THEO

THE ULTIMATE MINI GUIDE TO

VIDEO ENCODING

WWW.THEOPLAYER.COM

WHAT IS VIDEO ENCODING?

In simple terms, encoding is the process of **compressing** and **changing the format** of raw video content to a digital file or format, which will in turn make the video content compatible for different devices and platforms. The main goal of encoding is to compress the content to **take up less space**. We do this by getting rid of the information we don't need, also referred to as a **lossy process**.

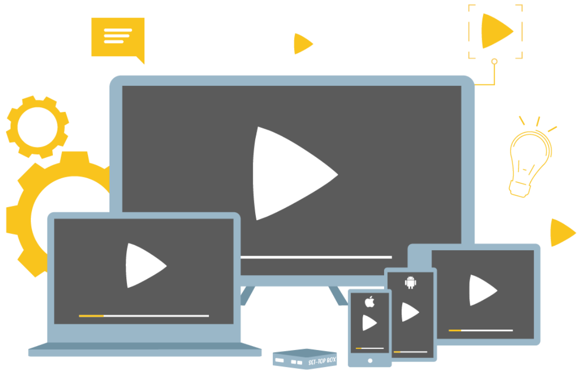
When the content is played back, it is played as an approximation of the original content. Of course, the more content information you get rid of, the worse the video that is played back will be, compared to the original. The process of video encoding is imposed by **codecs**, which we will discuss further in this Mini Guide.

WHY IS ENCODING IMPORTANT?

Video encoding is important because it allows us to more easily transmit video content over the internet. In video streaming, encoding is crucial because the compressing of the raw video **reduces the bandwidth** making it easier to transmit, while still maintaining a **good quality of experience** for end viewers.

If all the video content was not compressed, available bandwidth on the Internet would be **inadequate** to transmit all of it, and prevent us from deploying widespread, distributed video playback services.

The fact that we can stream video on **multiple devices** in our homes, **on-the-go using mobile**, or even while **video chatting** with loved ones across the globe, even with low bandwidth, is owed to **video encoding**.

