Vídeo Industrial para visão EXPANdida em operações à Distância

VIEXPAND
E6.1
CURRENT STANDARDISATION AND EVOLUTION I (EN)
ESTADO ATUAL DE ESTANDARDIZAÇÃO E EVOLUÇÕES I (PT)

Contractual Date of Delivery:	30/Nov/2020
Actual Date of Delivery:	30/Nov/2020
Editor:	Pedro A. Assunção, Lino Ferreira (IT)
Authors:	Pedro A. Assunção, Lino Ferreira (IT)
Internal reviewers:	João Gil, Carlos Ribeiro (TWEVO), Pedro A. Assunção (IT)
Workpackage (EN)/Atividade (PT):	6
Security:	PU
Version:	1.0
Total number of pages:	33

SUMMARY (EN):

This Deliverable reports the research and review work carried out within the scope of Activity 2 in regard to the standardisation in the field of video coding, presenting the state of the art and standard evolution over the past decades. The basic concepts and techniques used in video coding algorithms are presented along with an overview of current hybrid coding architectures, commonly used in the standards. Then, the Deliverable is mainly focused on the detailed description of the high-level syntax, data structures and coding tools of the High Efficiency Video Coding (HEVC)/H.265 standard, which implicitly define the technical aspects of possible implementation options and provide useful background information for defining suitable encoder configurations for the project. Coding methods for Regions of Interest (ROI) are also described within the scope of standard algorithms and some future perspectives of standards evolution are also highlighted. Finally, a discussion on the HEVC coding tools and their usefulness for the project is presented.

SUMÁRIO (PT):

Este Entregável relata o trabalho de investigação e revisão realizado no âmbito da Atividade 2 em relação à normalização no domínio da codificação de vídeo, apresentando o estado da arte e a evolução padrão ao longo das últimas décadas. São apresentados os conceitos e técnicas básicos utilizados nos algoritmos de codificação de vídeo, juntamente com uma visão geral das atuais arquiteturas híbridas de codificação, comummente utilizadas nas normas. Depois, o Entregável concentra-se principalmente na descrição detalhada da sintaxe de alto nível, estruturas de dados e ferramentas de codificação da norma HEVC/H.265, que define implicitamente os aspetos técnicos das possíveis opções de implementação e fornece informação de base útil para a definição de configurações de codificadores adequadas ao projeto. Os métodos de codificação para Regiões de Interesse (ROI) são também descritos no âmbito dos algoritmos padrão e algumas perspetivas futuras de evolução das normas são também destacadas. Finalmente, é apresentada uma discussão sobre as ferramentas de codificação HEVC e a sua utilidade para o projeto.

Keyword list: Standardisation, Video Coding, HEVC/H.265, ROI-based Coding

Cofinanciado por:

CENTRO 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

The VIEXPAND project is funded under Research and Technological Development Incentive
Scheme - CO-PROMOTION | Centro2020 | P2020 | European Regional Development Funds.

Table of Contents

List of Acronyms

AMVP	Advanced Motion Vector Prediction
ANI	Agência Nacional de Inovação
AO	Media Alliance for Open Media
AR/VR	Augmented/Virtual Reality
AVC	Advanced Video Coding
AVS	Audio Video Standard
BLA	Broken Link Access
CABAC	Context Adaptive Binary Arithmetic Coding
CCTV	Closed Circuit Television
CIF	Common Intermediate Format
CRA	Clean Random Access
CTC	Capture and Transmission Centre
CTU	Coding Tree Unit
CU	Coding Unit
DASH	Dynamic Adaptive Streaming over HTTP
DBF	Deblocking Filter
DCT	Discrete Cosine Transform
DPB	Decoded Picture Buffer
FGS	Fine Grain Scalability
FMO	Flexible Macroblock Ordering
FPGA	Field-Programmable Gate Array
FVC	Future Video Coding
HD	High Definition
HD TV	High-Definition Television
HEVC	High Efficiency Video Coding
HVS	Human Visual System
IDR	Instantaneous Decoder Refresh
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
ITU-T	International Telecommunication Union (Telegraphy section)
JCT-VC	Joint Collaborative Team on Video Coding
JVC	Joint Video Experts Team
JVET	Joint Video Experts Team
JVT	Joint Video Team
LCU	Largest Coding Unit
MB	Macroblock
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MTU	Maximum Transmission Unit
MV	Motion Vector
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
PDF	Probability Density Function
POC	Picture Order Count
PPS	Picture Parameter Set

PU	Prediction Unit
QCIF	Quarter Common Intermediate Format
QP	Quantisation Parameter
RADL	Random Access Decodable Leading
RAP	Random Access Point
RASL	Random Access Skipped Leading
RBSP	Raw Byte Sequence Payload
RD	Rate-Distortion
ROI	Region Of Interest
RPSet	Reference Picture Set
RTP	Real-Time Protocol
SAO	Sample Adaptive Offset
SEI	Supplemental Enhanced Information
SPS	Sequence Parameter Set
STSA	Step-Wise Temporal Sub-layer Access
SVC	Scalable Video Coding
TS	Transport Stream
TSA	Temporal Sub-Layer Access
TU	Transform Unit
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VPS	Video Parameter Set
VVC	Versatile Video Coding
WPP	Wavefront Parallel Processing

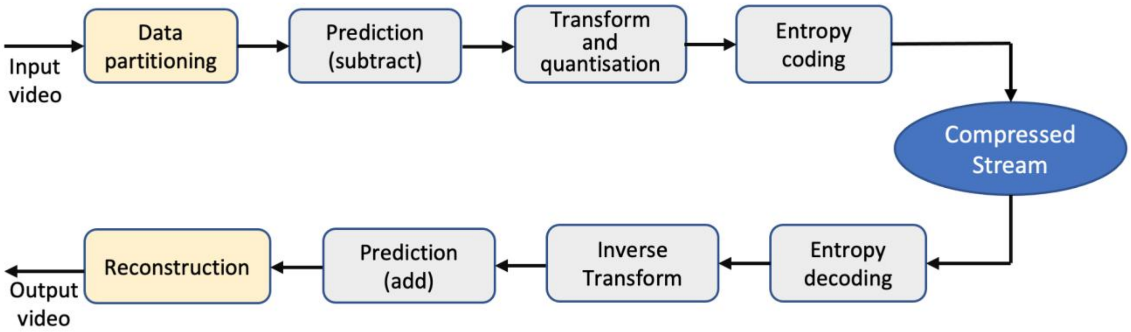
1. Introduction

This Deliverable presents an overview of standard video coding algorithms with particular emphasis on the coding tools used in High Efficiency Video Coding (HEVC) standard, as this is the most recent standard that can be found in the market, fully implemented on real-time hardware. After the standardisation context and evolution, the document describes the general structure of hybrid video encoders and main functions. The previous standard H.264 is used as reference, since most of the coding tools defined in HEVC/H.265[1] are inherited and further developed from its predecessor. Particularly important for this project is the high-level coding syntax, where the various elements are described. To a great extent, the coding mechanisms allowed by such syntax are the basis for the adaptive coding planned for the project, to deal with Regions of Interest (ROI). The underlying coding functions are presented to provide support for the ROI coding mechanisms. A literature review is presented for ROI coding, establishing the state of the art in this field. Finally standard evolution and future perspectives are discussed.

2. Video compression - basic concepts

Video compression is based on the elimination of redundant and irrelevant information of the source, input video. While de former is related to statistical data redundancies which can be removed without information loss, the latter consists in removing information from the video signal that is not perceived by the Human Visual System (HVS). This leads to either lossless compression (data reduction without loss of any information) or lossy compression (data reduction with information loss). Lossy compression is normally used in image and video coding by most video compression standards based on intra- and inter-frame coding. Spatial redundancy is due to correlation between pixels in the same image/frame. If correlation exists in the spatial domain (i.e., neighbouring pixels have similar values), redundancy can be reduced through intra-frame prediction. In the case of the temporal domain, the temporal redundancy is due to similarities between adjacent or near frames.

The main coding tools used in hybrid video coding are inter- and intra-frame prediction, transform, quantisation and entropy coding, which is illustrated in Figure 1. After partitioning,



[1] The HEVC was approved as the ITU-T Recommendation H.265.

Figure 1 – Processing pipeline of a standard video coder-decoder (codec) system.

the video data in small processing units comprising pixel blocks of different sizes, the prediction module is usually followed by the transform and quantisation of prediction residues, which is then multiplexed with signalling and header information before entropy coding. Entropy coding is used to exploit the statistical data redundancy. In hybrid video coding, each input frame is divided into blocks, in which the block size is dependent on the prediction mode used. In inter-frame prediction, each block is predicted with information used in other encoded frames, typically using motion compensation. Contrary to inter-frame prediction, in intra-frame prediction no information of other frames is used, i.e., each block is predicted from the information used in neighbouring blocks. Intra-frame coding ensures that systematic errors do not continuously propagate throughout the sequence, since an entire frame is periodically encoded on its own [1].

Motion Estimation / Compensation

Motion Estimation (ME) is used in inter-frame prediction to exploit the fact that, in most video sequences, the difference between two adjacent frames results from camera or object motion. Bi-directional ME uses two reference frames to search for predictions while simple unidirectional ME only uses one frame for prediction. By using ME, the encoder only encodes the prediction residual, discarding the redundant information between current and reference frames. The ME function finds Motion Vectors (MV) for each block, i.e., motion estimation of a block involves finding an N×N region in an arbitrary region of a reference frame that closely matches the current block. The MV and previously reconstructed frame are fed to a motion compensation function to create the inter-frame prediction Figure 2.

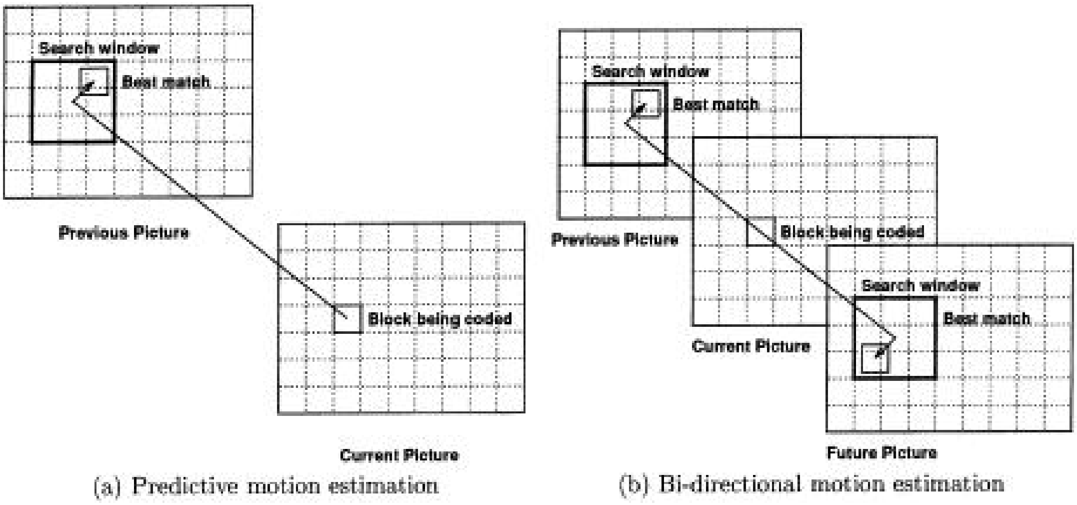


Figure 2 – Block-based motion estimation [2].

The prediction obtained from intra- and inter-frame unit is subtracted from the current block to produce a prediction residue or prediction error. Then the residue is transformed from the spatial domain to the frequency domain in order to de-correlate the signal and concentrate the energy in a few coefficients. Then, each sub-block is quantised, and the small values associated to spectral components that are not perceptually relevant are eliminated. Finally, the coefficients, motion vector and associated header information for each block are entropy encoded to produce the compressed bit stream.

