Video Codec Technology

I. Introduction to Video Codecs

A. Defining Video Codecs: The "Coder-Decoder" Paradigm

A video codec is a sophisticated hardware or software utility central to digital video processing. Its fundamental purpose is twofold: to encode (compress) raw video data into a more manageable format and subsequently to decode (decompress) this data for playback or further processing.[1] The term "codec" itself is a portmanteau, derived from "coder-decoder" or, alternatively, "compressor-decompressor".[1] This dual capability is the defining characteristic of a codec; a technology must be proficient in both compression and decompression to be classified as such.[1]

The mechanism underlying a video codec involves the application of complex algorithms designed to significantly reduce the voluminous size of raw video files. This reduction is paramount for efficient storage on various media and for effective transmission across networks, particularly the internet, where bandwidth is a critical constraint.[1] During playback, the codec reverses the process, decompressing the encoded data to reconstruct the video sequence for viewing.[1] This operational principle can be intuitively understood by analogy to the process of zipping files for smaller size and then unzipping them to restore the original content.[4]

The significance of video codecs in the contemporary digital landscape cannot be overstated. In their absence, the storage, distribution, and streaming of high-quality video content would be practically unfeasible due to the immense file sizes associated with uncompressed video data.[1] Codecs are, therefore, the bedrock of modern digital media, underpinning a vast array of applications ranging from global video streaming services and high-definition broadcasting to real-time video conferencing and personal video recording.[1] The seamless consumption of video content that characterizes modern digital interaction is largely attributable to the efficiency and efficacy of video codec technology. While these technologies are fundamental, their successful implementation often results in their "invisibility" to the end-user. A well-functioning codec delivers smooth video playback without interruptions or noticeable degradation in quality, making the intricate underlying technology transparent. It is typically only when issues arise, such as "codec not supported" errors or poor visual fidelity, that the user becomes aware of the complex processes involved. This inherent transparency, when achieved, is a hallmark of a successful codec, as the ultimate goal extends beyond mere technical metrics to the enhancement of user experience, abstracting the complexities of video delivery

entirely from the viewer.

B. The Imperative for Video Compression: Addressing Raw Video Data Overload

The primary impetus for the development and continuous refinement of video codecs stems from the extraordinary volume of data inherent in uncompressed video. Without compression, digital video files are impractically large, posing insurmountable challenges for storage, transmission, and processing.

1. Calculating Uncompressed Video File Sizes

The size of an uncompressed video file is a direct function of several parameters: the duration of the video, the frame rate (frames per second, fps), and the data size of each individual frame.[7] The frame size, in turn, is determined by multiplying the horizontal resolution (number of pixels per line), the vertical resolution (number of lines), and the color depth (number of bits used to represent the color of a single pixel).[7]

To illustrate the magnitude of uncompressed video data, consider the following examples:

- 1080p Full HD Video: A single minute of uncompressed video at a resolution of 1920x1080 pixels (1080p), a frame rate of 24 fps, and using 8-bit RGB color (which translates to 24 bits per pixel, or bpp), would result in a file size of approximately 8958 MB, or nearly 8.96 GB.7
  The calculation is as follows:
  Frame Size = 1920 pixels×1080 pixels×24 bits/pixel=49,766,400 bits
  Converting to bytes (8 bits = 1 byte): 49,766,400 bits/8 bits/byte=6,220,800 bytes/frame
  Data Rate per Second = 6,220,800 bytes/frame×24 fps=149,299,200 bytes/second
  Data Rate per Minute = 149,299,200 bytes/second×60 seconds/minute=8,957,952,000 bytes
  This is approximately 8.96 GB (using the metric definition where 1 GB = 109 bytes).
  A similar estimation indicates that raw HD (1080p) video at 30 fps requires approximately 150 MB of storage per second, equating to a staggering 540 GB per hour.8
- **4K UHD Video:** The data requirements escalate dramatically with higher resolutions. For instance, a REDCODE28 raw 4K video file can consume 1.76 GB per minute of footage at 24 fps. An Apple iPhone 13 capturing video in ProRes

format at 4K resolution can generate files as large as 5.5 GB per minute.[7]

These figures underscore the critical need for effective compression technologies. It is also pertinent to note the distinction in data unit prefixes: metric prefixes (such as kilo-, mega-, giga-) typically denote powers of 1000 (e.g., 1 MB = 106 bytes), whereas binary prefixes (kibi-, mebi-, gibi-) denote powers of 1024 (e.g., 1 MiB = 220 bytes).[7] For consistency with prevalent industry terminology, this report will primarily utilize metric prefixes unless explicitly stated otherwise.

The following table provides further illustration of uncompressed video data sizes for common formats:

Table 1: Illustrative Uncompressed Video Data Sizes

Resolution Name	Dimensions (Pixels)	Frame Rate (fps)	Color Depth (bits/pixel)	Data Rate (Mbps)	Size per Minute (GB)
SD (480p)	640 x 480	30	24	221.18	1.66
HD (720p)	1280 x 720	30	24	663.55	4.98
FHD (1080p)	1920 x 1080	30	24	1492.99	11.20
FHD (1080p)	1920 x 1080	60	24	2985.98	22.39
UHD (4K)	3840 x 2160	30	24	5971.97	44.79
UHD (4K)	3840 x 2160	60	24	11943.94	89.58
UHD (8K)	7680 x 4320	60	24	47775.74	358.32

*Note: Calculations assume uncompressed RGB color. Data rates are approximate and calculated as (Horizontal Pixels * Vertical Pixels * Bits per Pixel * Frame Rate) / 106. Size per Minute = (Data Rate * 60) / (8 * 1000).*

2. Impact on Storage and Transmission

The immense data volumes of uncompressed video directly translate into prohibitive storage requirements and unmanageable bandwidth demands for transmission.[3] Storing hours of uncompressed HD or 4K footage would necessitate petabytes of storage, a costly proposition even for large enterprises. Similarly, transmitting the

uncompressed 1080p video example (at 24 fps) would require a sustained network bandwidth of nearly 150 MB/sec (approximately 1.2 Gbps), excluding any audio data or transmission overhead.[5] Such bandwidth requirements are far beyond the capacity of typical internet connections and would render video streaming services economically and technically infeasible.

Video codecs address these challenges by dramatically reducing file sizes. This compression makes it feasible to store extensive video libraries, distribute content efficiently online, and stream video smoothly over existing network infrastructures.[1] The economic implications are profound; without efficient compression, the entire digital video industry, from content creation to distribution and consumption, would be unsustainable. The continuous drive for more efficient codecs is, therefore, not merely a technical pursuit but an economic imperative, enabling reduced operational costs for providers and enhanced accessibility for consumers, such as streaming high-quality video on mobile data plans.

C. Fundamental Goals of Video Codecs: Balancing Quality, Bitrate, and Complexity

The design and implementation of video codecs are governed by a fundamental set of objectives, often referred to as the "compression trilemma." This involves achieving an optimal balance between three critical, and often conflicting, parameters:

1. **Video Quality:** This refers to the perceptual fidelity of the compressed video when compared to the original, uncompressed source. Higher video quality generally implies that less information has been discarded during the compression process. The assessment of quality can be subjective (human visual perception) or objective (using metrics like PSNR or VMAF).
2. **Bitrate (File Size):** Bitrate is the amount of data used to represent the video per unit of time, typically measured in bits per second (bps) or megabits per second (Mbps). A lower bitrate directly translates to a smaller file size, which in turn means reduced storage requirements and lower bandwidth consumption for transmission.[8]
3. **Computational Complexity:** This pertains to the amount of processing power and memory required to perform the encoding (compression) and decoding (decompression) operations. More sophisticated compression algorithms may yield better quality at lower bitrates but often demand greater computational resources, potentially impacting real-time performance, power consumption (especially on mobile devices), and the cost of encoding/decoding hardware.

The core challenge in codec design is to navigate these trade-offs effectively. The

primary objective of an efficient delivery codec is to provide the highest possible perceived video quality at a specific target bitrate, ensuring that any data "lost" during the compression process is minimally perceptible to the viewer.[1] There is an inherent relationship: any modification or technique that makes the encoded video more closely resemble the original, uncompressed source will generally lead to an increase in the resulting data size.[5]

When lossy compression is applied too aggressively, or if the codec is not well-optimized for the content type, visible distortions known as "compression artifacts" can appear in the decoded video. Common artifacts include "blocking" (visible square patterns, especially in flat areas), "blurring" (loss of fine detail), "ringing" (oscillations or halos near sharp edges), and "pixelation" (coarsely visible pixels).[5] The art and science of codec development lie in minimizing these artifacts while maximizing compression efficiency.

II. Core Principles of Video Compression

Video compression technologies operate by identifying and reducing or eliminating various forms of redundancy present in raw video data. Video signals are inherently rich in such redundancies, and codecs are engineered to exploit these characteristics to achieve significant data reduction.[3] The two primary categories of redundancy are spatial and temporal, which are addressed by intra-frame and inter-frame compression techniques, respectively.

A. Redundancy in Video Data

1. Spatial Redundancy and Intra-frame Compression

Spatial redundancy refers to the correlation between neighboring pixels within a single video frame. In typical images, adjacent pixels often share similar or identical color and intensity values, particularly in areas of uniform texture or smooth gradients.[9] For instance, a large patch of blue sky in a frame will contain many pixels with very similar blue values.

Intra-frame compression, also known as "intra-coding," aims to reduce this redundancy *within* each individual frame, independently of any other frames in the video sequence.[4] This process is analogous to compressing a still image. Key techniques employed in intra-frame compression include:

- **Transform Coding:** This is a cornerstone of spatial compression. Techniques like the Discrete Cosine Transform (DCT) or Wavelet Transform convert blocks of pixel data from the spatial domain to the frequency domain.[9] In the frequency domain,

the energy of typical image blocks tends to be concentrated in a few low-frequency coefficients, while high-frequency coefficients, often representing fine details or noise, tend to have smaller magnitudes. These high-frequency components are often less perceptible to the human visual system and can be quantized more aggressively or even discarded, leading to data reduction.
- **Quantization:** Following the transform, quantization reduces the precision of the transform coefficients.[9] This is a crucial step where information loss typically occurs in lossy compression schemes. By representing coefficients with fewer bits, data volume is reduced.
- **Spatial Prediction:** Instead of directly encoding pixel values, intra-prediction modes predict the value of a block of pixels based on the values of already coded neighboring blocks within the same frame. Only the (hopefully small) difference between the predicted block and the actual block, known as the residual, is then transformed, quantized, and encoded.
- **Entropy Coding:** The quantized transform coefficients and prediction mode information are then further compressed using lossless entropy coding techniques, which assign shorter codes to more frequent symbols.

Intra-frame compression is particularly effective for regions of a video frame that exhibit high levels of detail or complex textures.[9] It also provides a baseline for inter-frame prediction, as the fully decoded intra-coded frames (I-frames) serve as reference points.

2. Temporal Redundancy and Inter-frame Compression

Temporal redundancy arises from the similarities between successive frames in a video sequence. In many videos, especially those with static backgrounds or objects exhibiting predictable motion, consecutive frames often share a significant amount of identical or nearly identical information.[9] For example, in a news broadcast, the background behind the anchor may remain unchanged for several seconds.

Inter-frame compression, or "inter-coding," is designed to exploit these similarities *between* frames.[4] Instead of encoding each frame in its entirety as if it were a new still image, inter-frame compression techniques encode only the differences between the current frame and one or more previously coded reference frames (or, in some advanced schemes, future frames). This typically leads to much higher compression ratios than intra-frame compression alone. The core techniques include:

- **Motion Estimation and Motion Compensation (MEMC):** This is the fundamental process in inter-frame coding.
    - **Motion Estimation (ME):** For a block of pixels in the current frame (the target

frame), the encoder searches for a closely matching block in a previously decoded frame (the reference frame). The displacement between the position of the current block and its best match in the reference frame is encoded as a "motion vector" (MV).[9]
    - **Motion Compensation (MC):** The decoder uses the received motion vector and the reference frame (which it has already decoded and stored) to create a prediction of the current block. The encoder then calculates the prediction error, or residual (the difference between the actual current block and the motion-compensated predicted block). This residual, which ideally contains much less information than the original block, is then transformed, quantized, and entropy coded.[10]
- **Frame Types:** To manage the dependencies and enable random access, inter-frame coding typically uses different types of frames:
    - **I-frames (Intra-coded frames or Keyframes):** These frames are coded entirely using intra-frame compression, without reference to any other frame.[10] They serve as starting points for prediction and allow a decoder to begin playback from these points. They generally have the largest size among the frame types.
    - **P-frames (Predicted frames):** These frames are coded using inter-frame prediction from a preceding I-frame or P-frame.[10] They store motion vectors and residuals relative to their reference frame(s). P-frames offer better compression than I-frames.
    - **B-frames (Bidirectional predicted frames):** These frames can use both past and future I-frames or P-frames as references for prediction.[10] By looking in two directions, B-frames can often find better matches and thus achieve the highest compression ratios. However, they introduce more complexity and delay in the encoding and decoding process.

