sum of Absolute Differences as the quality metric to find the best predictor for the DCT or DCT-like transform of the residual, which can lead to uniform coding errors across the intensity range. While the Gamma Curve is an acceptable approximation of the perceptually uniform colourspace for SDR images with a 8 bit range, it is completely non-uniform for HDR imagery. Using the wrong colourspace for encoding leads to reduced perceived quality for the same bitrate. Thirdly the current trends in technological development as evidenced by technical and industry publications suggested that in scenarios when HDR captured images are rendered on a SDR screen, an advanced spatially non-uniform tone mapping operator is more appropriate to be used. Therefore the research presented in this thesis proposed HDR image/video CODECs that support such class of non-uniform tone mapping operators.

7.2 High Dynamic Range Image Compression

Addressing the above issues a novel two layer backward compatible HDR image CODEC scheme was proposed in Chapter 4. This scheme allowed the use of spatially non-uniform tone mapping operators for the representation of the base layer as a LDR image. The use of the non-uniform operators required the introduction of the representation of non-uniform inverse tone mapping operators. Given the assumption that the tone mapping operator models spatially varying human eye adaptation, the adaptation parameters are expected to change slowly over the image. Hence the effective kernel can be assumed to be within ~25% of the image size, under normal viewing conditions according Fairchild and Johnson (2006). In this thesis an efficient spatially non uniform, piecewise linear representation of the inverse tone mapping operator was proposed.. The coefficients of the spatial lookup table were calculated using least squares regression and then delta encoded and Huffman entropy coded for transmission. The experiments showed that it was possible to transmit the content in a backward compatible manner using 120% of the bitrate of the benchmark non-backward compatible compression method. The proposed compression scheme also covered HDR-only usage scenarios. In the absence of a LDR stream the same codec can decode the HDR stream without any overhead. The tone mapping data can either be omitted to save space or used to represent the forward tone mapping operator for display purposes.

Further research was directed towards potential optimization methods of the proposed two layer HDR image compression scheme. At that time this research was conducted, the international image compression standard JPEG-XR was, proposed as a successor to the JPEG2000 standard, claiming similar compression efficiency at lower computation requirements. Therefore two separate sets of experiments were conducted to compare the compression efficiency obtainable by both JPEG2000 and JPEG-XR in the compression of the enhancement layer data and also for HDR-only content. In a further experiment the use of different colourspaces for HDR content representation and their associated use in compression were compared. The experiments revealed that using perceptual colourspaces is important for achieving compression efficiency. It was shown that using simple floating point mapping, as standardized in JPEG-XR, leads to 10-50% loss of bitrate compared to using the perceptual luv colourspace. On the other hand the use of other perceptual colourspaces such as Bef (luv and Bef) did not show significant further improvements in compression efficiency. The comparison of the the use of JPEG2000 and JPEG-XR showed that the use of JPEG2000 allowed achieving 30-50% better compression rate in following both the two layer and HDR-only scenarios. This result was consistent with previous research reported in literature, for example by Richter (2013).

7.3 High Dynamic Range Video compression

The research presented in Chapter 4 was extended to HDR video compression in Chapter 5. Initial experiments in extending the use of novel HDR image compression algorithms to the compression of video content revealed a further challenge: the proposed inverse tone mapping operator encoding HDR images proved to be inefficient for encoding HDR video. Detailed investigations revealed that as video frames typically have significant temporal redundancy the compression of the tone mapping operator should exploit this redundancy, otherwise it presents excessive overheads as compared to representing efficiently encoded B or P frames. Therefore the proposed scheme was designed to utilize both spatial and temporal redundancy by trying to predict the transform lookup table at a given position from already decoded neighbouring intra-frames and reference frames. Linear algebra based methods were used to find the best candidates that were subsequently refined using brute force search to reduce the number of vectors and to account for quantization. The residuals were arithmetically encoded. In order to reduce redundancy in the intensity domain, the lookup table was subjected to Karhunen-Loewe transform. Combination of these methods allowed increasing the compression ratio of the inverse tone mapping operator, three fold.

7.4 System design and Implementation

Research questions of practical implementation of the proposed algorithms were addressed in Chapter 6. It was shown that one of the most prominent implementation targets for the proposed CODEC is high speed system-on-chip platforms. The SoC platforms allow efficient implementation of the computationally intensive algorithms. By using a combination of conventional CPUs, vector processors and dedicated hardware blocks it was shown that one can exploit design trade-offs between processing time, power consumption, chip area and algorithm flexibility. Current trend in industry shows that the complexity of the SoC grows exponentially with time whereas the design cycles stay the same or become shorter. One of the answers to this challenge was standardization and automation of the design process. It was mentioned that the system architecture can be expressed in visual modelling languages, for example SysML or UML MARTE. They provide unambiguous symbology to express the model of system requirements, architecture and behaviour and verify that the model is logically consistent. The work presented in Chapter 6 showed that the model data can be converted to use in the tools for automated design and simulation and the simulation results can be annotated back to the model.

7.5 Conclusion

The research conducted within the EngD programme has resulted in the design, implementation, testing and evaluation of a number of novel approaches for HDR image/video compression as presented in Chapter 4, 5 and 6. The proposed systems can be further improved and potential exists for their extended use in the future.

Compression algorithms can benefit from improved schemes for tone mapping operator coding. The effect should be more pronounced for the lower bit rates.

Video compression algorithm can be rearranged to be more SoC friendly. One of the major problems for embedded application is external memory bandwidth. The two layer encoding can be done in parallel with macroblock or CTU level of parallelism: the base layer and the enhancement layer are encoded or decoded in the lock step. The inverse tone mapping operator model is pixel based, which means that the macroblock of the base layer can be transformed into macroblock of the prediction signal for the enhancement layer, without the need for storing extra data from adjacent blocks for spatial filtering.

From perspective of improving the visual quality of the HDR video CODEC the rate-distortion optimization should also include inverse tone mapping. The errors in prediction and approximation of the inverse tone mapping typically lead to extra low frequency residual in the enhancement layer. This residual will affect small number of coefficients and it can also be effectively intra- and inter- predicted using standard video coding tools.

References

Adobe (1992) *TIFF 6.0 Specification*. Available at: //partners.adobe.com/public/developer/tiff/index.html.

Alston, I. and Campbell, S. (2010) ‘Systems Engineering Approach For Security System Design’, in *International Conference on Emerging Security Technologies*.

Apical (2014) *Apical Sinter(r) Datasheet*.

Apical Ltd (2015) *Apical home page*. Available at: http://www.apical.co.uk (Accessed: 12 August 2015).

Apple (2007) *Apple Reinvents the Phone with iPhone*. Available at: http://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html.

Apple (2009) *Apple Announces the New iPhone 3GS—The Fastest, Most Powerful iPhone Yet*. Available at: http://www.apple.com/pr/library/2009/06/08Apple-Announces-the-New-iPhone-3GS-The-Fastest-Most-Powerful-iPhone-Yet.html.

Apple (2012) *Thinnest, Lightest iPhone Ever Features All-New Aluminum Design, Stunning 4-Inch Retina Display, A6 Chip & Ultrafast Wireless*. Available at: http://www.apple.com/pr/library/2012/09/12Apple-Introduces-iPhone-5.html.

Bahill, A. T. and Gissing, B. (1998) ‘Re-evaluating Systems Engineering Concepts Using Systems Thinking’, *IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews*. IEEE, 28(4), pp. 516–527. doi: 10.1109/5326.725338.

Barkowsky, M. and Callet, P. Le (2010) ‘On the perceptual similarity of realistic looking tone mapped high dynamic range images’, *Image Processing (ICIP)*, pp. 3245–3248. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5651143 (Accessed: 14 September 2014).

Berkel, C. Van (2009) ‘Multi-core for mobile phones’, *Proceedings of the Conference on Design, Automation ...*, pp. 1–6. Available at: http://dl.acm.org/citation.cfm?id=1874924 (Accessed: 17 September 2012).

Bezryadin, S. (2007) ‘New Generation of Image Editing Algorithms’, *SPIE-IS&T*, 6502.