3. Video coding standards

Video coding standards define the technology behind compression of video data that is generated by video cameras in raw format. These standards find application in multiple areas like digital television video calls, video games, video streaming services, surveillance and remote monitoring. The main purpose of video coding standardisation is to define the syntax and semantics of coded streams, in order to guarantee interoperability between video codecs independently developed by different manufactures [1]. In the last decades, the development of these standards has been carried out within the context of two main groups, as described in the next section.

3.1. A brief history of the H.26X and MPEG – X family standards

The Video Coding Experts Group (VCEG) of the International Telecommunication Union (Telegraphy section) (ITU-T) has been the standardisation group responsible for the ITU-T H.26x series of coding standards, including H.120, H.261, H.262, H.263, H.264 [3-9] and more recently the H.265 [9] and the H.266. The other group is the Moving Picture Experts Group (MPEG), working under the International Standards Organisation (ISO) and International Electrotechnical Commission (IEC), which is responsible for the popular MPEG-1, MPEG-2 and MPEG-4 standards and their various extensions and parts. It is worthwhile to notice that at some point, the standards developed by the MPEG group were adopted by the ITU. For this, there is a correspondence between the standards of these two groups. A timeline of the standards evolution is shown in Figure 3.

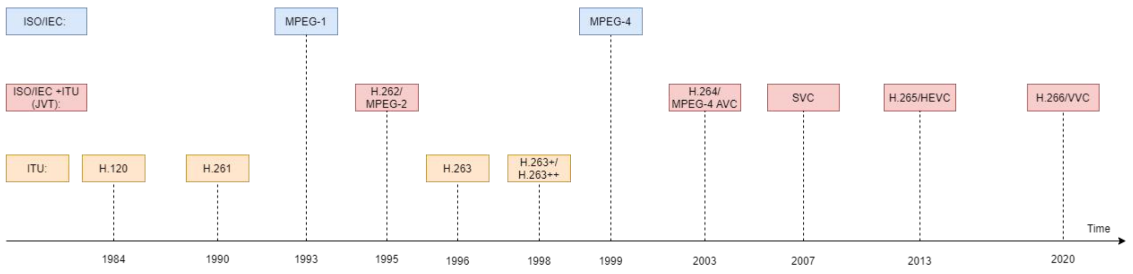


Figure 3 - Timeline of video coding standards – year of standard approval.

Note that this timeline shows the standard approval times, which are not synchronised with the availability of real-time operational technology in the market. For instance, the latest standard Versatile Video Coding (H.666/VVC) was recently approved but there are no hardware-based codecs in the market yet. The Fraunhofer HHI, which was one of the main active players in the standardisation process of H.266/VVC, announced in July 2020 that the first software implementations would be available in the autumn of 2020[2]. It is expected that the first chipsets will appear in 2021 while fully integrated implementations with all standard features shall be available in 2022. Therefore, the most advanced and mature standard technology, currently available in the market, is the H.265/HEVC.

The two groups, ITU and ISO/IEC, have been collaborating since they jointly developed the H.262/MPEG-2 standard in 1995. In 2001, they formed the Joint Video Team (JVT) and cooperated in the development of H.264/MPEG-4 Advanced Video Coding (AVC). Later, the Joint Collaborative Team on Video Coding (JCT-VC) developed the video coding standard H.265 in 2003, which is also called HEVC [9]. Table 1 presents a comparison of the main features of the MPEG/H.26X family of video coding standards, up to the H.265/HEVC.

	Video coding standard	Year	Features
MPEG family	MPEG-1 part-2	1993	Developed for video and audio storage on CD-ROMS; Supports YUV 4:2:0 with a resolution 352 × 288; Lossless motion vectors.
	MPEG-2 part-2	1995	Supports High Definition (HD) TV and video on DVDs; Introduction of profiles and levels; Nonlinear quantisation and data partitioning.
	MPEG-4 part-2 (Visual)	1999	Supports low bit-rate multimedia applications on mobile platforms; Shares subset with H.263; Supports object-based or content-based coding.
	MPEG-4 part-10 (AVC)	2003	Co-published with H.264/AVC.
H.26x family	H.120	1984	The first digital video coding standard
	H.261	1988	First block-based hybrid coding with integer pixel motion compensation; Support for CIF and QCIF resolutions. Developed for video conferencing over ISDN.
	H.262	1995	This is MPEG-2 part 2.
	H.263/H.263+	1996 1998	Improved quality to H.261 at lower bit rate; shares subset with MPEG-4 part 2.
	H.264 AVC	2003	Support video on the Internet, computers, mobile and HD TVs; Significant quality improvement with lower bit rates; Increased computational complexity; Improved motion compensation with variable block-size, multiple reference frames and weighted prediction.
	H.265/HEVC	2013	Support ultra-HD video up to 8k with frame rates up to 120 fps; Greater flexibility in prediction modes and transfer block sizes; Parallel processing; 50% bit-rate savings compared with H.264 for the same video quality.

Table 1 - The MPEG family and H.26X family video coding standards.

3.2. H.264/MPEG-4 AVC standard

The H.264 standard was initially developed between 1999 and 2003, as a fundamental technology to be adopted in a wide range of video applications, including broadcast of HD TV, camcorders, surveillance systems, Internet and cellular networked videos, real-time video chat, video conferencing, and Blu-ray Discs. The major video coding standards proposed after the 1990s are mostly based on a similar hybrid coding model that incorporates predictive coding,

[2] See the newsletter: https://newsletter.fraunhofer.de/-viewonline2/17386/465/11/14SHcBTt/V44RELLZBp/1

transform coding, and entropy coding. H.261[3], H.263 [4], MPEG-1 [5], MPEG-2 [6], MPEG-4 visual [7], and H.264/AVC [8] are all developed under this framework. Although there are differences in details, they share most of the basic functions. Worthy of notice is that H.265 [9] and H.264 achieve better compression efficiency and have greater flexibility in compressing, transmitting, and storing videos than their predecessors MPEG-1, MPEG-2, etc.

A standard H.264/AVC encoder processes the input video in data units defined as macroblocks (16×16 pixels). Inter-prediction uses a range of block sizes (from 16×16 to 4×4) to predict pixels in the current frame from similar regions in previously coded frames. Intra-prediction adopts a range of block sizes (from 16×16 to 4×4) to predict the macroblock from the previously coded pixels within the same frame. The encoder then subtracts the prediction from the current macroblock to form a residual and the block of residual samples is transformed using a 4×4 or 8×8 integer transformation yielding a set of Discrete Cosine Transformation (DCT) coefficients. The transformed coefficients and other information are then quantised and coded into the output stream using entropy coding.

At the decoder, the quantised, transformed coefficients and the prediction information are firstly extracted from the bit stream. The coefficients are then rescaled to restore each block of the residual data. These blocks are combined together to form a residual macroblock for frame reconstruction. The decoder adds the prediction to the decoded residual to finally reconstruct a decoded macroblock.

Scalable Video Coding (SVC) extensions

Scalability in video coding consists in producing streams where some form of layered representation of the video data is embedded, allowing partial decoding of the stream. Each layer corresponds to a substream which in turn allows to obtain a reduced resolution and/or quality or the input video, at lower bit rates. Such decoding flexibility enables seamless access to various versions of the same video content as well as bit stream truncation with very low computational demand, finding applications in user profile and networking adaptation among others. Previous video coding standards such as H.262|MPEG-2 Video, H.263 and MPEG-4 Visual define scalable profiles, but these have rarely been used because spatial and quality scalability features came along with a significant loss in coding efficiency, as well as a large increase in encoder complexity as compared to the corresponding single-layer profiles.

In the case of the H.264/MPEG-4 AVC video coding standard, substantial improvements have been obtained in its scalable extension, in terms of coding efficiency and scalability compared to scalable profiles of the previous standards [10]. Without significantly increasing the decoding complexity, the H.264/MPEG-4 SVC standard [10] provides the same compression functionality of the H.264/MPEG-4 AVC standard, but new coding tools for the generation of scalable bit stream were implemented. Following the conventional approach, H.264/MPEG-4-SVC is based on a layered scheme, in which the bit stream is coded into a base layer, H.264/MPEG-4 AVC compliant, and one or more enhancement layers, as it is shown in the block diagram of an SVC encoder of the Figure 4. Each version of the video signal with specific resolution is coded in a scalable bit stream and they are characterised by a layer identifier (layer 0 or base layer, layer 1, tiles, layer *n*). To exploit the dependencies between layers and to improve the coding efficiency of enhancement layers, the H.264/MPEG-4-SVC provides inter-layer motion prediction, inter-layer residual prediction, including the intra modes. These inter-layer predictions modes are represented in Figure 4. The main advances in this standard, which are responsible for improved efficiency, lie in the interlayer prediction modes that are capable of better exploiting the redundancies between different representation levels of the same video information.

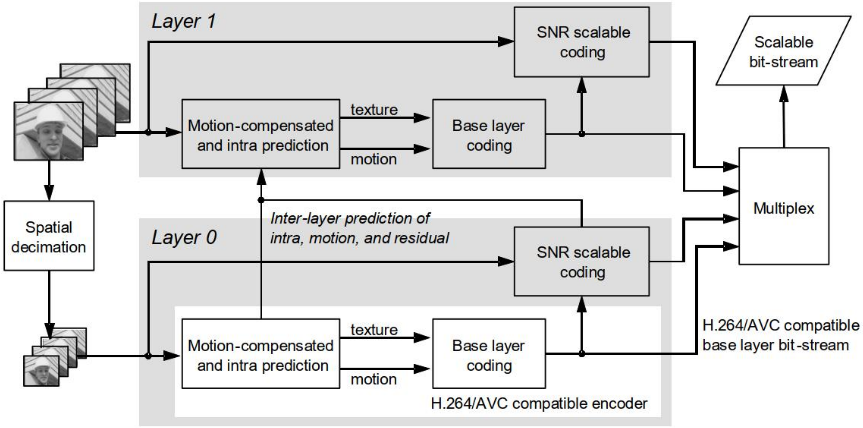

Layer 1
SNR scalable coding
Scalable bit-stream
texture
Motion-compensated and intra prediction
motion
Base layer coding
Multiplex
Spatial decimation
Inter-layer prediction of intra, motion, and residual
Layer 0
SNR scalable coding
texture
Motion-compensated and intra prediction
motion
Base layer coding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

Figure 4 – Simplified SVC encoder structure [11].

The SVC bit stream is organised in such a way that it enables a user to easily extract only a subpart of the data contained in the scalable bit stream while still being able to decode the original input video at a reduced spatial resolution, frame rate or quality. Furthermore, the H.264/MPEG-4 SVC supports temporal, spatial and quality scalability and each can be available with different granularity.

The Network Abstraction Layer (NAL)

The H.264/AVC standard introduced the concept of the Video Coding Layer (VCL) and a NAL, to decouple the coded video elements from the transport network. See Figure 5. The VCL includes all functions related to processing and coding the signal video, producing a compressed representation in the form of a standard-compliant stream. The NAL specifies an encapsulated format of the coded video data and defines header information in an appropriate manner for conveyance by the transport layers or storage media. All data is contained in NAL units (NALUs), each of which contains an integer number of bytes. The NAL concept is also used in HEVC for the same purpose, as described in the next section.

