

VIDEO COMPRESSION CODECS: A SURVIVAL GUIDE

Iain E. Richardson, Vcodex Ltd., UK

1. Introduction

Not another video codec!

Since the first commercially viable video codec formats appeared in the early 1990s, we have seen the emergence of a plethora of compressed digital video formats, from MPEG-1 and MPEG-2 to recent codecs such as HEVC and VP9. Each new format offers certain advantages over its predecessors. However, the increasing variety of codec formats poses many questions for anyone involved in collecting, archiving and delivering digital video content, such as:

- Which codec format (if any) is best?
- What is a suitable acquisition protocol for digital video?
- Is it possible to ensure that early 'born digital' material will still be playable in future decades?
- What are the advantages and disadvantages of converting (transcoding) older formats into newer standards?
- What is the best way to deliver video content to end-users?

In this article I explain how a video compression codec works and consider some of the practical concerns relating to choosing and controlling a codec. I discuss the motivations behind the continued development of new codec standards and suggest practical measures to help deal with the questions listed above.

2. Codecs and compression

2.1 What is a video codec?

'Codec' is a contraction of 'encoder and decoder'. A video encoder converts 'raw' or uncompressed digital video data into a compressed form which is suitable for storage or transmission. A video decoder extracts digital video data from a compressed file, converting it into a displayable, uncompressed form. It is worth noting that:

- The original and decoded video material may or may not be identical. If the output of the decoder is identical to the original video, the compression process is *lossless*. If the two videos are not identical, the compression process is *lossy* and it is (generally) not possible to recover the original data.
- There are many different codec formats which can provide widely varying amounts of compression.
- In general, higher compression can be achieved at the expense of reducing the quality of the decoded video.
- A video encoder and decoder must be compatible to work successfully, i.e. they must both conform to a common specification. The decoder needs to know the format in which the encoder compressed the video in order to successfully decompress it. This usually means that the encoder and decoder should use the same codec format.
- Newer codec formats such as HEVC and VP9 may require much more computational power than older formats such as MPEG-1. More processing power may translate into slower encoding and decoding.
- The amount of compression achieved can (and often does) vary dramatically between different codec formats and even between different versions or implementations of the same codec.
- A video codec may be implemented as a software application or it may be built into a device such as a smartphone, computer or camera.

2.2 Why compress video?

Figure 1 compares the resolution of popular video formats:

Format	Pixels per frame
Standard Definition (SD)	720x576 (PAL) or 720x486 (NTSC)
720p High Definition (HD)	1440x720
1080p High Definition (Full HD)	1920x1080
Ultra HD*	3840x2160

* Sometimes referred to as 4K, although the Digital Cinema Initiative 4K specification contains 4096x2160 pixels per frame.

Making certain assumptions about colour depth¹, one second of SD video recorded at 25 frames per second, requires 15.5 Mbytes of storage space. One second of Full HD video at 50 frames per second requires around 176 Mbytes of storage. This means that storing an hour-long Full HD video would require over 630 Gbytes of storage space. A key benefit of compression is that a compressed version of the same 1-hour file might require only a few Gbytes of space. The exact amount of space required depends on a number of factors including the content of the video and the chosen playback quality, as we shall see later. Compression of video can:

- Reduce the amount of storage space required per hour of video, making it possible to store more content in a certain storage capacity,
- Reduce the time taken to copy or move video files between locations by making the files smaller,
- Make it possible to access video material via the internet, by reducing the bitrate required to send or stream video.

Audio-visual material is increasingly captured in a compressed form. Unless your content is created using professional cameras and production tools, it is likely to be compressed during the recording process. For example, if you record video on a consumer device such as a digital camera, camcorder or smartphone, video is captured via the device's camera, encoded and stored on the device's internal storage in a compressed form. 'Born digital' audio-visual material is very often 'born compressed'.

¹ 4:2:0 sampling and 8 bits per sample, with PAL format for the SD example.