Inter-frame compression is highly effective for videos containing motion, as only the changes between frames and the motion information need to be explicitly encoded and transmitted.[9] The strategic placement and frequency of I-frames, P-frames, and B-frames within a Group of Pictures (GOP) significantly impact both compression efficiency and the ability to seek within the video.

B. Lossy vs. Lossless Compression

Video codecs employ one of two fundamental compression strategies: lossless or lossy.

- **Lossless Compression:**
    - **Definition:** This method reduces the file size of video data without discarding

any information from the original source. Consequently, when the compressed data is decompressed, the original data can be perfectly reconstructed, bit for bit.[1]
    - **Mechanism:** Lossless compression works by identifying and eliminating statistical redundancies in the data. For instance, if a particular sequence of pixel values or symbols occurs frequently, it can be represented by a shorter code. Techniques like Run-Length Encoding (RLE), Huffman coding, and Lempel-Ziv-Welch (LZW) are examples of lossless algorithms often applied to metadata, motion vectors, or transform coefficients after quantization (though quantization itself is lossy).[4]
    - **Use Cases:** Lossless compression is preferred in scenarios where absolute data fidelity is paramount. This includes professional video editing and mastering, medical imaging, satellite imagery, and the initial capture of video by some high-end cameras (often referred to as "raw" or "visually lossless" formats which might still involve some very light, imperceptible lossy steps or chroma subsampling).[1] The degree of file size reduction achieved by lossless compression is generally modest compared to lossy techniques, often in the range of 2:1 to 3:1.
- **Lossy Compression:**
    - **Definition:** This approach achieves significantly higher compression ratios by permanently discarding some of the data from the original video stream.[1] The key is that the discarded data is chosen to be information that is considered less significant or less perceptible to the Human Visual System (HVS).
    - **Mechanism:** Lossy compression techniques are designed to exploit perceptual redundancies. The HVS, for example, is less sensitive to very fine spatial details (high-frequency components), subtle variations in color compared to brightness, and rapid changes that occur too quickly to be fully processed.[4] Common lossy techniques include:
        - **Aggressive Quantization:** Reducing the precision of many transform coefficients, especially high-frequency ones, often rounding them to zero.
        - **Chroma Subsampling:** Reducing the resolution of color information relative to brightness information (discussed in Section III.A).
        - **Frame Skipping or Lowering Frame Rates:** Reducing temporal resolution, though this is more a pre-processing choice than a core codec mechanism for standard video.
    - **Use Cases:** Lossy compression is the dominant method used for the vast majority of video distribution and streaming applications, including online video platforms, broadcast television, and video conferencing.[1] The primary goal is to achieve a file size or bitrate that is practical for transmission and

storage, while minimizing any noticeable degradation in perceived video quality. Most widely used video codecs such as H.264, HEVC, VP9, and AV1 are fundamentally lossy codecs.[1]

- The "loss" in lossy compression is a carefully engineered compromise. While the term might sound detrimental, it is this controlled discarding of information that makes modern video streaming and distribution feasible.[1] The objective is to ensure that the "lost" data is information that viewers are unlikely to miss, thereby achieving a balance between efficiency and perceptual quality.

The effectiveness of lossy compression is not merely a result of mathematical data reduction; it is deeply intertwined with an understanding of human perception. Codecs are meticulously designed to discard information that the HVS is less sensitive to. This implies that the evolution of video codecs is not solely about developing more powerful computational algorithms but also about integrating more sophisticated models of human vision. Consequently, purely objective quality metrics like Peak Signal-to-Noise Ratio (PSNR) may not always perfectly align with subjectively perceived quality, which has led to the development and adoption of more perceptually-oriented metrics such as VMAF.[12]

C. Key Compression Techniques: An Overview

Modern video codecs typically employ a hybrid approach, integrating a suite of compression techniques to address the various redundancies present in video data. The core components of this hybrid model generally include:

1. **Partitioning:** Dividing each video frame into blocks (e.g., macroblocks, Coding Tree Units) for independent or semi-independent processing.
2. **Prediction:**
   - **Intra-frame Prediction:** Predicting a block from spatially neighboring, already coded blocks within the same frame.
   - **Inter-frame Prediction:** Predicting a block from temporally related blocks in other reference frames using motion estimation and compensation.
3. **Transform Coding:** Applying a mathematical transform (commonly DCT or its variants) to the pixel data (for intra-coded blocks) or the residual data (the difference after prediction for inter-coded blocks) to convert it into the frequency domain. This concentrates signal energy into fewer coefficients.
4. **Quantization:** Reducing the precision of the transform coefficients. This is the primary stage where information is lost in lossy compression.
5. **Entropy Coding:** Applying lossless compression techniques (e.g., Huffman coding, Arithmetic coding, CABAC, CAVLC) to the quantized transform

coefficients and other coded information (like motion vectors and prediction modes) to produce the final compressed bitstream.

6. **In-Loop Filtering:** Applying filters (e.g., deblocking filter, SAO) within the encoding loop to reduce compression artifacts on reconstructed frames. These cleaner frames can then be used as higher-quality references for predicting subsequent frames, improving overall efficiency and visual quality.

The sophisticated interplay and adaptive application of these techniques, based on the characteristics of the video content, define the efficiency and complexity of a video codec. The decision-making process within an encoder, often guided by Rate-Distortion Optimization (RDO) principles, determines how these tools are used to achieve the best possible trade-off between bitrate and quality for each segment of the video.

III. The Video Encoding Pipeline: A Technical Deep Dive

The transformation of raw, uncompressed video data into a compact, compressed bitstream is a multi-stage process known as the video encoding pipeline. While specific implementations vary between codecs, most modern standards, including H.264/AVC, H.265/HEVC, and AV1, adhere to a general framework often described as hybrid block-based transform-prediction coding.[11] This section elucidates the key stages of this pipeline.

A. Pre-processing: Color Space Conversion (RGB to YCbCr) and Chroma Subsampling

Before the core compression algorithms are applied, raw video data often undergoes pre-processing steps to make it more amenable to efficient compression.

- Color Space Conversion (RGB to YCbCr):
  Video is typically captured by cameras in the RGB (Red, Green, Blue) color space, where each pixel's color is represented by the intensities of these three primary colors. However, for compression purposes, RGB is often converted to a color space like YCbCr.14
  The YCbCr color space consists of three components:
  - **Y (or Y'):** Luma, representing the brightness or intensity information of the image. The prime symbol (') often indicates that the luma is non-linearly encoded, typically based on gamma-corrected RGB primaries.
  - **Cb (Chroma blue-difference):** Represents the difference between the blue component and a reference luma value ($CB=B'-Y'$).
  - **Cr (Chroma red-difference):** Represents the difference between the red

component and a reference luma value (CR=R′−Y′). The primary rationale for this conversion is that the Human Visual System (HVS) is significantly more sensitive to variations in luma (brightness) than it is to variations in chroma (color).[14] By separating luma from chroma, codecs can process these components differently, particularly enabling more aggressive compression of the chroma information without substantial perceptual loss.[15] The conversion formulas vary slightly depending on the specific standard (e.g., ITU-R BT.601 for standard-definition television, ITU-R BT.709 for high-definition television, ITU-R BT.2020 for ultra-high-definition television). For example, the BT.709 luma (Y') is calculated from linear R, G, B values (after gamma expansion) as: Y′=0.2126R+0.7152G+0.0722B The chroma components are then derived from the differences between the color channels and this luma value, and scaled.[15]

- Chroma Subsampling:
  Leveraging the HVS's lower acuity for color detail, chroma subsampling is a technique used to reduce the spatial resolution of the Cb and Cr components relative to the Y component.14 This directly reduces the amount of color information that needs to be encoded, leading to significant data savings with often minimal impact on perceived image quality.
  Chroma subsampling schemes are commonly denoted by a three-part ratio J:a:b (e.g., 4:4:4, 4:2:2, 4:2:0) 14:
  - **J:** Refers to the width of the sampling region (typically 4 pixels).
  - **a:** Indicates the number of Cb and Cr samples in the first row of J pixels.
  - **b:** Indicates the number of Cb and Cr samples in the second row of J pixels (changes in Cb, Cr relative to the first row for 4:2:0 and 4:1:1).

Table 2: Common Chroma Subsampling Formats

Format	Description of Sampling (for a Jx2 pixel region, e.g., 4x2)	Chroma Samples per J Luma Samples (Avg.)	Relative Chroma Data (vs 4:4:4)	Typical Use Cases	Perceptual Impact
4:4:4	No subsampling. Each luma sample has correspondi ng Cb and Cr samples.	4 Cb, 4 Cr per 4 Y (in each row)	100%	High-end production, graphics, mastering	None; full color fidelity.

4:2:2	Horizontal subsampling by 2. Cb and Cr are sampled at half the horizontal rate of Y.	2 Cb, 2 Cr per 4 Y (in each row)	50%	Professional video, some broadcast standards	Minimal loss for most content; slight color bleeding on sharp vertical color edges.
4:2:0	Horizontal and vertical subsampling by 2. Cb and Cr are sampled at half the rate of Y in both dimensions.	1 Cb, 1 Cr per 2x2 block of Y samples	25%	Most consumer video, streaming, broadcast (DVB, ATSC), Blu-ray	Generally imperceptible for natural video; can be visible on sharp color graphics.
4:1:1	Horizontal subsampling by 4. Cb and Cr are sampled at one-quarter the horizontal rate of Y.	1 Cb, 1 Cr per 4 Y (in each row)	25%	Older DV formats, some specialized applications	More noticeable color bleeding on vertical edges than 4:2:0 or 4:2:2.

4:2:0 is the most prevalent format for consumer video delivery and streaming services due to its excellent balance of compression efficiency and perceptual quality, saving 50% of the chrominance data compared to 4:4:4.[14] While effective, aggressive subsampling can sometimes lead to artifacts like color bleeding or reduced color vibrancy, especially in computer-generated graphics or content with sharp, saturated color transitions.

B. Frame Partitioning: From Macroblocks to Coding Tree Units (CTUs)

After pre-processing, individual video frames are spatially divided into smaller, non-overlapping regions or blocks for subsequent processing steps like prediction,

transform, and quantization.[11] This block-based processing is a fundamental characteristic of most video compression standards.