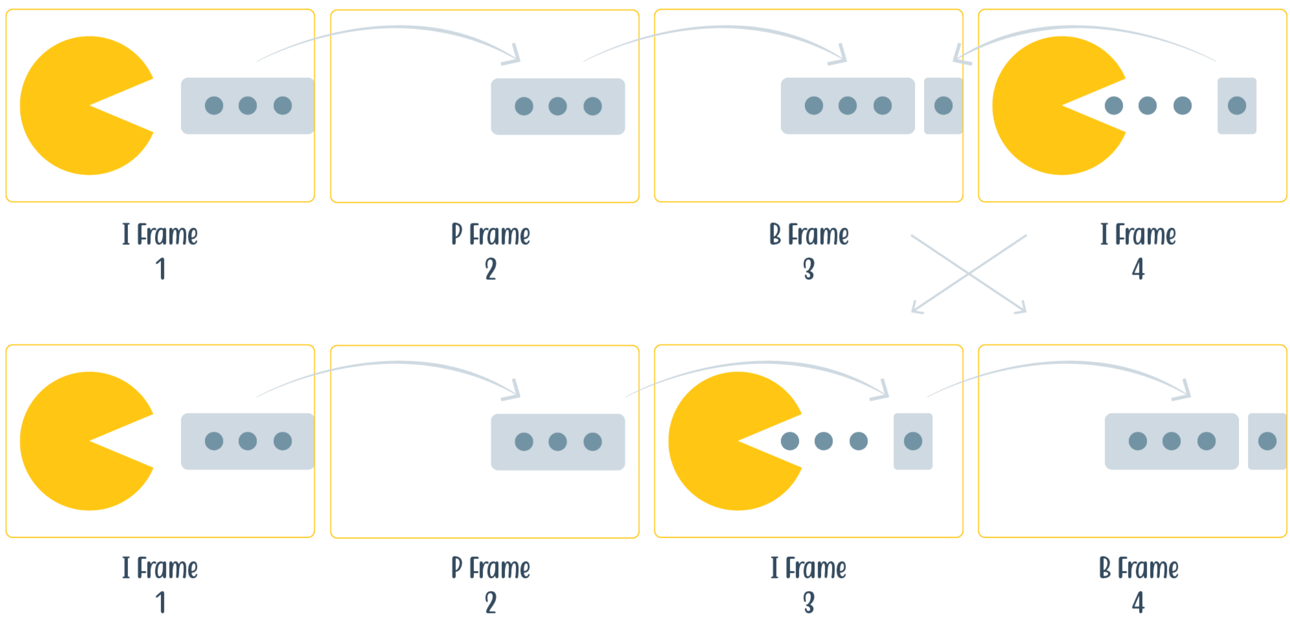


MOTION COMPENSATION

In video encoding, motion is very important. We most often express this in **I frame** (or keyframe), **P frame** and **B frame**. The keyframe stores the entire image.

In the next frame or the P frame, when noticing not much has moved or changed, the P frame can **refer to the previous keyframe**, as only some pixels have moved. **I, P and B frames form groups of pictures (GOPs)** together, and frames in such a group **can only refer to each other**, not frames outside of it.

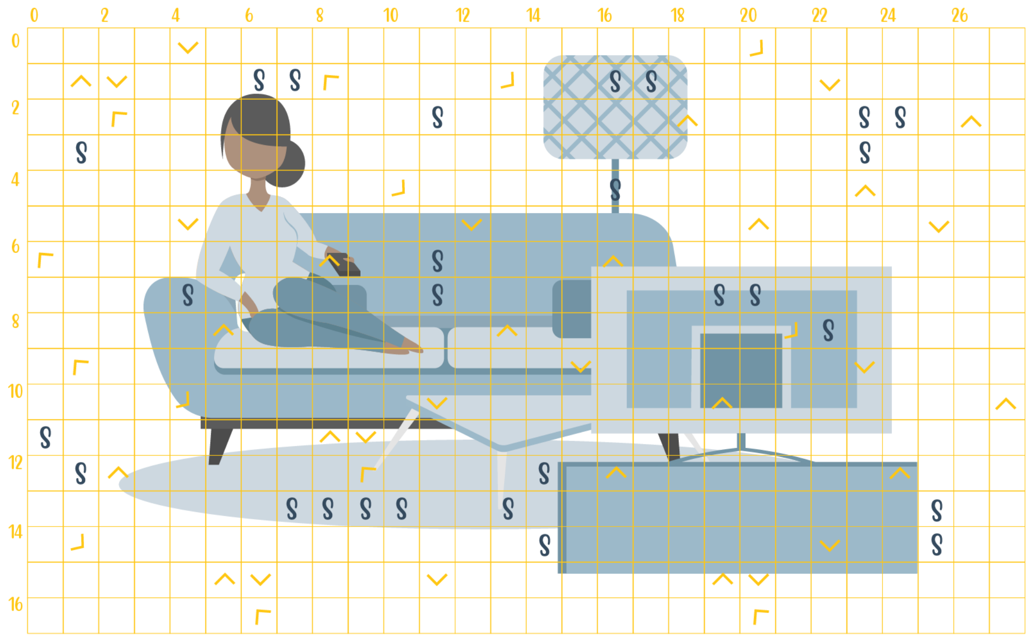
For frames with not a lot of movement, this can save about **90% of the data storage** that you would need to store a regular image (if it would use a lossless data compression like in PNG files), which are very big.



Predictive Frames

MACRO BLOCKS

Within each frame, there are **macroblocks**. Each block has a **specific size**, **colour** and **movement information**. These blocks are encoded somewhat separately, which leads to **proper parallelization**. Previously, codecs such as H.264 only allowed block sizes of 4x4 samples or 8x8 samples, however newer codecs have now allowed for **more block sizes** as well as r**ectangular form**. Large blocks are used when there isn't a lot of detail needed in the block, and only using a large block **saves a lot of space**, rather than just having many small blocks.