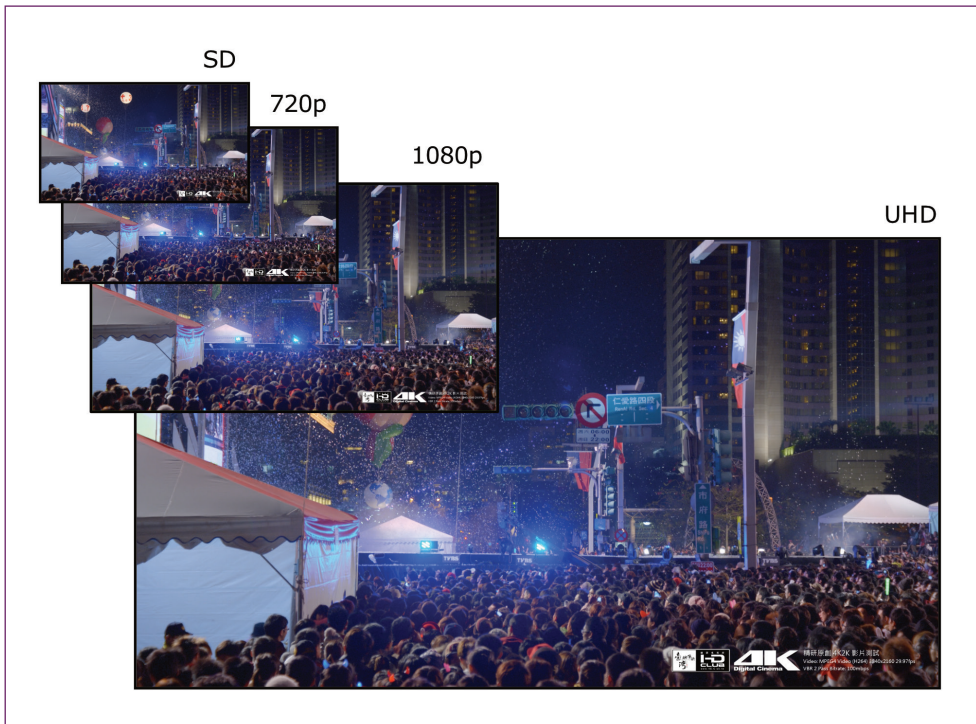


Figure 1 Video resolutions

2.3 How does a codec compress video?

A video codec can compress video material by exploiting two main factors:

1. The characteristics of a typical video scene, and
2. The way humans perceive visual material.

Most of the video material that we watch has certain predictable characteristics that can be used to help compress the video. Pixels or regions that are close to each other in space or time are likely to be correlated, i.e. similar. Within a single video frame, spatially neighbouring pixels are often the same or similar, particularly when they are all part of the same image feature or region. We can often find the same or very similar pixels in a video frame before or after the current frame, either (a) in the same place, if there is no movement from frame to frame, or (b) in a nearby location, if there is movement of the camera or the objects in the frame. A video encoder exploits these spatial and temporal similarities in several ways to compress video. For example, during prediction, each block of pixels in a frame is predicted from nearby pixels in the same frame or from pixels in another, previously processed frame.

When we look at a visual scene, we only take in or attend to a relatively small amount of information (Anderson, Charles, et al., 2005). Many factors are at work, including the sensitivity of the human visual system to detail and movement, our attention to and interest in what is actually in the scene and our innate response to unusual or unexpected details. A human observer is not capable of paying attention to every pixel in a high definition video display. A (lossy) video encoder exploits this by discarding much of the visual information in a scene, removing fine details and small variations that would typically not be noticed by the viewer.

2.4 What's inside a video encoder?

Most video compression encoders carry out the following steps to process and compress video, converting a series of video frames into a compressed bitstream or video file (Figure 3).

1. Partitioning: The encoder partitions the video sequence into units that are convenient for processing. A video clip is typically partitioned into -

- Groups of Pictures (GOPs) : Random access points, each including an independently-decodeable frame
- Frames :A complete video frame, sometimes described as a Picture (Figure 2)
- Slices or Tiles : Regions within a frame
- Macroblocks (or Coding Tree Units, CTUs) :The basic unit of processing, a square region of pixels, ranging in size from 16x16 up to 64x64 pixels depending on the codec format
- Block :A square or rectangular region within a macroblock or CTU.