- Macroblocks (Used in older standards like MPEG-2, H.261, H.263, and H.264/AVC): Traditionally, the basic processing unit was the macroblock, typically defined as a 16x16 array of luma pixels and corresponding chroma pixels (e.g., two 8x8 chroma blocks for 4:2:0 subsampling).10 In H.264/AVC, these 16x16 macroblocks could be further partitioned into smaller block sizes for motion estimation and transform coding, such as 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4 pixels.11 This variable block-size partitioning allowed the encoder to adapt to the local characteristics of the image content, using smaller blocks for areas with fine detail or complex motion and larger blocks for more uniform regions. Each macroblock is generally encoded separately, which facilitates parallel processing.14
- Coding Tree Units (CTUs) (Used in newer standards like H.265/HEVC and VVC): H.265/HEVC introduced the concept of the Coding Tree Unit (CTU), replacing the fixed-size macroblock as the fundamental processing unit.19 CTUs can be significantly larger than traditional macroblocks, with configurable sizes typically ranging from 16x16 up to 64x64 pixels (and even larger in VVC, e.g., 128x128). CTUs possess a hierarchical structure:
    1. A picture is first divided into non-overlapping CTUs.
    2. Each CTU can then be recursively partitioned into smaller square regions called Coding Units (CUs) using a quadtree structure. This means a CU can be split into four smaller CUs of half its width and height. This recursive splitting allows CUs to vary in size within a CTU, adapting to local image complexity.
    3. Each CU is then further partitioned into one or more Prediction Units (PUs). PUs define how prediction (intra or inter) is performed for that CU. PU shapes can be square or rectangular (e.g., symmetric and asymmetric partitions) to better match object boundaries or motion patterns.[19]
    4. Finally, for coding the residual signal (after prediction), each CU is partitioned into Transform Units (TUs), also using a quadtree-like structure. TUs are the blocks to which the discrete cosine transform (or a similar transform) and quantization are applied.[19]

The primary advantage of CTUs and their flexible partitioning scheme is improved compression efficiency, especially for high-resolution video (e.g., 4K, 8K). Larger block sizes (like 64x64 CUs) can more efficiently represent large, homogeneous areas of a frame, reducing the overhead associated with signaling block information. The adaptive quadtree-based partitioning allows the codec to allocate more bits and finer processing to complex regions while using fewer bits for simpler regions.[19]

Table 3: Evolution of Frame Partitioning Units

Standard	Primary Coding Unit Name	Typical/Max Luma Size(s) (pixels)	Key Partitioning Features
MPEG-2	Macroblock	16x16	Fixed size; limited partitioning for motion compensation (e.g., field/frame).
H.264/AVC	Macroblock	16x16	Partitions into 16x8, 8x16, 8x8 for ME; 8x8 and 4x4 sub-partitions for transform.
H.265/HEVC	Coding Tree Unit (CTU)	Up to 64x64 (e.g., 64x64, 32x32, 16x16)	Quadtree partitioning into CUs (down to 8x8); CUs into PUs (various symmetric/asymmetric shapes) and TUs (quadtree).
VP9	Superblock	Up to 64x64	Recursive partitioning from superblocks into smaller blocks (e.g., 64x64 down to 4x4) for prediction and transform.
AV1	Superblock	Up to 128x128	Highly flexible multi-type tree partitioning, allowing square and rectangular blocks of various sizes (4x4 to 128x128).
VVC (H.266)	Coding Tree Unit (CTU)	Up to 128x128	Quadtree plus Multi-Type Tree (QTMT) partitioning, allowing binary and

			ternary splits, more rectangular shapes.

This evolution towards larger and more flexible partitioning structures is a key factor in the enhanced compression performance of newer video codecs, enabling them to adapt more effectively to diverse video content and higher resolutions.

C. Prediction Mechanisms

Prediction is a cornerstone of video compression, aiming to reduce redundancy by creating an estimate of the current block to be coded. Only the difference between the actual block and its prediction (the residual) needs to be explicitly coded, which typically requires far fewer bits than coding the original block.

1. Intra-frame Prediction

Intra-frame prediction, or simply intra-prediction, reduces spatial redundancy by predicting the pixel values of a current block using information from already coded and reconstructed neighboring pixels within the *same* frame.[11] It does not rely on any information from other frames.

- **Mechanism:** The encoder selects one of several predefined prediction modes. Each mode specifies a direction or method for extrapolating pixel values from adjacent (top and/or left) reconstructed blocks to form a prediction for the current block.
- **H.264/AVC Example:** For luma blocks, H.264 supports 9 prediction modes for 4x4 and 8x8 blocks (including vertical, horizontal, DC (average), and 6 diagonal modes) and 4 prediction modes for 16x16 blocks (vertical, horizontal, DC, planar).[17] Chroma blocks typically have a similar, often simpler, set of prediction modes. The encoder usually tries all available modes and selects the one that minimizes the energy of the prediction residual (after subtraction from the original block), often using a rate-distortion optimization criterion.
- **HEVC and VVC Enhancements:** Newer standards like HEVC and VVC significantly increase the number of intra-prediction modes (e.g., HEVC supports up to 35 modes for luma, including planar, DC, and 33 angular directions) and refine the prediction process.[20] This allows for more accurate predictions, especially for textured regions and edges, leading to smaller residuals and better compression. VVC further extends this with techniques like Wide-Angle Intra Prediction (WAIP) and Matrix-based Intra Prediction (MIP).

2. Inter-frame Prediction: Motion Estimation and Motion Compensation (MEMC)

Inter-frame prediction, or inter-prediction, is designed to exploit temporal redundancy between frames.[9] It predicts a block in the current frame (the *target* frame) by finding a similar block in one or more previously coded and reconstructed frames (the *reference* frames).

- **Motion Estimation (ME):** This is the process performed by the encoder to find the best-matching block in the reference frame(s) for each block in the current frame being coded. The search is typically conducted within a defined search window in the reference frame(s). The displacement between the coordinates of the current block and its best match in the reference frame is represented by a **motion vector (MV)**.[9] The "best match" is usually determined by a cost function, such as Sum of Absolute Differences (SAD) or Sum of Squared Differences (SSD), often as part of a rate-distortion optimization process.
- **Motion Compensation (MC):** Once the motion vector(s) and reference frame(s) are determined by the encoder, this information is transmitted to the decoder. The decoder uses the received MV(s) to fetch the corresponding block(s) from its stored reference frame(s) and form a predicted block for the current position.[10] The encoder calculates the **prediction residual** (the pixel-wise difference between the original current block and its motion-compensated prediction). This residual is what gets transformed, quantized, and entropy coded.
- **Reference Frames:**
    - **P-frames (Predictive-coded frames):** Blocks in P-frames are predicted from one or more previously decoded I-frames or P-frames (uni-directional prediction).
    - **B-frames (Bi-predictive-coded frames):** Blocks in B-frames can be predicted from one or more past frames, one or more future frames, or an average/weighted average of predictions from both past and future frames (bi-directional prediction).[11] B-frames often achieve the highest compression efficiency because they have more reference options.
- **Advanced Techniques in Modern Codecs:**
    - **Sub-pixel Motion Vectors:** MVs can point to positions between pixels in the reference frame (e.g., half-pixel or quarter-pixel accuracy).[11] This requires interpolation of pixel values in the reference frame to generate the predicted block, but it allows for more precise motion representation and better prediction accuracy.
    - **Multiple Reference Frames:** Encoders can choose from a list of several previously decoded frames to find the best match, improving robustness to occlusions or complex motion.
    - **Variable Block Sizes for Motion:** As with partitioning, motion estimation and

compensation can be performed on various block sizes (e.g., H.264 allows partitioning a 16x16 macroblock into smaller units for ME).[11] This allows the codec to adapt to different scales of motion. HEVC and VVC extend this with more flexible PU sizes.
    - **Advanced Motion Vector Prediction (AMVP) and Merge Mode (HEVC/VVC):** These techniques improve the efficiency of coding motion vectors by predicting the current MV from spatially or temporally neighboring MVs, or by directly inheriting MVs from neighbors.
    - **Affine Motion Compensation (VVC, AV1):** Allows for more complex motion models than simple translation, such as rotation and scaling, which can better represent certain types of motion (e.g., zooming, rotating objects).

The accuracy and efficiency of these prediction mechanisms are critical to the overall performance of a video codec. Better predictions lead to lower-energy residuals, which in turn require fewer bits to encode, resulting in higher compression efficiency.

D. Transform Coding: Discrete Cosine Transform (DCT) and its Variants

After prediction (either intra or inter), the resulting data (original pixel block for intra-coding, or residual block for inter-coding) undergoes a transform coding stage. The primary purpose of transform coding is to decorrelate the pixel data and compact its energy into a smaller number of coefficients, making it more suitable for quantization and entropy coding.[9]

- **Discrete Cosine Transform (DCT):** The DCT is the most widely used transform in video compression standards. It converts a block of spatial-domain pixel values (or residual values) into a block of frequency-domain coefficients of the same size.[22]
    - **Mechanism:** The DCT decomposes the input block (e.g., 8x8 or 4x4 pixels) into a weighted sum of basis functions, which are cosine functions of different spatial frequencies. The output is a set of DCT coefficients, each representing the amplitude or contribution of a specific basis function (frequency component) to the original block.[22]
    - **Energy Compaction Property:** For typical image or residual blocks, which often exhibit strong spatial correlation, the DCT has excellent energy compaction properties. This means that most of the signal's energy tends to be concentrated in a few low-frequency coefficients (particularly the DC coefficient, representing the average value, and nearby AC coefficients).[23] High-frequency coefficients, which represent finer details or noise, generally have smaller magnitudes.
    - **Benefit for Compression:** This concentration of energy is highly beneficial

for compression. Many of the high-frequency coefficients can be quantized more coarsely (or even to zero) with minimal perceptual impact on the reconstructed image, as the HVS is less sensitive to these high-frequency components.[23]

- **Transform Block Sizes:**
  - Older standards like MPEG-2 and early H.264 primarily used an 8x8 DCT.
  - H.264/AVC introduced a 4x4 integer transform (an approximation of DCT) and an optional 8x8 integer transform for some profiles. Integer transforms are used to avoid potential mismatches between floating-point implementations in encoders and decoders.[18]
  - H.265/HEVC supports larger transform unit (TU) sizes, including 4x4, 8x8, 16x16, and 32x32 integer transforms. Larger transforms are generally more effective at energy compaction for larger, flatter regions of an image, which are common in high-resolution video.[20]
  - VVC further extends the range of transform sizes and introduces more specialized transforms.
  - AV1 also uses DCT-based transforms of various sizes (up to 64x64) and can even use an identity transform (no transform) for certain blocks. It also incorporates asymmetric DCTs (e.g., 16x32, 32x8).
- **Other Transforms:** While DCT and its integer approximations are dominant, some codecs or specific modes might use other transforms. For example, VP9 was mentioned as using Discrete Wavelet Transform (DWT) in [29], although DCT is its primary transform; DWT is more commonly associated with standards like JPEG 2000. Some newer research explores learned transforms using neural networks.

The output of the transform stage is a block of coefficients that, due to energy compaction, is typically easier to compress further in the subsequent quantization and entropy coding stages.

E. Quantization: The Art of Information Reduction

Quantization is a critical step in the video encoding pipeline, and it is the primary stage where information is irreversibly lost in most lossy video compression schemes.[9] Its main purpose is to reduce the precision of the transform coefficients generated by the DCT (or other transform) stage, thereby significantly reducing the amount of data that needs to be entropy coded and transmitted.

- **Mechanism:** Each transform coefficient in a block is divided by a corresponding **quantization step size** (also known as a quantizer value), and the result is typically rounded to the nearest integer.[24] The quantization step sizes are often derived from a **Quantization Parameter (QP)**. A single QP value can be used to

scale a default quantization matrix, or different step sizes can be applied to different frequency coefficients (e.g., using larger step sizes for high-frequency coefficients, to which the HVS is less sensitive). The formula is generally: Qcoeff=round(DCTcoeff/QPstep_size)

- **Impact of Quantization Parameter (QP):**
  - **Larger QP / Larger Step Size:** Results in more aggressive quantization. More coefficients will be rounded to smaller integer values, and many will be rounded to zero. This leads to higher compression (fewer bits needed to represent the coefficients) but also greater information loss and thus lower reconstructed video quality.[24]
  - **Smaller QP / Smaller Step Size:** Results in finer quantization. Coefficients are represented with more precision, leading to less information loss and better video quality, but at the cost of lower compression (larger file size/higher bitrate).[24]
- **Lossy Nature:** Because of the division and rounding, the original transform coefficient values cannot be perfectly recovered from the quantized values during decoding. This is why quantization is the main source of loss in lossy codecs.
- **Adaptive Quantization:** Modern codecs often employ adaptive quantization strategies. This means the QP or the quantization matrix can be varied for different parts of the video:
  - **Frequency-dependent quantization:** Different quantization step sizes can be used for different DCT coefficients within a block. Typically, high-frequency coefficients are quantized more coarsely than low-frequency coefficients.
  - **Spatially adaptive quantization:** Different QPs can be applied to different blocks or regions within a frame based on their visual importance or complexity. For example, visually important regions (like faces) might be quantized more finely (lower QP) than less important background regions.
- **Rate Control:** The QP is a key parameter used by encoders in rate control algorithms. By adjusting the QP, the encoder can try to achieve a target bitrate for the compressed video stream while balancing the resulting video quality.
- **The Trade-off:** Quantization directly embodies the fundamental trade-off in lossy compression: higher compression (lower bitrate) versus higher quality (lower distortion).[24] The choice of quantization strategy and QP values is crucial for achieving the desired balance for a given application.