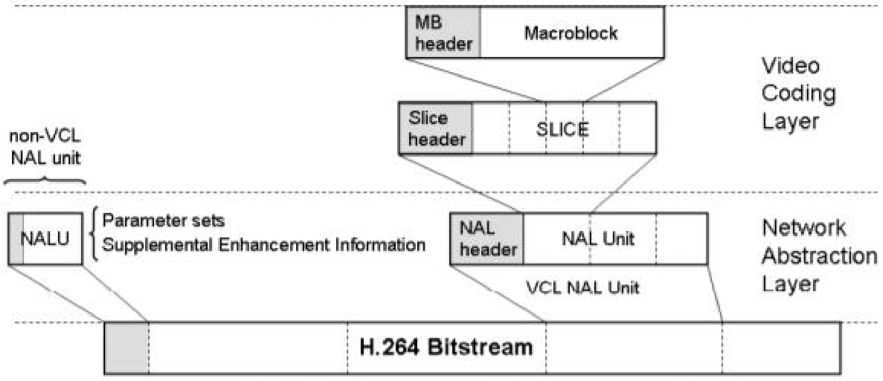


Figure 5 - H.264 bitstream encapsulation [11].

4. High Efficiency Video Coding/H.265

The HEVC/H.265, as other video coding standards, defines the syntax and semantics of a coded stream containing a compressed representation of video sequence [9]. The constrains and functionalities of an encoding system are defined through profiles and levels, while the syntax elements along with semantics define the standard-compliant decoding process, which ensures that every decoder conforming to the standard is capable of reconstructing the same coded video frames. Since the actual encoding algorithm is not standardised, it allows manufacturers to develop proprietary optimisations, competing in the global market with different solutions in terms of compression ratio, coding complexity, coding rates, etc.

The functional diagram of Figure 6 applies to HEVC, which follows a conventional hybrid coding architecture. The encoder uses both intra and inter-picture prediction to exploit spatial and temporal redundancies, respectively. Subsequently, the prediction residue is transformed by a linear spatial transform, and the resulting coefficients are then quantised and entropy coded, further reducing data redundancy. In the coding process, the Quantisation Parameter (QP) is of utmost importance, since this is used to control the bit rate and has direct impact on the video quality. In HEVC, the range of usable QPs is defined between 0 and 51 and an increase of 6 doubles the actual quantisation step size used in the quantisation function. The final coded stream comprises the video data (i.e., quantised coefficients) multiplexed with signalling information, which includes prediction modes, motion vectors, in-loop filtering modes and well as other control data related to data structures, bitstream headers, etc.

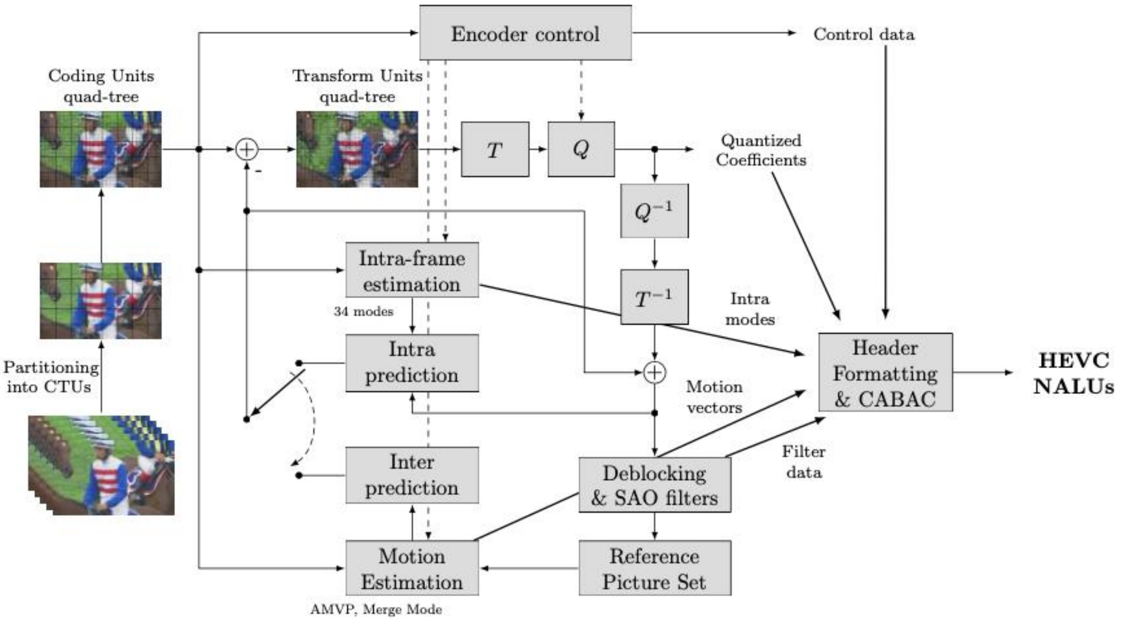


Figure 6 – Functional diagram of a hybrid video encoder, such as HEVC [12].

The HEVC standard brings some new data structures and tools, evolving from previous encoders. As shown in Figure 6, the encoding process follows a conventional block-based approach, where each picture is partitioned into Coding Tree Units (CTUs) with a maximum size of 64×64 pixels. Then, a dynamic quad-tree partitioning is used to divide CTUs into Coding Units (CUs), which are the basic processing units for prediction. For the transform and quantisation operations, Transform Units (TUs) are used. The prediction residuals are computed by using one out of 34 possible intra-prediction modes or symmetric/asymmetric motion compensated partitions from inter-prediction. Motion estimation is performed in the pictures listed in the Reference Picture Set, which was introduced in the HEVC to increase error robustness and easy synchronisation with reference pictures in case of data loss. To smooth the coding artefacts and improve the overall quality of the reconstructed pictures, the HEVC not only uses a deblocking filter but also a Sample Adaptive Offset (SAO) filter. Finally, the control data, quantised coefficients, prediction modes, motion vectors and filter information are multiplexed and entropy-coded using Context Adaptive Binary Arithmetic Coding (CABAC). The compressed bitstream is then packed into NALUs.

4.1. High level syntax

High level syntax elements of HEVC are relevant to provide tools for signalling and stream partitioning, which combine coding features with robust transmission. The concept of VCL and NAL allows to organise the syntax structures into logical units for transport, i.e., the NALUS. HEVC inherits several high-level syntax elements from H.264, such as the NAL, parameter sets, the use of Picture Order Count (POC) and Supplemental Enhanced Information (SEI) messages for auxiliary data. However, some important features defined in H.264 were not included, such as Flexible Macroblock Ordering (FMO), redundant slices, arbitrary slice order, data partitioning and switching slices. In addition, several new high-level features are introduced, such as the

Video Parameter Set (VPS), tiles and wavefront tools for parallel processing, dependent slices for reduced delay, and a new reference picture management concept. Moreover, new picture types were introduced to increase the random-access flexibility and temporal sub-layer switching [13].

Parameters set

The encoder uses a set of signalling packets to share data with the decoder, referred to as parameter set units. This information was introduced in previous standards to deal with vulnerabilities in the transport of signalling information. In this approach, such set of parameters is multiplexed in the bitstream and repeated as many times as required by the application, e.g., streaming, storage among others. The HEVC includes three sets of parameters, Sequence Parameter Set (SPS), Picture Parameter Set (PPS) and the newly introduced VPS [9]. The new set of VPS parameters contains information related to the different layers of the video signal, avoiding duplications by providing signalling for every layer, capable of supporting multiple layers. Moreover, VPS also carries information for session negotiation (e.g., profile, level and tier). The SPS contains information related to the whole sequence (i.e., it should affect all coded slices) and related to the decoder operation point, as well as flags to control optional tools and scalability. The PPS is responsible to carry information that may change for every picture, such as, initial QP, flags for picture related tools and tilling information.

Picture types

The HEVC standard supports several picture types, which are classified as (i) Random-Access Points (RAP), (ii) leading pictures and (iii) temporal sub-layer access pictures. The use of different types of pictures allows the coded stream to be more flexible, supporting a wider range of applications and robust communications. They are important to provide random-access, which is also a critical feature for channel switching, seek operations and dynamic streaming services. Moreover, the use of RAP also increases error robustness as they provide reset points for starting a new decoding sequence after data loss [12].

Random-Access Points

RAPs consist of pictures used to provide access points in the stream with no dependencies from other pictures, thus allowing to start the decoding process at such points [14]. A standard compliant stream must start with a RAP picture, which must belong to temporal sub-layer 0 and should be intra-coded, i.e., it must not use previously coded frames as reference for prediction. One should note that there may be frames compressed with intra-prediction mode only, but not marked as RAP. It is always possible to start decoding from a RAP frame onwards, and to output any subsequent pictures in the display order, even if all pictures that precede the RAP in the decoding order are discarded from the stream. An Instantaneous Decoding Refreshing (IDR) picture can be used for random access, since this is an intra-coded picture, which means that no dependencies exist from other pictures. However, if an IDR picture is an RAP, then pictures following the IDR in decoding order cannot use pictures decoded prior to the IDR picture as reference, i.e., neither the IDR nor any other subsequent frame in the decoding order can have prior dependencies. IDR frames are allowed to have leading pictures, i.e., frames that follow the RAP picture in the decoding order but precede it in the display order. Nevertheless, they must be decodable on their own to create a true RAP, where all subsequent frames in the decoding order can be decoded and displayed.

Introducing RAPs reduces the coding efficiency since no prior reference frames can be used. In order to reduce this impact, the HEVC standard introduces a new RAP picture, referred to as a Clean Random Access (CRA) picture. These frames do not refresh the decoding process, allowing leading pictures to depend on frames that precede the CRA picture in the decoding order. This enables a more efficient prediction structure [14], while maintaining an access point to begin the decoding process. However, it should be noted that some leading pictures may not be decodable due to prior dependencies.

The HEVC specification supports stream splicing, which allows taking a particular stream from a RAP (both IDR and CRA frames) and inserting it into another bitstream at a random-access point. In case of bitstream splicing, starting from an IDR frame, the leading pictures do not have dependencies from frames prior the RAP, thus all frames are decodable. However, whenever this starts from a CRA frame, the leading associated frames might not be decodable, because some of their references are not present in the combined stream. To make the splicing operation straightforward, the NAL units containing the CRA picture are changed to a type named as Broken Link Access (BLA).

Leading and trailing pictures

Leading pictures correspond to frames that follow a particular RAP frame in the decoding order, but preceding it in the display order. A trailing picture is a frame that follows a particular RAP frame in both decoding and display order. Figure 7 shows examples of leading and trailing pictures. In the figure, frames are illustrated in the display order and the number associated with the frame type (i.e., I, P and B) represents the decoding order. In this example, frame I$_1$ corresponds to a CRA picture. Leading and trailing pictures are considered to be associated with the closest previous RAP picture in decoding order, such as frame I$_1$. Also, all leading frames of a RAP frame must precede, in decoding order, all trailing frames that are associated with the same RAP. This means that the following order is imposed by the HEVC standard: 1) RAP picture, 2) associated leading pictures and 3) associated trailing pictures. Due to the introduction of more flexible random-access points using CRA pictures, it is important to mark the leading and trailing pictures, so the decoder can be aware of which frames can be correctly decodable whenever it starts decoding the bitstream.

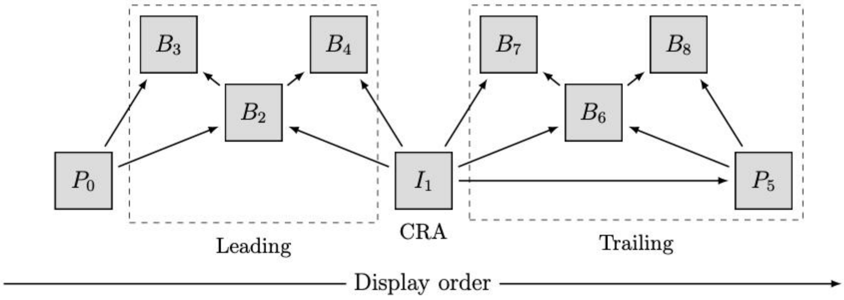


Figure 7 – Leading and Trailing Pictures in regard to an access point CRA [13].