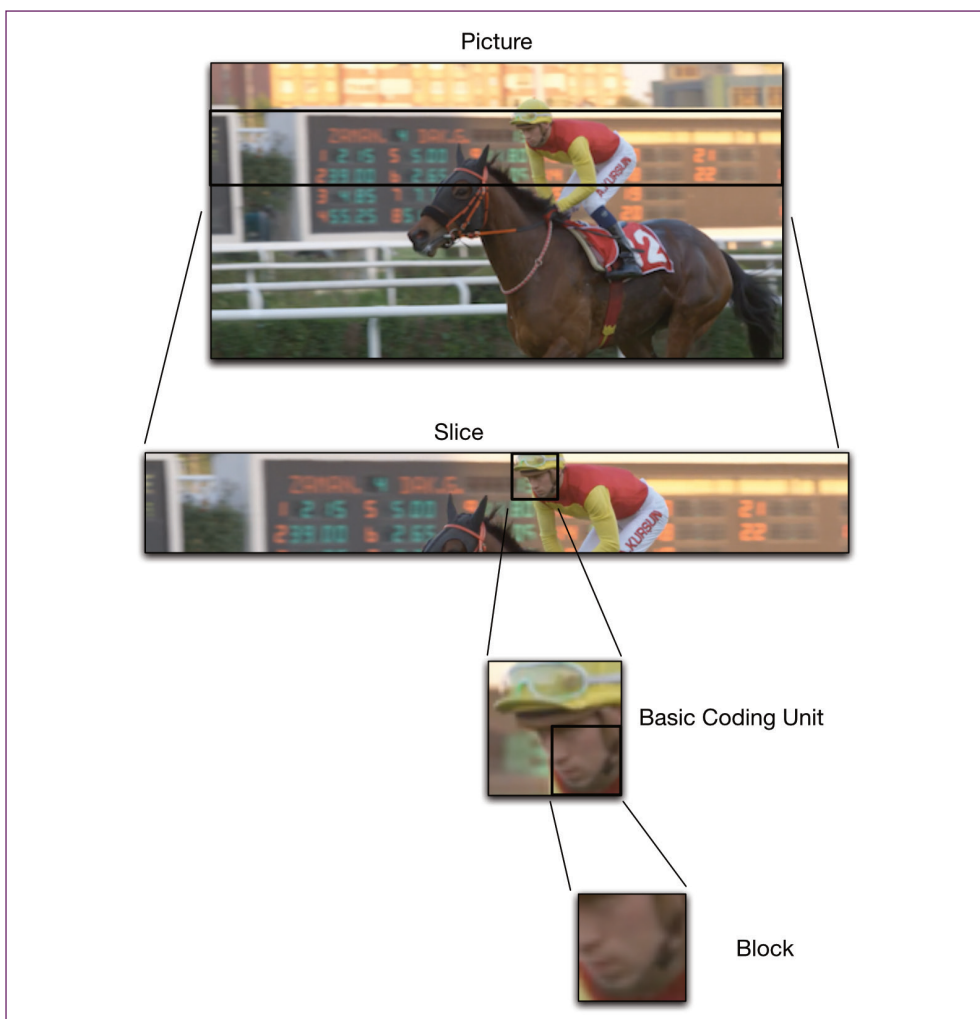


Figure 2 Partitions

Many of the partitions within a frame are square or rectangular, with dimensions that are powers of two (16, 32, 64 etc). These dimensions are (a) easy for electronic devices to process efficiently using digital logic and processors and (b) easy to indicate or signal in the encoded file.

2. Prediction: Each basic unit or block is predicted from previously-coded data such as neighbouring pixels in the same frame (intra prediction) or pixels in previously coded frames (inter prediction). A prediction block is created that is as close a match as possible to the original block. The prediction is *subtracted* from the actual block to create a difference or residual block.

3. Transform and quantize: Each residual block is transformed into a spatial frequency domain using a two-dimensional transform such as a Discrete Cosine Transform (DCT) or a Discrete Wavelet Transform. Instead of storing each sample or pixel in the block, the block is converted into a set of coefficients which correspond to low, medium and high frequencies in the block. For a typical block, most of the medium and high frequency coefficients are small or insignificant. Quantization removes all of the insignificant, small-valued coefficients in each block. The *quantization parameter (QP)* controls the amount of quantization, i.e. how much data is discarded.

4. Entropy encoding: The video sequence is now represented by a collection of values including quantized coefficients, prediction information, partitioning information and 'side' or header information. All of these values and parameters are entropy encoded, i.e. they are converted into a compressed binary bitstream. An entropy encoder such as a Huffman or Arithmetic encoder represents frequently-occurring values and parameters with very short binary codes and less-frequent values and parameters with longer codes.

The output of all these steps is a compressed bitstream - a series of binary digits (bits) that takes up much less space than the original video frames and is suitable for transmission or storage.

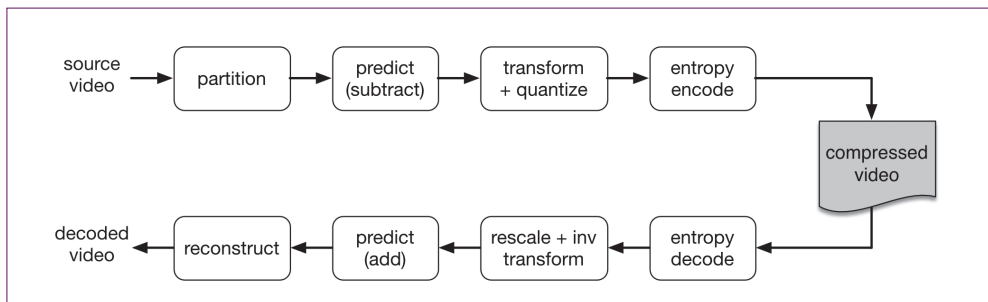


Figure 3 Encoding and decoding steps

Most video encoders in current use carry out the steps described above. However, there is considerable variation within each of the steps, depending on the codec standard and on the particular software or hardware implementation.

2.5 What's inside a video decoder?

A video decoder reverses the steps carried out by the encoder, converting a compressed bitstream into a displayable series of video frames. To decode a bitstream created as described in section 2.4 above, the decoder carries out the following steps:

1. Entropy decoding: The decoder processes the compressed bitstream and extracts all the values and parameters required to re-create the video clip.