After quantization, many of the transform coefficients, especially the high-frequency ones, will be zero or small integers, making the data highly suitable for efficient representation by entropy coding techniques.

F. Entropy Coding: Efficient Data Representation

Entropy coding is the final stage in the main encoding path for video data, applied after the transform coefficients have been quantized and other syntax elements (like prediction modes, motion vectors) have been determined. It is a form of lossless compression that further reduces the size of the data to be transmitted or stored, by representing the symbols (quantized coefficients, MVs, etc.) more efficiently based on their statistical properties.[4]

- **Purpose:** To assign shorter binary codes to symbols that occur more frequently and longer codes to symbols that occur less frequently, thereby minimizing the average number of bits required to represent the data.[9]
- **Mechanism:** Entropy coders analyze the probability distribution of the input symbols.
- **Common Techniques:**
  1. **Variable-Length Coding (VLC):** This is a general category of codes where different symbols are mapped to codewords of different lengths.
     - **Huffman Coding:** A well-known VLC method that constructs an optimal prefix code based on the estimated probabilities of symbols.[9] It is used in some older codecs or for certain syntax elements.
  2. **Arithmetic Coding:** Instead of assigning a specific integer number of bits to each symbol, arithmetic coding represents an entire sequence of symbols as a single floating-point number within the range [0, 1). It can achieve compression ratios closer to the theoretical limit (entropy) than Huffman coding, especially when symbol probabilities are not close to powers of two or when dealing with adaptive models.[27]
  3. **Context-Adaptive Schemes:** The efficiency of entropy coding can be significantly improved if the probability models adapt to the local statistics of the data being coded.
     - **Context-Adaptive Variable Length Coding (CAVLC):** Used in the Baseline profile of H.264/AVC for coding quantized transform coefficients.[11] CAVLC selects different VLC tables based on the context, which is determined by the values of previously coded neighboring coefficients. This adaptation allows it to better match the changing statistics of the coefficients.
     - **Context-Adaptive Binary Arithmetic Coding (CABAC):** A more advanced and generally more efficient entropy coding method used in the Main and High profiles of H.264/AVC, and as the primary entropy coding method in H.265/HEVC and VVC.[11] CABAC typically provides a bitrate reduction of 5-15% over CAVLC for the same video quality, but at the cost

of increased computational complexity.

- **CABAC Process** [26]**:**
  1. **Binarization:** Non-binary syntax elements (e.g., transform coefficients, motion vector differences) are first converted into a sequence of binary symbols (bins).
  2. **Context Model Selection:** For each bin, a context model is selected from a set of available models. The context model stores the probability of the bin being a '1' or a '0'. The selection is based on the values of previously coded syntax elements (the "context"), allowing the probability estimation to adapt to local statistics.
  3. **Arithmetic Encoding:** Each bin is then encoded by a binary arithmetic coder using the probability estimate provided by the selected context model.
  4. **Probability Update:** After encoding a bin, the statistics of the selected context model are updated based on the actual value of the bin. This makes the probability model adaptive.

The context modeling in CABAC is a key reason for its superior compression performance, as it allows the coder to exploit higher-order statistical dependencies in the video data.[26] The output of the entropy coding stage is the final compressed bitstream that represents the video.

G. In-Loop Filtering (e.g., Deblocking Filter, SAO)

Block-based video coding, particularly with quantization, can introduce visible artifacts in the reconstructed video, most notably "blocking artifacts" which appear as artificial discontinuities or edges along the boundaries of coding blocks. To mitigate these artifacts and improve overall visual quality, modern video codecs incorporate in-loop filters. These filters are applied *within* the prediction loop, meaning the filtered, cleaner reconstructed frames are stored in the Decoded Picture Buffer (DPB) and can be used as reference frames for the prediction of subsequent frames. This not only enhances the visual quality of the output video but can also improve compression efficiency by providing higher-quality reference pictures.

- **Deblocking Filter (e.g., in H.264, HEVC, VVC):**
  - **Purpose:** To smooth the sharp edges that can form between adjacent transform blocks due to independent quantization of their coefficients.[17]
  - **Mechanism:** The deblocking filter adaptively processes pixels on and near block boundaries. The strength and mode of filtering are typically determined based on factors like the quantization parameter (QP) used for the blocks, the coding mode, the presence of motion, and the magnitude of pixel differences

across the boundary. Stronger filtering is applied to more noticeable artifacts.
    - **Benefits:** Reduces visible blocking artifacts, leading to a subjectively more pleasant viewing experience. By improving the quality of reference frames, it can also lead to slightly better prediction accuracy for subsequent frames, contributing to overall compression gains.[18]
- **Sample Adaptive Offset (SAO) (e.g., in HEVC, VVC):**
    - **Purpose:** SAO is an additional in-loop filter, typically applied after the deblocking filter, designed to further reduce distortion and improve the fidelity of the reconstructed samples.[28] It aims to correct systematic errors or biases in the reconstructed pixel values.
    - **Mechanism:** SAO classifies reconstructed pixels into categories based on their characteristics (e.g., edge orientation/type, or pixel intensity band). For each category, an offset value is calculated and signaled by the encoder. The decoder adds this offset to the pixel values in that category to bring them closer to the original values. The offsets are determined by the encoder to minimize distortion.
    - **Benefits:** SAO can reduce various types of artifacts, including ringing and contouring, and generally improve the sharpness and detail of the reconstructed image.
- **Adaptive Loop Filter (ALF) (e.g., in VVC, and proposed/used in some AV1 development):**
    - ALF is a more complex filter that uses Wiener-filter-based techniques or other adaptive filter coefficients. The filter coefficients are designed by the encoder and signaled to the decoder. ALF can adapt its filtering characteristics on a region-by-region basis, providing more targeted artifact reduction.
- **Constrained Directional Enhancement Filter (CDEF) and Loop Restoration Filter (AV1):**
    - AV1 employs a sophisticated set of in-loop filtering tools. CDEF is a conditional directional filter designed to remove ringing and other artifacts around sharp edges without excessive blurring.[29] The Loop Restoration Filter includes options for Wiener filtering and self-guided filtering to further enhance the quality of reconstructed frames by reducing noise and other impairments.[29]

The inclusion and refinement of these in-loop filtering techniques represent a significant area of advancement in modern video codecs. They play a crucial role in achieving high subjective video quality, especially at lower bitrates where compression artifacts are more likely to occur.

The video encoding pipeline is a sophisticated cascade of interdependent stages. The efficiency of each stage profoundly influences the subsequent ones; for instance, superior prediction yields a smaller residual, simplifying the task for transform and quantization stages, ultimately requiring fewer bits post-entropy coding.[17] This interconnectedness means that codec enhancement is not about isolated tool improvement but about optimizing the entire chain and the synergistic interactions between its components. Rate-Distortion Optimization (RDO) is a common strategy employed by encoders to make informed decisions at various points in the pipeline—such as selecting prediction modes or motion vectors—by evaluating the impact on both the final bitrate and the perceived quality.

Comparing the evolution from older codecs like MPEG-2, with its simpler macroblock structures, to newer standards such as HEVC/VVC with their complex Coding Tree Unit (CTU) partitioning [19], expanded intra-prediction modes [17], advanced motion vector prediction, and sophisticated in-loop filters like SAO [28] or AV1's CDEF [29], reveals a clear trajectory. Newer codecs incorporate more granular tools and more adaptive decision-making processes. This escalating complexity allows for a more profound exploitation of data redundancies and a finer attunement to perceptual nuances, thereby achieving superior compression efficiency. However, this advancement inherently increases computational demands for both encoding and decoding, a critical trade-off in codec design. The gains in compression are realized by meticulously identifying and removing every possible bit of redundancy and by tailoring the compression process more precisely to the specific characteristics of the video content and the limitations of the human visual system.

A central theme in the efficiency battle is the "residual"—the signal remaining after prediction. The overarching goal of the prediction stages (both intra- and inter-frame) is to minimize the energy and information content of this residual signal.[17] A smaller, less complex residual means that the subsequent transform stage can more effectively compact its energy, and the quantization stage can represent it with fewer bits. Consequently, innovations in prediction methodologies—such as more accurate prediction modes or more sophisticated motion compensation techniques—are paramount. A better prediction directly translates to a simpler residual, making the tasks of transformation, quantization, and entropy coding more effective and less data-intensive. This explains the extensive research and development efforts consistently focused on enhancing prediction capabilities within new and evolving video codec standards.

IV. Video Codecs and Container Formats

Understanding the distinction between video codecs and video container formats is fundamental to comprehending how digital video files are structured, stored, and played back. While often used in proximity, they serve distinct and complementary roles in the digital video ecosystem.

A. Distinguishing Codecs from Container Formats: An Analogy and Technical Explanation

A **video codec** (coder-decoder) is an algorithm or a software/hardware implementation that is responsible for the actual **compression and decompression** of the video (and often audio) data streams.[1] The codec dictates *how* the raw pixel and audio sample data is transformed into a more compact representation to reduce file size, and how it is reconstructed back into a viewable form. Examples of video codecs include H.264/AVC, H.265/HEVC, VP9, and AV1.

A **video container format** (also known as a wrapper or multimedia container) is a file structure that **holds or packages** these compressed video and audio data streams. Beyond the media streams themselves, a container also typically encapsulates metadata, subtitles, chapter information, and synchronization data necessary for coherent playback.[1] The container defines *how* these various elements are organized and interleaved within a single file. Examples of container formats include MP4, MKV (Matroska), WebM, MOV (QuickTime), AVI, and TS (Transport Stream).

An effective analogy helps clarify this distinction 4:
Imagine sending a gift. The codec is akin to the method used to carefully pack the gift items into the smallest possible box without damaging them – perhaps by disassembling an item, using bubble wrap efficiently, or vacuum-sealing clothes. It's the "recipe and cooking process" that makes the contents compact.
The container format is the actual physical box or gift wrapping you put the packed items into. It holds everything together, has a label (filename and extension), and provides instructions for opening (how a player should read the file). Just as you might use a cardboard box, a padded envelope, or decorative gift wrap depending on the item and destination, different video containers serve different needs. The container is the "lunchbox" holding the prepared "meal."
Another way to conceptualize it is that the codec determines how the video and audio "stuff/data" is laid out or arranged in a compressed form, while the container is the "box" that holds this arranged data along with other necessary information.[30]

B. How Codecs and Containers Work in Unison

Codecs and container formats are not independent entities; they work synergistically to create a playable video file.[33]

1. **Encoding and Muxing:** During the creation of a video file, the raw video data is first processed by a video codec to produce a compressed video stream (often called an elementary stream). Similarly, raw audio data is processed by an audio codec to produce a compressed audio stream. These compressed elementary streams, along with any subtitles, metadata (like title, artist, resolution, frame rate), and chapter information, are then **multiplexed** (or "muxed") together and encapsulated into a single file according to the specifications of the chosen container format.[33] The container provides the structure for interleaving these different data types and includes timing information to ensure they can be synchronized during playback.
2. Demuxing and Decoding: When a user wants to play the video file, a media player application first interacts with the container format. The player demultiplexes (or "demuxes") the container to separate the individual elementary streams (video, audio, subtitles) and extract the metadata.31 The metadata within the container usually includes information about which specific codecs were used to compress the video and audio streams.
   Based on this codec information, the media player then invokes the appropriate video decoder to decompress the video stream and the appropriate audio decoder to decompress the audio stream(s). These decompressed streams are then rendered and played back in synchronization, ideally providing a seamless viewing experience.

The container format must effectively signal which codecs are used for the enclosed streams so that the playback device or software can select and utilize the correct decoders.[32] While a single container format can typically support multiple different video and audio codecs, not all playback platforms (devices or software players) support every possible combination of container and codec.[1] This compatibility aspect is a crucial consideration in video distribution.

C. Overview of Common Video Container Formats (MP4, MKV, WebM, MOV, AVI, TS)

Several container formats are prevalent in digital video, each with its own characteristics, strengths, and common use cases.