Bezryadin, S., Burov, P. and Tryndin, I. (2008) ‘Chromatic coordinates in HDR image coding’, *Proc SPIE-IS&T*, 6817 68170.

Brock, D. C. and Moore, G. E. (2006) *Understanding Moore’s law: four decades of innovation*. Chemical Heritage Foundation. Available at: http://www.chemheritage.org/Downloads/Publications/Books/Understanding-Moores-Law/Understanding-Moores-Law_Chapter-07.pdf.

Chen, Z., Lin, W., Ngan, K. and Ngi Ngan, K. (2010) ‘Perceptual video coding: Challenges and approaches’, in *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*, pp. 784–789. Available at:

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5582549 (Accessed: 14 September 2014).

Chesnokov, V. N. (2011) 'Dynamic Range Management for Displays for Reduced Power and Improved Ambient Contrast', in *SID Int. Symp. Digest Tech. Papers*, pp. 436–439.

Chou, W.-F., Yeh, S.-F., Chiu, C.-F. and Hsieh, C.-C. (2014) 'A Linear-Logarithmic CMOS Image Sensor With Pixel-FPN Reduction and Tunable Response Curve', *IEEE Sensors Journal*, 14(5), pp. 1625–1632. doi: 10.1109/JSEN.2013.2294740.

Daedalus (2012) *Daedalus: System-Level Design For Multi-Processor System-on-Chip, Daedalus*. Available at: http://daedalus.liacs.nl/details.html (Accessed: 10 August 2012).

Denman, S., Bialkowski, A., Fookes, C. and Sridharan, S. (2011) 'Determining operational measures from multi-camera surveillance systems using soft biometrics', in *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2011*, pp. 462–467.

Densmore, D. and Passerone, R. (2006) 'A platform-based taxonomy for ESL design', *Design & Test of Computers, IEEE*, (October), pp. 359–374. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1704727 (Accessed: 17 September 2012).

Dolzhenko, V., Chesnokov, V. N. and Edirisinghe, E. A. (2010) 'Backwards compatible High Dynamic Range Image Compression', in *IASTED CGIM2010*.

dpreview staff (2003) *Fujifilm announce SuperCCD SR*. Available at: http://www.dpreview.com/articles/6851251325/fujisuperccdsr (Accessed: 10 July 2014).

Ericsson (2012) *MPEG issues video compression draft, Ericsson news centre*. Available at: http://www.ericsson.com/news/120814_mpeg_244159018_c (Accessed: 15 November 2012).

Espinoza, H., Cancila, D., Selic, B. and Gérard, S. (2009) 'Challenges in combining SysML and MARTE for model-based design of embedded systems', *Model Driven Architecture-Foundations and Applications.*, pp. 98–113. Available at: http://link.springer.com/chapter/10.1007/978-3-642-02674-4_8 (Accessed: 3 November 2014).

Ford, A. and Roberts, A. (1998) 'Colour space conversions', *Westminster University, London*, 1998, pp. 1–31. Available at: http://herakles.fav.zcu.cz/research/night_road/westminster.pdf.

H. 264 Advanced Video Coding for Generic Audiovisual Services (2013).

Hertel, D., Betts, A., Hicks, R. and Brinke, M. (2008) 'An adaptive multiple-reset CMOS wide dynamic range imager for automotive vision applications', in *IEEE Intelligent Vehicles Symposium*, pp. 614–619.

High Efficiency Video Coding (HEVC) (2013) *Fraunhofer Heinrich Hertz Institute*. Available at: http://hevc.hhi.fraunhofer.de/.

INCOSE (2012) *A Consensus of the INCOSE Fellows*. Available at: http://www.incose.org/practice/fellowsconsensus.aspx.

Johnson, G. M. and Fairchild, M. D. (2006) ‘Rendering HDR images’, *IS&T/SID 11th Color Imaging Conference*, pp. 36–41.

Kahn, G. (1974) ‘The semantics of a simple language for parallel programming’, in *Information Processing’74: Proceedings of the IFIP Congress*, pp. 471–475. Available at: http://www.citeulike.org/group/872/article/349829.

Kaida, H. and Okuda, M. (2008) ‘Image Compression Suitable for High Dynamic Range Image Rendering’, in *IEEE 22nd International Conference on Advanced Information Networking and Applications*, pp. 1029–1033.

Koh, C. C. and Mitra, S. K. (2003) ‘Compression of Bayer color filter array data’, *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, 2. doi: 10.1109/ICIP.2003.1246665.

Koz, A. and Dufaux, F. (2012) ‘Optimized Tone Mapping With Perceptually Uniform Luminance Values for Backward-Compatible High Dynamic Range Video Compression’, in *IEEE VCIP*, pp. 1–6. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:OPTIMIZED+TONE+MAPPING+WITH+PERCEPTUALLY+UNIFORM+LUMINANCE+VALUES+FOR+BACKWARD-COMPATIBLE+HIGH+DYNAMIC+RANGE+VIDEO+COMPRESSION#0 (Accessed: 14 September 2014).

Lee, S. H. and Cho, N. I. (2010) ‘H.264/AVC Based Color Filter Array Compression with Inter-Channel Prediction Model’, in *IEEE 17th International Conference on Image Processing*.

Leszczuk, M. I., Stange, I. and Ford, C. (2011) ‘Determining image quality requirements for recognition tasks in generalized public safety video applications: Definitions, testing, standardization, and current trends’, in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2011 - Conference Programme*.

List, P., Joch, A., Lainema, J., Bjøntegaard, G. and Karczewicz, M. (2003) ‘Adaptive deblocking filter’, *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp. 614–619.

Liu, S., Kim, W.-S. and Vetro, A. (2008) ‘Bit-depth Scalable Coding for High Dynamic Range Video’, *Proc SPIE*, 6822(68220O).

Lukac, R. and Plataniotis, K. N. (2006) ‘Single-sensor camera image compression’, *IEEE Transactions on Consumer Electronics*, 52(2), pp. 299–307.

Mangiat, S. and Gibson, J. (2011) ‘Inexpensive high dynamic range video for large scale security and surveillance’, in *Proceedings - IEEE Military Communications Conference*

MILCOM, pp. 1772–1777.

Mantiuk, R., Daly, S. J., Myszkowski, K. and Seidel, H.-P. (2005) ‘Predicting visible differences in high dynamic range images: model and its calibration’, in *Proc. SPIE*, pp. 204–214. doi: 10.1117/12.586757.

Mantiuk, R., Kim, K. J., Rempel, A. G. and Heidrich, W. (2011) ‘HDR-VDP-2 : A calibrated visual metric for visibility and quality predictions in all luminance conditions’, *Image Rochester NY*. ACM Press, 1(212), p. 1. doi: 10.1145/1964921.1964935.

Mantiuk, R., Krawczyk, G., Myszkowski, K. and Seidel, H.-P. (2007) ‘High Dynamic Range Image and Video Compression - Fidelity Matching Human Visual Performance’, in *2007 IEEE International Conference on Image Processing*, pp. I9–12. doi: 10.1109/ICIP.2007.4378878.

Mantiuk, R., Myszkowski, K. and Seidel, H.-P. (2006) ‘Lossy compression of high dynamic range images and video’, in *Electronic Imaging 2006*, p. 60570V–60570V–10.

Mitsunaga, T. and Nayar, S. K. (1999) ‘Radiometric self calibration’, *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1. doi: 10.1109/CVPR.1999.786966.

Mura, M., Panda, A. and Prevostini, M. (2008) ‘Executable models and verification from MARTE and SysML: a comparative study of code generation capabilities’, *Proceedings of MARTE Workshop (DATE08)*, (March). Available at: https://bearspace.baylor.edu/Paul_Grabow/public/CPS_refs/MARTE/Marte2.pdf (Accessed: 2 November 2014).

Murakami, T., Sugimoto, K., Sekiguchi, S., Yamagishi, S., Kato, Y. and Asai, K. (2010) *Consideration on guideline of Complexity/Performance balance for HEVC Test Model Definition*. JVCJT-B065.