2. Re-scaling and inverse transform: Quantized coefficients are scaled up to their original range and each block is transformed back into a set of image samples or pixel differences. It is important to note that in a lossy codec, the information that was removed by the quantizer cannot be restored, i.e. the output of this stage is not identical to the original difference block.

3. Prediction: The decoder creates the same prediction as the encoder, based on spatial or temporal values that have previously been decoded, and *adds* it to the decoded residual block to create an output block.

4. Reconstruction: Each video frame is processed block by block to reconstruct the video clip.

3. Video codec formats and standards

A codec standard makes it possible for encoders and decoders to communicate with each other successfully. Consider the example of a video that is recorded on a smartphone, encoded (compressed) and emailed to a PC where it is decoded and played back. The smartphone and the PC may be designed and manufactured by different companies. If the encoder and decoder conform to the same codec standard, we can ensure that the decoder will be able to successfully extract and play back the video clip, regardless of how the source and destination devices were designed.

3.1 What's in a standard?

A standard is a specification document created by a committee of technical experts. A video coding standard defines at least the following –

1. The format of a compressed video stream or file, i.e. exactly how each part of the coded file is represented.
2. A method of decoding the compressed file.

Typically, a video coding standard does *not* define an encoder (Figure 4). It is up to each manufacturer to decide how to design an encoder. The only requirement is that the bitstream produced by the encoder must be compliant with the standard, i.e. it has to conform to the format described by the standard and it must be decodable by the method defined in the standard.

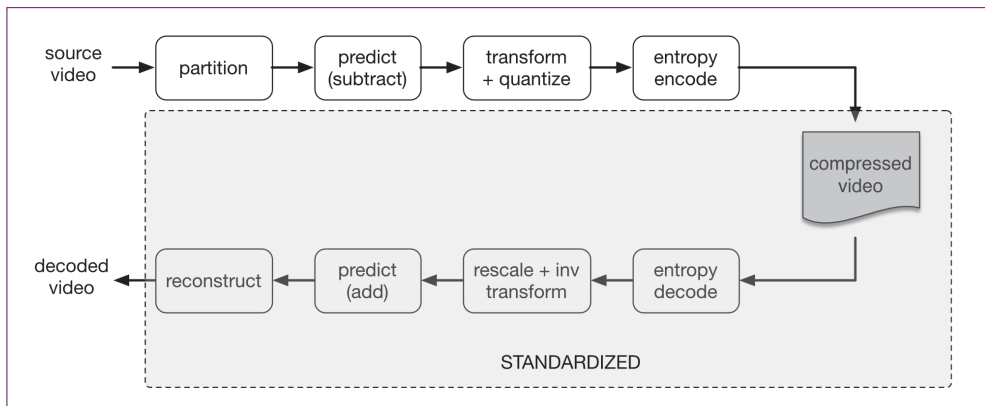


Figure 4 What a video coding standard covers

3.2 Why are there so many standards?

Since the first digital video coding standards were developed in the late 1980s/early 1990s, there have been a surprising number of standards released. Figure 5 shows some (but not all) of the key standards released over the last 25 years. Many were developed by working groups of the ISO/IEC and ITU-T international standards organisations. ISO/IEC standards include MPEG-2 (ISO/IEC 13818-2 and ITU-T Recommendation H.262, 1995), the standard used for the first digital TV services and for DVD Video. Some standards have been co-published by ISO/IEC and ITU-T, which is why H.264 (ITU-T Recommendation H.264, 2003) (for example) is also known as MPEG-4 Part 10. The most recent publication of the ISO/IEC and ITU-T is H.265 / HEVC, High Efficiency Video Coding, first released in 2013 (ITU-T Recommendation H.265, 2013).

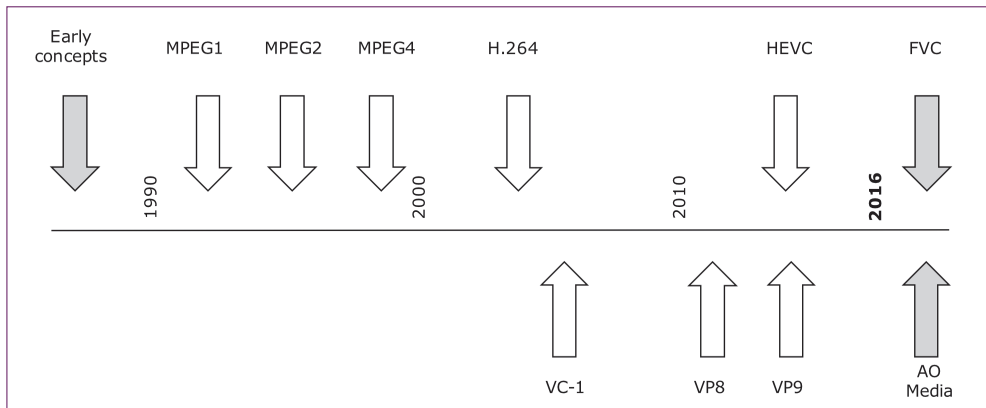



Figure 5 Timeline of selected video coding standards

The VP8 (IETF Request for Comments 6386, 2011) and VP9 formats were published by Google as part of the open source WebM initiative. At the time of writing (late 2016), new codec formats continue to be developed. The Joint Video Exploration Team (JVET) of ITU-T and ISO/IEC is considering new technologies as part of its Future Video Coding (FVC) exploration. The Alliance for Open Media (AOM), a consortium of companies including Google, Cisco, Microsoft, Mozilla, Netflix and others, is developing the AV1 codec format.

