The VP9 Codec: How Does It Stack Up?

Written by:
James Konik

June 23rd, 2021

Uncertain if he's a coder who writes or a writer who codes, James tries to funnel as much of this existential tension as possible into both of his passions but finds it of more benefit to his writing than his software. When he occasionally hops out from behind his keyboard, you can find him jogging and cycling around suburban Japan.

If you work with video, picking a codec is a critical decision that will affect both your workflows and the quality of your output. There are many different video codecs available, and as they all offer different things, there's no clear right choice. As devices and platforms evolve, the sands are ever-shifting. Today, I'll go into detail on the VP9 codec and see what it offers versus the competition. I'll talk about its history, then look at its pros and cons, so you can make a decision on whether VP9 is a good fit for your video projects.

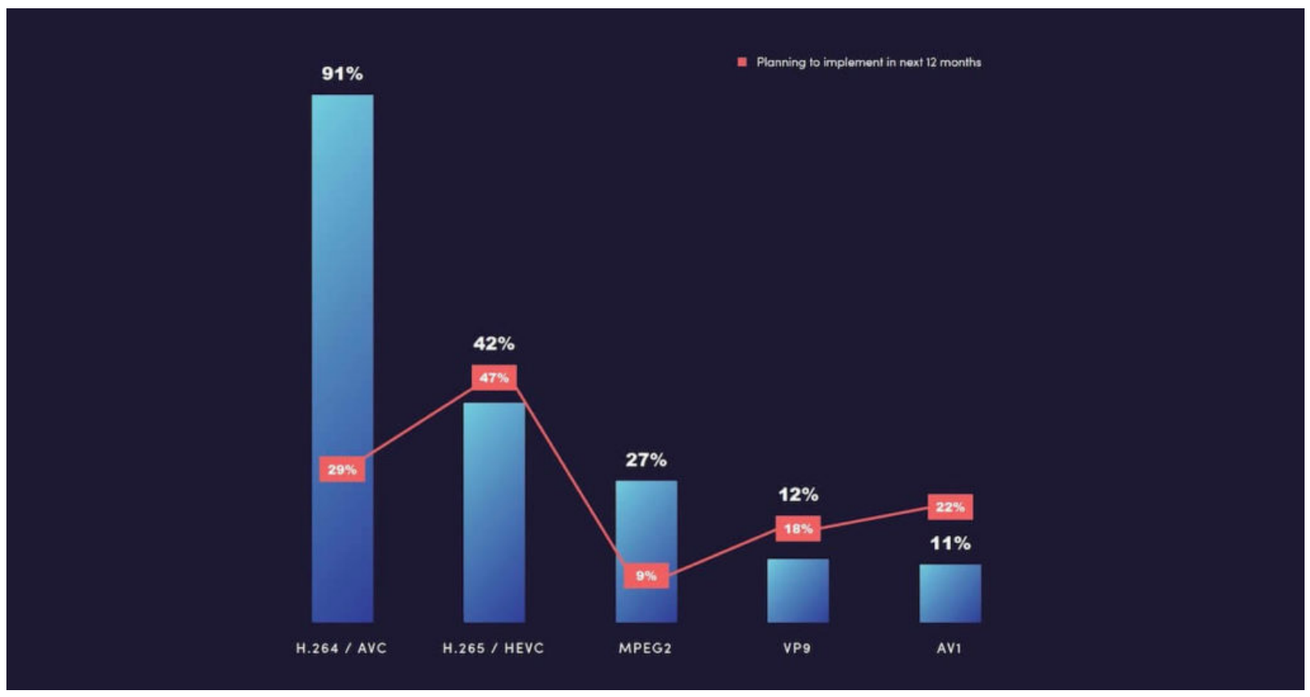
What Are Codecs, and Why Are They Important in OTT Video?

Codecs are responsible for compressing and decompressing video when it is created and played back. Your choice of codec will affect your compression ratio, encoding, and playback speed. It will also affect your videos' quality and which devices and players are able to view them. It can affect their maximum resolution and features, too. In OTT video these are critical, as they directly affect your server and bandwidth costs. They also define your potential customers. Not every device can view every codec, and quality can vary even on supported devices.

You want to deliver your video to as many people as possible, with the best quality, at the lowest cost. Since there isn't a dream codec that's best in class for all of these, you need to compromise on at least one. To figure out which, you need the details.

What Codecs are Popular Today and why?

Things move quickly in the world of video, and what's popular today may not be so for long. The current state of play is as follows. According to Bitmovin's Video Developer Report, H.264 (AVC) is the world's most popular codec with over an estimated 90% of video using it. Most modern devices have hardware that can decode AVC, making it easy to play, making it a great choice if you want the broadest possible user base.



This codec's licensing system is complicated, though, with thousands of patents involved. That means you need to pay royalties if you want to use it for hardware or paid content. Despite this, it is used by many services, as well as Blu-ray discs, and was the codec of choice for many early HDTV broadcasts.