- **MP4 (MPEG-4 Part 14):**
  - **Overview:** One of the most widely supported and versatile container formats, MP4 is ubiquitous across a vast range of devices, web browsers, and software platforms. It is often the default choice for online video streaming and distribution.[1]
  - **Origins:** MP4 is based on Apple's QuickTime File Format (MOV) and is part of

the MPEG-4 standard (ISO/IEC 14496-14).[36]
    - **Supported Codecs:** It commonly contains video encoded with H.264/AVC or H.265/HEVC, and increasingly AV1 or VP9. For audio, AAC is a frequent companion, but MP3 and FLAC are also supported.[1]
    - **Features:** Well-suited for streaming due to features like "hint tracks" for network delivery and robust support for metadata, subtitles, and chapter markers. It is also a common format for adaptive bitrate streaming segments (e.g., in DASH and HLS, often as fragmented MP4 or fMP4).[33]
    - **Licensing:** May involve licensing considerations for certain uses, as it is part of the MPEG standards.[33]
- **MKV (Matroska):**
    - **Overview:** An open-standard, royalty-free container format known for its extreme versatility and rich feature set.[2]
    - **Supported Codecs:** MKV is designed to be a universal container and can hold virtually any type of video codec (e.g., H.264, HEVC, AV1, VP9, DivX, Xvid) and audio codec (e.g., AAC, MP3, DTS, Dolby TrueHD, FLAC, Opus).[33]
    - **Features:** Excels at storing multiple video and audio tracks (e.g., for different languages or commentaries), multiple subtitle tracks (including advanced formats like ASS/SSA), chapter points, extensive metadata, and even attachments.
    - **Use Cases:** Popular for storing high-definition movies and TV shows, especially for personal media libraries, due to its ability to package a complete viewing experience. While highly capable, native playback support might be less universal than MP4 on some consumer devices without third-party software.[33]
- **WebM:**
    - **Overview:** An open, royalty-free container format specifically designed for use with HTML5 video on the web. It is based on a profile of the Matroska (MKV) container format.[2]
    - **Supported Codecs:** Primarily uses VP8, VP9, or AV1 video codecs, and Vorbis or Opus audio codecs, all of which are also open and royalty-free.[32]
    - **Features:** Optimized for streaming, with low computational overhead for playback.
    - **Use Cases:** Intended for web-based video delivery. Supported by most modern web browsers like Chrome, Firefox, Edge, and Opera.[36]
- **MOV (QuickTime File Format):**
    - **Overview:** Developed by Apple Inc., MOV is the native container format for the QuickTime framework. It enjoys excellent support within the Apple ecosystem (macOS, iOS).[1]

    - **Supported Codecs:** Can contain a wide variety of codecs, including Apple's own ProRes (popular in professional editing), H.264, H.265, as well as various audio formats like AAC and PCM.
    - **Features:** Robust metadata support, suitable for editing workflows.
    - **Use Cases:** Commonly used in professional video production and editing environments, particularly those centered around Apple software and hardware. Also used for distributing video content for Apple devices.[33]
- **AVI (Audio Video Interleave):**
    - **Overview:** An older container format developed by Microsoft in the early 1990s.[31]
    - **Supported Codecs:** Can contain video encoded with various codecs, but is often associated with older codecs like DivX, Xvid, and uncompressed video.
    - **Features:** Was once a de facto standard on Windows PCs but has significant limitations compared to modern containers, such as poor support for modern codecs (e.g., H.265, AV1), limited support for variable bitrate (VBR) audio, lack of native support for aspect ratio information, and poor streaming capabilities.[31]
    - **Use Cases:** Largely superseded by MP4 and MKV for most modern applications, but may still be encountered with older video files or in specific legacy workflows.
- **TS (MPEG Transport Stream):**
    - **Overview:** A standard container format specified in MPEG-2 Part 1, designed primarily for digital broadcasting (e.g., DVB, ATSC) and for storage on Blu-ray discs (often with the.m2ts extension).[1]
    - **Supported Codecs:** Typically contains video encoded with MPEG-2, H.264/AVC, or H.265/HEVC, and audio codecs like AC-3 (Dolby Digital), AAC, or MPEG audio layers.
    - **Features:** Characterized by its use of fixed-size (188-byte) packets, which makes it robust against data loss and suitable for transmission over error-prone channels. Includes features for multiplexing multiple programs, synchronization, and program-specific information (PSI) tables.
    - **Use Cases:** Digital television broadcasting (terrestrial, satellite, cable), IPTV, and Blu-ray Discs.

The choice of container format is often a pragmatic decision, balancing the need for rich features and broad codec support against the requirement for widespread compatibility across target playback devices and platforms. For instance, while MKV offers unparalleled versatility for archival and feature-rich local playback, MP4 often prevails for broader distribution due to its near-universal device and browser

compatibility, even if it presents slightly more restrictions in codec support or metadata features compared to MKV. This practical consideration highlights that container selection is not solely based on technical prowess but also on the intended audience and delivery ecosystem.

Furthermore, the evolution of container formats mirrors the changing patterns of media consumption. Older formats like AVI were adequate for local playback of standard-definition content. However, the ascent of internet streaming and high-definition/ultra-high-definition content necessitated the development and adoption of containers like MP4 and WebM, which are optimized for streaming functionalities such as adaptive bitrate delivery and possess lower overhead.[33] Similarly, Transport Streams (TS) were engineered for broadcast environments where error resilience is a paramount concern.[37] As new consumption paradigms emerge, such as interactive video or virtual reality experiences, container formats are likely to continue evolving, or entirely new formats may be developed, to cater to the specific metadata, synchronization, and streaming demands these advanced applications entail. The Common Media Application Format (CMAF), built upon the ISO Base Media File Format (ISOBMFF, which also forms the basis of MP4), exemplifies ongoing standardization efforts aimed at simplifying and unifying streaming workflows across different platforms.[38]

D. Metadata in Container Formats

Beyond encapsulating the primary audio and video data streams, container formats play a crucial role in storing a wide array of **metadata**.[1] Metadata, in this context, is "data about data"—information that describes the content, structure, and playback characteristics of the media file.

The types of metadata commonly found in video containers include [33]:

- **Descriptive Metadata:** Information that describes the content itself, such as the title of the video, episode name, series title, director, actors, genre, summary or plot description, creation date, copyright notices, and album/artist information for music videos.
- **Technical Metadata:** Details about the encoded streams, essential for decoders and players. This includes:
    - Video codec used (e.g., H.264, HEVC, AV1)
    - Video resolution (width and height)
    - Frame rate
    - Aspect ratio (pixel aspect ratio and display aspect ratio)
    - Color space (e.g., Rec.709, Rec.2020)

    - Bit depth (e.g., 8-bit, 10-bit)
    - Bitrate (average or variable)
    - Audio codec used (e.g., AAC, AC3, Opus)
    - Number of audio channels (e.g., stereo, 5.1 surround)
    - Audio sample rate
    - Duration of the media.
- **Structural Metadata:** Information that defines the internal structure or navigation of the content, such as:
    - Chapter markers or cue points, allowing users to jump to specific sections.
    - Playlists or edit decision lists (EDLs) in some professional formats.
- **Synchronization Metadata:** Timestamps (e.g., Presentation Time Stamps - PTS, Decode Time Stamps - DTS) associated with video frames and audio samples. This is critical for ensuring that audio and video are played back in sync, and for synchronizing subtitles with the dialogue.
- **Accessibility Metadata:** Information to support accessibility features, most notably:
    - Subtitle tracks (in various formats like SRT, WebVTT, ASS/SSA) for different languages or for the hard of hearing (SDH).
    - Closed captions (e.g., CEA-608/708 embedded data).
    - Audio descriptions for the visually impaired.
- **Dynamic and Streaming-Related Metadata:**
    - **In-band events:** Such as emsg (event message) boxes in fragmented MP4 (fMP4) used in DASH/HLS streaming. These are timed metadata events synchronized with the media timeline, often used to trigger client-side actions like ad insertion, display of interactive overlays, or dynamic content replacement.[38]
    - Information for Digital Rights Management (DRM) systems.
    - Hint tracks for RTP streaming (less common in modern HTTP-based streaming).

The importance of metadata cannot be understated. It is essential for media players to correctly interpret, decode, and render the video and audio content. It enables features like seeking to specific points in the video, selecting different audio tracks or subtitle languages, and displaying informative details about the media. In the context of streaming, metadata is vital for adaptive bitrate (ABR) switching, content protection, and dynamic ad insertion. The richness of metadata support within a container format can therefore significantly influence the user experience, the accessibility of the content, and the monetization strategies available to content providers. As video services evolve towards greater personalization and interactivity,

the complexity and significance of metadata managed by container formats are poised to increase further.

Table 4: Common Video Container Formats: Features and Codec Support

Container Format	Key Features	Commonly Supported Video Codecs	Commonly Supported Audio Codecs	Typical Use Cases
MP4 (.mp4)	Widely compatible, streaming optimized (fMP4), good metadata/subtitle support, part of MPEG standards.	H.264/AVC, H.265/HEVC, AV1, VP9, MPEG-4 Part 2	AAC, MP3, AC-3, Opus, FLAC	Web streaming (HTML5, DASH, HLS), mobile devices, general distribution, digital downloads.
MKV (.mkv)	Open standard, royalty-free, highly versatile, multiple A/V/subtitle tracks, chapters, extensive metadata.	Virtually any (H.264, HEVC, AV1, VP9, Xvid, DivX, ProRes etc.)	Virtually any (AAC, MP3, AC-3, DTS, TrueHD, FLAC, Opus)	Storing movies/TV shows, personal media libraries, high-fidelity archives.
WebM (.webm)	Open standard, royalty-free, designed for HTML5 web video, based on Matroska.	VP8, VP9, AV1	Vorbis, Opus	Web video streaming, online multimedia.
MOV (.mov)	Apple QuickTime format, excellent in Apple ecosystem, strong metadata, good for editing.	ProRes, H.264/AVC, H.265/HEVC, MPEG-4 Part 2	AAC, PCM, MP3	Professional video editing, Apple device distribution, multimedia applications.

AVI (.avi)	Older Microsoft format, once widespread on Windows, limited modern features.	DivX, Xvid, MJPEG, older codecs, some H.264 (less common)	MP3, PCM, AC-3	Legacy video files, some specific older workflows.
TS (.ts,.m2ts)	MPEG Transport Stream, designed for broadcast, error resilience, fixed-size packets, multiple program muxing.	MPEG-2, H.264/AVC, H.265/HEVC	MPEG Audio (Layer I/II), AC-3, AAC	Digital TV broadcast (DVB, ATSC), Blu-ray Discs, IPTV.
3GP (.3gp)	Derived from MP4, optimized for mobile networks, lower bandwidth scenarios.	H.263, MPEG-4 Part 2, H.264/AVC, VP8	AMR-NB/WB, AAC-LC, HE-AAC	Multimedia messaging (MMS), older mobile device video.
Ogg (.ogv)	Open standard, royalty-free, Xiph.Org Foundation, often with Theora video.	Theora, VP8, VP9	Vorbis, Opus, FLAC	Open-source multimedia projects, some web use (less common than WebM).

Sources: [1]

V. Evolution and Key Standards in Video Codecs

The history of video codecs is a narrative of continuous innovation, driven by the relentless demand for higher video quality, increased resolutions, and more efficient use of storage and bandwidth. Each generation of codecs has built upon the successes and addressed the limitations of its predecessors, leading to the sophisticated compression technologies available today.