There are two types of leading pictures:

- The Random Access Decodable Leading (RADL) pictures, which only depend on the associated RAP picture and do not have any prior dependencies to trailing pictures associated with the previous RAP picture;
- The Random Access Skipped Leading (RASL) pictures, which may have dependencies to prior trailing pictures. When random-access is performed at the associated RAP frame, these frames cannot be correctly decoded, therefore they have to be skipped. In the example of Figure 7, the leading pictures correspond to RASL pictures.

Temporal sub-layer switching pictures

A Temporal Sub-Layer Access (TSA) picture is a trailing picture that marks a temporal layer switching point. When decoding a sub-set of temporal layers, if a TSA picture is found in the temporal layer just above the maximum temporal layer currently being decoded, then it is possible to start decoding any number of additional temporal layers. For example, frame P_6 in Figure 8 is considered a TSA picture since it only depends on a frame belonging to temporal layer 0, as well as any subsequent frame predicted from the TSA frame P_6. In this example, if the decoder is only decoding temporal layer 0, then after decoding P_6 it can also decode layer 1, or decode all the available three layers. Although these frames allow higher flexibility in terms of layer switching, they severely constrain the prediction of frames following the TSA picture.

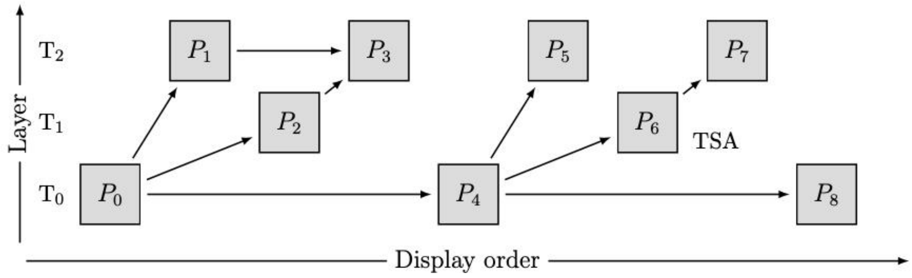


Figure 8 – Temporal Layers and access pictures: example with 3 layers T0 to T2 [13]

To reduce the constrains in frame prediction due to layer switching, HEVC introduces the Step-wise Temporal Sub-layer Access (STSA) picture. These pictures have similar purpose as the TSA, but they only guarantee that frames in the same temporal layer as the STSA frame are decodable, by not using frames preceding the STSA as reference in its temporal layer. One example of a STSA frame is shown in Figure 8 in frame P_2. This frame can be used to switch to layer 1, because there are not dependencies to any prior frame in the same layer (P_2 only depends on P_0). However, it cannot be classified as TSA picture because P_3 has dependencies to a prior frame in the decoding order (P_2). Summarising, STSA frames can be used to switch to a particular layer, while TSA frames can be used to switch to any layer.

4.2. Picture partitioning

The high-level segmentation of a picture in HEVC is achieved by using four different approaches associated to different data structures: regular slices, dependent slices, tiles and Wavefront

Parallel Processing. Picture partitioning normally serves one or more of the following three purposes:

- Error robustness: partitioning the picture into smaller self-contained units in order to increase error robustness, allowing to re-synchronise both the parsing and decoding processes in case of data losses;
- Network adaptation: adapting to the network constraint of the Maximum Transmission Units (MTU) size, found for example in Internet Protocol (IP) networks. Such packetisation scheme restricts the maximum number of payload bits within a slice regardless the size of the coded frame. To keep each slice within this limit and minimise the packetisation overhead, a variable number of coding units is used for each slice;
- Parallel processing: partitioning the coded frame into processing data units, which can be encoded in parallel. This is achieved by dividing the coding units, such that they can be encoded and decoded independently of each other.

Slices

One slice in HEVC may comprise either the entire frame or a section of it, and all the associated data (i.e., entropy symbols, prediction and residue information) can be independently decoded. Each slice is transported in a different NALU. As some dependencies across slice boundaries are disabled, each slice can be independently reconstructed, regardless of whether previous slices were lost or incorrectly decoded. Since each slice is independent from each other, it is straightforward to process multiple slices in parallel, without inter-process communication (except for inter-frame prediction). Although this is a simple approach, it incurs in substantial coding overhead due to the higher number of slice headers, and reduction of causal neighbours for prediction (due to lack of predictions across slice boundaries). Another disadvantage of using slices for parallel processing is that they are also used to meet the MTU size constrains. Therefore, it might be impossible to meet both requirements using only one technique.

HEVC also supports partitions of dependent slices. Such type of slices allows the partitioning of a frame at the coding unit boundaries without breaking intra-frame predictions. Moreover, as they have a smaller slice header, they are less costly in terms of signalling overhead than regular slices. Dependent slices are normally used to reduce the end-to-end delay by allowing part of the slice to be transmitted while the rest of its data is still being processed.

Tiles

Tiles are a new data structure that was included in the HEVC standard, which enable frame partition into groups of CTU using horizontal and/or vertical boundaries [15]. Figure 9 shows two examples of slice partitioning using tiles. In both cases the frame is partitioned into three tiles by using two vertical boundaries. By only using boundaries to define tiles, signalling is achieved with a small overhead. These boundaries are defined in the PPS NALUs.

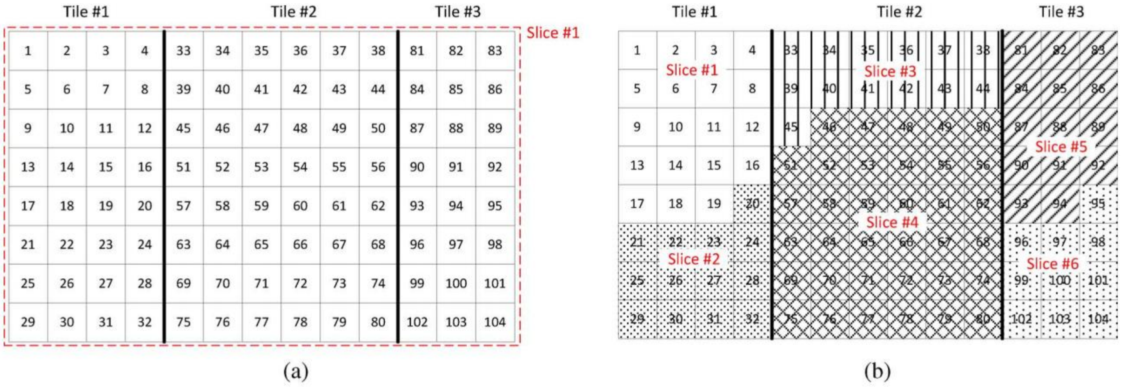


Figure 9 – Tile and slices in HEVC [15].

Although the HEVC standard enables the simultaneous use of slices and tiles, some constrains are imposed in order to reduce complexity. One of two cases must be used: (i) all CTUs within a tile belong to the same slice, or (ii) all CTUs within a slice belong to the same tile. These cases are illustrated in Figure 9 (a) and (b), respectively.

The use of tiles provides several advantages because they enable frame splitting and processing in a non-rigid manner. Specifically, they improve frame partitioning for parallel processing when compared to slices, by reducing the required overhead. Moreover, they also improve the MTU size matching and reduce the line buffer memory. Finally, as tiles allow flexible partitioning, they enable the definition of ROIs for non-uniform video coding.

Wavefront parallel processing

The use of either slices or tiles to process each frame using parallel threads requires the use of different entropy coding contexts for each slice/tile, in order to make them independently decodable. As a consequence, the compression efficiency decreases. In order to overcome this issue, the concept of Wavefront Parallel Processing (WPP) was introduced to enable efficient parallel processing, where each slice (or tile) is divided into single rows of CTUs which can be processed in parallel. However, entropy coding and prediction are enabled across CTUs of different rows, in order to avoid loss in coding efficiency. Figure 10 illustrates an example of WPP using at least five different threads. Each square corresponds to a CTU and the numbers correspond to the processing time instance of each CTU. Using this staggered start, which appears like a wavefront, the parallelisation may use as many threads as the number of CTU rows available in the video frame. However, although the number of threads does not affect the coding efficiency, the required inter-process communication substantially increases.

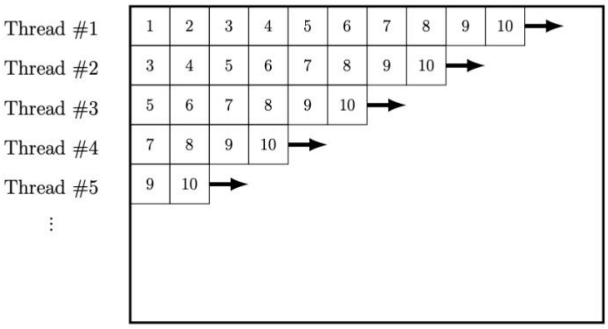


Figure 10 – Wavefront parallel processing (numbers indicate the processing instant of each CTU) [15].

4.3. Data structures

Despite the fact that HEVC standard follows a traditional hybrid video coding architecture, it introduces significant changes in the data structures when compared to the previous standards (e.g., H.264/AVC). Each picture is divided into CTUs, where the maximum CTU size is an encoder configuration parameter that is signalled to the decoder and should be one of the following: 64×64, 32×32 and 16×16 pixels. Each CTU can be recursively partitioned into four smaller units, referred to as CUs until it reaches the minimum unit size, which cannot be smaller than 8×8 pixels. Similar to the maximum CTU size, the minimum size is also defined at the encoder. For instance, in case of homogeneous regions, large CUs can be used to represent such regions by using a smaller number of symbols than in the case of using several small units. Figure 11 illustrates an example of CTU partitioning into CUs and the quad-tree structure represented on the left side with the corresponding coding order shown on the right side. This structure allows a multilevel hierarchical quadtree structure to be specified in a simple and elegant way, with size-independent syntax representation [16].

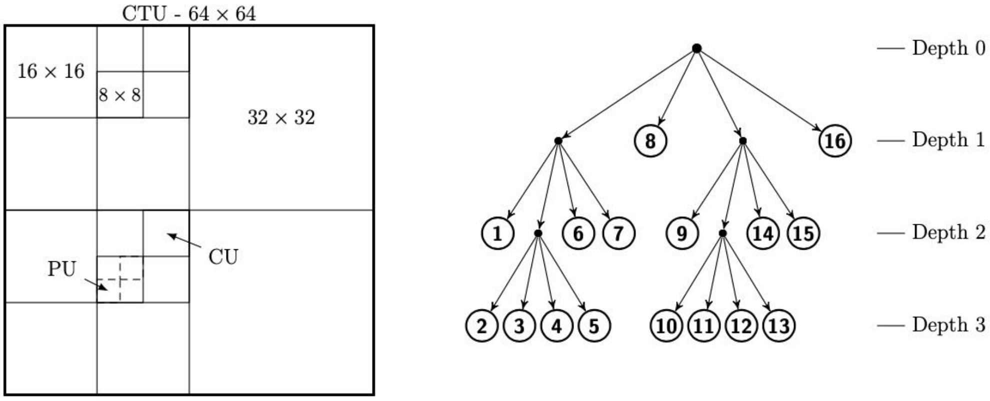


Figure 11 – The structure of a CTU, quadtree and CUs.

The CU is the basic processing unit in HEVC that is further partitioned into two other types of units, specifically Prediction Units (PUs) and TUs, which are data blocks used for prediction and transform operations, respectively. The decision whether to use intra- or inter-frame prediction modes is taken at the CU level. However, depending on the partitioning of the CU into PUs, different intra- or inter-coding modes can be used for each PU.

4.4. Prediction modes

Prediction techniques play an important role in image and video coding standards, due to their ability to reduce the signal redundancy based on the previously encoded pixels. These techniques include the directional intra-frame prediction used to reduce spatial redundancy, and inter-frame motion compensation to remove the temporal redundancies.