Nayar, S. K. and Mitsunaga, T. (2000) ‘High dynamic range imaging: spatially varying pixel exposures’, *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. IEEE Comput. Soc, 1, pp. 472–479. doi: 10.1109/CVPR.2000.855857.

Ohm, J. R., Sullivan, G. J., Schwarz, H., Tan, T. K. and Wiegand, T. (2012) ‘Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC)’, *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), pp. 1669–1684.

Okuda, M. and Adami, N. (2007a) ‘Effective color space representation for wavelet based compression of HDR images’, in *Proceedings - 14th International conference on Image Analysis and Processing, ICIAP 2007*, pp. 388–392.

Okuda, M. and Adami, N. (2007b) ‘Two-layer coding algorithm for high dynamic range images based on luminance compensation’, *Journal of Visual Communication and Image*

Representation, 18(5), pp. 377–386.

OMG (2012) *OMG Systems Modeling Language*. Available at: www.omgsysml.org.

OMG (2014) *The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems*. Available at: http://www.omgmarte.org/.

Pattanaik, S. N., Ferwerda, J. A., Fairchild, M. D. and Greenberg, D. P. (1998) 'A multiscale model of adaptation and spatial vision for realistic image display', in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*. ACM Press, pp. 287–298. doi: 10.1145/280814.280922.

Pereira, F. (2009) 'Distributed video coding: Basics, main solutions and trends', in *Proceedings - 2009 IEEE International Conference on Multimedia and Expo, ICME 2009*, pp. 1592–1595.

Pimentel, A. D., Erbas, C. and Polstra, S. (2006) 'A systematic approach to exploring embedded system architectures at multiple abstraction levels', *IEEE Transactions on Computers*. Published by the IEEE Computer Society, pp. 99–112. doi: 10.1109/TC.2006.16.

Pioppo, G., Ansari, R., Khokhar, A. A. and Masera, G. (2006) 'Low-complexity video compression combining adaptive multifoveation and reuse of high-resolution information', in *Proceedings - International Conference on Image Processing, ICIP*, pp. 3153–3156.

Poynton, C. (1995) 'A Guided Tour of Color Space', in *Proceedings of the SMPTE Advanced Television and Electronic Imaging Conference*, pp. 167–180. Available at: http://www.poynton.com/PDFs/Guided_tour.pdf.

Punchihewa, A., Hamamoto, T. and Kojima, T. (2011) 'From a review of HDR sensing and tone compression to a novel imaging approach', in *Fifth International Conference on Sensing Technology*, pp. 40–46. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6137010 (Accessed: 14 September 2014).

Rahman, Z., Jobson, D. J. and Woodell, G. A. (1996) 'Multi-scale retinex for color image enhancement', in *Proceedings of 3rd IEEE International Conference on Image Processing*. doi: 10.1109/ICIP.1996.560995.

Reinhard, E., Ward, G., Pattanaik, S. and Debevec, P. E. (2005) *High dynamic range imaging*. San Francisco, CA: Morgan Kaufmann Publisher.

Richter, T. (2013) 'Backwards Compatible Coding of High Dynamic Range Images with JPEG', *2013 Data Compression Conference*. Ieee, pp. 153–160. doi: 10.1109/DCC.2013.24.

Schwarz, H., Marpe, D. and Wiegand, T. (2007) 'Overview of the Scalable Video Coding Extension of the H.264/AVC Standard', *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), pp. 1103–1120. doi: 10.1109/TCSVT.2007.905532.

Segall, A. (2006) 'Scalable coding of high dynamic range video', in *Proceedings -*

International Conference on Image Processing, ICIP, pp. 7–10.

Segall, A., Zhao, J. and Rubinstein, R. (2009) ‘Video coding for the mobile capture of higher dynamic range image sequences’, in *2009 Picture Coding Symposium, PCS 2009*.

Seshadrinathan, K., Soundararajan, R., Bovik, A. C. and Cormack, L. K. (2010) ‘A subjective study to evaluate video quality assessment algorithms’, *Database*, 7527, p. 75270H–75270H–10. doi: 10.1117/12.845382.

Seshadrinathan, K., Soundararajan, R., Bovik, A. C. and Cormack, L. K. (2010) ‘Study of subjective and objective quality assessment of video’, *IEEE Transactions on Image Processing*, 19(6), pp. 1427–1441.

Shams, R. and Kennedy, R. a (2007) ‘Efficient Histogram Algorithms for NVIDIA CUDA Compatible Devices’, *Memory*, pp. 1–5. Available at: http://users.rsise.anu.edu.au/~ramtin/papers/2007/ICSPCS_2007.pdf.

Shibata, S., Ando, Y., Honda, S., Tomiyama, H. and Takada, H. (2012) ‘A Fast Performance Estimation Framework for System-Level Design Space Exploration’, *IPSJ Transactions on System LSI Design Methodology*, 5, pp. 44–54. doi: 10.2197/ipsjtsldm.5.44.

Smith, T. and Guild, J. (1931) ‘The C.I.E. colorimetric standards and their use’, *Transactions of the Optical Society*, pp. 73–134.

Sony (2013) *Diagonal 8.92 mm (Type 1/1.8) 6.44M-Effective Pixel Color CMOS Image Sensor IMX178LQJ*. Available at: http://www.sony.net/Products/SC-HP/new_pro/september_2013/imx178lqj_e.html (Accessed: 11 July 2014).

Springer, D. and Kaup, A. (2009) ‘Lossy Compression of Floating Point High Dynamic Range Images Using JPEG2000’, *Proc SPIE Visual Communications and Image Processing*, 7257.

Srinivasan, S., Tu, C., Regunathan, S. L. and Sullivan, G. J. (2007) ‘HD Photo: A new image coding technology for digital photography’, *Proc. of SPIE*, 6696(66960A).

Srinivasan, S., Zhou, Z., Sullivan, G. J., Rossi, J., Regunathan, S., Tu, C. and Roy, A. (2008) ‘Coding of High Dynamic Range Images in JPEG XR / HD Photo’, *Proc. of SPIE*, 7073(707315-1).

Sühring, K. (2013) *H.264/AVC Software Coordination, Fraunhofer Heinrich Hertz Institute*. Available at: http://iphome.hhi.de/suehring/tml/.

Sullivan, G. J., Ohm, J. R., Han, W. J. and Wiegand, T. (2012) ‘Overview of the high efficiency video coding (HEVC) standard’, *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), pp. 1649–1668. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6316136 (Accessed: 14 September 2014).

Sullivan, G. and Wiegand, T. (2005) ‘Video compression-from concepts to the H. 264/AVC standard’, *Proceedings of the IEEE*, 93(1). Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1369695 (Accessed: 14 September 2014).

Teich, J. (2012) ‘Hardware/Software Codesign: The Past, the Present, and Predicting the Future’, *Proceedings of the IEEE*, 100, pp. 1411–1430. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6172642 (Accessed: 17 September 2012).

TOPCASED (2012) *TOPCASED The Open Source Toolkit for Crtical Systems*. Available at: www.topcased.org.

Velastin, S. A. and Remagnino, R. (eds) (2006) *Intelligent Distributed video Surveillance Systems*. London: IEE.

Viehl, A., Sander, B., Bringmann, O. and Rosenstiel, W. (2008) ‘Integrated requirement evaluation of non-functional system-on-chip properties’, *Forum on Specification and Design Languages*, pp. 105–110. doi: 10.1109/FDL.2008.4641430.

VQEG (2013) *VQEG HDTV video sequences*. Available at: ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/1080p50_CgrLevels_SINC_FILTER_SVTdec05_/ (Accessed: 17 February 2013).

Wallace, G. K. (1992) ‘The JPEG still picture compression standard’, *IEEE Transactions on Consumer Electronics*, 38(1). doi: 10.1109/30.125072.

Walters, P. E. (1995) ‘CCTV Systems Thinking – Systems Practice’, in *European Convention on Security and Detection, pub. 408*.

Ward, G. (1998) ‘Overcoming gamut and dynamic range limitations in digital images’, in *6th Color Imaging Conference - Color Science, Systems and Applications*, pp. 214 – 219.

