

Ticao Zhang and Shiwen Mao Dept. Electrical & Computer Engineering, Auburn University

Editors: Michelle X. Gong and Shiwen Mao

AN OVERVIEW OF EMERGING VIDEO CODING STANDARDS

Today's popular video coding standards, such as H.264/AVC, are widely used to encode video into bit streams for storage and transmission. With the explosive growth of various video applications, H.264/AVC may not fully satisfy their requirements anymore. There is an increasing demand for high compression efficiency and low complexity video coding standards. In this article, we provide an overview of existing and emerging video coding standards. We review the timeline of the development of the popular H.26X family video coding standards, and introduce several emerging video coding standards such as AV1, VP9 and VVC. As for future video coding, considering the success of machine learning in various fields and hardware acceleration, we conclude this article with a discussion of several future trends in video coding.

Video coding standard; H.264 Advanced video coding (AVC); High Efficiency Video Coding (HEVC); AOMedia Video 1 (AV1); Machine learning; Hardware acceleration.

Various industrial studies and reports have predicted the drastic increase of video traffic in the Internet [1] and wireless networks [2]. Today's popular video coding standards, such as H.264 Advanced video coding (AVC), are widely used in video storage and transmission systems. With the explosive growth of video traffic, it has been recognized that video coding technology is crucial in providing a more engaging experience for users. For users that are often restricted by a limited data plan and dynamic wireless connections, high-efficiency video coding standards are essential to enhance their quality of experience (QoE). On the other hand, network service providers (NSP) are constrained by the scarce and expensive wireless spectrum, making it challenging to support emerging data-intensive video services, such as high-definition (HD) video, 4K ultra high definition (UHD), 360-degree video, augmented reality (AR), and virtual reality (VR). Efficient video coding standards are indispensable for enabling such video applications.

The most widely adopted video coding standard nowadays is H.264 Advanced video coding (AVC) [3], which has been proposed for 15 years. So far, most hardware manufacturers support H.264 AVC and almost all the major platforms, such as YouTube, Netflix, and Samsung VR, adopt this standard. However, the efficiency of H.264 is still not sufficiently high especially for today's heterogeneous wireless links, scarce wireless spectrum, and data-intensive video applications. For example, a high-definition (HD) video (1080p) will generate four times more data traffic than a standard resolution video (480p). A YouTube 360-degree video generates 4 to 5 times data traffic of a standard YouTube video of the same resolution. The maximum resolution H.264 can support is 4K (4096×2304), while 6K and 8K videos, which the next generation VR systems promise to deliver, are obviously out of the scope of H.264. Finally, the quality of H.264 videos may not always be satisfactory (e.g., with blocking, ringing, and flickering effects). It becomes more and more challenging for H.264/AVC to fully satisfy the requirements of emerging multimedia applications, and there is an increasing demand for developing new video coding standards with a high compression ratio and low complexity.

On the other hand, the computational power of contemporary computers keeps rising over the past 15 years. Parallel computing together with hardware acceleration make it possible to process more tasks that were computationally prohibitive in the past. For example, the use of graphics processing unit (GPU) allows large blocks of data to be processed in parallel, which greatly saves time and improves efficiency. Furthermore, the advances in algorithms (e.g., machine learning) have greatly changed the traditional way that signals are processed. As a result, emerging standards begin to challenge the status of H.264. For example, H.265 with parallel computing and an adoption of larger blocks now achieves a coding efficiency that is twice of H.264, which can thus support videos of up to 8K resolution.

In this article, we first introduce the fundamental concept of video coding in Section 1. We review the basics of H.264 and the timeline of the development of the H.26X family video coding standards



FIGURE 1. Video encoder and decoder (codec).

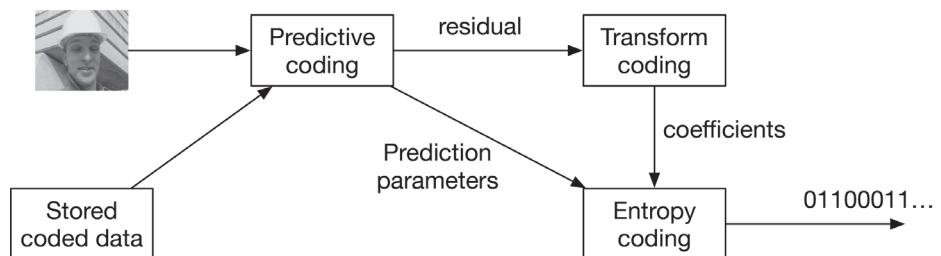


FIGURE 2. Architecture of a video encoder.

in Section II. We then introduce several emerging video coding standards, such as VVC, VP9, and AV1 in Section III, which are quickly gaining popularity. For future video coding, considering the success of machine learning in various fields and hardware acceleration, we discuss several future trends in video coding in Section IV. Continuous efforts are in great demand for high efficiency video coding standards to enable future multimedia applications.