Intra-frame prediction

Intra-frame prediction is used to efficiently reduce the spatial redundancy within video frames. Intra-predicted blocks are obtained using the previously encoded pixels around the current PU. When the intra prediction mode is chosen for a CU, the size of the PU needs to be the same as the CU except when the smallest CU size is selected and further division into four PUs is allowed. For example, when the smaller CU size is 8×8, the encoder may decide to divide it into four PUs of 4×4 pixels, each one having its own intra-prediction mode. This is signalled using a binary flag. The use of small PUs is useful for regions with many details that require fine-granularity prediction.

In HEVC, there are 35 intra-frame prediction modes: Planar, DC and 33 angular modes, as shown on the left side of Figure 12. For each PU, the encoder chooses the best mode among all the available intra-prediction modes, based on the previously reconstructed pixels. The reconstructed reference pixels used in the prediction process belong to the neighbouring blocks located at left-down, left, top-left, top and top-right positions, as can be seen in the example shown on the right in Figure 12. Given an N×N PU, intra prediction requires the top neighbouring row of 2N pixels, the left column of 2N pixels and the top-left neighbouring pixel. Due to data unit boundaries (e.g., slice or tile boundaries) and constrained intra-prediction (i.e., reference pixels belonging to inter-predicted PUs are not used in order to remove error propagation from potentially erroneous reconstructed reference frames), neighbouring pixels might not all be available to be used as reference for intra-prediction, resulting in an incomplete set of neighbouring reference pixels. In order to overcome this issue and to allow all possible modes, HEVC uses reference sample substitution to replace the unavailable reference pixels with the closest available ones, in order to allow the use of all intra-prediction modes.

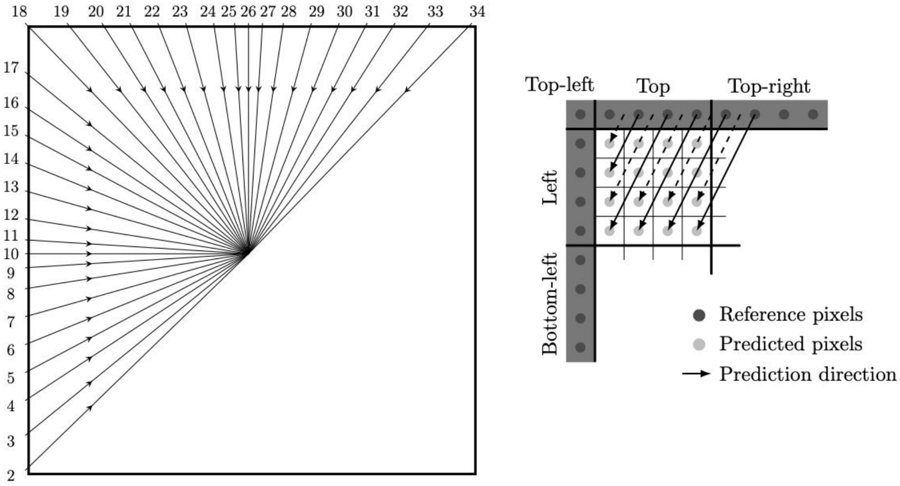


Figure 12 - Directional intra prediction modes in HEVC (left) and example of Mode 32.

As shown in Figure 12, the angular modes are designed to provide a dense coverage near the horizontal (mode 10) and vertical (mode 26) directions. Moreover, the angular directions are coarser as it gets closer to the diagonals. This reflects the observed statistical prevalence of the angles and the prediction efficiency. To improve the efficiency of angular prediction, when the reference samples need to be projected, a bilinear interpolation is used from the two closest pixels using an accuracy of 1/32. The dashed arrows on the right of Figure 12 correspond to a projection of interpolated pixels.

The Planar and DC modes are particularly efficient in the prediction of smooth regions. The DC mode uses the average or the neighbouring reference samples to generate a constant prediction for the current PU as a whole, i.e., all pixels are predicted from the same value. The Planar mode predicts the PU through a linear interpolation from the four closest neighbouring reference pixels.

Sample smoothing

In HEVC, sample smoothing is applied in two steps to improve the overall performance of the prediction accuracy. Firstly, the reference samples are filtered and, secondly, the PU is filtered after prediction. The reference sample smoothing is conditionally applied, based on the PU and intra-prediction mode. The second filter in HEVC is the boundary smoothing, aiming at removing the discontinuities along block boundaries due to intra-prediction.

Inter-frame prediction

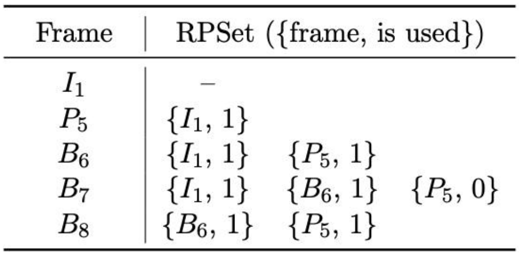
The efficiency of video coding algorithms consistently relies on inter-frame prediction techniques to reduce temporal redundancy. The underlying idea of inter-frame prediction is to estimate the current frame from one or more previously encoded frames used as reference with block-based motion compensation. The most common technique used for motion estimation in HEVC is the block matching algorithm, previously presented. The HEVC standard defines an Advanced MV Prediction (AMVP), adopted for efficient MV coding, which in turn may use the so-called Merge Mode or differential coding of the motion information [16].

Motion estimation and compensation

Motion estimation consists in searching for a block with the highest similarity with the current PU to be predicted, using block-matching algorithms over previously encoded frames, i.e., the reference frames. Typically, reference frames correspond to past or future temporally adjacent frames. The frame encoding order determines which references are selected and which future frames are available. The reference block that results in the lowest error is selected as the best block for prediction. To identify the best matching block to be used for motion compensation, the difference between the target and the reference block positions is encoded as a two-dimensional MV. The HEVC allows for either one or two MVs to be used for each PU. Video frames of type P are predicted from one single reference (one MV) while frame of type B are predicted from two references (two MVs). This results in uni- or bi-directional coding, respectively. Furthermore, weighted prediction can also be used, which consists in scaling and offset operations performed on the prediction block to improve efficiency. The MVs only refer to frames included in a reference frame list, where each frame is identified by an index. Therefore, to fully describe inter-frame prediction, a combination of two parameters is required: MV and reference frame index.

Reference picture management

HEVC introduces the Reference Picture Set (RPSet) concept, which defines how previously decoded pictures are managed in the Decoded Picture Buffer (DPB). Decoded frames in the DBP are grouped in one of the following categories: (i) short-term reference, (ii) long-term reference and (iii) unused for reference. Once a frame is marked as unused for reference it is no longer used for motion compensation and it will be discarded from the DPB after being displayed. The status of the DPB is encoded for every slice, instead of the implicit management used in previous standards. With the RPSet concept, no information from earlier frames in decoding order is needed to maintain the correct status of the DPB. Figure 13 shows the RPSet for some frames illustrated in Figure 7. As expected, for the CRA frame (I_1) no information is sent to the decoder because there is no inter-prediction and the DPB should be empty. For the remaining frames, two types of information are encoded: frame number and a binary flag to indicate whether it is currently used. This is required because any frame not listed in a given RPSet will be discarded by the decoder and then subsequent frames cannot use them as reference.



Frame	RPSet ({frame, is used})		
I_1	–		
P_5	$\{I_1, 1\}$		
B_6	$\{I_1, 1\}$	$\{P_5, 1\}$	
B_7	$\{I_1, 1\}$	$\{B_6, 1\}$	$\{P_5, 0\}$
B_8	$\{B_6, 1\}$	$\{P_5, 1\}$	

Figure 13 - An example of a Reference Picture Set [16].

The RPSet concept provides a basic level of error robustness to the reference picture management, since the decoder is always able to detect loss events and correctly identify the missing frames.

4.5. Transform and quantisation

Similarly, to any other hybrid encoder, in HEVC a transform is applied to the residual signal followed by quantisation. The residual block is partitioned into multiple TUs using core transform matrices from the DCT basis functions to define an integer matrix of 32×32 points. Then sub-sampled versions of this matrix are used to derive the remaining transform matrix sizes down to 4×4.

The HEVC follows the same principle as previous standards in regard to quantisation. After applying the transform operation to the prediction residue, quantisation is applied to the resulting coefficients using a uniform-reconstruction scalar quantiser scheme based on QP. The standard supports QP values ranging from 0 to 51, where an increase of 6 in the QP corresponds to doubling the quantisation step. This procedure is the main cause for the coding distortion, as it performs a many-to-one mapping, which cannot be reversed.

4.6. Entropy coding

HEVC only uses CABAC which is an arithmetic coding method that only uses binary symbols, considering different probability models for each symbol. Entropy coding efficiency is closely linked with the context model selected. Therefore, this was carefully designed in the HEVC standard, extending the functionality previously defined for H.264/AVC. For example, the depth of the partition tree or residual transform tree is exploited in order to derive the context models for several syntax elements. Regarding the transform coefficient coding, CABAC uses a scanning method to firstly organise the coefficients and then encodes the position of the last non-zero transform coefficient. Moreover, a significance map is also encoded along the sign bits and the levels of the transform coefficients for which three scanning methods are available: the diagonal up-right, the horizontal and the vertical scan. The coefficient scanning is implicitly selected and always performed in 4×4 sub-blocks.

4.7. In-loop filters

The HEVC standard uses two filtering procedures over the reconstructed pixels before writing them into the frame buffer, namely the Deblocking Filter (DBF) and the SAO filter. The DBF aims to reduce blocking artefacts and it is applied to samples that are spatially adjacent to the PU or PU boundaries, considering a 8×8 grid. Such grid-based restriction reduces the computational complexity and facilitates parallel-processing. Three strength levels can be used by the DBF depending on the coded block characteristics.

The SAO is a non-linear filter, that is adaptively applied to all samples of the image, after the deblocking filter. The SAO filter modifies the samples by adding an offset value, extracted from lookup tables transmitted by the encoder. For each CTU, the encoder decides whether to apply the SAO filter or not, and if used, one out of two filter types is applied, namely the band offset or the edge offset. In the band offset mode, the added offset value depends on the sample amplitude, while the edge offset uses the gradient to classify and derive the offset value.

4.8. Network abstraction layer

In HEVC, the NAL is defined to decouple the video coding algorithm, also known as the VCL, from the transport layer where different transport protocols may be used. The NAL concept is used for stream partitioning and synchronisation, which was found useful to deal with lossy network environments in previous standards. Using such structure, the compressed video stream is composed of different data units, the NALUs, which can be used to map the VCL data, comprised of coded video slices (or frames), on various transport protocols, such as Real-Time Protocol (RTP), MPEG-2 Systems and MPEG DASH (Dynamic Adaptive Streaming over HTTP) [17]. Figure 14 shows the general protocol stack that can be used in different application scenarios. Particularly interesting for the project are the following options NAL/RTP/UDP/IP or NAL/TS/UDP/IP, as they provide the timing and transport requirements for real-time multi-stream video transmission.

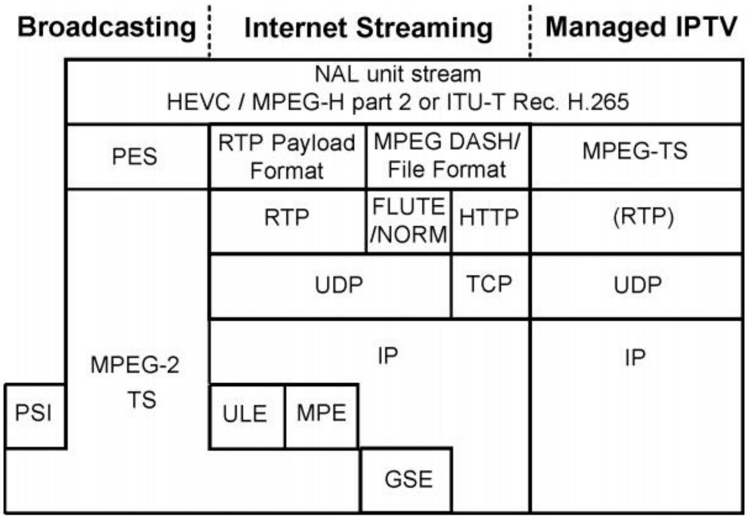


Figure 14 – General protocol stack for HEVC [17].