Its successor, H.265 (or HEVC), is also common, though less popular, and uses up to 50% less bandwidth than H.264. However, much like its predecessor, H.264, HEVC can be expensive, with patents and fees required to use it.

The VP9 Codec

Widely supported and open, the VP9 codec has several key advantages over the competition, with a few caveats.

History of VP9

Google purchased On2, creators of VP8 video codec, for $124.6 million in 2010. They began work on VP9 soon afterward, eventually releasing it in late 2012. Support for it spread quickly, and by 2017 it was used by Chrome, Firefox, and Edge. It had hardware support from NVidia, ARM, and others. It is widely supported on Android, but iOS has added support only in its most recent version.

The base software used to encode VP8 and VP9 is called libvpx (it's also possible to encode VP8 and VP9 using vpxenc.exe). Fortunately, it's free, easy to use, well documented, and can handle real-time encoding should you need it. Refinements to the libvpx library used to encode videos have improved VP9 over time, with a 50-70% speed boost, multithreading, and support for extra bit depths and chroma subsampling modes among the additions.

How Is VP9 Different?

There are several major differences between VP9 and its rival video codecs. Take a look at the pros and cons below.

Pros of VP9 vs. Other Codecs

Like HEVC, VP9 has a 50% bitrate advantage over H.264, but VP9 outperforms HEVC at resolutions above HD. That makes it a great choice if you're working with video at different sizes, or if you're focusing specifically on HD. VP9 can use hardware decoding on Intel and ARM devices. Thanks to widespread hardware support, it gives you smoother playback on PCs in most scenarios. That makes it ideal for general use. It's also the best codec choice for high-quality YouTube videos. If you want to watch 4K on YouTube, you need VP9. Now even Safari has added support for the codec, making it the clear choice for high-resolution video in the macOS version of the browser.

Android has supported VP9 since 4.4. "KitKat" launched in 2013, so all but the most ancient devices can use it. VP9 is open, making it free and convenient to deploy. By contrast, HEVC is mired in patent issues, making it expensive—and potentially troublesome—to use. It costs $0.20 per device to include HEVC support, which discourages adoption and may well give VP9 a longer tail for hardware manufacturers, even as new codecs eventually take over.

Early comparisons suggested VP9 had a bitrate disadvantage over its competitors, but Netflix claimed that this could be eliminated or reversed by tuning the settings. With settings making a big difference to the quality, not to mention costs, you should be prepared to experiment. Companies are likely to keep most of their findings to themselves, but information on the relative performance in different scenarios is available.

Cons of VP9 vs. Other Codecs

Not all players work well with VP9. Older devices that don't support hardware decoding are still in use. If you're a developer, your friends' phones may not be representative of the wider population. Until recently, iOS devices haven't supported VP9, but that's changed with the recent update to version 14. They have supported H.265 since iOS 11. The older codec, H.264, runs smoothly on practically all devices, so if you want a single codec to run as close to everywhere as possible—and can budget for the monetary costs—that should be your preference.

There are some issues with 360 videos and VP9. While YouTube works well with VP9 360-degree video, its metadata injector doesn't. Other publishers may have similar issues. It is important to confirm that your target platform has support for the specific type of video you want to deploy, so testing before making a commitment is recommended. VP9 is also slow to encode. The more videos you have to deal with, the more of an issue that will be.

Internet Explorer (now Edge) doesn't currently support VP9. That may not be a problem for you personally, but it's possible that businesses running legacy software might be stuck with it, so do be careful to check what your clients use at work. An excuse to upgrade to a better browser may even be

viewed as a positive. AV1 is already en route to replacing VP9. This codec builds on work done towards VP10 and could potentially halve video file sizes. So you don't want to put all your eggs in one basket.

Businesses Currently Using VP9

VP9 can be found in many places. Google uses it on YouTube and Google Stadia, as mentioned earlier. It is also used for videos on WikiMedia Commons. Netflix uses it, though not exclusively, giving it a 35% bandwidth saving at equivalent quality.

Use Cases/Business Situations Where VP9 Might Be a Good Choice of Video Codec

VP9 works smoothly on PCs at high resolutions, so if you're delivering OTT video to desktop, it's a great choice. It can save you bandwidth over other codecs while delivering the same or better quality. If you need control over your codec or even want to make a few customizations, then open source has obvious advantages over the competition. The lack of fees can add up to significant savings if you are delivering paid content. The better support for HD+ and 360-degree videos also make it a great choice if those are your specialties. Though it isn't yet widely supported on iOS, you can still use it alongside other codecs if you want a cross-platform strategy, and enjoy the advantages for those platforms that support it fully.

The Future of VP9

If H.264's longevity is any indication, VP9 will be around for some time. In the near future, codecs such as VVC (released in 2020) and AV1 (released in 2018) promise improvements. It will take time for them to spread, but all developers need to keep an eye on the horizon. VVC, the successor to H.265, promises a 50% bitrate improvement. AV1, building on VP10, is promising a 30% compression rate improvement over H.265. Both of these are worth getting excited about, but of course, browser and device support will be crucial when making a decision. If the pattern holds, neither of these relatively new video codecs are likely to completely replace VP9 any time soon.