Ward, G. (2001) ‘High dynamic range imaging’, *Color and Imaging Conference*. Available at: http://www.ingentaconnect.com/content/ist/cic/2001/00002001/00000001/art00003 (Accessed: 21 September 2014).

Ward, G. (2005) ‘High Dynamic Range Image Encodings’, *Anyhere Software*, 38, pp. 1046–1048. doi: Doi 10.1889/1.2785485.

Ward, G. (2006) ‘A General Approach to Backwards-Compatible Delivery of High Dynamic Range Images and Video’, in *Proceedings of the Fourteenth Color Imaging Conference*.

Ward, G. and Simmons, M. (2004) ‘Subband encoding of high dynamic range imagery’, in *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pp. 83–91. doi: 10.1145/1012551.1012566.

Wiegand, T., Sullivan, G. J., Bjøntegaard, G. and Luthra, A. (2003) ‘Overview of the H.264/AVC video coding standard’, *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp. 560–576.

Winken, M., Marpe, D., Schwarz, H. and Wiegand, T. (2007) ‘Bit-Depth Scalable Video Coding’, in *2007 IEEE International Conference on Image Processing*. IEEE, pp. I – 5–I – 8. doi: 10.1109/ICIP.2007.4378877.

Xu, R., Pattanaik, S. N. and Hughes, C. E. (2005) ‘High-dynamic-range still-image encoding in JPEG 2000’, *IEEE Computer Graphics and Applications*, 25(6), pp. 57–64. doi: 10.1109/MCG.2005.133.

Yang, S.-H., Lee, S., Lee, J. Y., Cho, J., Lee, H.-J., Cho, D., Heo, J., Cho, S., Shin, Y., Yun, S., Kim, E., Cho, U., Pyo, E., Park, M. H., Son, J. C., Kim, C., Youn, J., Chung, Y., Park, S. and Hwang, S. H. (2012) ‘A 32nm High-k Metal Gate Application Processor’, in *ISSCC 2012*, pp. 264–266.

Zhang, Y., Reinhard, E. and Bull, D. (2011) ‘Perception-based high dynamic range video compression with optimal bit-depth transformation’, in *Proceedings - International Conference on Image Processing, ICIP*, pp. 1321–1324. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6115679 (Accessed: 14 September 2014).

Appendix 1

Perceptual Colourspaces and Encodings

A colourspace is a method by which one can specify, create and visualize colour. A colour is usually specified using three coordinates or parameters.(Ford and Roberts, 1998). The encoding is related to how this information is stored as a sequence of bits in the computer memory,

CIE XYZ colourspace

The XYZ colourspace was introduced by Commission Internationale de L'Éclairage (CIE) in 1931 as a result of a series of colour matching experiments (Smith and Guild, 1931). The human colour vision system has 3 kinds of colour sensitive receptors and one high sensitivity wide band receptor for night vision. The X,Y,Z components are linear combinations for the responses of the human photoreceptors. The Y component is chosen to closely resemble intensity perception across the visible spectrum. The Z component is close to the response of a shortwave photoreceptor and X is a linear combination to avoid negative spectral responses The colour matching functions for these components are presented in Figure A1.1.

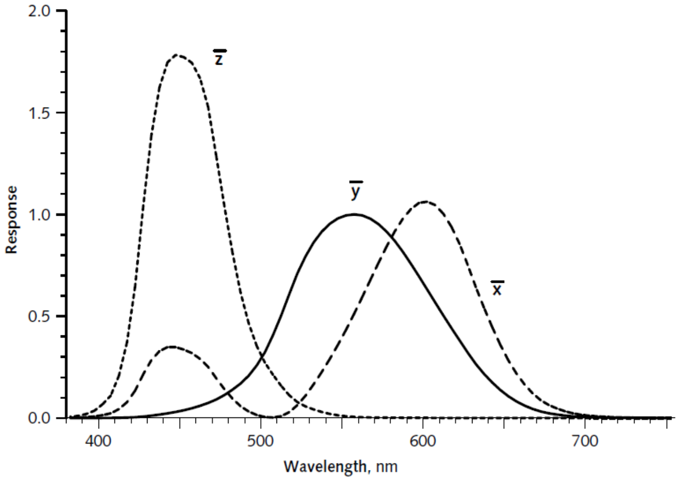


Figure A1.1 x,y,z colour matching functions (Poynton, 1995)

If the light source spectral energy distribution is $E(\lambda)$ where λ is the light wavelength then the X,Y,Z coordinates can be calculated as:

$$X = \int_{780nm}^{380nm} \overline{x}(\lambda)E(\lambda)d\lambda$$
$$Y = \int_{780nm}^{380nm} \overline{y}(\lambda)E(\lambda)d\lambda \qquad \text{(A1.1)}$$
$$Z = \int_{780nm}^{380nm} \overline{z}(\lambda)E(\lambda)d\lambda$$

CIE xyY colourspace

The xyY colourspace separates the colour into intensity information Y and colour components x,y.

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z} \qquad \text{(A1.2)}$$

where X and Z components are defined as in the CIE XYZ colourspace.

CIE L*a*b* colourspace

The L*a*b* colourspace is an attempt by CIE to introduce a perceptually uniform colourspace. It improves perceptual non-uniformity from 80:1 in CIE XYZ colourspace to about 6:1.

$$L^* = 116 f(Y/Y_n) - 16 \qquad \text{(A1.3)}$$

$$a^* = 500\left[f(X/X_n) - f(Y/Y_n)\right] \qquad \text{(A1.4)}$$

$$b^* = 200\left[f(Y/Y_n) - f(Z/Z_n)\right] \qquad \text{(A1.5)}$$

$$f(t) = \begin{cases} t^{1/3} & t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & t \le \left(\frac{6}{29}\right)^3 \end{cases} \qquad \text{(A1.6)}$$

where X_n, Y_n, Z_n are the colour coordinates of the reference point. For example D65 (daylight) illuminant is $X_n = 31.28, Y_n = 32.90, Z_n = 35.82$

CIE L*u*v* colourspace

CIE L*u*v* is an alternative perceptually uniform colourspace that uses a different non-linear mapping.

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 Y/Y_n & Y/Y_n \leq \left(\frac{6}{29}\right)^3 \\ 116(Y/Y_n)^{1/3} - 16 & Y/Y_n > \left(\frac{6}{29}\right)^3 \end{cases} \qquad \text{(A1.7)}$$

$$u^* = 13L^* \cdot (u' - u'_n) \qquad \text{(A1.8)}$$

$$v^* = 13L^* \cdot (v' - v'_n) \qquad \text{(A1.9)}$$

$$u' = \frac{4X}{X + 15Y + 3Z} \qquad \text{(A1.10)}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \qquad \text{(A1.11)}$$

For illuminant C, $u_n = 0.2009, v_n = 0.4610$

Log Luv colouspace

The Log Luv colourspace was proposed by Greg Ward Larson (1998). The L component is logarithmically encoded (base 2) and quantized with a step of 1/256 and stored as a 16 bit number; u,v components are quantized with a step of 1/410 and stored as 8 bit number.

Bef colourspace

Bef is another perceptually uniform colourspace based on the Cohen metric (Bezryadin, Burov and Tryndin, 2008) as defined below.

$$\begin{pmatrix} D \\ E \\ F \end{pmatrix} = \begin{pmatrix} 0.2053 & 0.7125 & 0.467- \\ 1.8537 & -1.2797 & -0.4429 \\ -0.3655 & 1.0120 & -0.6104 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \qquad \text{(A1.12)}$$

$$\begin{aligned} B &= \sqrt{D^2 + E^2 + F^2} \\ e &= E/B \\ f &= F/B \end{aligned} \qquad \text{(A1.13)}$$

RGB888 colourspace

This colourspace is also known as the TrueColor colourspace. It uses gamma companding and 8 bits per colour component. One of the most used encodings of this colourspace used extensively in computer graphics is the sRGB colourspace. An additional encoding used in HDTV video is the ITU-R BT.709 colourspace that uses the REC709 gamma curve.