Macroblocks consist of multiple components. There are **sub-blocks** whose purpose is to **give pixels colour information**. There are also sub-blocks that **give the vector for motion compensation** compared to the previous frame.

Due to this macroblock structure, **in low bandwidth situations** there can be sharp edges or "**blockiness**" visible within the video content. There are ways around this by adding a filter that smooths out these edges. The filter is called an **"in-loop" filter**, and is used in the encoding and decoding processes to ensure the video content stays close to the source material.

COLOR IN VIDEO

In most cases, we divide a color in RGB channels, however, the human eye detects changes in brightness much more quickly than changes in color, especially in moving images. **Therefore, in video, we use a different color space called YCbCr.** This colour space divides into:

- Y (luminance or luma)
- Cb (blue-difference chroma)
- Cr (red-difference chroma)
- Luma = brightness, Chroma = color

CHROMA SUBSAMPLING

In chroma subsampling, we split images in their Y channel and their CbCr channel. For example, from an image we take a grid of two rows with four pixels each (4x2). In the subsampling we define a ratio as **j:a:b**.

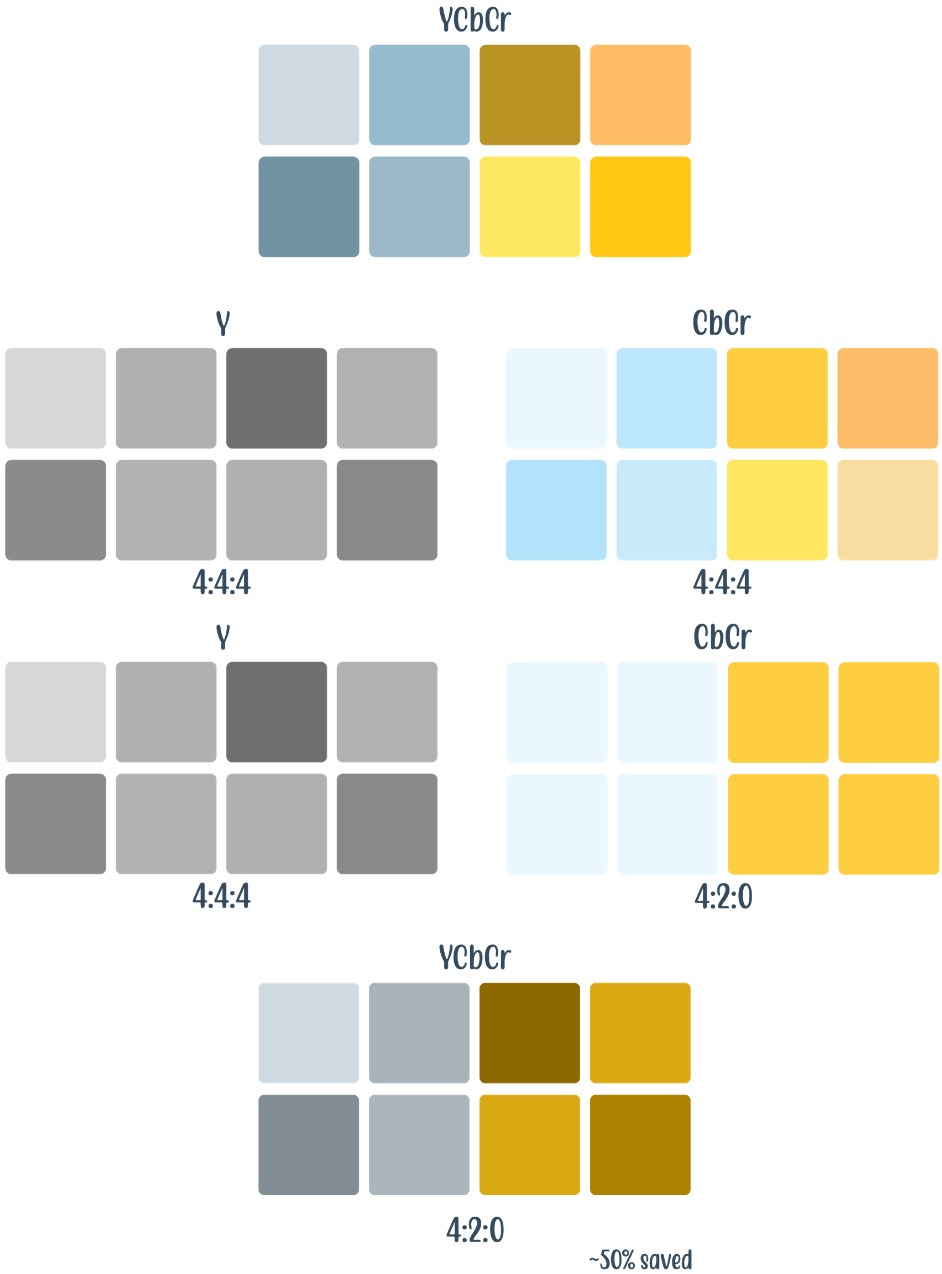
j: amount of pixels sampled (i.e. width of grid)