Compiling a Video Using FFMpeg and Libvpx

It's very simple to use VP9. You can download FFmpeg here and start encoding right away. To encode a video in VP9 in the command line, just type:

```
ffmpeg -i input.mp4 -c:v libvpx-vp9 -c:a libopus output.webm
```

You can control the quality using a flag as in the following example. The available settings are Good, Realtime, or Best. Best is extremely slow, and as you can see in the following table, the gains are marginal. To get a better idea of relative encoding speeds, I did some testing. The results are in the table below. I used an i5-7600 based system using a small 3 MB video. All times are in seconds.

Quality	Time
Good	67.34
Best	398.623
Realtime	96.037
Threads	**Time**
1	166.615
2	97.764
4	66.669

VP9 Codec Encoding Speeds by Quality

```
ffmpeg -i input.mp4 -quality good -c:v libvpx-vp9 -c:a libopus output.webm
```

You can also specify the number of threads used, though by default all threads in my test system's quad-core processor were taken advantage of. Specifying the thread count is likely better if you want to reserve threads for other things. You can do that as follow:

```
ffmpeg -i input.mp4 -threads 4 -c:v libvpx-vp9 -c:a libopus output.webm
```

You can also use the benchmark flag if you want to confirm how long encoding is taking.

```
ffmpeg -benchmark -i input.mp4 -quality good -c:v libvpx-vp9 -c:a libopus output.webm
```

```
libssh --enable-libzmq --enable-avisynth --enable-sdl2 --enable-libwebp --enable-libx264 --enable-libx265 --enable-libx
id --enable-libaom --enable-libopenjpeg --enable-libvpx --enable-libass --enable-libfreetype --enable-libfribidi --enab
e-libvidstab --enable-libvmaf --enable-libzimg --enable-amf --enable-cuda-llvm --enable-cuvid --enable-ffnvcodec --enab
e-nvdec --enable-nvenc --enable-d3d11va --enable-dxva2 --enable-libmfx --enable-libgme --enable-libopenmpt --enable-libo
pencore-amrwb --enable-libmp3lame --enable-libtheora --enable-libvo-amrwbenc --enable-libgsm --enable-libopencore-amrnb
--enable-libopus --enable-libspeex --enable-libvorbis --enable-librubberband
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'input.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf57.84.100
  Duration: 00:00:19.98, start: 0.000000, bitrate: 1416 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720, 1281 kb/s, 60 fps, 60 tbr, 15360 tbn, 1
20 tbc (default)
    Metadata:
      handler_name    : VideoHandler
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 120 kb/s (default)
    Metadata:
      handler_name    : SoundHandler
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> vp9 (libvpx-vp9))
  Stream #0:1 -> #0:1 (aac (native) -> opus (libopus))
Press [q] to stop, [?] for help
[libopus @ 000001acd64d84c0] No bit rate set. Defaulting to 96000 bps.
[libvpx-vp9 @ 000001acd64d6a00] v1.9.0-158-gebefb90b7
[libvpx-vp9 @ 000001acd64d6a00] Neither bitrate nor constrained quality specified, using default CRF of 32
Output #0, webm, to 'output.webm':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: vp9 (libvpx-vp9), yuv420p, 1280x720, q=-1--1, 60 fps, 1k tbn, 60 tbc (default)
    Metadata:
      handler_name    : VideoHandler
      encoder         : Lavc58.91.100 libvpx-vp9
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
    Stream #0:1(und): Audio: opus (libopus), 48000 Hz, stereo, flt, 96 kb/s (default)
    Metadata:
      handler_name    : SoundHandler
      encoder         : Lavc58.91.100 libopus
frame=  674 fps= 36 q=0.0 size=     768kB time=00:00:11.27 bitrate= 558.1kbits/s speed=0.596x
```

Command line output using libvpx in ffmpeg to create a VP9 video file

Conclusion

VP9 has several advantages over other codecs. As well as being open-source, it works well with higher resolutions and enjoys support on many devices. It also has drawbacks, such as slow encoding speed and limited iOS support. If you're delivering high-resolution videos—and aren't worried about iOS—VP9 is the strongest video codec choice, at least for now. If you want to target every device going, you'll need to use something else, either exclusively or alongside VP9. You should consider which of these will make the most difference to your team and your customers when deciding what to use. If you need a hand implementing modern codecs, Bitmovin can help: https://bitmovin.com/encoding-service/

If you do choose VP9, you can take advantage of a huge customer base, high-quality encoding, and well-developed supporting tools. It's a great way to deliver videos and can help you produce content that works for you and your clients.

Latest from our blog

Everything you need to know about Video Watermarking

September 1, 2021

The State of Streaming and Content Piracy In the last 10 years, over 100 unique […]