I. VIDEO CODING PRELIMINARIES

A digital video consists of a sequence of frames sampled in spatial and temporal domains, as a direct representation would require an enormous amount of bits. For efficient video storage and transmission, it is essential to compress the digital video signal. For example, a 150-minute colored movie at 30 frames per second (fps) with a spatial resolution 720×480 is 280GB. Without compression, it would be challenging to be transmitted through a bandwidth limited wireless link, let alone HD movies. Video coding is a necessary step to reduce the bandwidth requirement and storage space for digital videos.

Video coding refers to a process in which the videos are compressed to a certain degree to achieve a high transmission and storage efficiency. Generally, there are two kinds of methods: (i) entropy coding: to compress video data towards the Shannon limit in a

lossless manner; and (ii) lossy coding: to remove the redundancy and less important information in video data. It turns out that with entropy coding only moderate compression ratios can be achieved due to the Shannon limit. Lossy compression of videos, on the other hand, is generally more efficient since the human visual system is tolerant to loss of details.

Many of today's basic video coding concepts were developed in the 1970s and 1980s. An encoder converts a video into a compressed format and a decoder restores a compressed video back to an uncompressed format, which collectively form the term codec (encoder/decoder) as shown in Figure 1. Figure 2 illustrates the structure of a typical video encoder. It is composed of three parts: (i) a predictive coding unit, (ii) a transform coding unit, and (iii) an entropy coding unit.

(i) Predictive coding: The predictive coding unit reduces the video's redundancy by exploiting temporal (inter-prediction) and spatial redundancies (intra-prediction). Generally, there are two ways of predictive coding, motion estimation (ME) and motion compensation (MC). They are illustrated in Figures 3 and 4, respectively.

- Motion estimation (ME) refers to finding a suitable match between regions in the reference frame and that in the past or future frames.

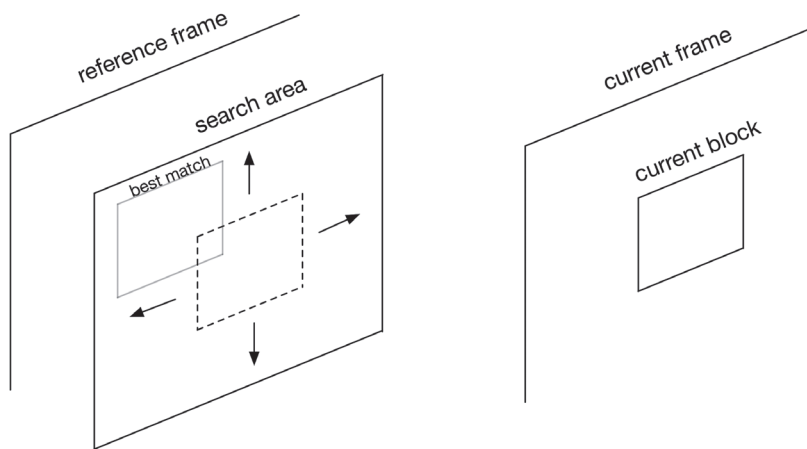


FIGURE 3. Motion estimation.

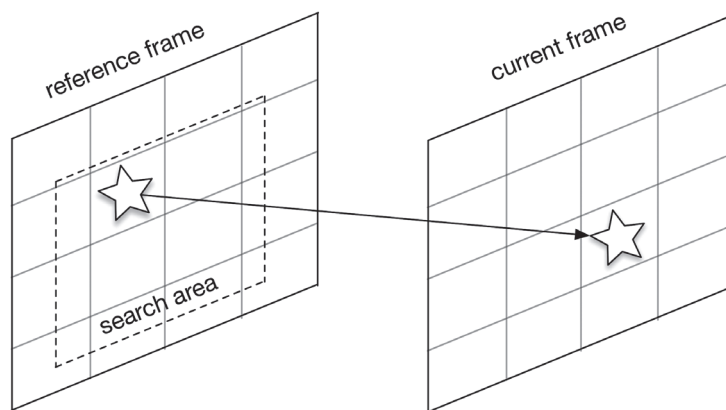


FIGURE 4. Motion compensation.