NALUs are classified into two categories: (i) VCL and (ii) non-VCL, according to type of payload data, either coded video or associated data, respectively. Each NALU has a fixed-length header code of two bytes to describe its content and to allow extra error robustness at the cost of a small bit-rate overhead. Figure 15 shows the structure of the HEVC NALU header. The first bit F, referred to as *forbidden_bit*, is always zero in order to ensure that a start code can be detected across different protocol standards. This is followed by the NAL type, which identifies the content, enabling the decoder to either use it, if required, or simply discard it. Another syntax element introduced by the HEVC standard is the *temporal_id_plus_one* (TIDP), which defines the temporal layer where the NAL unit belongs. This allows implicit temporal scalability (layers ranging from 0 to 6) by immediately discarding the NALU, if it belongs to a layer that is higher than the pre-defined one. In the header, six bits are reserved for future extensions, preventing the creation of extra NALUs for scalable and multiview extensions.

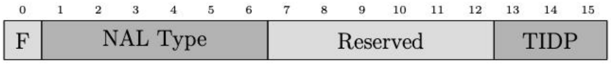


Figure 15 – NALU header in HEVC.

The encapsulation structure of NALUs is shown in Figure 16. Coded slices are the basic data unit from VCL that are encapsulated as NALUs. The figure shows both non-VCL NALUs (e.g., SPS and PPS) and VCL NALUs, which comprise the Raw Byte Sequence Payload (RBSP). RSBP is byte-aligned with the NALU by inserting one stop bit and then filling the remaining ones with zeros.

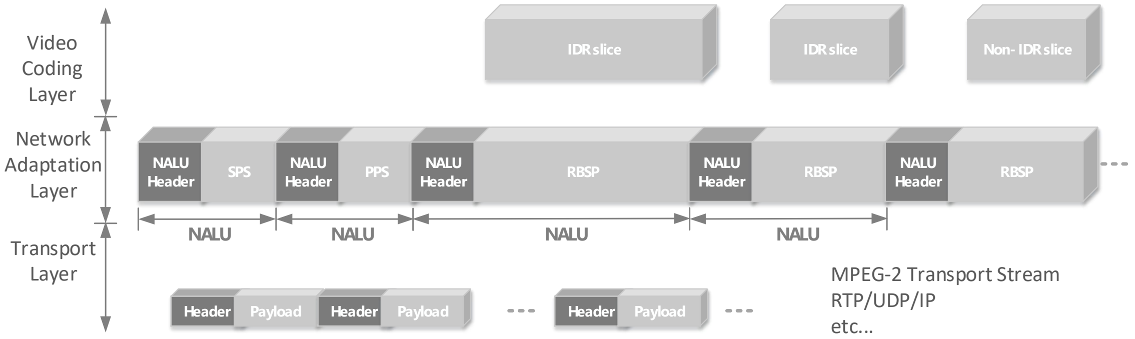


Figure 16 – NALU encapsulation.

5. ROI-based video coding

As mentioned before, ROIs are defined as image regions where some relevant visual content may be located. The shape of ROIs can be either regular shape, such as rectangular, or completely arbitrary. They can be defined manually through user interaction, or computed through visual attention models, saliency detection algorithms, object detection, etc. ROI coding aims at providing better quality and/or better protection against the errors in the ROI, as opposed to the rest of the visual scene. This differentiated coding can be used in various types of video applications where some kind of perceptual coding may be useful or to increase the coding efficiency in constrained video delivery systems. Some coding systems allow assigning higher priority and quality to a ROI over the rest of the frame (non-ROI). The quality and priority difference between the ROI and non-ROI will depend on the video application. For example, in video communications, the use of the available bit rate should be maximised to provide the best quality to viewers. In this case, lower quality non-ROI is acceptable by the human visual system since the viewers pay less attention to the non-ROI than to the ROI. Therefore, more bits can be allocated to the ROI without reducing the overall quality of the video. In this case, the ROI quality can be increased. In the literature, there are different approaches to achieve this goal: (i) use more bits in ROI (use a fine QP); (ii) reduce the bits in the non-ROI (use a great QP) and (iii) code the non-ROI regions in skip-mode. In addition to these approaches, several works proposed to adjust the QP of the ROI and non-ROI according to certain principles, such as the visual sensitivity of the human visual system or the ROI quality target. Chen *et al.* proposed a method that minimises the non-ROI information based on application of low-pass filters to the non-ROI region. In this case, it is not necessary to do any modification to the coder, since the technique is applied before the coding [18]. The same approach was used by Karlsson *et al.*, applying a spatio-temporal filter to re-allocate bits from the non-ROI to the ROI, after the ROI detection step was proposed [19].

ROI coding with H.264/MPEG-4 AVC and H.265/HEVC

In the most recent standards, the slice structures provide support for ROI coding. Although there are no coding tools specifically designed for ROIs, the flexibility of existing ones allows the implementation of efficient methods. For instance, in H.264/MPEG-4 AVC, each slice is identified with slice group ID, which determines the coding order of the MBs. In the case of two slice groups, all the MBs of slice group 0 are coded before the coding the MBs of slice group 1. Conventionally, video coding standards support encoding MBs only in raster order. However, with the introduction of FMO in H.264/MPEG-4 AVC standard, the order of assigning the MBs into slices has been liberalised. Here, the slides are coded into separate NAL units [20] making them totally autonomous from others. Seven different types of FMO modes are defined: Interleaved, Dispersed, Foreground with left-over, Box-out, Raster-scan, Wipe, and Explicit. The FMO type 2 has usually been used in ROI coding, but it is only suitable for ROIs with rectangular shapes and cannot represent irregular regions in an efficient manner. In this case, the type 6 is the most general one, where the ROI shape is user-defined. The ROI position must be firstly detected with a pre-processing algorithm.

Leuven *et al.* [21] presents an implementation of multiple ROI models in H.264/MPEG-4 AVC standard to enhance the quality of video surveillance. The ROIs are user-defined, i.e., no detection algorithm is used. The results show that a convenient selection of the ROI, in combination with a suitable choice of quantisation parameter and the FMO type can reduce bandwidth usage while maintaining the same video quality. More recently, Peng *et al.* [22] present a ROI privacy protection scheme for H.264/MPEG-4 AVC video in Closed Circuit Television (CCTV) based on FMO. To encrypt the ROI, the FMO technology of the H.264/MPEG-4 AVC is used. Firstly, the human face regions in the video are detected and extracted. Then, ROIs are mapped into slice groups and such regions are encrypted using selective video encryption based on chaos.

Since the standardisation of H.265/HEVC other works have addressed the problem of ROI coding. Thus, Xu *et al.* [23] proposed a ROI based HEVC coding approach for videos with a novel hierarchical perception model of a face. Moreover, a weight-based unified rate-quantisation scheme is proposed to adaptively adjust the value of QP rather than the conventional pixel-based unified rate-quantisation scheme. In [24], Xing *et al.* proposed a surveillance video coding method with HEVC quadtree partition-based ROI extraction. In this method, ROI-layer and background-layer videos are produced with the help of background modelling and ROI extraction, and then encoded into ROI stream and background stream, respectively. At the decoder side, the ROI-layer video can be selectively decoded also using the background data, that is almost static. A fast ROI-based HEVC coding system is also proposed for surveillance in [25], with the objective of reducing the bit rate and computational cost of non-ROI regions. The method is based on fast ROI detection as pre-processing, i.e., before coding, producing a binary mask to identify the different regions. Then the bit rate is allocated to such different regions following a smart strategy where the ROI quality is enhanced in comparison with non-ROI image regions.

Meddeb et al. [26] developed a rate control scheme for HEVC standard with the aim of improving the perceptual quality of ROI. Here, the algorithm was developed for a videoconferencing system, where the ROIs (typically, faces) are automatically detected and each CTU is classified in a region of the interest map. After that, this map is given as input to the rate control algorithm and the bit allocation is made accordingly. In the systems described above, a common aspect is the fact that they all attempt to allocate more coding rate to the ROI in order to obtain better quality in the such most relevant regions than in the remaining non-ROI regions. This is usually achieved by controlling the QPs assigned to ROI vs non-ROI regions. Zhang *et al.* [27] built Rate Distortion (RD) models for ROIs and non-ROIs in the HEVC. The rate control scheme is proposed for ROI mode coding based on DCT coefficient model. CUs are categorised by their depth levels and whether they belong to ROI or non-ROI group. The proposed R-D model takes considerations of various statistical characteristics of transformed coefficient residues for CUs by multiple Laplacian Probability Density Functions (PDF). For the sake of improving the estimation of distortion, a machine learning approach is adopted by using historical results and parameters as the training sequence, and the distortion is predicted as the output of a neuron network. The rate control scheme is designed from GOP level to CU level. More recently, Chai *et al.* [28] proposed a ROI encoding solution to accelerate the processing and reduce the bitrate based on the H.265/HEVC with FPGAs.

ROI coding using scalability

In addition to spatial, temporal and quality scalability, the H.264/MPEG-4 SVC also supports ROI scalability. This type of scalability is appropriate to many scalable video coding applications. For instance, a mobile phone user may be required to extract only a particular ROI in a video; at the same time, other user with large portable device screen can extract another ROI to receive better video stream resolution. Thus, to fulfil these requirements, it would be necessary to transmit or store a scalable bit stream with different ROIs. Grois *et al.* [29] present a scalable ROI video coding algorithm which enables the adaptation to the position, size and resolution of the ROI. This algorithm has two methods for ROI coding, the first is based on inter-layer prediction and the second the uses FMO. In the first proposed method, the authors cropped the ROI from the original sequence and used it as a base layer and increased the ROI resolution in the enhancement layer. After this, inter-layer prediction is applied in the cropping areas. Compared to conventional single layer coding, the method incurs in low bit-rate overhead. However, this approach does not allow the existence of non-ROIs in the base layer and the size of ROI, in this layer, is constant along the sequence. In the second proposed method, the ROI is encoded with FMO type 2 (only to enhancement layers) and each ROI is represented by a rectangular shape with the ROI and non-ROI regions coded in separate slices. The algorithm does not implement the ROI detection method, meaning that the ROI is pre-defined by the user. Further, Lee *et al.* in [30] propose a scalable ROI algorithm (H.264/MPEG-4 SVC) which used the FMO with the Box-Out method in the coding process. Two methods for ROI detection were proposed, the passive (pre-defined by the user) and active setting of ROI (based on motion vectors). The active selection method produced better subjective quality than the passive selection method. The algorithm supports Fine Grain Scalability (FGS) in ROIs with low computing complexity in order to achieve better objective and subjective video quality.

6. Beyond HEVC

Beyond HEVC, the new VVC standard, also known as H.266, has been approved quite recently. In parallel with the open video coding process of JVET/MPEG, a few companies and the audio and video coding standard workgroup of China are also developing their own video codecs, such as VP10 [31], Daala [32], Thor [33], AVS3[34] and AV1, as shown in Figure 17. A description of the available coding tools in AV1 is provided in [35]. Syntax element definition and decoder operation logic in AV1 are available in [36]. More information of this standard is available in the next section.