The primaries and white point for both colourspaces are defined in the Table A1.1:

Table A1.1 Primaries and white point for sRGB colourspace

	Red	Green	Blue	White point
X	0.6400	0.3000	0.1500	0.3127
Y	0.3300	0.6000	0.0600	0.3290
Z	0.0300	0.1000	0.7900	0.3583

The gamma curve for sRGB is defined as:

$$R' = \begin{cases} 12.92\,R & when\ R < 0.00304 \\ 1.055\,R^{1/2.4} - 0.055 & when\ R \geq 0.00304 \end{cases} \qquad \text{(A1.14)}$$

The gamma curve for ITU-R BT.709 colourspace is defined as

$$R' = \begin{cases} 4.5\,R & when\ R < 0.018 \\ 1.099\,R^{0.45} - 0.099 & when\ R\ \geq 0.018 \end{cases} \qquad \text{(A1.15)}$$

As this colourspace is additive with 3 primaries, there are saturated colours that can not be correctly rendered in this colourspace as they would require negative components or components above 1.0.

RGBE colourspace

RGBE is an HDR extension to the RGB888 colourspace where the 4th component E is the base 2 exponent for the R, G and B components. The exponent is typically relative to number 128:

$$\begin{aligned} R' &= R \cdot 2^{E-128} \\ G' &= G \cdot 2^{E-128} \\ B' &= B \cdot 2^{E-128} \end{aligned} \qquad \text{(A1.14)}$$

JPEG-XR float colourspace

The JPEG XR format uses two variants of floating point encoding – 16 bits and 24 bits (Srinivasan *et al.*, 2008).

The 16 bit format is:

15	11	10	0
Exponent(E)		Mantissa(M)	

The encoded value is defined by:

$$X = \begin{cases} \left(1 + \dfrac{M}{2^{11}}\right) \cdot 2^{E-1} & E > 0 \\ \dfrac{M}{2^{11}} & E = 0 \end{cases} \qquad \text{(A1.15)}$$

The 24 bit format is:

23	22	16	15	0
S	Exponent (E)		Mantissa (M)	

The encoded value is defined by:

$$X = \begin{cases} (-1)^S \left(1 + \dfrac{M}{2^{16}}\right) \cdot 2^{E-1} & E > 0 \\ (-1)^S \dfrac{M}{2^{16}} & E = 0 \end{cases} \qquad \text{(A1.16)}$$

Appendix 2

Extra syntax elements for HDR extension of SVC

A2.1 Format of the extra syntax elements.

A2.1.1 Format of the compressed tone mapping operator.

Table A2.1 Syntax elements for the metadata encoding

itm_data(nx,ny,ni) {		
delta_qp		ae
for (i=0 ; i < nx*ny ; i++) {		
if (i > 0 || frame_type != I_Frame) {		
num_pred_minus_one[i]		au(3)
for(j=0 ; j <= num_pred_minus_one[i] ; j++) {		
idx_pred[i,j]		ae
weight[i,j]		ae
}		
}		
for (j=0;j<ni ; j++) {		
residual[i,j]		ae
}		
}		

itm_data_rbsp {		
seq_parameter_set_idc		Ue
frame_num		Fl
itm_data_present		u(1)
if (itm_data_present) {		
cabac_alignment_one_bit		U
itm_data(nx, ny, ni)		
}		
}		

A2.1.2. Format of the SEI message for KLT coefficients

Table A2.2. SEI message format

sei_message_klt {		
nx_minus16		se

ny_minus16		se
ni_minus16		se
table_variant_idc		ue
table_qp_minus26		se
if (table_variant_idc == 0) {		
for(i=0;i<ni_minus16+16;i++) {		
for(y=0 ; y < ny_minus16+16 ; y++){		
for(x=0 ; x < nx_minus16+16 ; x++) {		
klt_coeff[x,y,i]		se
}		
}		
}		
}		
if (table_variant_idc == 1) {		
for(i=0;i<ni_minus16+16;i++) {		
klt_coeff[i]		Se
}		
}		
}		

scalability_info_hdrext() {		
/* scalability_info */		
hdr_scheme_idc		Ue
hdr_bit_depth_luma_minus8		Ue
hdr_bit_depth_chroma_minus8		Ue
klt_message_flag		u(1)
if (klt_message_flag) {		
sei_message_klt()		
}		
}		

A2.2. Binarization and the context mapping of the extra syntax elements.

delta_qp – EGk0, first bin context H+0, second bin context H+1 remaining bypass, including sign num_pred – TU, uses bins H+2..H+4, remaining bits in bypass mode

idx_pred – UEGk0, H+5..H+7

weight – UEgk0, H+8..H+10 for unary prefix, bypass for suffix

Residual – EGk0, H+11..H+13 for unary prefix, bypass for suffix

A2.3. Semantics of extra syntax elements

delta_qp – difference between TM quantization step and corresponding frame QP.

num_pred_minus_one – number of the predictors used for node, 0..3

idx_pred – index of predictor node in the remap table.

weight – weight of the predictor in 1/8 increments

residual – quantized residual

seq_parameter_set – as in H.264

frame_num – fixed with log2_frame_num_minus4+4 bits

itm_data_present – flag whether tone mapping operator is present. If not present the settings from previous frame in the decoding order are reused.

cabac_alignment_one_bit – bit ‘1’

nx_minus16, ny_minus16, ni_minus16 – number of mesh nodes in horizontal, vertical and intensity direction minus 16 correspondingly.

table_variant_idc – the tone mapping table encoding scheme. Variant 1 is defined as KLT based in all three directions. Variant 0 is KLT in intensity direction only. Variant 2 does not use KLT on coefficients.

table_qp_minus26 – quantization parameter minus 26

klt_coeff – coefficient of KLT basis vectors for tone mapping operator.

hdr_scheme_idc – HDR encoding scheme. Scheme 0 is the algorithm described in Chapter 6. All others are reserved.

hdr_bit_depth_luma_minus8, hdr_bit_depth_chroma_minus8 – bit depth of the enhancement layer

klt_message_flag – if 0 the KLT data is skipped.

A2.4. Decoding of the syntax elements

A2.4.1. Mapping of idx_pred

The idx_pred is decoded using move-to-front decoder.

The remap_table[] is initialized with the entries 0..num_neighbours-1. The neighbour index numbering is defined in table A2.2 and the initialization of the remap_table[] is defined in talbe A2.3.

Table A2.2. Neighbour position numbering

Current frame

0	1	2
3	X	

L0 frame

4	5	6
7	8	9
10	11	12

L1 frame

13	14	15
16	17	18
19	20	21

Table A2.3. Initial values in remap_table[] for I/P/B frames

		Position in remap_table[]										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Frame type	**I**	3	1	0	2							
	P	8	3	1	9	11	5	7	0	2	4	6
	B	8	17	3	1	9	11	18	20	5	7	14

		Position in remap_table[] (cont'd)										
		11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Frame type	**I**											
	P	10	12									
	B	16	0	2	4	6	10	12	13	15	19	21

The translation algorithm is following

```
for (i  = 0 ; i  < nx*ny ; ++i)
   for (j = 0 ; j <= num_pred_minus_one[i] ; ++ j)
    t = remap_table[idx_pred[i,j]]
    for ( k = idx_pred[i,j]-1 ; k > 0 ; --k)
        remap_table[k] = remap_table[k-1]
    remap_table[0] = t
    neighbour[i,j] = t
```

A2.4.2. Decoding of the coefficients

Let nb[i,j,k] are the neighbours of the coeff_matrix[*,k] taken from currently decoding matrix or coeff_matrix[] of already decoded reference frames.

```
qd_table[6] = {13,14,16,18,20,23};
effective_qp = delta_qp + frame_qp
qp_i = effective_qp//6 - 4
qp_f = effective_qp - qp_i*6;
for (k = 0 ; k < ni_minus16+16 ; ++k)
   s = 0
   for (j = 0 ; j < ni_minus16+16 ; ++j)
      s = s + klt[j] * residual[I,k]
   s =((qd_table[qp_f] * s) <<qp_i)
   for (j = 0 ; j <= num_pred_minus_one[i] ; ++j)
      s = s + nb[i,j,k] * weight[i,j]
   coeff_matrix[i,k] = (s + 4) >> 3
```