- Motion compensation (MC) refers to finding the difference (residual) between the matching regions and the target region.

Predictive coding generates residuals and the motion vectors with ME and MC. The residuals are created by subtracting the prediction from the actual current frame, while the motion vector is generated by computing the offset between the current block and the position of the candidate region. Motion vector indicates the block's moving direction.

(ii) Transform coding: The transform coding module generates a set of coefficients, each of which is a weight for the standard basis pattern. Followed by a quantizer, a reduced precision yet bits-saving, quantized coefficient is obtained. For example, discrete

cosine transformation (DCT), developed in 1974, is a widely used transform coding technique. In H.264, a block of residual samples is converted to DCT coefficients with transform coding. This process helps to reduce the dependency between the sample points.

(iii) Entropy coding: Entropy coding produces a compressed bit stream that is suitable for transmission and storage. Recall that predictive coding and transform coding remove a significant amount of redundancy. To further remove the redundancy in the coded data itself, the parameters of the prediction coding and most of the quantized transformed coefficients are encoded with entropy coding. Common entropy coding methods include variable length coding (VLC), arithmetic coding, and Huffman coding.

Motion vectors are coded with another table of VLC. For example, in Huffman coding, the most frequent symbols are encoded with the shortest bit stream, while the least frequent symbols are encoded with the longest bit stream. By entropy coding, the average length of the encoded bit stream is minimized and compressed.

The video decoding process is the inverse of the encoding process. The video decoder recovers the coefficients and prediction parameters from the compressed bit stream with an entropy decoder. The spatial decoder is then used to reconstruct the residual frame. Finally, a prediction decoder uses the parameters and the previously decoded pixels to reconstruct the frame.

II. H.264 AND H.265/HEVC

The H.264 standard was initially developed between 1999 and 2003, and was then extended in 2003 to 2009. It is now a fundamental technology that has been 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 1990s are mostly based on a similar codec model that incorporates predictive coding, transform coding, and entropy coding. H.261, H.263, MPEG-1, MPEG-2, MPEG-4 visual, and H.264/AVC are all developed under this framework. Although there are differences in details, they share most of the basic functions. Generally, H.264 achieves a better compression efficiency and has greater flexibility in compressing, transmitting, and storing videos than MPEG-1, MPEG-2, etc.

A. A Brief History of the H.26X Family Standards

Before looking into the details of H.264, we briefly introduce two important video standardization groups. One is the video coding experts group (VCEG) of the international telecommunication union (telegraphy section) (ITU-T). It is responsible for standardization of the H.26x series of coding standards, including H.120, H.261 [4], H.263 [5] and H.263+, as shown in Figure 5. The other group is the moving picture experts group (MPEG) under the international standards organization

(ISO) and international electrotechnical commission (IEC). This group developed the popular MPEG-1 [6], and MPEG-4 [7] standards.

These two groups, i.e., ITU and ISO/IEC, then collaborated and jointly developed H.262/MPEG-2 [8] in 1995. In 2001, these two groups formed the joint video team (JVT) and cooperated in the development of H.264/MPEG-4 advanced video coding (AVC) [3]. Later, the joint collaborative team on video coding (JCT-VC) developed a new generation video coding standard H.265 in 2003, which is also called high efficiency video coding (HEVC) [9]. HEVC achieves a 50% bit-rate reduction for the same perceptual video quality over H.264. Their main features are compared in Table I.

B. H.264/MPEG-4 AVC Standards

Conceptually, the H.264 standard consists of a video coding layer (VCL) and a network abstraction layer (NAL). The VCL converts a video into a bit stream. The NAL specifies the format of the data and defines header information.

1. Video coding layer (VCL): The H.264 standard codec is illustrated in Figure 6. The encoder processes a video input in units of macroblock (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

TABLE 1. The MPEG family and H.26X family video coding standards			
	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 HDTV and video on DVDs; Introduction of profiles and levels; Nonlinear quantization 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 (AVS)	2003	Co-published with H.264/AVC
H.26X family	H.120	1984	The first digital video coding standard
	H.261	1988	Developed for video conferencing over ISDN; Block-based hybrid coding with integer pixel motion compensation; Support for CIF and QCIF resolutions
	H.262	1995	See 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 HDTVs; 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

then subtracts the prediction from the current macroblock to form a residual. A block of residual samples is transformed using a 4×4 or 8×8 integer transformation and outputs a set of DCT coefficients. The transformed coefficients and other

information are then quantized and coded into bit streams using entropy coding. At the decoder, the quantized, transformed coefficients and the prediction information are firstly extracted from the bit stream. The coefficients are then rescaled to restore