In parallel with the development of the AV1, the third generation of Audio Video Standard (AVS3) of China has been created. This standard is built on the top of its predecessor AVS2 [37]. Currently, AVS3 has adopted many novel coding techniques including block partitioning structure, intra/inter and transform coding tools. The compression performance of the AVS3 is about 24.3% and 26.88% bit-rate reduction against AVS2 and HEVC, respectively, under 4K resolution sequences. These coding standard evolutions have software-based implementations, but still lack standalone real-time hardware implementations. They might be viable options for future versions of VIEXPAND. But, at their current market status, they do not present technical advantages over HEVC/H.265.

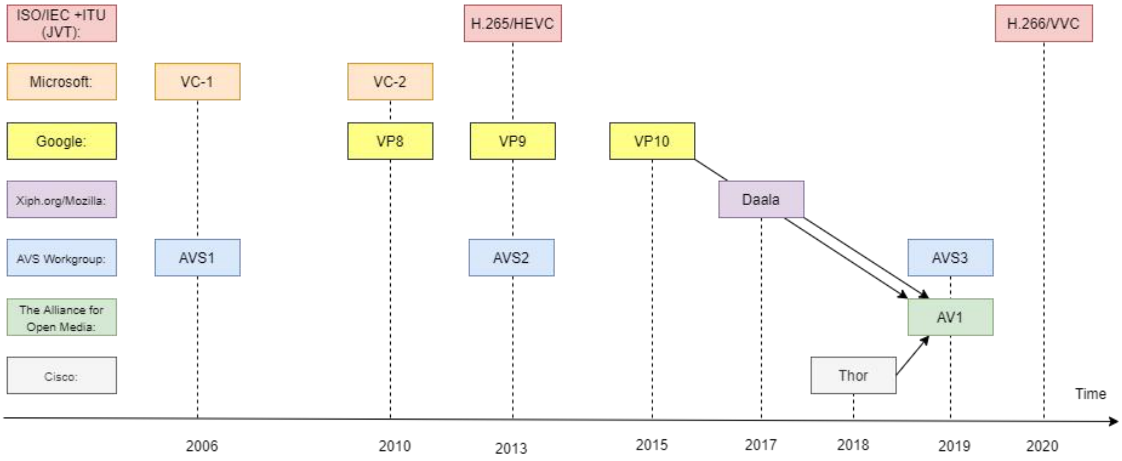


Figure 17 - Timeline of the new video code standards.

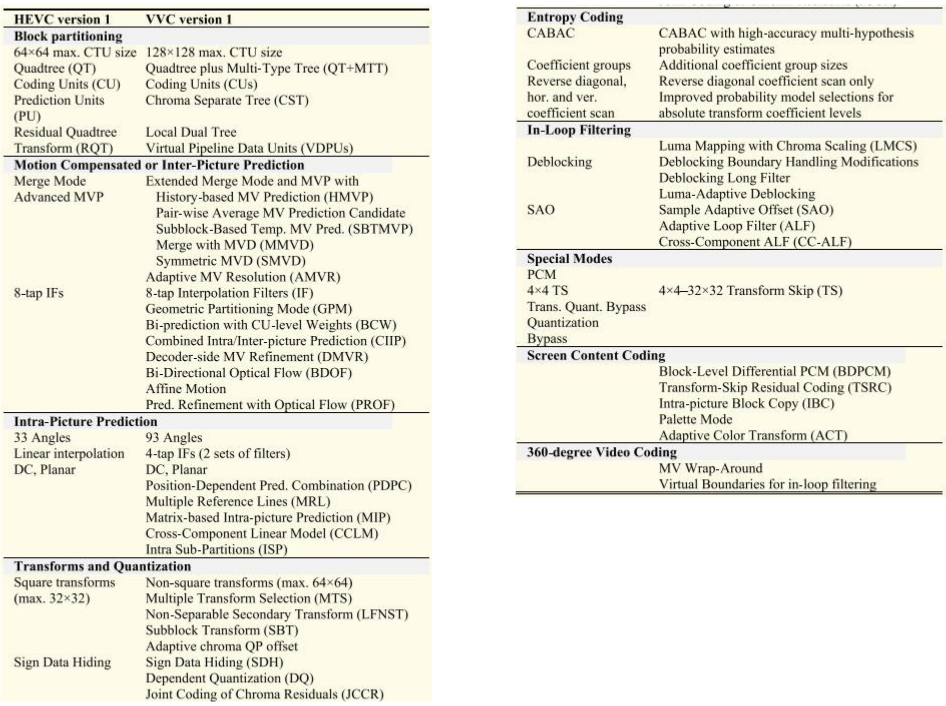
6.1. Versatile Video Coding

This section provides a brief overview of the most recent standard, VVC [38], also known as H.266, ISO/IEC 23090-3, MPEG-I Part 3 and Future Video Coding (FVC). This new video compression standard was finalised on 6 July 2020, by the Joint Video Experts Team (JVET), a joint video expert team of the VCEG working group of ITU-T Study Group 16 and the MPEG working group of ISO/IEC JTC 1. It is the successor to HEVC. It is formally approved as ITU-T H.266 and ISO/IEC 23090-3. The standardisation process started in 2015 and finished in 2020.

The primary target of VVC is to provide a significant performance improvement over HEVC. Apart from achieving major bit-rate savings over its HEVC and H.264 AVC predecessors, VVC was designed to provide and improve functionalities and coding efficiency for a broadened range of existing and emerging applications, including [33]:

- Video beyond the standard and high definitions;
- Computer-generated or screen content;
- Ultra low-delay streaming;
- Adaptive streaming;
- 360° video;
- Multilayer coding.

A VVC standard encoder follows the classic block-based hybrid video coding architecture known from its predecessors, HEVC and H.26X (see Figure 1). Although the same framework is applied, novel coding tools are included in each basic building block to further improve the compression. Table 2 provides an overview of the coding tools in HEVC version 1 and VVC version 1.



HEVC version 1	VVC version 1
Block partitioning	
64×64 max. CTU size	128×128 max. CTU size
Quadtree (QT)	Quadtree plus Multi-Type Tree (QT+MTT)
Coding Units (CU)	Coding Units (CUs)
Prediction Units (PU)	Chroma Separate Tree (CST)
Residual Quadtree Transform (RQT)	Local Dual Tree Virtual Pipeline Data Units (VDPUs)
Motion Compensated or Inter-Picture Prediction	
Merge Mode Advanced MVP	Extended Merge Mode and MVP with History-based MV Prediction (HMVP) Pair-wise Average MV Prediction Candidate Subblock-Based Temp. MV Pred. (SBTMVP) Merge with MVD (MMVD) Symmetric MVD (SMVD) Adaptive MV Resolution (AMVR)
8-tap IFs	8-tap Interpolation Filters (IF) Geometric Partitioning Mode (GPM) Bi-prediction with CU-level Weights (BCW) Combined Intra/Inter-picture Prediction (CIIP) Decoder-side MV Refinement (DMVR) Bi-Directional Optical Flow (BDOF) Affine Motion Pred. Refinement with Optical Flow (PROF)
Intra-Picture Prediction	
33 Angles	93 Angles
Linear interpolation	4-tap IFs (2 sets of filters)
DC, Planar	DC, Planar Position-Dependent Pred. Combination (PDPC) Multiple Reference Lines (MRL) Matrix-based Intra-picture Prediction (MIP) Cross-Component Linear Model (CCLM) Intra Sub-Partitions (ISP)
Transforms and Quantization	
Square transforms (max. 32×32)	Non-square transforms (max. 64×64) Multiple Transform Selection (MTS) Non-Separable Secondary Transform (LFNST) Subblock Transform (SBT) Adaptive chroma QP offset
Sign Data Hiding	Sign Data Hiding (SDH) Dependent Quantization (DQ) Joint Coding of Chroma Residuals (JCCR)
Entropy Coding	
CABAC	CABAC with high-accuracy multi-hypothesis probability estimates
Coefficient groups	Additional coefficient group sizes
Reverse diagonal, hor. and ver. coefficient scan	Reverse diagonal coefficient scan only Improved probability model selections for absolute transform coefficient levels
In-Loop Filtering	
	Luma Mapping with Chroma Scaling (LMCS)
Deblocking	Deblocking Boundary Handling Modifications Deblocking Long Filter Luma-Adaptive Deblocking
SAO	Sample Adaptive Offset (SAO) Adaptive Loop Filter (ALF) Cross-Component ALF (CC-ALF)
Special Modes	
PCM	
4×4 TS	4×4–32×32 Transform Skip (TS)
Trans. Quant. Bypass Quantization Bypass	
Screen Content Coding	
	Block-Level Differential PCM (BDPCM) Transform-Skip Residual Coding (TSRC) Intra-picture Block Copy (IBC) Palette Mode Adaptive Color Transform (ACT)
360-degree Video Coding	
	MV Wrap-Around Virtual Boundaries for in-loop filtering

Table 2 - Overview of Coding Tools in HEVC and VVC [2].

In VIEXPAND, the VVC should be the natural successor of HEVC. But concerning the current status of the technology available in the market, real-time hardware standalone VVC implementations do not exist yet. Therefore, the VVC standard is not n viable option for VIEXPAND, at this stage of the development.

6.2. From V10, Daala and Thor to AV1

Even though the performance of VP9 [39] is satisfactory, continued growth of the demand for high efficiency video applications, such as Augmented/Virtual Reality (AR/VR) and 360° video, calls for more efficient video coding standards. The latest VP10 standard achieves modest gains in coding efficiency. In late 2015, Google cooperated with the Alliance for Open Media (AOMedia) [40], which is a forum of more than 30 leading tech companies such as Microsoft and Mozilla, to jointly develop a royalty-free codec called AOMedia Video 1 (AV1). The evolution towards AV1 is shown in Figure 17. Most of the code of AV1 is based on Google's VP10 with minor additions from Cisco's Thor and Mozilla's Daala. It was finalised in 2019. The main goal of AV1 is to achieve a substantial compression gain over state-of-the-art codecs and scalability to modern devices with various link bandwidths, with a practical decoding complexity and hardware feasibility. Presently, AV1 can achieve an almost 30% reduction in average bitrate with the same quality when compared with the VP9 encoder. Moreover, compared with HEVC, AV1 has the following additional advantages:

- Royalty free: AV1 will be completely royalty free;
- Better compression: AV1 can save up to 30% in bandwidth for the same video quality over H.265/HEVC;
- Play everywhere: with support of Apple, Google, Microsoft, and Mozilla, all major web browsers will support this new codec.

However, since AV1 is a new codec, the hardware support it receives would not be as good as that of HEVC and the decoder may be energy inefficient. Presently, for live encoding of HD video on most devices, HEVC is definitely the only choice available in the market due to the existence of flexible standalone hardware solutions.

7. The HEVC/H.265 in VIEXPAND

Given the current status of standard video coding technology, the HEVC/H.265 is the best choice for the VIEXPAND project, due to its high coding efficiency and availability of real-time hardware solutions in the market. As pointed out before, despite a more recent standard exists (VVC/H.266), approved in 2020, no viable implementation solutions exist in the market yet. Moreover, the wide variety of coding tools and configuration options enables a high level of flexibility, which is also useful for the project to allow possible definition of different operational modes.

The different Profiles, Levels and Tiers allowed by the HEVC standard provide support for quite flexible implementations easily adapted to applications with very different requirements, in terms of video resolutions, colour formats, optional coding tools and bit rates [41]. In general, these options are included in the configuration setup of encoders, allowing operation in different modes and using computational resources accordingly. In VIEXPAND, these flexible setup configurations allow the use of the hardware system to be developed in quite different applications, without being strongly tied to the demo configuration planned for the end of the project. Taking into account the fact that the number of cameras to be used is also flexible (i.e., not fixed), by using an encoder supporting various Profiles and Levels, the prototype to be developed in the project has great potential for diverse industrial applications beyond the project lifetime.