There are at least three factors behind the continued development and publication of new video codec formats and standards.

1. The demand for storing and transmitting increasingly high-resolution video content continues to rise. In the early days of broadcast and internet digital video, resolutions tended to be limited to Standard Definition or lower, and the volume of content was significantly lower.
2. This increase in high-resolution video content puts significant pressure on network and storage capacity, despite continuing increases in bandwidth. According to Cisco (Cisco Visual Networking Index, 2015), video data is increasingly dominating internet use and will make up over 80% of all consumer internet traffic by 2020.
3. Processing power continues to increase, so that it becomes feasible to carry out more complex processing of video, even on a mass-market device such as a smartphone.

Furthermore, new formats and usage scenarios are continuing to emerge. For example, 360 degree video involves an array of cameras that simultaneously capture video in all directions from a single central point. Playback on a conventional screen allows the viewer to move their viewpoint around to any angle, from within the scene. Free Viewpoint Video gives the viewer the freedom to observe a scene from the outside, selecting to view the scene from angle or



viewpoint. These new modes may have particular advantages for capturing events where a single, conventional viewpoint only records a small part of what is happening. These and other scenarios such as stereoscopic video, animation and screen sharing, may require new or modified standards.

Putting these factors together, there is a continued demand for better compression of video to support the increase in created, stored and transmitted video. Increasing processing power on consumer devices makes it possible to use new, more sophisticated video coding standards to meet this demand.

As new standards are released, manufacturers build support for new formats into devices such as tablets and smartphones. Typically, older standards such as MPEG-2 and H.264 continue to be supported, so that a newly-manufactured device may be capable of decoding video in multiple formats including MPEG-2, H.264, HEVC and VP9. In a similar way, software players such as VLC and web browsers increasingly support a range of codec standards.

4. Practical concerns

4.1 Quality, compression and computation

Coding video involves a trade-off between many different factors, including:

- **Quality and fidelity.** What is the resolution of the video image? How good is the quality, compared with the original captured image?
- **Compression.** How much space does the compressed file occupy? how many bits per second does it take to stream or transmit the coded file?
- **Computation.** How quickly can we compress video? Can it be processed in real time, or faster than real time? How expensive is the hardware for compressing video?

The *fidelity* of a video image depends on factors such as spatial resolution, frame rate and colour depth. Higher spatial resolutions such as 1080p and UHD can give the appearance of a sharper, more detailed video image. However, there is some debate as to whether a human observer can actually tell the difference between 1080p and UHD video at longer viewing distances (Le Callet and Barkowsky, 2014). As humans, our sensitivity to fine detail is limited and at a certain viewing distance from a screen, it is no longer possible to observe the extra details added by a UHD display. Increased frame rates (e.g. 50 or 60 frames per second) can represent fast and complex motion with better fidelity. Higher colour depths, in which each colour component of a pixel is represented using 10 or more bits instead of the widely-used 8 bits per colour component, may give a more vivid impression of colours and ranges of brightness, depending on the capability of the display.

The *quality* of a decoded video image depends on how it was compressed. Lossless coding involves retaining all of the visual information present in the original video sequence. However, the amount of compression is likely to be limited to 2-3 times. Lossy coding offers the potential for much higher compression ratios, often 100 times or more, at a cost of a reduction in visual quality. So-called 'visually lossless' compression may be a suitable compromise for some archival scenarios, in which the compression ratio of a lossy codec is kept deliberately low, perhaps reducing the file size by a factor of 20 times or more, whilst maintaining visual quality at a level that is indistinguishable to a human observer.

Compression determines the size of the encoded video file and the bitrate (number of bits per second) required to stream or transmit the file. Many factors affect the amount of compression achieved by a particular codec for a particular video clip. The encoder *quantization parameter* (QP) is often used to control the amount of compression and the quality of the decoded video clip. A higher QP tends to produce more compression but also lower video quality. A lower QP gives less compression but maintains a higher video quality.

The amount of *computation* required to compress a video file determines how long it will take to process the file. In general, more compression can be achieved at the expense of increased computation. Encoding a video sequence involves many choices and repeated computation steps, such as finding the best prediction for each block of a video frame. Video compression software often has different options such as 'fast', 'slow' or 'very slow' encoding presets. This makes it possible to choose whether to encode slowly and achieve better compression, or to compress a file more quickly at the expense of a lower compression ratio. Newer standards such as HEVC or VP9 typically require more computation than older standards such as MPEG-2 or H.264.