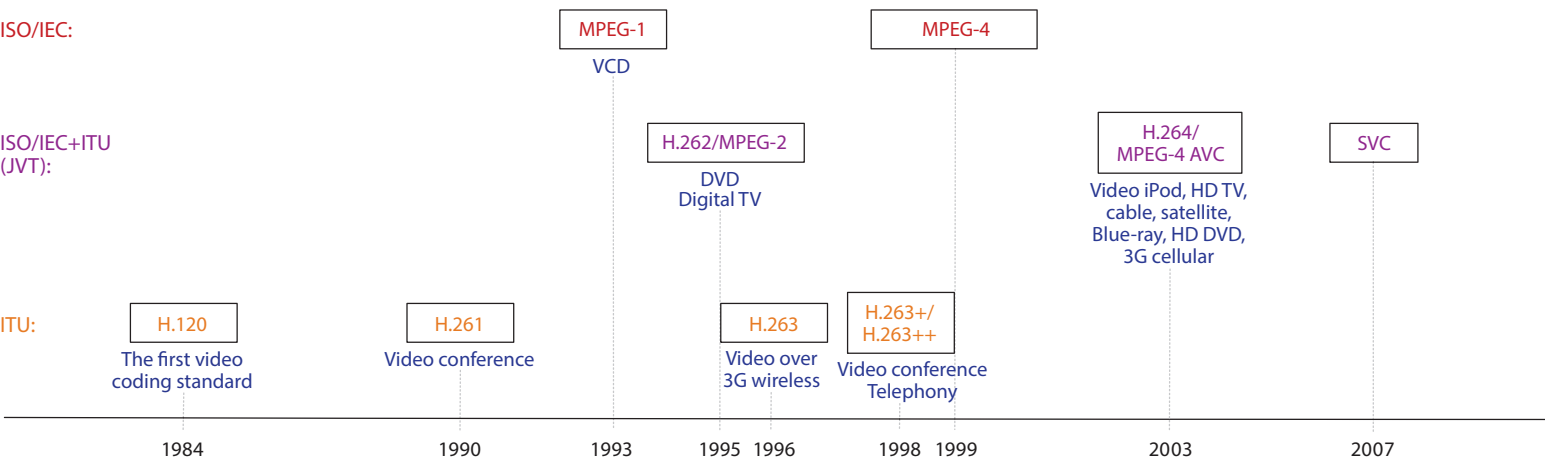


FIGURE 5. A timeline of video codex standards proposed by ITU + ISO/IEC.

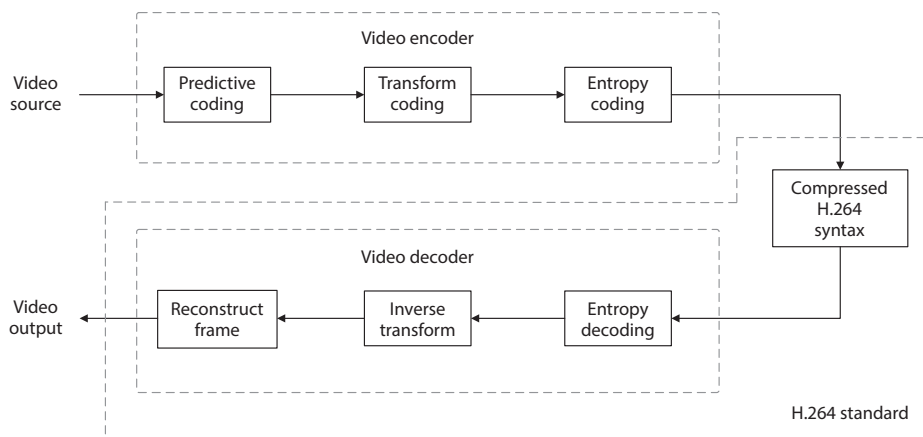


FIGURE 6. The H.264 codec architecture.

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 reconstruct a decoded macroblock.

2. SVC extension: The JVT developed a scalable video coding (SVC) extension to the H.264 standard in 2007 [10]. The key idea is to divide a video stream into one base layer and multiple enhancement layers. The base layer provides the essential information, while the enhancement layers preserve detailed information. The more enhancement layers, the better the video quality. Hence SVC enables a graceful degradation in reconstructed video quality when the video is transmitted in lossy channels; it also enables bit rate and power adaptation in heterogeneous network environments.