Several features and coding tools defined in the HEVC standard are relevant for VIEXPAND in regard to the implementation of efficient ROI coding. One of the requirements in ROI coding is the possibility of unequal bit allocation in across each video image and along the time, i.e., throughout different consecutive images. This is necessary to provide higher quality in ROIs than in the remaining regions of the image, in order to preserve the fidelity of relevant details while keeping the global bit rate lower than agnostic encoding of the whole image area. The HEVC standard enables efficient encoding of ROIs through smart use and configuration of encoding options such as picture partitioning and data structures, presented in the Sections 4.2 and 4.3, combined with the possibility of controlling the quantisation parameters at CTU level.

In VIEXPAND the ROIs are user-defined, but other automatic techniques can be used to detect salient regions or objects, such as attention models, saliency detection algorithms through machine learning approaches. In this project, a ROI is defined as an arbitrary rectangular shape containing the image area of interest. For efficient encoding, such image area can be defined by the data structures available in HEVC, such slices, tiles or CTUs, depending on the ROI requirements. For instance, the inherent characteristics of tiles allow independent encoding/decoding of ROIs that are fully enclosed in a tile or slice. Moreover, since each ROI can be defined through a binary map (ROI map) associated to each video frame and comprising an integer number of CTUs, differentiated quality and bit rate control can be supported by dynamic rate-distortion control CTUs, located either inside or outside of the ROI. Therefore, picture partitioning, combined with the data structures of HEVC and ROI maps, allow a differentiated coding between ROI and background, to obtain significant coding efficiency gains over default coding configurations. The binary maps can be given as input parameters to the HEVC encoder and to the rate control module. Note that the Largest Coding Unit (LCU) in HEVC can be defined as a configuration parameter, allowing to use boundaries with different levels of granularity. Overall, the flexibility of HEVC standard includes a wide range of configuration and parametrisation options, which allow to further investigate the best methods for real-time ROI encoding, subject to highly dynamic conditions and optimised compression efficiency.

Another strategy is near-lossless encoding of a ROI, while background regions are coarsely encoded, based on pre-processing of the video frames, i.e., before entering the HEVC video encoder. In this case, the ROI quality can be either decoupled or loosely coupled with the encoding process and standard features of the compression algorithms. In this case, the pre-processing is responsible for filtering each input image, guided by a ROI map, such that regions outside the ROI are strongly low-pass filtered, to remove those signal components that require more encoding bits (i.e., high spatial frequencies). A spatio-temporal filter may also be considered for better temporal consistency. In VIEXPAND, this type of ROI coding strategy may be exploited for application scenarios where the ROI is defined as a region with irregular shape rather than rectangular, such as an arbitrary object. In this case, the ROI map can be defined as a gray-level map (i.e., not binary) to accommodate pixels of different relevance in the filtering process. Likewise, the HEVC encoder may not even be aware of the ROI because a “flat” QP configuration for all CTUs in each image will implicitly result in regions of different quality and reduced bit rate. However, the bit rate control will turn into a research challenge because it is planned to be implemented through an outer feedback loop, from the output encoding bit rate to the control of the pre-processing filter parameters, according to some target bit rate given as input parameter.

Finally, at the transport level, the HEVC/H.265 also presents the necessary features to support the project goals. For instance, NALUs of variable size are used to transport self-contained HEVC tiles or slices, which in turn may be independent ROIs. Multi-stream multiplexing is also supported by TS or RTP protocols, which are independent from the encoding process (i.e., VCL). Nevertheless, the different picture types supported by HEVC/H.265, in combination with the prediction modes, provide a valuable set of options to deal with harsh transmission environments, such as those that can be found in industrial plants. In particular, careful choice of RAP pictures allow to control and limit the effects of possible transmission errors when wireless networks are used between the planned VIEXPAND Capture and Transmission Centre (CTC) and the Monitoring and Control Centre (MCC) (from the Viexpand Technical Annex).

8. Conclusions

This Deliverable presents the main aspects of video coding technology that is behind the solution to be adopted in the VIEXPAND Project. The main coding tools used in the HEVC standard were described in detail, providing the necessary background information for control and parametrisation of the encoding system. Some recent developments in the field of ROIs coding using standard encoders were also presented. These are research results and proposals of coding methods particularly suited for ROIs, which bring up-to-date technical information and useful bibliographic references for the possible future developments within the Project. Finally, the technical characteristics of HEVC/H.265 are discussed within the scope of the VIEXPAND project, highlighting advantages and linking the most relevant options available in the standard with the specific objectives of the project, namely the support for efficient ROI encoding.

9. References

[1] Richardson, I.E.: The H.264 advanced video compression standard. Wiley, New York (2010). (ISBN: 978-0-470-51692-8).

[2] Ali j. Tabatabai, Radu S. Jasinschi, T.Na Veen, Motion Estimation Methods for Video Compression—A Review, Journal of the Franklin Institute, Volume 335, Issue 8, 1998. Pages 1411-1441.

[3] B. Bross, J. Chen, J. -R. Ohm, G. J. Sullivan and Y. K. Wang, "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC)," in Proceedings of the IEEE.

[4] ITU, Video Codec for Audiovisual Services AT PX 64kbits, 1994.

[5] ISO/IEC 11172-2 (MPEG-1), Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s Part 2: Video, 1993.

[6] K. Rijkse, "H. 263: Video coding for low-bit-rate communication," IEEE Communications Magazine, vol. 34, no. 12, pp. 42-45, Dec. 1996.

[7] ITU-T, Generic Coding of Moving Pictures and Associated Audio Information Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 138182 (MPEG 2 Video), 1994.

[8] ISO/IEC, Coding of Audio-Visual Objects— Part 2: Visual, ISO/IEC 144962 (MPEG-4 Visual version 1), 1999.

[9] ITU-T, Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), 2003.

[10] H. Schwarz, T. Marpe, and T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103–1120, 2007.

[11] Schwarz, Heiko, and Mathias Wien. "The scalable video coding extension of the H. 264/AVC standard [standards in a nutshell]." IEEE Signal Processing Magazine 25.2, 135-141, 2008.

[12] João Carreira, Error Resilience and Concealment Techniques for High Efficiency Video Coding, PhD Thesis, Loughborough University London, 2018.

[13] R. Sjoberg, Y. Chen, A. Fujibayashi, M. Hannuksela, J. Samuelsonn, T. Tan, Y-K Wang and S. Wenger, Overview of HEVC High Level Syntax and Reference Picture Management, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No 12, Dec. 2012.

[14] A. Fujibayashi and T. Tan, "Random access support for HEVC, document JCTVC-D234," JCT-VC, Daegu, Korea, Tech. Rep., Jan. 2011.

[15] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth and M. Zhou, "An Overview of Tiles in HEVC," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 969-977, Dec. 2013.

[16] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[17] Thomas Schierl, Miska M. Hannuksela, , Ye-Kui Wang, Stephan Wenger, System Layer Integration of High Efficiency Video Coding, IEEE Trans. on Circuits and Systems for Video technology, Vol. 22, No 12, Dec. 2012.

[18] Mei-Juan Chen, Ming-Chieh Chi, Ching-Ting Hsu, and Jeng-Wei Chen, "ROI video coding based on H.263+ with robust skin-color detection technique," IEEE International Conference on Consumer Electronics, ICCE, pp. 44–45, 2003.

[19] L.S. Karlsson and M. Sjostrom, "Region-of-interest 3D video coding based on depth images," in 3DTV Conference: The True Vision - Capture, Transmission and Display of

3D Video, pp. 141–144, 2008.

[20] T. Stockhammer, M.M. Hannuksela, and Stephan Wenger, “H.26L/JVT coding network abstraction layer and IP-based transport,” in Image Processing. 2002. Proceedings. 2002 International Conference on, 2002, vol. 2, pp. II–485–II–488.

[21] Sebastiaan Van Leuven, Kris Van Schevensteen, Tim Dams, and Peter Schelkens, “An implementation of multiple region-of-interest models in H.264/AVC,” in Signal Processing for Image Enhancement and Multimedia Processing, pp. 215–225. Springer, 2008.

[22] F. Peng, X. Zhu, and M. Long, “A ROI privacy protection scheme for H.264 video based on FMO and chaos,” Information Forensics and Security, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2013.

[23] M. Xu, X. Deng, S. Li and Z. Wang, "Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face," in IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 3, pp. 475-489, June 2014.

[24] Xing, P.; Tian, Y.; Huang, T.; Gao, W.: “Surveillance video coding with quadtree partition based ROI extraction, in Proc. of the IEEE Picture Coding Symposium (PCS), Dec. 2013, 157–160.

[25] H. Xue, Y. Zhang and Y. Wei, Fast ROI-based HEVC coding for surveillance videos, 2016 19th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 299-304, 2016.

[26] M. Meddeb, M. Cagnazzo and B. Pesquet-Popescu, "Region-of-interest based rate control scheme for high efficiency video coding," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 7338-7342, doi: 10.1109/ICASSP.2014.6855025.

[27] Z. Zhang, T. Jing, J. Han, Y. Xu and F. Zhang, "A New Rate Control Scheme For Video Coding Based On Region Of Interest," in IEEE Access, vol. 5, pp. 13677-13688, 2017.

[28] Chai, Z., Li, S., He, Q., Chen, M., & Chen, W. (2020). FPGA-Based ROI Encoding for HEVC Video Bitrate Reduction. Journal of Circuits, Systems and Computers, 29(11), 2050182.

[29] D. Grois, E. Kaminsky, and O. Hadar, “Dynamically adjustable and scalable ROI video coding,” in Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on, pp. 1–5, 2010.

[30] Jung-Hwan Lee and C. Yoo, “Scalable ROI algorithm for H.264/SVC-based video streaming,” Consumer Electronics, IEEE Transactions on, vol. 57, no. 2, pp. 882–887, 2011.

[31] Parker, Sarah, et al. "On transform coding tools under development for VP10." Applications of Digital Image Processing XXXIX. Vol. 9971. International Society for Optics and Photonics, 2016.

[32] Valin, J.-M.; Terriberry, T.B.; Egge, N.E.; Deade, T.; Cho, Y.; Montgomery, C.; Bebenita, M.: Daala: Building A Next-Generation Video Codec From Unconventional Technology, arXiv:1608.01947, Aug. 2016.

[33] A. Fuldseth, G. Bjontegaard, S. Midtskogen, T. Davies and M. Zanaty. Thor Video Codec Internet-Draft. Technical report, Cisco, 2016. Accessed: 04/2021. [Online]. Available: https://tools.ietf.org/html/draft-fuldseth-netvc-thor-03.

[34] AVS workgroup, “AVS3 video requirement V3.0”, AVS-Doc. N2747, Shenzhen 2019.12.

[35] Y. Chen et al., “An overview of coding tools in AV1: The first video codec from the alliance for open media,” APSIPA Trans. Signal Inf. Process., vol. 9, no. 6, pp. 1–15, 2020.

[36] P. de Rivaz and J. Haughton. AV1 Bitstream & Decoding Process Specification. Accessed: 04/2021. [Online]. Available: https://aomediacodec.github.io/av1-

spec/av1-spec.pdf.

[37] Wen Gao and Siwei Ma. “An overview of AVS2 standard. In Advanced Video Coding Systems”, pages 35–49. Springer, 2014.

[38] Versatile Video Coding, Recommendation ITU-T H.266 and ISO/IEC 23090-3 (VVC), ITU-T and ISO/IEC JTC 1, Jul. 2020.

[39] D. Mukherjee et al., “A technical overview of VP9—The latest open-source video codec,” SMPTE Motion Imag. J., vol. 124, no. 1, pp. 44–54, 2015.

[40] Alliance for Open Media. Accessed: 04/2021. [Online]. Available: https://aomedia.org/.

[41] ITU-T Recommendation H.265, International Standard ISO/IEC 23008-2, High Efficiency Video Coding, ed. 2.0, Oct. 2014.