A2.5. Predicting the HDR content from LDR

```
nx = nx_minus_16+16
step_x = (frame_width+nx-2)//(nx-1)
step_y = (frame_height+ny_minus_16+14)//(ny_minus_16+15)
step_i = (256+ni_minus_16+14)//(ni_minus_16+15)
ix = x//step_x
fx = x mod step_x
rx = step_x-fx
iy = y//step_y
fy = y mod step_y
ry = step_y-fy
ldr_data = ldr_frame[x,y]
ii = ldr_data // step_i
fi = ldr_data mod step_i
ri = step_i - fi
pos=iy*nx+ix
pred_hdr_l[x,y] = (coeff_matrix[pos,ii]*rx*ry*ri +
coeff_matrix[pos,ii+1]*rx*ry*ri +
coeff_matrix[pos+1,ii]*fx*ry*ri +
coeff_matrix[pos+1,ii+1]*fx*ry*ri +
coeff_matrix[pos+nx,ii]*rx*fy*ri +
coeff_matrix[pos+nx,ii+1]*rx*fy*ri +
coeff_matrix[pos+nx+1,ii]*fx*fy*ri +
coeff_matrix[pos+nx+1,ii+1]*fx*fy*ri +
(step_x*step_y*step_i>>1)) / (step_x*step_y*step_i)
```

Appendix 3

Experimental Data.

A3.1. Chapter 4 tables

Table A3.1. VDP and MDLab for different table sizes

Table nodes	VDP 75%	MDLab
20	5.09E-06	0.14
15	1.98E-04	0.15
10	6.13E-04	0.16
5	9.73E-03	0.19
2	9.09E-02	0.24
1	9.97E-02	0.24

Table A3.2. VDP and MDLab for different algorithms

algorithm	bpp	VDP 75%	MDLab
two layer	1.20	1.04E-01	1.11
	1.83	3.35E-02	0.79
	2.22	8.74E-03	0.62
	3.06	8.08E-04	0.50
	4.66	0.00E+00	0.28
	5.56	0.00E+00	0.28
	11.05	0.00E+00	0.18
Ward	1.10	5.59E-01	1.92
	1.63	2.81E-01	1.72
	1.76	2.28E-01	1.71
	2.40	1.92E-01	1.66
	2.51	1.88E-01	1.66
	3.42	1.78E-01	1.61
	7.53	1.73E-01	1.57
luv in JPEG2000	0.40	3.23E-01	1.19
	0.61	2.25E-01	0.97
	0.80	1.39E-01	0.80
	1.19	4.93E-02	0.62
	1.60	1.98E-02	0.48
	2.33	1.13E-03	0.34
	3.12	1.03E-04	0.26
	4.54	0.00E+00	0.17
	6.18	0.00E+00	0.12
	7.48	0.00E+00	0.10

Table A3.3. VDP and MDLab for two layer codec and residual in JPEG2000

bpp	VDP 75%	MDlab

1.15	3.62E-01	2.50
1.44	1.63E-01	2.26
1.64	1.36E-01	2.07
1.86	6.01E-02	2.07
2.05	5.12E-02	1.90
2.45	3.28E-02	1.65
3.15	1.42E-02	1.26
3.57	4.99E-03	1.19
4.48	7.27E-04	1.09
6.72	0.00E+00	0.65
10.85	0.00E+00	0.60
14.86	0.00E+00	0.29

Table A3.4 VDP and MDLab for two layer CODEC and residual in JPEG-XR

bpp	VDP 75%	MDLab
1.08	4.41E-01	2.81
1.24	3.02E-01	2.68
1.34	2.36E-01	2.61
1.73	1.16E-01	2.46
5.23	9.67E-03	0.99
5.92	4.28E-03	0.96
9.71	2.25E-03	0.81
16.77	0.00E+00	0.20

Table A|3.5 VDP and MDLab for two layer CODEC and different colourspaces. Residual in JPEG2000

colourspace	bpp	vdp 75%	MDLab
lef	0.30	7.65E-01	3.40
	0.35	7.35E-01	3.25
	0.40	6.87E-01	3.03
	0.48	6.29E-01	2.79
	0.60	5.37E-01	2.52
	0.69	4.57E-01	2.35
	0.80	3.98E-01	2.19
	0.97	3.17E-01	1.96
	1.19	2.43E-01	1.74
	1.60	1.06E-01	1.44
	2.32	3.69E-02	1.09
	4.56	2.54E-06	0.56
	8.48	0.00E+00	0.24
luv	0.30	7.82E-01	3.49
	0.35	7.38E-01	3.32
	0.40	7.05E-01	3.13
	0.48	6.37E-01	2.88
	0.60	5.50E-01	2.58
	0.69	4.65E-01	2.41
	0.80	4.07E-01	2.26
	0.96	3.19E-01	2.04
	1.19	2.49E-01	1.79
	1.57	1.17E-01	1.50
	2.32	3.79E-02	1.13
	4.31	0.00E+00	0.61
	7.34	0.00E+00	0.30
float	0.30	7.32E-01	5.87
	0.35	6.97E-01	5.47
	0.40	6.54E-01	4.85
	0.48	5.96E-01	4.46
	0.61	4.85E-01	3.94
	0.69	4.46E-01	3.65
	0.80	3.93E-01	3.42
	0.96	3.19E-01	3.07
	1.19	2.53E-01	2.75
	1.60	1.28E-01	2.32
	2.31	5.75E-02	1.80
	4.52	6.46E-04	1.06
	8.28	0.00E+00	0.53

Table |A3.5. VDP and MDLab for single layer CODEC with floating point encoding for JPEG2000 and JPEG-XRcompression

compression	bpp	vdp 75%	MDLab
JPEG2000	0.30	7.32E-01	5.87
	0.35	6.97E-01	5.47
	0.40	6.54E-01	4.85
	0.48	5.96E-01	4.46
	0.61	4.85E-01	3.94
	0.69	4.46E-01	3.65
	0.80	3.93E-01	3.42
	0.96	3.19E-01	3.07
	1.19	2.53E-01	2.75
	1.60	1.28E-01	2.32
	2.31	5.75E-02	1.80
	4.52	6.46E-04	1.06
	8.28	0.00E+00	0.53
JPEG-XR	0.05	9.96E-01	32.08
	0.12	9.59E-01	10.57
	0.27	8.73E-01	6.75
	0.68	6.11E-01	4.26
	1.62	2.53E-01	2.68
	3.49	2.93E-02	1.56
	6.43	4.02E-04	0.84
	10.14	0.00E+00	0.42
	14.05	0.00E+00	0.19
	18.05	0.00E+00	0.08
	21.91	0.00E+00	0.03
	24.13	0.00E+00	0.02

A3.3. Chapter 5 tables

Table A3.6 VDP and MDLab for CODECs based on H.265 compression

algorithm	bpp	VDP 75%	MDLab	iTMO bpp	LDR bpp	HDR bpp
H.265 two layer	0.25	7.88E-01	11.70	0.12	0.13	0.00
	0.25	7.88E-01	10.89	0.12	0.13	0.01
	0.26	7.91E-01	10.40	0.12	0.13	0.01
	0.26	7.61E-01	9.84	0.12	0.13	0.01
	0.27	7.38E-01	9.26	0.12	0.13	0.02
	0.36	4.97E-01	8.20	0.12	0.22	0.02
	0.38	4.91E-01	7.34	0.12	0.22	0.04
	0.52	2.75E-01	6.47	0.12	0.36	0.03
	0.55	2.28E-01	5.76	0.12	0.36	0.06
	0.77	1.19E-01	5.02	0.12	0.60	0.06
	0.84	7.93E-02	4.26	0.12	0.60	0.12
	1.19	1.31E-02	3.68	0.12	0.96	0.11
	1.27	6.71E-03	3.08	0.12	0.96	0.19
	1.73	2.03E-03	2.68	0.12	1.44	0.17
	1.85	1.58E-03	2.19	0.12	1.44	0.29
	2.39	9.98E-04	1.98	0.12	2.01	0.26
	2.56	6.80E-04	1.62	0.12	2.01	0.43
H.265 HDR	1.65	1.31E-02	1.81			
	1.12	4.31E-02	2.49			
	0.72	1.19E-01	3.36			
	0.45	3.28E-01	4.56			
	0.26	5.30E-01	6.08			
	0.14	7.61E-01	8.07			