Basically, there are three types of scalability, i.e., temporal, spatial, and quality

scalability [10]. Spatial scalability and temporal scalability refer to the approach that a source content with reduced picture size or reduced frame rate is decoded from a subset of the bit stream, respectively. Quality scalability refers to the approach that a subset of the total stream provides the same spatial-temporal resolution as the complete bit stream, but with a lower fidelity.

C. H.265/ High Efficiency Video Coding (HEVC)

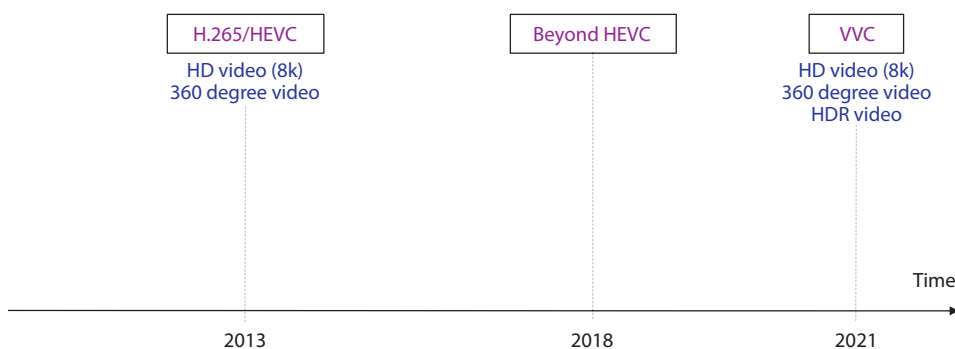
(i) Video coding layer (VCL): The encoder part of HEVC employs the same structure (inter/intra prediction and transform coding) as in H.264. Each frame of the input video sequence is divided into block shaped regions. The first picture of the video sequence is coded using only intra-picture prediction. All remaining pictures are coded using inter-picture predictive coding. The macroblock concept adopted in H.264 is now translated into a coding

tree unit (CTU) in HEVC. A CTU can be of size 64×64 , 32×32 , or 16×16 . Each CTU is organized in a quad-tree form to further partition to smaller sized coding units (CU). Each CU can be predicted via intra-prediction or inter-prediction. The prediction residual is coded using block transforms. Context adaptive binary arithmetic coding (CABAC) is used in the entropy coding module. The decoding process is an inverse procedure of the coding process.

(ii) Profiles and levels: A profile specifies a set of coding tools to generate a conforming bit stream while a level restricts a certain key parameter of the bit stream based on the decoder's processing load and capability. The high-level syntax of HEVC helps to improve the flexibility for operation in various environments and makes the system more robust to data losses during transmission.

(iii) Parallel processing: The picture is partitioned into rectangular regions called tiles, which can be independently decoded. Tiles make it possible to adopt parallel processing, but a more sophisticated synchronization is needed. In addition, the wave-front parallel processing (WPP) provides better compression performance when it is enabled, while dependent slice segment allows data in a wave-front or tile to be carried in a separate NAL unit. These mechanisms all contribute to better parallelism in HEVC.

The question is, since 2013 when HEVC was approved as a video coding standard, why it is not widely adopted? One reason is H.264 has only one patent pool while H.265 has three patent pools with different pricing structures and terms and conditions. The unclarity of royalties makes many web browsers (e.g., Chrome and Firefox) even not support it. Another reason is HEVC requires nearly 10x more computing power for decoding. Generally, when a computer attempts to play an HEVC video, it may become quite slow if it only adopts software decoding with CPU. But if a computer can hand off the computational load to the graphics card (or the integrated graphics chip in the CPU), i.e., with hardware decoding, the processing could become much faster and more energy efficient. HEVC would gain higher popularity as more hardware markets adopt it.



III. BEYOND HEVC

Beyond HEVC, the joint video experts team (JVC) has launched a new video coding standard, i.e., versatile video coding (VVC). In parallel with the open video coding process of JVC, a few companies are also developing their own video codecs, such as VP9, VP10, and AV1.

A. Versatile Video Coding (VVC)

Started in 2015, the JVC began to consider video coding standard with capabilities beyond HEVC. VVC is said to be the next video coding standard as illustrated in Figure 7. The primary target of VVC is to provide a significant performance improvement over HEVC. This standard is expected to be completed by 2020. VVC will enable a 128×128 CTU with a recursive quadtree partitioning (QT) and nested recursive multi-type tree partitioning (MTT). It will enable high-quality video services and many emerging applications, such as 360-degree omnidirectional immersive multimedia and high-dynamic-range (HDR) videos.