a: amount of colors (CbCr) in the first row

b: amount of color (CbCr) changes in the second row, in comparison to the first row.

CHROMA SUBSAMPLING IN STREAMING

In streaming video, **4:4:4 is the full colour space**. Reducing it to **4:2:2 saves 30% of space**, and reducing it further to **4:2:0 saves 50% of space**. In video streaming (think your Netflix and Hulu TV shows), the most commonly and widely used is **4:2:0**. In the video editing space however, 4:4:4 is the most common.

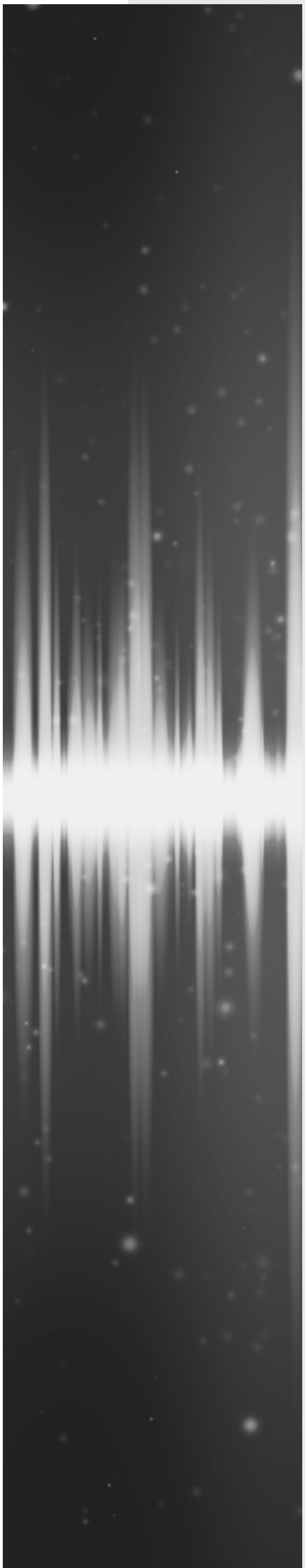


QUANTIZATION

When discussing the encoding of video, we refer to more than just saving space with the image components, but also with the **audio components**. **Audio is a continuous analog signal**, but for encoding, we need to **digitise** this.

Once the audio is digitised, we split it up into **multiple sinusoids**, or **sinus waves**, each of which represents an **audio frequency**. To save space, we can **discard frequencies that we do not need**.

If we take an image, we can also see rows of pixels behind one another as one large signal. Just like audio we remove frequencies in the image, known as **frequency domain masking**. Removing frequencies does lead to a loss of detail, but you can remove a fair amount of frequencies without it being noticeable to the end viewer. This process is known as **quantization**.



WHAT ARE CODECS?

Codecs are essentially standards of video content compression.

Codecs are made up of two components, an encoder to compress the content, and a decoder to decompress the video content and play an approximation of the original content. An enCOder and a DECoder, hence the name codec.