A. Historical Milestones in Video Compression

The journey of video compression spans several decades, marked by pivotal standards and technological breakthroughs:

- **Pre-1970s (Analog Era):** Prior to widespread digital technology, video was predominantly stored and transmitted in analog formats on magnetic tape.[41]
- **1974 - Discrete Cosine Transform (DCT) Conceptualized:** A seminal moment occurred when Nasir Ahmed, T. Natarajan, and K. R. Rao introduced the Discrete Cosine Transform (DCT).[41] This mathematical technique for converting signals into frequency components would become a cornerstone of nearly all subsequent digital video (and image) compression standards.
- **1988 - H.261:** Developed by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), H.261 is widely regarded as the first practical digital video coding standard.[6] It was primarily designed for videoconferencing and videotelephony over ISDN networks. H.261 introduced fundamental concepts like the macroblock structure (16x16 pixels), motion compensation for inter-frame prediction, DCT for spatial redundancy reduction, quantization, and Huffman coding for entropy coding. It typically supported resolutions like CIF (352x288 pixels) at frame rates up to 30 fps.[6]
- **1993 - MPEG-1:** The Moving Picture Experts Group (MPEG), a working group of ISO/IEC, released the MPEG-1 standard (ISO/IEC 11172). It was famously designed for storing video on Video CDs (VCDs) at quality comparable to VHS tapes, typically at resolutions like SIF (352x240 pixels for NTSC, 352x288 for PAL) at 30/25 fps.[6] MPEG-1 Part 3 (Audio Layer III), or MP3, became a revolutionary audio compression format.
- **1994-1996 - MPEG-2 (H.262):** A collaborative effort between MPEG and ITU-T (where it was standardized as H.262), MPEG-2 (ISO/IEC 13818) represented a significant advancement.[6] It became the dominant standard for digital television broadcasting (e.g., DVB, ATSC), DVD-Video, and early standard-definition digital video. Key improvements over MPEG-1 included support for interlaced video (essential for broadcast TV), higher resolutions (up to Main Level @ High Profile for HDTV), and more flexible bitrate allocation. It introduced program streams and transport streams for different applications.[37]
- **1999 - MPEG-4 Part 2 (Advanced Simple Profile):** MPEG-4 Part 2 (ISO/IEC 14496-2), often referred to as MPEG-4 Visual, offered improved compression efficiency over MPEG-2.[41] It introduced tools like global motion compensation, quarter-pixel (QPel) motion estimation, and support for object-based coding. It saw use in some early web video applications, portable media players, and DivX/Xvid codecs are based on it. It is distinct from, and often confused with, MPEG-4 Part 10 (H.264/AVC).

- **2003 - H.264/AVC (MPEG-4 Part 10, Advanced Video Coding):** This standard, jointly developed by the ITU-T's Video Coding Experts Group (VCEG) and ISO/IEC MPEG as the Joint Video Team (JVT), marked a watershed moment in video compression.[11] H.264/AVC achieved a dramatic improvement in compression efficiency, typically offering about a 50% bitrate reduction compared to MPEG-2 for the same perceptual quality, or significantly better quality at the same bitrate. Its advanced features (detailed in section V.C) led to its widespread adoption across a vast range of applications, including Blu-ray Discs, HDTV broadcasting, internet streaming services, mobile video, and digital cameras.
- **2010 - VP8:** Following Google's acquisition of On2 Technologies, VP8 was released as an open-source, royalty-free video codec, positioned as an alternative to the patent-encumbered H.264.[1] It formed the video basis for the WebM project.
- **2012-2013 - H.265/HEVC (High Efficiency Video Coding, MPEG-H Part 2):** Developed as the successor to H.264/AVC by the Joint Collaborative Team on Video Coding (JCT-VC), a collaboration between ITU-T VCEG and ISO/IEC MPEG.[1] HEVC aimed for another ~50% improvement in compression efficiency over H.264, making it suitable for emerging 4K and 8K Ultra High Definition (UHD) resolutions, as well as High Dynamic Range (HDR) video. While technically superior, its adoption in some sectors was complicated by a more complex and costly patent licensing landscape.[41]
- **2013 - VP9:** Google's successor to VP8, VP9 was also released as an open-source, royalty-free codec.[1] It offered compression efficiency comparable to HEVC and became a key codec for YouTube, particularly for high-resolution content, and gained significant traction in web browsers and Android devices.
- **2018 - AV1 (AOMedia Video 1):** Developed by the Alliance for Open Media (AOMedia), a consortium of major technology companies (including Amazon, Apple, Google, Intel, Meta, Microsoft, Mozilla, Netflix, NVIDIA, Samsung).[2] AV1 was designed as a high-efficiency, royalty-free video codec intended to surpass both HEVC and VP9 in compression performance and to provide an open alternative for internet video and beyond.
- **2020 - VVC (Versatile Video Coding, H.266, MPEG-I Part 3):** The designated successor to HEVC, developed by the Joint Video Experts Team (JVET), the successor to JCT-VC.[42] VVC aims for a further 30-50% improvement in compression efficiency over HEVC, with enhanced support for a wide range of video types, including 8K, HDR, 360° video, and screen content.
- **Ongoing - AV2:** The Alliance for Open Media is currently developing AV2 as the successor to AV1, aiming for further improvements in compression efficiency and

features.[50]

This timeline illustrates a consistent trend: roughly every decade, a new generation of video codec emerges, offering approximately double the compression efficiency (or a 50% bitrate reduction for similar quality) compared to its mainstream predecessor. This progression has been vital in enabling higher resolutions, better quality, and more diverse video applications within the constraints of available storage and bandwidth.

Table 5: Timeline of Major Video Codec Standards

Year	Codec Standard	Key Innovations Introduced	Primary Applications/Impact
1988	H.261	Macroblocks, DCT, motion compensation, first practical digital video standard.	Videoconferencing over ISDN.
1993	MPEG-1	DCT, motion compensation, GOP structure, I/P/B frames.	Video CD (VCD), MP3 audio.
1994-1996	MPEG-2 (H.262)	Support for interlaced video, higher resolutions/bitrates, program/transport streams.	DVD-Video, digital TV broadcast (DVB, ATSC), Super Video CD (SVCD).
1999	MPEG-4 Part 2 (Visual)	Improved efficiency over MPEG-2, object-based coding, QPel motion compensation, global motion compensation.	Early web video, DivX/Xvid, some mobile applications.
2003	H.264/AVC (MPEG-4 Part 10)	Advanced intra-prediction, flexible macroblock partitioning, 1/4-pixel	Blu-ray, HDTV broadcast, streaming (YouTube, Netflix), mobile video, digital

		ME, in-loop deblocking filter, CABAC/CAVLC, multiple reference frames.	cameras. Ubiquitous standard.
2010	VP8	Open-source, royalty-free, block-based DCT, intra/inter prediction.	WebM project, early HTML5 video, WebRTC.
2012-2013	H.265/HEVC (MPEG-H Part 2)	Coding Tree Units (CTUs up to 64x64), advanced intra/inter prediction, SAO filter, parallel processing tools (tiles, WPP), Main 10 profile for HDR.	4K/8K UHD streaming, UHD Blu-ray, modern broadcast (ATSC 3.0), high-end mobile.
2013	VP9	Open-source, royalty-free, superblocks (up to 64x64), improved prediction, 10/12-bit support, HDR.	YouTube (primary HD/UHD codec), Chrome, Firefox, Android, WebRTC.
2018	AV1 (AOMedia Video 1)	Royalty-free, superblocks (up to 128x128), advanced prediction & filtering (CDEF, Loop Restoration), film grain synthesis, designed for internet video.	Major streaming services (Netflix, YouTube, Meta), web browsers, emerging hardware support.
2020	VVC (H.266 / MPEG-I Part 3)	QTMT partitioning, affine motion, adaptive loop filter (ALF), matrix-based intra prediction (MIP), further efficiency gains over HEVC.	Next-generation UHD (8K+), HDR, 360° video, screen content, broadcast (future DVB/ATSC updates).

Ongoing	AV2	Successor to AV1, aims for further compression improvements and feature enhancements.	Future internet video, streaming.

Sources: [6]

B. MPEG-2: The Standard for DVD and Early Digital Broadcast

MPEG-2, formally known as ISO/IEC 13818 and also standardized by the ITU-T as Recommendation H.222/H.262, was a pivotal standard in the transition from analog to digital video.[37] Released around 1995-1996, its primary objective was to provide a "generic coding of moving pictures and associated audio information," enabling the efficient storage and transmission of video content for applications like DVD-Video and digital television broadcasting.[37]

- Key Innovations over MPEG-1:
  MPEG-2 built upon the foundations of MPEG-1 but introduced several crucial enhancements:
  1. **Support for Interlaced Video:** Analog television systems predominantly used interlaced scanning. MPEG-2's native support for interlaced video was critical for its adoption in digital broadcast (DVB, ATSC) and for DVD content derived from broadcast sources.[37]
  2. **Higher Resolutions and Bitrates:** MPEG-2 supported a wider range of resolutions and bitrates, making it suitable for standard-definition television (SDTV) and early high-definition television (HDTV) applications.[58]
  3. **Stream Types:** It defined two main types of bitstreams:
     - **Program Streams (PS):** Designed for relatively error-free environments like storage media (e.g., DVD-Video VOB files). PS combines one or more packetized elementary streams (PES) with a common time base.
     - **Transport Streams (TS):** Designed for error-prone transmission environments like broadcasting. TS multiplexes one or more programs (each consisting of video, audio, and data PES packets) into fixed-size (188-byte) packets, incorporating timing information and error resilience mechanisms.[37] This is the format used in DVB and ATSC systems.
  4. **Profiles and Levels:** MPEG-2 defined various profiles (sets of coding tools) and levels (constraints on parameters like resolution, bitrate) to cater to different application requirements. For example, "Main Profile at Main Level" (MP@ML) was common for SDTV, while "Main Profile at High Level" (MP@HL)

was used for HDTV.

5. **Audio Enhancements:** MPEG-2 Part 3 (Audio) extended MPEG-1 audio capabilities, notably by allowing the coding of multi-channel audio programs (up to 5.1 surround sound) in a backward-compatible manner with MPEG-1 stereo decoders.[37] It also standardized Advanced Audio Coding (AAC) as a more efficient, non-backward-compatible audio codec option.[37]

- Video Coding Scheme (MPEG-2 Part 2, H.262):
  The video coding part of MPEG-2 employed a hybrid block-based DCT approach similar to MPEG-1 but with refinements:
  - **Macroblocks:** 16x16 luma blocks with corresponding chroma blocks.
  - **Prediction:** I-frames (intra-coded), P-frames (uni-directionally predicted), and B-frames (bi-directionally predicted).
  - **Motion Compensation:** Using motion vectors to predict blocks from reference frames. MPEG-2 allowed for frame-based or field-based motion prediction for interlaced content.
  - **DCT:** Typically 8x8 DCT applied to pixel blocks or prediction residuals.
  - **Quantization:** Using quantization matrices to control bit allocation.
  - **Entropy Coding:** Variable-length coding (Huffman coding). While effective for its time, MPEG-2 video was not highly optimized for very low bitrates (e.g., below 1 Mbps at standard definition) compared to later standards.[37]
- **Use Cases** [29]**:**
  - **DVD-Video:** MPEG-2 is the mandatory video codec for DVD-Video, with specific constraints on resolution (e.g., 720x480 for NTSC, 720x576 for PAL), aspect ratios, and GOP (Group of Pictures) structure.[37]
  - **Digital Television Broadcasting:** It was the foundational video codec for major digital broadcast standards worldwide, including DVB (Europe, Asia, etc.) and ATSC (North America, South Korea) for both SD and early HD transmissions.
  - **Super Video CD (SVCD):** An optical disc format that used MPEG-2 video.
  - **HDV:** Some early high-definition consumer and prosumer camcorder formats (like HDV) used MPEG-2 compression for HD video.
  - **Blu-ray Disc:** While H.264/AVC and VC-1 are more common for HD content on Blu-ray, MPEG-2 is also a supported codec and is sometimes used, especially for standard-definition bonus material or older content transfers.[59]
- Limitations:
  Compared to modern codecs like H.264/AVC and H.265/HEVC, MPEG-2 is significantly less efficient in terms of compression.37 It requires a higher bitrate to achieve the same visual quality. Its support for advanced features like flexible block partitioning and sophisticated entropy coding was limited. Nevertheless, its

widespread deployment in hardware and software for over a decade solidified its place in video history, and it remains in use in some legacy systems and broadcast applications.

C. H.264/AVC (MPEG-4 Part 10): The Ubiquitous Codec

H.264/AVC, also known as MPEG-4 Part 10, Advanced Video Coding, represents one of the most significant and widely adopted video compression standards in history. Standardized in 2003 through a joint effort by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) under the Joint Video Team (JVT), H.264/AVC was designed to provide substantially better compression efficiency than its predecessors like MPEG-2 and MPEG-4 Part 2, without an unmanageable increase in computational complexity.[11] Its success is evident in its pervasive use across nearly all digital video applications.