4.2 Files and containers

A coded video clip is typically stored in a *container file*, together with associated audio track(s) and side information such as metadata. Just as there are many video coding standards, there are a number of file format standards including:

- MPEG-2 Systems : Part of the MPEG-2 family of standards, file extensions include .MP2, .TS and .VOB
- MPEG-4 File Format : Part of the MPEG-4 family of standards, file extensions include .MP4, .M4V and .MOV
- Flash file format : A proprietary format with file extensions including .FLV, .SWF
- Matroska file format: An open-source format with file extension .MKV
- WebM file format: An open-source format with file extension .webm

In general, a container file will include:

- A header indicating the type of container, the type of coded video and audio within it, the number of tracks, etc
- Metadata
- One or more coded video streams
- One or more coded audio streams.

The audio and video streams are often interleaved, i.e. chunks of coded video and associated audio are interspersed within the file (Figure 6).

There is considerable flexibility in the choice of container format, video format and audio format. For example, an MP4 file can contain video formats such as MPEG-2, MPEG-4, H.264 or H.265 and audio formats including MP3, AAC and others.

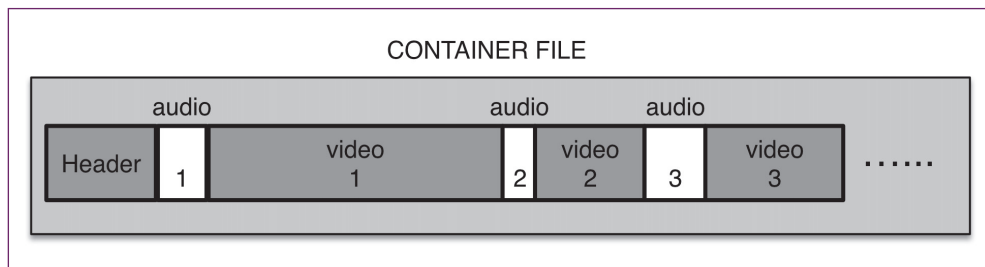


Figure 6 Container file

4.3 Transcoding

In a simple scenario (Figure 7), a video source is encoded into a compressed file (Format 1) and decoded in order to play back the video. However, often it may be necessary to convert from one compressed format into another (Figure 8). Here, the compressed file (Format 1) is converted into a new format, Format 2. Video and perhaps audio is decoded then re-encoded into the new format. This conversion process is known as *transcoding*. Transcoding may be necessary for a number of reasons, for example:

- To convert between resolutions and bitrates
- To convert video from multiple sources and formats into a common format for storage in an archive
- To convert from a high-quality archive source (e.g. visually lossless, high definition) into one or more lower-bitrate versions for streaming or delivery to end-users
- To upgrade from an older, legacy format such as MPEG-2 into a newer format such as H.265/HEVC.

It is important to be aware that each transcoding step - more specifically, each encoding step - can introduce quality loss into the video and/or audio content. If the encoding step is lossy, then degradation is introduced every time the content is re-coded. This may lead to generation loss (Figure 9), such that the visual quality of the material progressively degrades with each conversion.