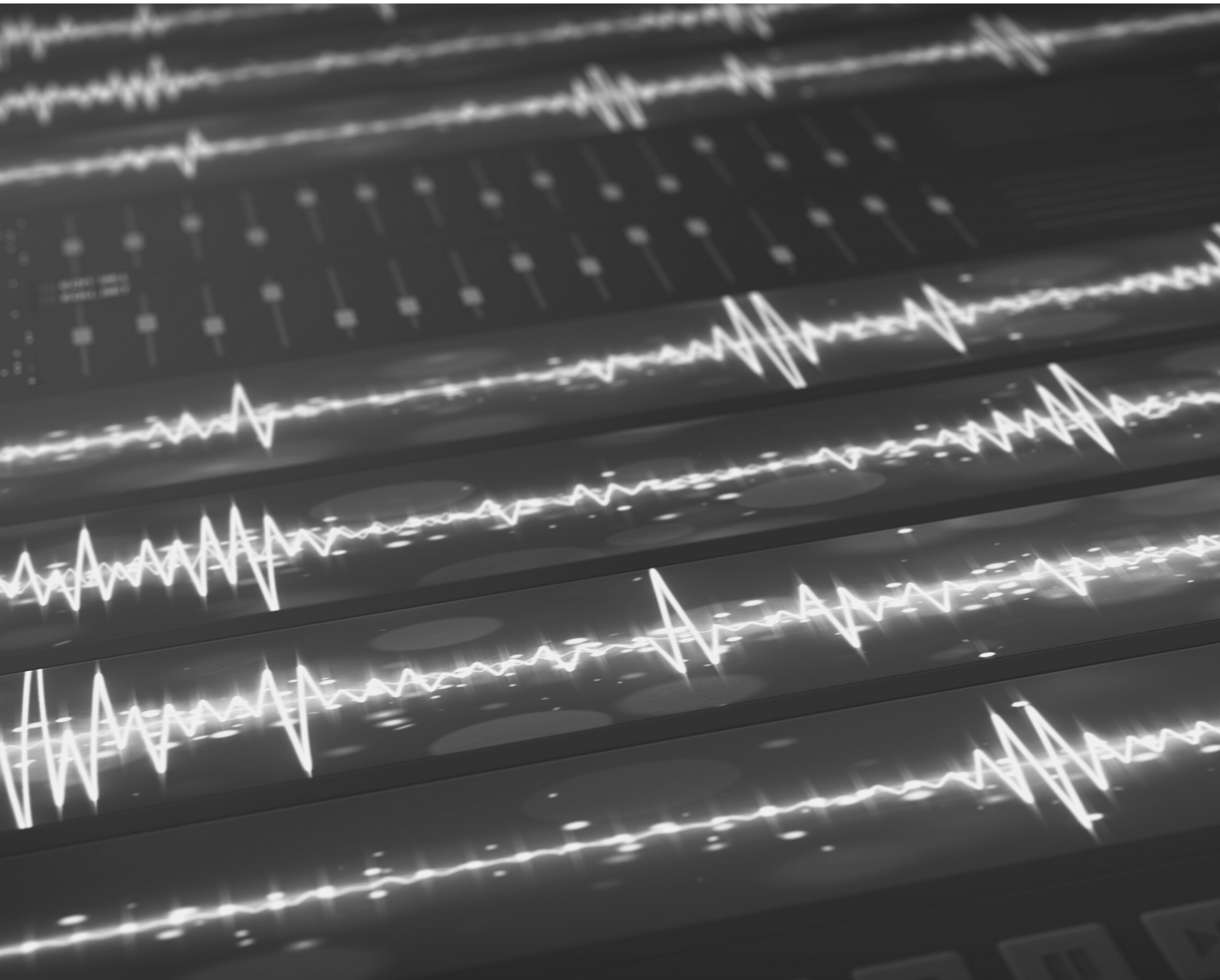
This Mini Guide will delve into the codecs you should know about, including:

- Audio Codecs
- Video Codecs
- Future of Codecs
- Hybrid Codecs

AUDIO CODECS

For audio, **AAC** is seen as the **de-facto standard** in the industry. AAC is essentially supported everywhere, and has the largest market share. Other audio codecs include: **Opus**, **Flac** and **Dolby Audio**.