B. VP9, VP10 and AV1

VP9, a powerful sibling of VP8, is now used on Google's video platform YouTube and serves billions of views every day. The VP9 encoder incorporates a larger

prediction block size, up to 64×64, and allows breakdown using a recursive decomposition all the way down to 4×4 blocks. It can thus achieve a high efficiency. VP9 supports 10 intra-prediction modes and four inter-prediction modes. It uses DCT, Asymmetric Discrete Sine Transform (ADST), and Walsh-Hadamard Transform (WHT) for transform coding. The bit stream is arithmetically encoded. A loop filter is designed to eliminate the blocking artifacts on the boundary.

VP9 supports high-dynamic-range (HDR) videos and enables lossless compression. VP9, launched in 2013, is as competitive in coding efficiency as the state-of-the-art HEVC codec. Furthermore, a VP9 bit stream is error resilient and the decoding process can be conducted in a parallel mode. It also allows both temporal/spatial scalabilities.

Even though the performance of VP9 is satisfactory, continued growth of the demand for high efficiency video applications, such as VR and 360-degree 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), which is a forum of more than 30 leading tech companies such as Microsoft and Mozilla,

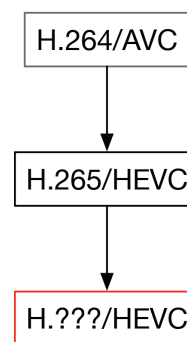


FIGURE 7. Evolution towards versatile video coding.

to jointly develop a royalty-free codec called AOMedia Video 1 (AV1).

The evolution towards AV1 is shown in Figure 8. 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 finalized in early 2018. 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.

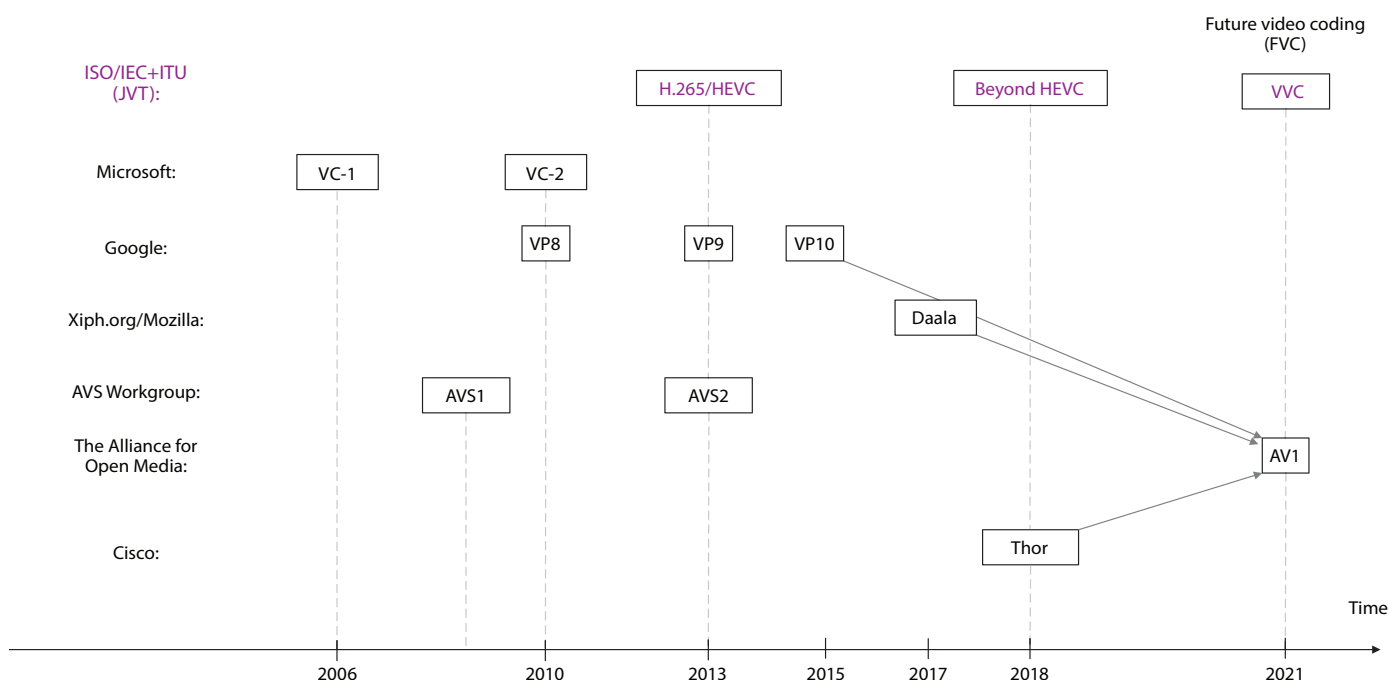


FIGURE 8. Future video coding.