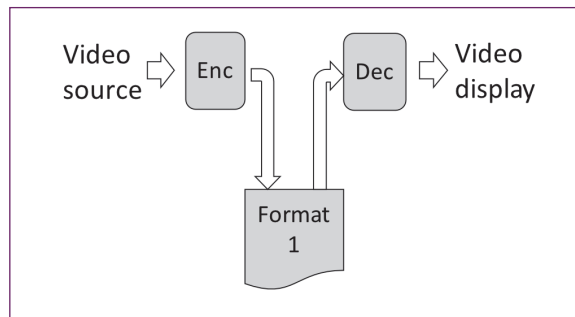


Figure 7 Single encode and decode

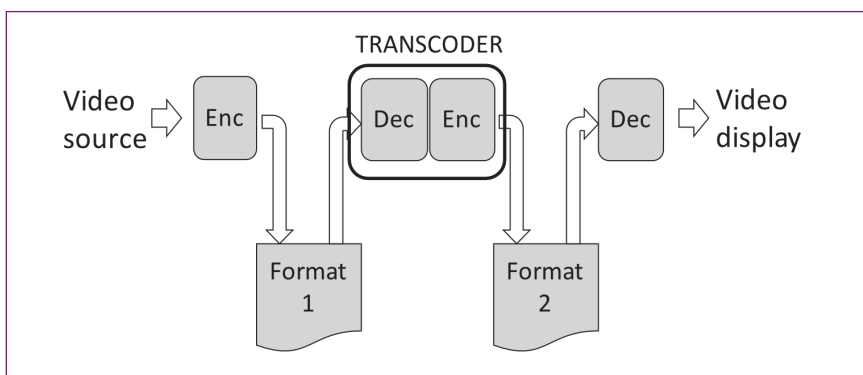


Figure 8 Transcoding

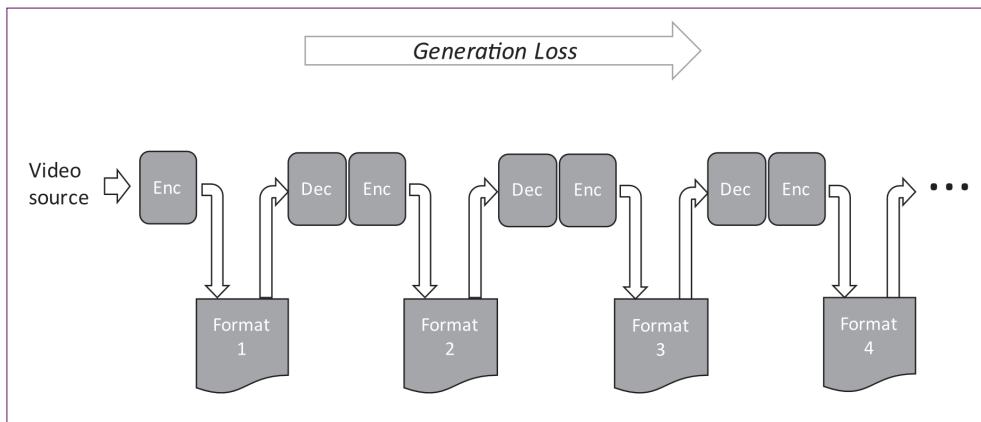


Figure 9 Generation loss

4.4 Storage requirements

How much space is required to store an hour of compressed video? The answer depends on many factors, including:

- Resolution and frame rate: Higher resolutions (HD, UHD) and higher frame rates (50 or 60 frames per second) will require more storage space than lower resolutions and frame rates.
- Choice of codec standard: In general, video compressed using newer standards and formats such as HEVC and VP9 takes up less space than video coded with older standards such as MPEG-2 or H.264. However, this depends on the next factor...
- Codec implementation: Not all codecs are created equal. For example, a recent study found significant variation between different implementations of the same coding standards (MSU Graphics and Media Lab, 2016). In some tests, a highly optimised software version of the older H.264 standard out-performed some implementations of the newer HEVC standard.
- Quantization / bit rate control: In a video encoder, the quantizer parameter QP acts as a control dial. A higher QP results in more compression and reduced quality; a lower QP gives less compression but higher quality. Setting the QP, or setting a target compressed size or target bitrate for the encoder, affects the size and also the quality of the compressed file.
- Video content: Some types of video sequence are harder to compress than others. For example, a clip with predictable motion such as a slowly panning camera is relatively easy to predict and therefore will tend to take up less space once it is compressed. A clip with complex motion, such as explosions or steam clouds, is much harder for the encoder to predict and will tend to take up more space after compression. Similarly, scenes with simple, smooth textures are easier to compress than scenes with complex detail.

With the correct choice of bitrate settings and/or quantization settings, it is generally possible to produce compressed files with either (a) a predictable file size or (b) a predictable visual quality, but not necessarily both at the same time.

4.5 Delivery

Acquiring, encoding and perhaps transcoding video material is one side of the story. The other side is providing access to the video content once it is stored. It may be sufficient to simply provide the encoded file to the intended viewer, for example by copying the file onto portable media or delivering it via file transfer. However, if the stored file is maintained at a high fidelity and therefore has a large size, it may be necessary to derive a version that is more suitable for transfer or streaming.

Proxy versions : The archived file may be transcoded to a lower-resolution and/or lower-quality proxy version for delivery to an end user. Reducing resolution and/or quality will make the compressed file smaller and can be a simple way of controlling or limiting access to full resolution versions.

Streaming : Container formats such as MP4 can be constructed to be 'streaming ready', such that the audio and video samples are interleaved (Figure 6). The file is streamed by transferring it in a sequence of packets, each containing one or more chunks of audio and/or video data. The receiver stores incoming packets in a buffer and once enough data is available (say, a few seconds of video), playback can commence. The well-known phenomenon of buffering occurs when the stream of packets does not arrive quickly enough to maintain constant decoding and playback.

Adaptive streaming : The buffering problem can be mitigated or avoided by using an adaptive streaming protocol such as DASH (Dynamic Adaptive Streaming over HTTP) (ISO/IEC 23009-1, 2014). A DASH server maintains multiple copies or representations of the video scene, each at a different bitrate (Figure 10). For example, the lowest-bitrate version might have a low spatial resolution (e.g. SD or lower) and may be encoded with a high QP so that the bitrate and the quality is low. Higher bitrate versions may have higher resolutions and/or lower QP settings. In a typical scenario, the receiver requests the lowest bitrate version first, so that playback can start quickly after a relatively small number of packets have been received. If packets are arriving quickly enough, the receiver requests a higher bitrate version and switches seamlessly to this version at certain switching points, e.g. every few seconds.