1. Technical Innovations and Encoder Architecture

The primary goal of H.264/AVC was to deliver good video quality at roughly half the bitrate required by MPEG-2, or significantly improved quality at similar bitrates.[61] This was achieved through a range of technical innovations built upon the established hybrid block-oriented, motion-compensated transform coding framework.[11]

The general H.264/AVC encoder architecture (illustrated in Fig.3 of 17) involves the following key processing steps for each macroblock:
Input Frame → Macroblock Partitioning → Prediction (Intra or Inter) → Residual Calculation (Original - Prediction) → Transform → Quantization → Entropy Coding → Bitstream.
A reconstruction loop within the encoder mirrors the decoder process: Inverse Quantization → Inverse Transform → Add Prediction → In-Loop Deblocking Filter → Store as Reference Frame.
Key technical features and innovations include [11]:

- **Flexible Macroblock Partitioning:** While the basic processing unit is a 16x16 luma macroblock, H.264 allows it to be partitioned into smaller sizes for motion estimation and compensation. These include 16x8, 8x16, and 8x8. An 8x8 partition can be further subdivided into 8x4, 4x8, or 4x4 blocks. This adaptability allows the encoder to better match the shapes and motion of objects, leading to more accurate prediction and smaller residuals.[11]
- **Advanced Intra Prediction:** For blocks coded without reference to other frames (I-slices or intra macroblocks in P/B-slices), H.264 provides a rich set of directional prediction modes. For 4x4 and 8x8 luma blocks, there are 9 modes (DC, vertical, horizontal, and 6 diagonal). For 16x16 luma blocks, there are 4 modes (DC, vertical, horizontal, planar). Chroma blocks also have prediction

modes, typically DC, horizontal, vertical, and planar. The encoder selects the mode that yields the best prediction (lowest residual energy).[11]

- **Quarter-Pixel Accurate Motion Compensation:** Motion vectors can specify motion with a precision of up to one-quarter of the distance between luma pixels.[11] This requires interpolation using multi-tap filters to generate the sub-pixel samples in the reference picture, but it significantly improves the accuracy of motion-compensated prediction, especially for slow-moving objects or fine details.
- **Multiple Reference Frames:** Encoders can select from a list of previously decoded frames (both past and, for B-slices, future in display order) as references for inter-prediction. This improves prediction robustness, especially in scenes with occlusions or repetitive motion, as a better match might be found in a frame other than the immediately preceding one.
- **Weighted Prediction:** Allows for changes in brightness and color between the current block and its reference, useful for fading effects or lighting changes.
- **Integer Transform:** H.264 uses 4x4 and (in High profiles) 8x8 integer transforms that are approximations of the DCT.[18] These transforms are precisely defined using integer arithmetic, which avoids potential mismatches between encoder and decoder implementations that could arise from differing floating-point precision.
- **In-Loop Deblocking Filter:** A crucial innovation was the inclusion of an adaptive deblocking filter applied *within* the prediction loop.[17] This filter smooths artifacts that appear at block boundaries due to quantization. By cleaning up the reconstructed frames before they are stored as references, the deblocking filter not only improves the visual quality of the output video but also enhances the efficiency of predicting subsequent frames. The filter strength is adapted based on coding parameters and image content.
- **Entropy Coding:** H.264 supports two main entropy coding methods:
    - **Context-Adaptive Variable Length Coding (CAVLC):** Used for transform coefficients in the Baseline and some other profiles. It is less complex than CABAC but also less efficient.[11]
    - **Context-Adaptive Binary Arithmetic Coding (CABAC):** Used in Main and High profiles. CABAC provides significantly better compression (typically 5-15% bitrate reduction over CAVLC for the same quality) by using adaptive probability models based on the context of previously coded symbols and employing efficient binary arithmetic coding.[11] It is, however, more computationally demanding.
- **Slice-based Coding:** Frames can be divided into one or more slices. Slices are independently decodable sequences of macroblocks (though inter-slice

prediction can occur unless restricted). Slices provide error resilience, as corruption in one slice does not necessarily affect others, and allow for parallel processing.[62]
- **Network Abstraction Layer (NAL):** The H.264 bitstream is organized into NAL units, which package the coded video data in a way that is suitable for transmission over various networks or storage in different file formats.

2. Profiles and Levels

H.264/AVC defines a set of **profiles** and **levels** to cater to different application requirements and device capabilities.[62]

- **Profiles:** A profile specifies a subset of the coding tools and algorithms available in the H.264 standard. Different profiles offer varying trade-offs between compression efficiency, computational complexity, and features. Key profiles include:
  - **Baseline Profile (BP):** Designed for low-cost applications with limited processing power, such as mobile devices and some video conferencing. It uses a restricted set of tools (e.g., I and P slices only, CAVLC for transform coefficients, no B-slices, no weighted prediction).
  - **Main Profile (MP):** Intended for standard-definition digital TV broadcasts. It adds support for B-slices, CABAC (optional), and interlaced coding tools.
  - **Extended Profile (XP):** Designed for streaming video, with improved error resilience tools.
  - **High Profile (HiP):** The most common profile for broadcast and disc storage applications (e.g., Blu-ray, HDTV). It builds on the Main Profile by adding support for 8x8 transform, custom quantization matrices, and making CABAC mandatory for transform coefficients, offering the best compression efficiency among these.
  - Other profiles exist for professional applications (e.g., High 10, High 4:2:2, High 4:4:4 Predictive) supporting higher bit depths and chroma formats.
- **Levels:** A level specifies a set of constraints on parameters such as maximum resolution, frame rate, bitrate, macroblock processing rate, and the size of the decoded picture buffer (DPB).[62] For example, Level 3.1 might support up to 720p at 30fps, while Level 4.1 supports up to 1080p at 30fps or 720p at 60fps, and Level 5.1 can handle 4K resolutions. Adherence to a specific level ensures interoperability between encoders and decoders.

3. Widespread Applications

The combination of significantly improved compression efficiency, a flexible toolset,

and well-defined profiles and levels led to the unprecedented adoption of H.264/AVC across a vast spectrum of video applications [1]:

- **Blu-ray Discs:** H.264/AVC is one of the three mandatory video codecs for Blu-ray Discs and is the most commonly used format for encoding HD movies on this medium.[18]
- **Internet Streaming:** It became the de facto standard for video streaming services like YouTube, Netflix, Amazon Prime Video, Hulu, and Vimeo, often serving as the primary codec or a widely compatible fallback option.[41]
- **HDTV Broadcasting:** Adopted by numerous digital television standards worldwide, including ATSC (North America), DVB-T/T2 (Europe, Asia, Australia), DVB-S/S2 (satellite), and DVB-C (cable) for HD transmissions.[61]
- **Video Conferencing and Real-Time Communication:** The Constrained Baseline Profile of H.264 is a mandatory-to-implement codec for WebRTC-compliant browsers, ensuring interoperability for real-time video calls.[18] Many dedicated video conferencing systems also rely heavily on H.264.
- **Mobile Devices:** Smartphones, tablets, and portable media players universally support H.264 decoding, and many support hardware-accelerated encoding.
- **Digital Cameras and Camcorders:** From consumer point-and-shoot cameras to professional video cameras, H.264 is a common recording format.
- **Surveillance Systems:** IP-based security cameras widely use H.264 for compressing video feeds due to its balance of quality and bitrate.[18]
- **Digital Signage and Video Players:** Broad support in various embedded systems and media players.

Despite the emergence of newer, more efficient codecs, H.264/AVC's vast installed base of compatible hardware and software ensures its continued relevance and widespread use for many years to come, particularly for HD content and applications prioritizing maximum compatibility.

D. H.265/HEVC: Efficiency for Higher Resolutions (4K/8K, HDR)

H.265, also known as High Efficiency Video Coding (HEVC) or MPEG-H Part 2, was developed as the successor to H.264/AVC by the Joint Collaborative Team on Video Coding (JCT-VC), a partnership between the ITU-T VCEG and the ISO/IEC MPEG.[1] Standardized in 2013, HEVC was designed to address the growing demand for higher resolution video, such as 4K (3840x2160) and 8K (7680x4320) Ultra High Definition (UHD), as well as content with High Dynamic Range (HDR) and wider color gamuts.

1. Key Features and Improvements over H.264/AVC

The primary design goal of HEVC was to achieve approximately a 50% reduction in

bitrate compared to H.264/AVC for the same level of subjective video quality, or substantially improved quality at the same bitrate.[28] This significant leap in compression efficiency was realized through several key architectural and algorithmic enhancements:

- **Coding Tree Units (CTUs):** HEVC replaces H.264's 16x16 macroblocks with more flexible Coding Tree Units (CTUs). CTUs serve as the basic processing unit and can range in size from 16x16 up to 64x64 luma pixels.[19] Larger CTUs are particularly beneficial for efficiently coding large, uniform regions common in high-resolution video.
    - **Hierarchical Partitioning:** Each CTU can be recursively partitioned into smaller square Coding Units (CUs) using a quadtree structure, down to a minimum size (e.g., 8x8). This allows the encoder to adapt the block size to local image complexity more effectively than H.264's macroblock partitioning.
    - **Prediction Units (PUs) and Transform Units (TUs):** CUs are further divided into Prediction Units (PUs) for intra/inter prediction and Transform Units (TUs) for the transform and quantization stages. PUs can have various symmetric and asymmetric shapes to better capture object boundaries. TUs also follow a quadtree structure within a CU and can range in size from 4x4 up to 32x32 pixels.[19]
- **Advanced Intra Prediction:** HEVC significantly expands the number of intra-prediction modes compared to H.264. For luma blocks, it supports planar, DC, and up to 33 angular prediction modes, allowing for more accurate prediction of diverse textures and edge orientations.[20]
- **Improved Motion Compensation and Prediction:**
    - **Advanced Motion Vector Prediction (AMVP):** Motion vectors are predicted from spatial and temporal neighbors, and the difference is coded.
    - **Merge Mode:** Allows a PU to inherit motion information (motion vectors, reference indices, prediction direction) from neighboring PUs, reducing signaling overhead.
    - These techniques, combined with the flexible PU partitioning, lead to more efficient motion representation.
- **Parallel Processing Capabilities:** HEVC incorporates features designed to facilitate efficient parallel processing on multi-core architectures:
    - **Tiles:** A picture can be divided into rectangular regions (tiles) that can be encoded and decoded independently, or with limited dependencies, allowing for parallel processing of different picture areas.[20]
    - **Wavefront Parallel Processing (WPP):** Allows rows of CTUs to be processed in a pipelined parallel fashion, with dependencies managed to enable parallel

decoding threads.[20]

- **Sample Adaptive Offset (SAO):** An in-loop filter applied after the deblocking filter.[28] SAO classifies reconstructed pixels and adds learned offsets to them to reduce distortion and improve perceptual quality by better matching the original signal characteristics.
- **Larger Transform Units (TUs):** HEVC supports TU sizes of 4x4, 8x8, 16x16, and 32x32 for the integer transform (an approximation of DCT). Larger TUs provide better energy compaction for larger, smoother CUs.
- **Profiles for Advanced Video Features:** HEVC defines several profiles, with **Main Profile** supporting 8-bit 4:2:0 video, and **Main 10 Profile** supporting 10-bit 4:2:0 video.[28] The Main 10 profile is crucial for HDR content (like HDR10 and HLG) and wider color gamuts, as 10-bit precision is needed to represent the increased dynamic range and color volume without banding artifacts. Other profiles cater to still pictures, screen content, and scalable/multi-view coding.