TABLE 2. A Comparison of H.265/HEVC, VP9, and H.264/AVC

Standard	Coding speed	Coding efficiency	Hardware support	Maximum resolution	Image quality	License fee	Device support	Decoding power
H.264/AVC	Fast	Medium	Well	4k	Not very good (black ends up looking gray and blocky encoding artefacts)	Not expensive	Widely supported by most devices	Low
H.265/HEVC	Slow	Twice as H.264	Poor	8k	Color looks better, and artefacts are less pronounced	Expensive and patent risks	Not widely supported	High
VP9	Slow	Twice as H.264	Poor	8k		Royalty free and open source	Supported by YouTube and Netflix	High

- 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 would definitely be the only choice. However, if one prefers a royalty-free and highly-efficient codec, AV1 may be the way to go.

IV. FUTURE TRENDS

According to a streaming media survey, by the year 2020, close to 40% of all video distributed will be 4K or higher resolutions [12]. HEVC will steal market shares from H.264 and will account for half of the market. VP9 will account for another 10% of the total market. From the estimated bitrates for HD videos shown in Figure 9, we can see that the performance of video coding standards was improved for about 50% every nine years, with an increase in computational complexity and memory requirement. It is not clear as such whether Moore's law will hold true in the future. But presently, considering the fast development of machine learning technologies and hardware acceleration, it is highly promising to fully harvest the potential of new video standards to satisfy the requirements of emerging video applications.

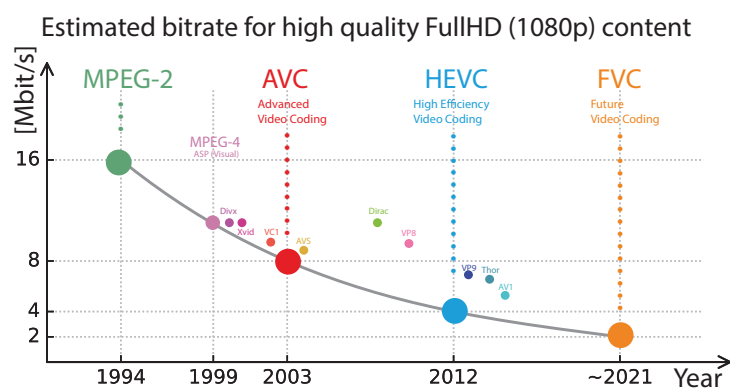


FIGURE 9. Illustration of performance of successive generations of video encoders. Figure courtesy of Figure 1 in [11])

A. Machine Learning for Video Coding

Machine learning has found great success in image recognition and classification. Video compression is fundamentally a prediction/regression problem. Thus there is a great potential to incorporate machine learning in new video coding standards. In VVC, an ad-hoc group has been set up for further study of deep learning-based video coding.

The first work of learning based image compression was published in 2016 [13], which demonstrates a better performance than the image coding standard JPEG. However, at present there are very few published works for video compression with learning-based methods. A big challenge is that it is extremely difficult to train a neural network for motion compensation (MC). Several future research areas, including learning-based intra-prediction, inter-prediction, sub-pixel interpolation, transformation and quantization, and entropy coding, need to be explored.

Presently, there are two sets of works that attempt to apply machine learning to video coding. One is enhancing traditional codecs with ML, while the other is pure learning-based compression.

(i) Codec enhancement with machine learning: The key idea is to use a machine learning based approach to replace the previously manually designed functions to achieve a performance improvement or a complexity reduction [14,15].

(ii) Pure learning-based video compression: In [16], the authors propose a deep learning approach for video compression with bidirectional recurrent convolutional neural networks (RCNN). Due to the large computational burden, the proposed method only works for a toy dataset, but not full video frames. Moreover, it fails to achieve the state-of-the-art

compression ratio and speed. The authors in [17] propose a learning-based framework for video compression with the VoxelCNN model. Although the work fails to implement functions such as entropy coding, it provides a possible direction for developing new learning-based codecs.

B. Hardware Acceleration

Thanks to the progress in hardware acceleration, more and more advanced algorithms can be deployed to achieve better efficiency with parallel computing and more memory. Recently, graphic processing units (GPUs) have emerged to accelerate various numerical and signal processing applications. These units are designed to support massive computations and they are highly cost-effective for arithmetic computation. Thus, GPUs are helpful to process the huge amount of data transferred in certain video applications. However, only certain types of computation in video coding are suitable for GPU processing. One challenge is how to enhance the parallelism of video codecs.