Opus excels in voice and is also used by YouTube, seemingly the only large service using it, but still it falls back to AAC. Dolby Audio, also known as AC3 is sometimes still used in instances for surround sound, as some older surround sound systems don't play AAC.



VIDEO CODECS

H.264/AVC

H.264, also called AVC (**Advanced Video Coding**) or MPEG-4 AVC, was standardised in 2003. The codec was developed by MPEG and ITU-T VCEG, under a partnership known as JVT (Joint Video Team). It is **supported virtually everywhere**, **on any device**, while still providing a **quality video stream**, and is seen as a **baseline** for newer codecs. It also is relatively easy concerning royalty fees.

H.265/HEV

H.265, also known as HEVC, which stands for **High Efficiency Video Encoding**, is a standard by MPEG and ITU-T VCEG (under a partnership known as JCT-VC). This codec was first standardized in 2013, and was eventually expanded on from 2014 through 2016. The goal with H.265 is to improve the content compression by **50% compared to H.264**, all while keeping the same quality. A Netflix study showed improvements that ranged from **35-53%** when comparing it to **H.264**, and improvements of **17-21%** when comparing it to **VP9**.

***keep in mind that the encoder, content, etc., matters a lot for these kinds of comparisons.*

These improvements were achieved by optimising the techniques that already existed in H.264. Essentially, H.265 compresses the content into **even smaller files** than possible in H.264, and in return **decreases the required bandwidth** that would be needed to play the video content. Although this is all great news, H.265 is still very rarely used. Why? The main issue is the uncertainty around **licensing** and **royalties**.

VP9

VP9, the successor of VP8, was developed by On2Technologies, which is owned by Google. VP9 was standardised in 2013. This codec is similar to HEVC, however **no royalties are required**. Where those in the industry run into difficulties with VP9 is that, while it is widely supported on all major browsers and android devices, it is **not supported by Apple or any of Apple's devices**. They instead support H.264 and H.265.

AV1

In 2015, the Alliance for Open Media (AOMedia) was formed. During this time Google was working on VP10, Mozilla (Xiph) was working on Daala and Cisco was working on Thor. Instead of creating three separate codecs and frustrated by the limitations of royalties, they decided to join, therefore AV1 was created. Together, their goal was to attain 30% more efficiency than shown previously in VP9, but just like VP9, to also remain royalty free. All AOMedia members offered up their related patents to contribute to a patent defense programme.

While the AV1 codec is finalised, there is still work being done, but it seems the codec is starting to be adopted by big industry players and will continue to be in the future.

THE FUTURE OF CODECS

The current state of codecs seems relatively simple: **AAC for audio**, and **H.264 are necessary for a broader reach**. **H.264 is a fallback** for older devices, **VP9 has no iOS support** and **AV1 still has a way to go** before it can be put into regular production and usage.

So in the cases where just one codec won't suffice, what is the solution?

The **multi-codec approach** is a must in these situations. For example, **Twitch**, today's most popular video live streaming service, currently uses **H.264 and VP9**, but are hoping to **switch to AV1** in the future.

Another example is **Netflix**, who have just recently announced that they have started streaming **AV1 on their Android App**, which boasts a **20% increased compression efficiency** over VP9. However, it's only enabled in places where CPU power is way cheaper than bandwidth, for example when a viewer is streaming over 4G. Netflix is **aiming to use AV1 for all platforms** in the future.

VVC

Versatile Video Coding, or **VVC** is a new standard by MPEG. The goal of VVC is to achieve **50% efficiency over HEVC**, along with other improvements. However, in reality, most as **expecting** to see a **30-40% efficiency over HEVC**. VVC will provide better support for **gaming and screen sharing**, **360° video** and **resolution switching** in video.