Example: The receiver of the stream shown in Figure 10 starts decoding and playback of the Medium Quality representation (Section 1). The first section is received before playback is completed and so the receiver switches to the High Quality representation (Section 2). The channel bitrate drops significantly and so the receiver requests to switch to the Low Quality representation (Section 3). The viewer experiences continuous video delivery, albeit with a reduction in quality if the network rate drops.

DASH and other adaptive streaming technologies require multiple transcoded versions of the video clip to be created, with each section stored in a container file so that the required switchover points are available.

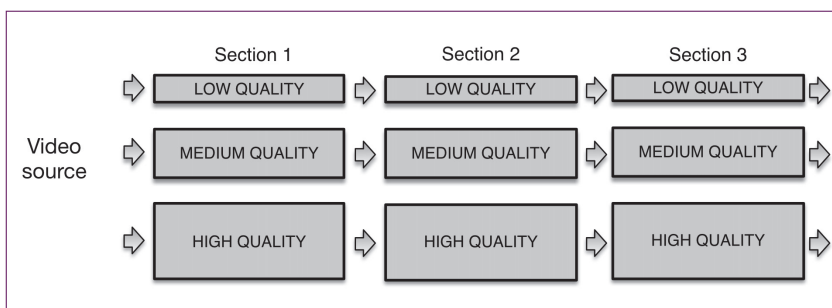


Figure 10 Adaptive streaming



5. Conclusions

Designing and specifying systems and protocols for archiving and retrieving coded video is a challenging task, as standards, electronic devices and user behaviours continue to change. Video resolution is increasing with each new generation of devices. For example, UHD resolution video recording is now supported by smartphones such as the iPhone 7 and Xperia Z5, and by an increasing range of consumer and professional cameras. As well as the challenges of higher resolutions and new codec formats, the way in which video is captured and disseminated continues to evolve.

Video footage of significant events is increasingly captured on a smartphone. With the rapid rise in user-generated video content, it is no longer possible to assume that content will be professionally captured in a well-lit environment. Content created on consumer devices such as smartphones and low-cost cameras is 'born' in an already compressed form.

Most video footage is still shot in the familiar format of a rectangular window. However, new ways of capturing video are beginning to emerge, such as 360 degree, stereoscopic and Free Viewpoint video, as discussed in Section 3.2. These and other departures from the traditional rectangular video scene offer particular challenges for coding, storing and delivering video.

Is it possible to future proof video archiving and delivery? The answer is probably not, since codec formats and usage patterns continue to evolve. However, the challenge of future proofing can be met at least partially by taking practical measures. During acquisition, it may be desirable specify an up-to-date codec that is likely to be supported for some time to come and a resolution such as 1080p that preserves visual information without taking up excessive storage space. Visually lossless rather than fully lossless compression may be an acceptable compromise between retaining important visual information and achieving reasonable compression. It is important to be aware that each transcoding or conversion process can introduce progressive degradation into audio-visual material. Finally, delivery or access to end users may be provided by deriving a reduced-quality, streamable version of stored content.

The rapid evolution of video capabilities, usage and formats in the last two decades implies that digital video technology will continue to change and develop for the foreseeable future. The only certainty is that further change is inevitable. However, by developing an understanding of the underlying principles and practical considerations of video compression coding, it is possible to specify and implement systems for the acquisition, storage and delivery of audio-visual media that can provide a good quality of service today and can adapt to the constantly changing landscape of digital video technology.

References

- Anderson, Charles, David Van Essen, and Bruno Olshausen. (2005). 'Directed visual attention and the dynamic control of information flow.' In *Neurobiology of Attention*, Elsevier.
- Cisco. (2015). Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf> (Date Accessed: 2016-12-28).
- IETF Request for Comments 6386. (2011). 'VP8 Data Format and Decoding Guide'.
- ISO/IEC 13818-2 and ITU-T Recommendation H.262, (1995). 'Generic coding of moving pictures and associated audio information:Video', (MPEG-2 Video).
- ISO/IEC 23009-1. (2014). 'Dynamic Adaptive Streaming over HTTP – Part 1'.
- ITU-T Recommendation H.264. (2003). 'Advanced video coding for generic audiovisual services'.
- ITU-T Recommendation H.265. (2013). 'High efficiency video coding'.
- Le Callet, Patrick and Marcus Barkowsky. (2014). 'On viewing distance and visual quality assessment in the age of Ultra High Definition TV', VQEG eLetter, Video Quality Expert Group, 2014, Best Practices for Training Sessions, 1 (1), pp.25-30.
- MSU Graphics and Media Lab. (2016). 'HEVC/H.265 Video Codecs Comparison', August 2016. http://compression.ru/video/codec_comparison/hevc_2016/ (Date Accessed: 2016-12-28).