Table A3.7. VDP and MDLab for CODECs based on H.264 compression

algorithm	bpp	VDP 75%	MDLab
H.264 two layer	0.29	7.83E-01	12.25
	0.29	7.82E-01	11.79
	0.29	7.83E-01	11.37
	0.30	7.54E-01	10.30
	0.31	7.37E-01	9.33
	0.45	5.01E-01	8.29
	0.48	4.89E-01	7.33
	0.71	1.90E-01	6.43
	0.75	1.69E-01	5.63
	1.08	5.80E-02	4.96
	1.16	3.10E-02	4.10
	1.59	1.26E-02	3.66
	1.71	3.78E-03	2.91
	2.27	2.75E-03	2.63
	2.44	1.32E-03	2.05
	3.14	1.34E-03	1.90
	3.37	5.50E-04	1.44
H.264 HDR	2.31	8.61E-04	1.38
	1.57	1.06E-02	2.12
	0.99	7.12E-02	3.15
	0.59	2.66E-01	4.50
	0.31	4.94E-01	6.26
	0.16	7.43E-01	8.36

Appendix 4

List of publications

Dolzhenko, V., Chesnokov, V. N., & Edirisinghe, E. A. (2010). Backwards compatible High Dynamic Range Image Compression. In *IASTED CGIM2010*.

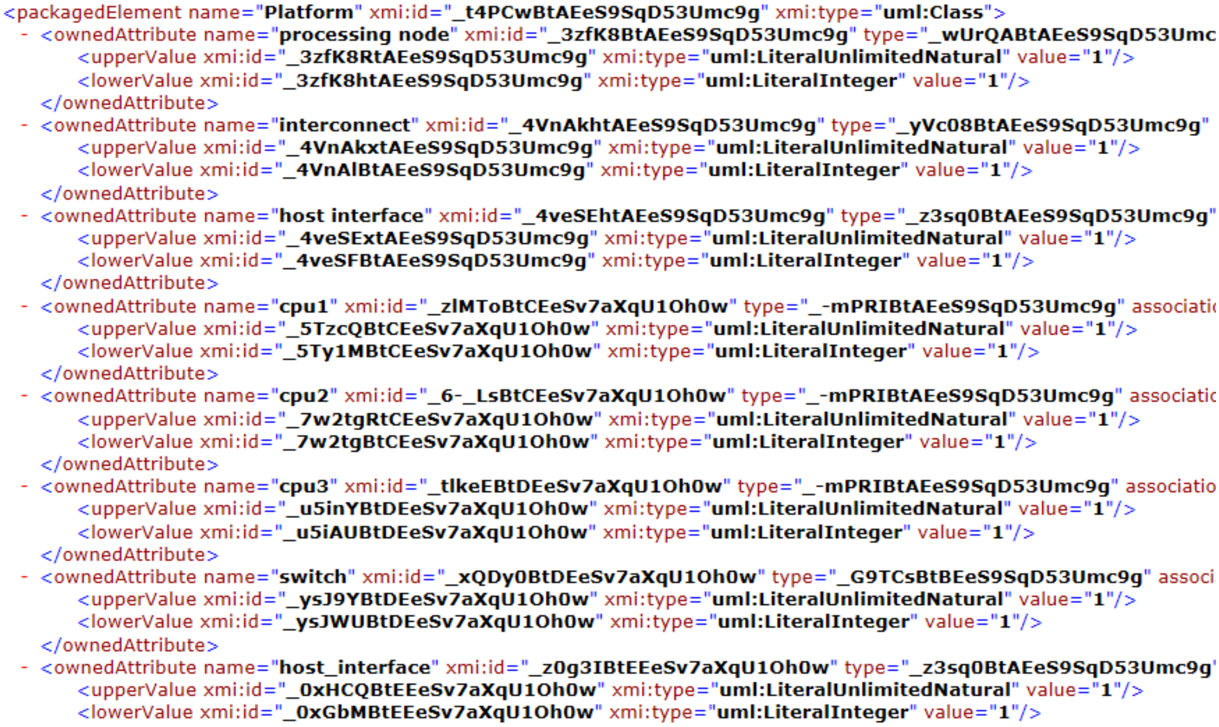
Dolzhenko, V., Chesnokov, V., & Edirisinghe, E. A. (2012). Perceptual colour spaces for high dynamic range image compression. In *Proc. SPIE* (Vol. 8436, pp. 843608–843611). doi:10.1117/12.922848

Dolzhenko, V., Chesnokov, V., & Edirisinghe, E. A. (2014). Backwards compatible high dynamic range video compression. In *Proc. SPIE* (Vol. 9029, pp. 902907–902909). doi:10.1117/12.2042513

Dolzhenko, V., Chesnokov, V. N., & Edirisinghe, E. A. (2016). Integration of a SysML framework with a domain-specific modelling system. *Design Automation for Embedded Systems*. (ready for submision)

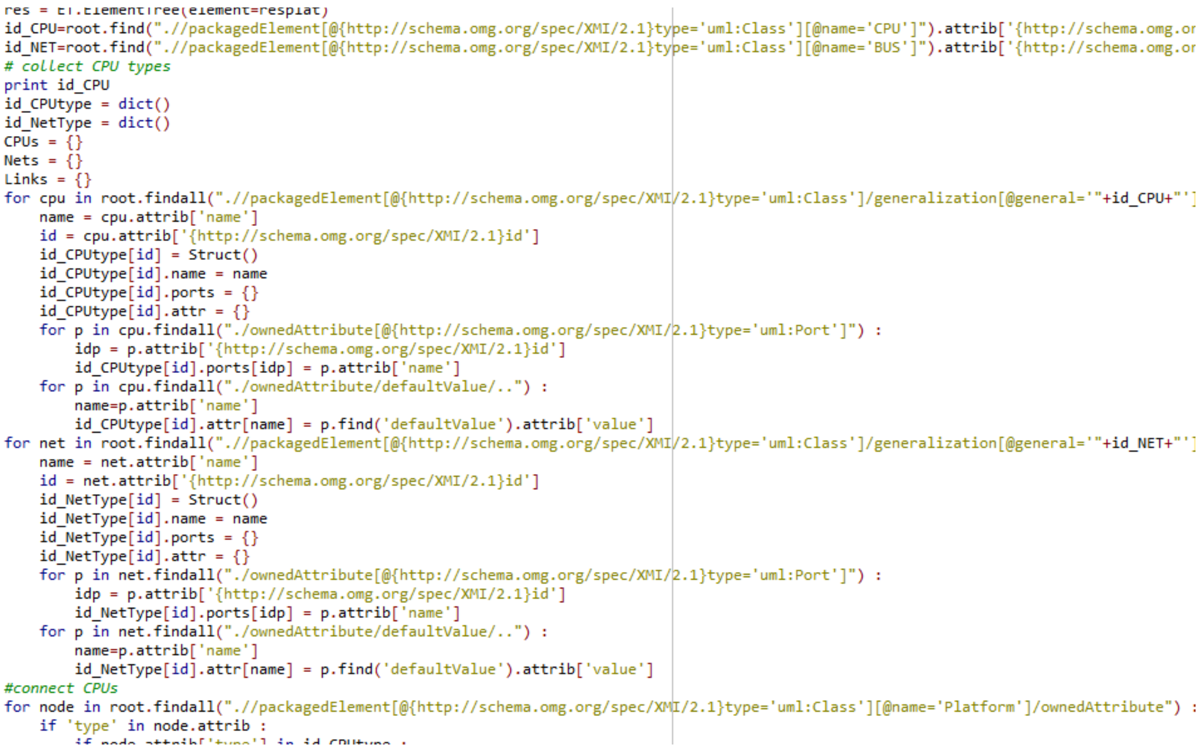
Appendix 5

Examples of the System Generation Tool Input and Output.