EVC

Essential Video Coding, or **EVC** is another new standard by MPEG, with the goal to **achieve the same or similar efficiency as that of HEVC**. EVC will be suitable for **realtime encoding** and **decoding**. This standard is made **specifically for offline encoding for streaming VOD and live OTT streaming,** and aims to be a "**licensing-friendly**" **codec**.

HYBRID CODECS

A hybrid codec is essentially a codec which works on top of another codec. The process usually follows these steps:

- Take an input video
- Use proprietary downscaler on the video (e.g. go from 1080p to 480p)
- Encode downscaled video
- Decode the video on the player
- Upscale the decoded image using proprietary upscaler

This can often result in a 20-40% bitrate reduction.

P+ (PREVIOUSLY KNOWN AS PERSEUS PLUS)

P+, created by V-Nova, is being actively used today. For example, the use of this hybrid codec can give a **30-50% bandwidth reduction while using it on top of the H.264 codec**.

Perseus has the advantage of saving in the scale of HEVC without having to redo the whole encoding pipeline, and it also has hardware decoding on **iOS** and **Android**.

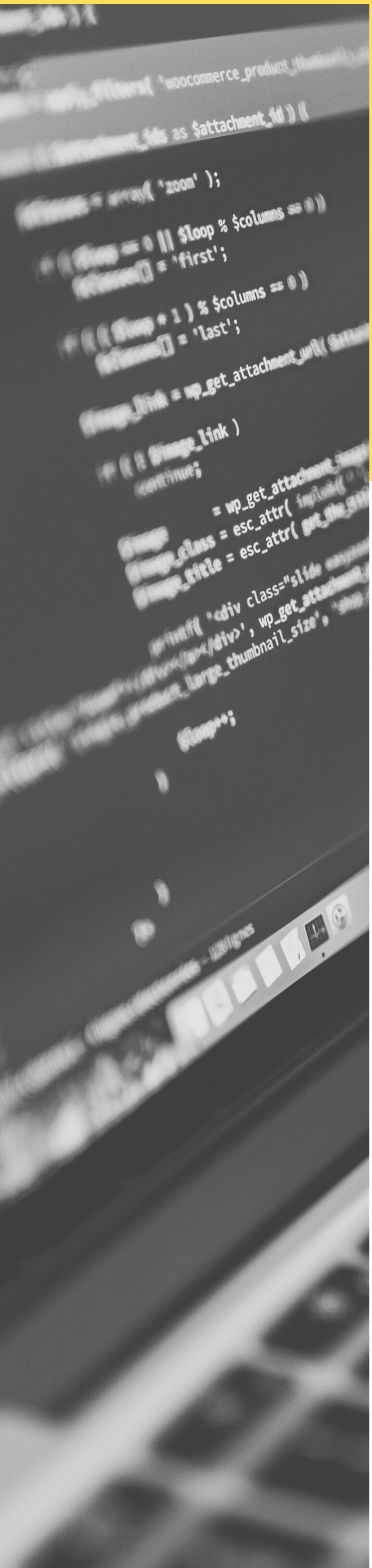
With that said, using P+ to decode on the web can be challenging, drains the battery, and does not currently work with DRM, but that will change.

This hybrid codec is currently being rebranded as **LCEVC**, which should work with DRM once integrated into hardware.

ENHANCEPLAYER

ENHANCEplayer is a joint and still ongoing innovation project between **Artomatix** and **THEO Technologies**. The hybrid codec utilises Artomatix's 'Artomatix Enhance AI' to **upscale** the **image resolution**, **remove compression artifacts** and **remove** the **noise**. This essentially means "**superresolution**" generation with a neural network (NN) or an AI.

The NN is trained to **enhance** the image and **add details which were lost during compression**. The goal is to reach **40ms** to get a framerate of **25fps**. The biggest challenges are with fidelity, meaning the image before compression is the same after enhancement, and it is still difficult to do while DRM is in play.



INTERESTED IN FINDING OUT MORE ABOUT VIDEO ENCODING?

Get in contact with one of our THEO experts.

Pieter-Jan Speelmans

Willem De Saegher

Ask an expert

THEO Technologies NV
www.theoplayer.com
contact@theoplayer.com