C. Emerging Applications

The development of video coding standards has been driven by emerging video applications. Imagine playing a basketball game in your sitting room along with other remote players using a VR system. Once such new applications are available, the way we watch and play sport games will be revolutionized. However, such new applications often generate a massive amount of data. A higher efficiency video coding standard should be able to support such data-intensive, delay sensitive applications while guaranteeing a satisfactory performance. Examples of such emerging applications include virtual reality, stereoscopic videos/Multi-view video/360 vision video, HDR video, 8K video, video surveillance, telemedicine, video games, etc.

D. Energy-aware Video Coding Standards

Finally, energy-aware video coding standards are necessary to reduce the energy consumption and extend the device's battery life. Current video coding standards, such as H.264, are suitable for the case where the encoder has sufficient power. However, the encoder complexity and energy consumption are becoming limiting factors for mobile

devices for real-time video communications. User QoE would be degraded by the lower rate video frames or limited conversation time, as a result of conserving battery power. For battery-operated video cameras, it is also critical to adopt a more energy efficient codec design for extended operational time.

Similarly, energy efficient video decoders are highly desirable. As more and more videos will be displayed on mobile screens, low complexity, low bit rate, and low distortion video decoder will be instrumental to extend the battery life of mobile devices.

CONCLUSION

This article presented an overview of the existing video coding standards. The features and applications of current video coding standards, such as H.264/MPEG-4 AVC and HEVC, were introduced. The field of video coding is developing rapidly especially with the recent interest in machine learning and hardware acceleration. Great efforts are needed for developing high efficiency and

adaptive video coding standards to meet the growing demand of emerging multimedia applications. ■

Ticao Zhang received a BE in 2014 and MS degree in 2017 from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. He is currently pursuing a PhD in Electrical and Computer Engineering at Auburn University. His research interests include video coding and communications, and optimization and design of wireless multimedia networks.

Shiwen Mao received a PhD in electrical and computer engineering from Polytechnic University, Brooklyn, N.Y., in 2004. He is the Samuel Ginn Professor and Director of Wireless Engineering Research and Education Center at Auburn University. His research interests include wireless networks, multimedia communications, and smart grid. He is a recipient of the Auburn University Creative Research & Scholarship Award in 2018, the NSF CAREER Award in 2010, and The 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a Fellow of the IEEE.

REFERENCES

- [1] Cisco, "The zettabyte era: Trends and analysis." <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>.
- [2] Ericsson, "Mobile traffic analysis by application." <https://www.ericsson.com/en/mobility-report/reports/november-2017/mobile-traffic-analysis-by-application>.
- [3] ITU-T, Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), 2003.
- [4] ITU, Video Codec for Audiovisual Services AT PX 64 Kbits. ITU, 1994.
- [5] K. Rijkse, "H. 263: Video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42–45, Dec. 1996.
- [6] 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.
- [7] ISO/IEC, Coding of Audio-Visual Objects—Part 2: Visual, ISO/IEC 144962 (MPEG-4 Visual version 1), 1999.
- [8] 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.
- [9] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand et al., "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.
- [11] D. Karwowski, T. Grajek, K. Klimaszewski, O. Stankiewicz, J. Stankowski, and K. Wegner, "20 years of progress in video compression—from MPEG-1 to MPEG-h HEVC. General view on the path of video coding development," in *Proc. The 2016 International Conference on Image Processing and Communications*, Bydgoszcz, Poland, Sept. 2016, pp. 3–15.
- [12] J. Ozer, "Encoding 2020: Experts predict the future of video encoding," *Streaming Media Magazine*, Sept. 2015. <http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Encoding-2020-Experts-Predict-the-Future-of-Video-Encoding-106619.aspx>.
- [13] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015, accessed on Nov. 1, 2018.
- [14] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2225–2238, July 2015.
- [15] R. Song, D. Liu, H. Li, and F. Wu, "Neural network-based arithmetic coding of intra prediction modes in hevc," in *Proc. IEEE VCIP'17*, Saint Petersburg, FL, Dec. 2017, pp. 1–4.
- [16] C. Y. S. Kin and B. Coker, "Video compression using recurrent convolutional neural networks," [online] Available: <http://cs231n.stanford.edu/reports/2017/pdfs/423.pdf>.
- [17] Z. Chen, T. He, X. Jin, and F. Wu, "Learning for video compression," *arXiv preprint, arXiv:1804.09869*, 2018.