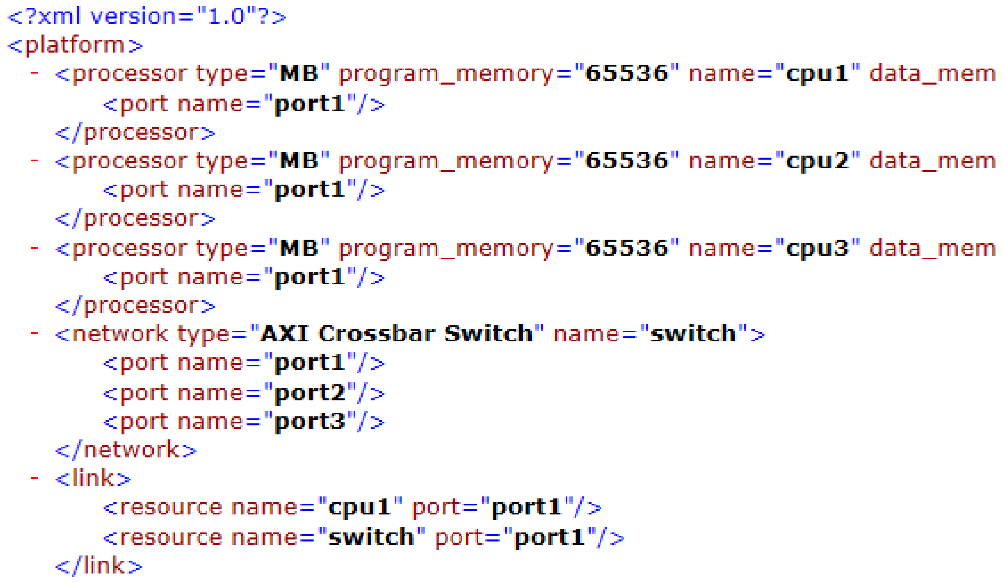
```
<packagedElement name="Platform" xmi:id="_t4PCwBtAEeS9SqD53Umc9g" xmi:type="uml:Class">
 - <ownedAttribute name="processing node" xmi:id="_3zfK8BtAEeS9SqD53Umc9g" type="_wUrQABtAEeS9SqD53Umc
      <upperValue xmi:id="_3zfK8RtAEeS9SqD53Umc9g" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_3zfK8htAEeS9SqD53Umc9g" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="interconnect" xmi:id="_4VnAkhtAEeS9SqD53Umc9g" type="_yVc08BtAEeS9SqD53Umc9g"
      <upperValue xmi:id="_4VnAkxtAEeS9SqD53Umc9g" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_4VnAlBtAEeS9SqD53Umc9g" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="host interface" xmi:id="_4veSEhtAEeS9SqD53Umc9g" type="_z3sq0BtAEeS9SqD53Umc9g"
      <upperValue xmi:id="_4veSExtAEeS9SqD53Umc9g" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_4veSFBtAEeS9SqD53Umc9g" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="cpu1" xmi:id="_zlMToBtCEeSv7aXqU1Oh0w" type="_-mPRIBtAEeS9SqD53Umc9g" associati
      <upperValue xmi:id="_5TzcQBtCEeSv7aXqU1Oh0w" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_5Ty1MBtCEeSv7aXqU1Oh0w" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="cpu2" xmi:id="_6-_LsBtCEeSv7aXqU1Oh0w" type="_-mPRIBtAEeS9SqD53Umc9g" associatic
      <upperValue xmi:id="_7w2tgRtCEeSv7aXqU1Oh0w" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_7w2tgBtCEeSv7aXqU1Oh0w" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="cpu3" xmi:id="_tlkeEBtDEeSv7aXqU1Oh0w" type="_-mPRIBtAEeS9SqD53Umc9g" associatio
      <upperValue xmi:id="_u5inYBtDEeSv7aXqU1Oh0w" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_u5iAUBtDEeSv7aXqU1Oh0w" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="switch" xmi:id="_xQDy0BtDEeSv7aXqU1Oh0w" type="_G9TCsBtBEeS9SqD53Umc9g" associ
      <upperValue xmi:id="_ysJ9YBtDEeSv7aXqU1Oh0w" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_ysJWUBtDEeSv7aXqU1Oh0w" xmi:type="uml:LiteralInteger" value="1"/>
   </ownedAttribute>
 - <ownedAttribute name="host_interface" xmi:id="_z0g3IBtEEeSv7aXqU1Oh0w" type="_z3sq0BtAEeS9SqD53Umc9g'
      <upperValue xmi:id="_0xHCQBtEEeSv7aXqU1Oh0w" xmi:type="uml:LiteralUnlimitedNatural" value="1"/>
      <lowerValue xmi:id="_0xGbMBtEEeSv7aXqU1Oh0w" xmi:type="uml:LiteralInteger" value="1"/>
```

Figure A5-0-1 Example of source XMI code for the platform definition



```python
res = ET.ElementTree(element=resplat)
id_CPU=root.find(".//packagedElement[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Class'][@name='CPU']").attrib['{http://schema.omg.or
id_NET=root.find(".//packagedElement[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Class'][@name='BUS']").attrib['{http://schema.omg.or
# collect CPU types
print id_CPU
id_CPUtype = dict()
id_NetType = dict()
CPUs = {}
Nets = {}
Links = {}
for cpu in root.findall(".//packagedElement[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Class']/generalization[@general='"+id_CPU+"']
    name = cpu.attrib['name']
    id = cpu.attrib['{http://schema.omg.org/spec/XMI/2.1}id']
    id_CPUtype[id] = Struct()
    id_CPUtype[id].name = name
    id_CPUtype[id].ports = {}
    id_CPUtype[id].attr = {}
    for p in cpu.findall("./ownedAttribute[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Port']") :
        idp = p.attrib['{http://schema.omg.org/spec/XMI/2.1}id']
        id_CPUtype[id].ports[idp] = p.attrib['name']
    for p in cpu.findall("./ownedAttribute/defaultValue/..") :
        name=p.attrib['name']
        id_CPUtype[id].attr[name] = p.find('defaultValue').attrib['value']
for net in root.findall(".//packagedElement[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Class']/generalization[@general='"+id_NET+"']
    name = net.attrib['name']
    id = net.attrib['{http://schema.omg.org/spec/XMI/2.1}id']
    id_NetType[id] = Struct()
    id_NetType[id].name = name
    id_NetType[id].ports = {}
    id_NetType[id].attr = {}
    for p in net.findall("./ownedAttribute[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Port']") :
        idp = p.attrib['{http://schema.omg.org/spec/XMI/2.1}id']
        id_NetType[id].ports[idp] = p.attrib['name']
    for p in net.findall("./ownedAttribute/defaultValue/..") :
        name=p.attrib['name']
        id_NetType[id].attr[name] = p.find('defaultValue').attrib['value']
#connect CPUs
for node in root.findall(".//packagedElement[@{http://schema.omg.org/spec/XMI/2.1}type='uml:Class'][@name='Platform']/ownedAttribute") :
    if 'type' in node.attrib :
        if node.attrib['type'] in id_CPUtype :
```

Figure A5-0-2 Example of the conversion script



```
<?xml version="1.0"?>
<platform>
 - <processor type="MB" program_memory="65536" name="cpu1" data_mem
       <port name="port1"/>
   </processor>
 - <processor type="MB" program_memory="65536" name="cpu2" data_mem
       <port name="port1"/>
   </processor>
 - <processor type="MB" program_memory="65536" name="cpu3" data_mem
       <port name="port1"/>
   </processor>
 - <network type="AXI Crossbar Switch" name="switch">
       <port name="port1"/>
       <port name="port2"/>
       <port name="port3"/>
   </network>
 - <link>
       <resource name="cpu1" port="port1"/>
       <resource name="switch" port="port1"/>
   </link>
```

Figure A5-0-3 Example of the resulting ESPAM platform definition