2. Adoption and Licensing Challenges

HEVC has seen significant adoption, particularly in applications where its compression efficiency for high-resolution and HDR content provides substantial benefits. However, its rollout was notably more complex than H.264's due to its patent licensing situation.[41]

- **Adoption and Use Cases** [45]**:**
  - **4K/8K Ultra HD Streaming:** Major streaming services like Netflix and Amazon Prime Video utilize HEVC for delivering 4K UHD and HDR content to compatible devices (e.g., Smart TVs, streaming boxes).[45]
  - **UHD Blu-ray:** HEVC is a mandatory codec for UHD Blu-ray discs, enabling the storage of 4K HDR movies.
  - **Digital Broadcasting:** Newer broadcast standards, such as ATSC 3.0 in North America and some DVB deployments, mandate or support HEVC for UHD transmissions.[73]
  - **Mobile Devices:** High-end smartphones and tablets increasingly feature hardware HEVC decoding and encoding capabilities, driven by the desire to capture and play 4K video. Apple devices, for example, have broadly adopted HEVC.
  - **Professional Video:** Used in some professional cameras and production workflows for high-quality acquisition and contribution.
- Licensing Challenges 41:
  Unlike H.264, which had a relatively consolidated patent pool managed by MPEG LA (now Via Licensing Alliance), the patent landscape for HEVC became

fragmented and significantly more complex. Multiple patent pools emerged, including:

- **MPEG LA:** One of the initial pools for HEVC.
- **HEVC Advance (now part of Access Advance):** Formed by several major patent holders, initially with more aggressive royalty demands, including potential royalties on content distribution (though this was later revised for free internet content).
- **Velos Media:** Another pool comprising significant patent holders. This multiplicity of pools, coupled with some patent holders choosing to license independently, created considerable uncertainty and increased the cumulative royalty burden for implementers (device manufacturers, software developers, and potentially content distributors).[47] Estimates suggested that the total per-unit HEVC patent royalty could be substantially higher than for H.264.[70] For example, MPEG LA's HEVC device royalty was $0.20/unit (after the first 100,000 units, capped at $25M annually), while Access Advance's rates were in a similar range but with potentially higher caps or different regional pricing.[69] The perceived high cost and complexity of HEVC licensing are widely cited as factors that slowed its adoption in certain royalty-sensitive areas, particularly for open web platforms and software applications. This environment directly contributed to the industry-led effort to develop AV1 as a high-performance, royalty-free alternative.[41] Despite these challenges, HEVC's technical merits for UHD and HDR video have ensured its strong presence in markets where these features are paramount and where licensing costs can be absorbed (e.g., premium devices and services).

E. VP9: Google's Royalty-Free Alternative

VP9 is an open and royalty-free video compression standard developed by Google. It was released in 2013 as a successor to VP8 and was designed to offer compression efficiency comparable to H.265/HEVC, providing a high-quality, patent-unencumbered option primarily for web-based video delivery.[1]

1. Development, Features, and Use Cases

- Development 49:
  Google began developing VP9 (initially under codenames like Next Gen Open Video - NGOV and VP-Next) in late 2011. The primary goal was to achieve a significant reduction in bitrate (around 50%) compared to VP8 while maintaining similar video quality, thus positioning it as a direct competitor to HEVC. Profile 0 of VP9 was finalized in June 2013, with Google Chrome quickly adding support. Mozilla Firefox followed in March 2014. Google later introduced profiles for higher

bit depth (Profile 2 and Profile 3 for 10/12-bit color and HDR) in 2014.

- **Key Features** [29]**:**
  - **Royalty-Free and Open Source:** This is a defining characteristic of VP9, eliminating patent licensing fees and encouraging broad adoption, particularly by web platforms and open-source projects.[48]
  - **Compression Efficiency:** VP9 offers substantial compression gains over VP8 and H.264/AVC, and its efficiency is generally considered to be in the same league as H.265/HEVC.[1] It can achieve up to 50% bitrate savings compared to VP8 for similar quality.[49]
  - **Superblocks:** VP9 uses a flexible block structure based on "superblocks," which can be as large as 64x64 pixels (similar to HEVC's CTUs).[29] These superblocks can be recursively partitioned into smaller blocks (down to 4x4) for prediction and transform, allowing adaptation to varying image complexity.
  - **Prediction Techniques:** VP9 employs advanced intra-prediction modes (e.g., DC, TrueMotion, and 8 directional modes) and sophisticated inter-prediction techniques, including multiple reference frames, sub-pixel motion compensation (up to 1/8th pixel accuracy), and specialized motion vector prediction.
  - **Transform Coding:** Primarily uses Discrete Cosine Transform (DCT) of various sizes (4x4, 8x8, 16x16, 32x32). [29] mentions DWT for VP9, but DCT is the standard transform; this might refer to specific research or experimental versions.
  - **Entropy Coding:** Uses arithmetic coding for efficient representation of quantized coefficients and other syntax elements.
  - **Color Depth and HDR Support:** Supports multiple profiles, including Profile 0 (8-bit, 4:2:0), Profile 1 (8-bit, 4:2:2/4:4:0/4:4:4), Profile 2 (10/12-bit, 4:2:0, for HDR), and Profile 3 (10/12-bit, 4:2:2/4:4:0/4:4:4, for HDR).[49] This enables support for High Dynamic Range video.
  - **Resolutions:** Supports a wide range of resolutions, from standard definition up to 8K.[48]
  - **Loop Filtering:** Includes an adaptive in-loop filter to reduce compression artifacts.
- **Use Cases** [1]**:**
  - **YouTube:** VP9 is extensively used by YouTube for streaming a vast amount of its content, especially for HD, 4K, and HDR videos.[48] Its royalty-free nature and efficiency made it an attractive choice for Google's high-volume platform.
  - **Web Browsers:** Enjoys strong native support in major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera.[48]
  - **Android:** VP9 decoding has been supported in Android since version 4.4

(KitKat).[49] iOS/iPadOS added support in version 14.[49]
    - **WebM Container:** VP9 video is typically packaged in the WebM container format, which also uses Vorbis or Opus for audio.
    - **WebRTC:** VP9 is an optional codec for WebRTC, offering better compression than the mandatory VP8 for real-time communication applications.[63]
    - **Other Streaming Services:** While not as universally adopted as H.264 or HEVC by all streaming services, its presence on YouTube and strong browser support make it significant for web video.

While VP9 offered a compelling royalty-free alternative with performance rivaling HEVC, hardware encoding support for VP9 did not become as widespread as hardware decoding support.[49] Nevertheless, its impact, particularly through YouTube, has been substantial in promoting high-quality, efficient video on the open web.

F. AV1: The Alliance for Open Media's Next-Generation Codec

AOMedia Video 1 (AV1) is an open, royalty-free video coding format developed by the Alliance for Open Media (AOMedia), a consortium founded in 2015 by leading technology companies including Amazon, Apple, ARM, Cisco, Google, Intel, Meta (Facebook), Microsoft, Mozilla, Netflix, NVIDIA, Samsung, and Tencent.[2] AV1 was designed with the ambitious goals of providing a next-generation codec that surpasses the compression efficiency of H.265/HEVC and VP9, while remaining entirely royalty-free to foster widespread adoption and innovation, particularly for internet-delivered video.[44] The final bitstream specification for AV1 was released in March 2018.

1. Royalty-Free Model and Industry Support

The primary motivation behind AV1's development was to create a high-performance video codec unencumbered by the complex and potentially costly patent licensing schemes associated with HEVC.[2] AOMedia's members, many of whom are major content distributors and hardware/software vendors, pooled their intellectual property and expertise to achieve this.

- **Licensing Model:** AV1 is distributed under a royalty-free patent license, meaning that implementers (e.g., browser vendors, streaming services, hardware manufacturers) do not have to pay royalties for using the codec technology itself.[2] This is intended to accelerate adoption, reduce costs for service providers, and ensure that the codec can be freely integrated into open-source software.
- **Industry Support:** The backing of AOMedia by such a broad coalition of influential tech companies provides AV1 with significant industry momentum.[50] These companies are actively contributing to the development of AV1 encoders

and decoders, promoting its use in their products and services, and working to ensure hardware support.

However, it's important to note that while the AV1 codec itself is royalty-free from AOMedia, some patent pools (like Access Advance's Video Distribution Patent Pool) have indicated intentions to seek royalties from content distributors for streaming video encoded in various formats, including AV1, based on patents they claim cover aspects of video distribution or playback systems, rather than the codec technology directly.[72] This remains an evolving area.

2. Technical Advancements and Performance

AV1 incorporates a wide array of advanced coding tools, many of which are enhancements of techniques found in VP9 and HEVC, along with novel approaches. The goal is to achieve a bitrate reduction of approximately 30% or more compared to HEVC or VP9 for the same perceptual video quality.[2]

Key technical features include [2]:

- **Flexible Block Partitioning:** AV1 uses "superblocks" of size 64x64 or 128x128 pixels. These can be recursively partitioned using a highly flexible scheme that includes not only quad-tree splits but also binary and ternary splits, and rectangular partitions (e.g., 4:1, 1:4 aspect ratios). This allows for block sizes ranging from 128x128 down to 4x4, enabling very fine adaptation to content features.[29]
- **Advanced Intra Prediction:** Includes a rich set of directional intra-prediction modes (e.g., 56 directional modes, plus DC, planar, and specialized modes like Paeth predictor), as well as intra block copy (where a block is predicted from an already coded region within the same frame). Chroma from Luma (CfL) prediction is also supported.
- **Sophisticated Inter Prediction:**
    - **Compound Prediction Modes:** Multiple ways to combine predictions from different reference frames (e.g., weighted averaging, difference-based).
    - **Warped Motion / Affine Motion Compensation:** Allows for more complex motion models than simple translation, capturing deformations like rotation, scaling, and shear. This is particularly useful for non-rigid motion or camera zooms.
    - **Overlapped Block Motion Compensation (OBMC):** Smooths block boundaries in motion-compensated prediction by extending PUs and blending with neighboring predictions.
    - **Global Motion Models:** Can estimate and compensate for dominant camera

motion (pan, tilt, zoom, rotation) across an entire frame.

- **Transform Coding:** Uses a variety of 2D DCT-like transforms (DCT, ADST, Identity Transform) with sizes from 4x4 up to 64x64, including non-square transforms (e.g., 64x32, 16x4).
- **In-Loop Filtering:** AV1 employs a sophisticated suite of in-loop filters to enhance reconstructed quality:
  - **Deblocking Filter:** Similar to those in H.264/HEVC but adapted for AV1's block structures.
  - **Constrained Directional Enhancement Filter (CDEF):** A conditional directional filter applied after deblocking to remove ringing artifacts and enhance detail around edges without excessive blurring.[29]
  - **Loop Restoration Filter:** Uses Wiener filtering or self-guided filtering to further reduce noise and other impairments in the reconstructed frame.[29]
- **Entropy Coding:** Uses highly efficient symbol-by-symbol adaptive multi-symbol arithmetic coding (ANS-based).
- **Film Grain Synthesis:** A unique tool where film grain characteristics are analyzed, removed before encoding, and then parameters are sent to the decoder to synthesize perceptually similar grain during playback. This can save significant bits, as encoding actual film grain is very costly.
- **Scalability and Screen Content Coding:** AV1 includes tools for scalable video coding (SVC) and efficient coding of screen content (e.g., computer desktops, presentations).
- **Bit Depth and Color Space Support:** Supports 8, 10, and 12-bit color depths, and various chroma subsampling formats including 4:2:0, 4:2:2, 4:4:4, and monochrome. It is well-suited for HDR and Wide Color Gamut (WCG) content.[5]

Performance:
Numerous studies and real-world deployments have demonstrated AV1's superior compression efficiency.

- NVIDIA's NVENC AV1 hardware encoder shows approximately 1.5-2 dB higher PSNR than their NVENC H.264 encoder at the same bitrate, translating to about 40% bitrate savings for AV1 over H.264 at 1080p60 for similar PSNR. VMAF scores also consistently show AV1 outperforming H.264, especially at lower bitrates.[12]
- Compared to HEVC, AV1 generally achieves the targeted 30% (or more) bitrate reduction for similar visual quality, particularly evident in tests by Moscow State University and Netflix.[50]
- The efficiency gains are often more pronounced at lower bitrates, which is critical for streaming over constrained networks.[78]

The combination of a royalty-free licensing model and strong technical performance

positions AV1 as a key codec for the future of internet video, though its adoption trajectory is still influenced by factors like hardware support evolution and encoding complexity.

The history of video codec development reveals a fascinating "pendulum swing" in licensing models. Standards emerging from formal bodies like MPEG (MPEG-2, H.264, HEVC) have typically involved patent pools and associated royalty obligations.[37] However, when these licensing terms become perceived as overly complex or expensive, as was notably the case with HEVC [44], it often catalyzes the development and adoption of royalty-free alternatives. Google's VP8 and VP9, and subsequently AOMedia's AV1, are prime examples of this reaction.[44] This dynamic creates a continuous push-and-pull within the industry, influencing research investment, standardization strategies, and ultimately, which codecs achieve mainstream adoption. The success of AV1, for instance, may further solidify the demand for royalty-free options, especially for foundational web technologies.

It is also evident that codec development is an evolutionary, rather than revolutionary, process. Core principles such as block-based processing, transform coding (DCT), motion estimation/compensation, and entropy coding have remained central from early standards like H.261 through to the latest ones like VVC.[11] Each new standard typically refines these existing tools—for example, by introducing larger and more flexible block structures [19], a greater variety of prediction modes [17], or more effective in-loop filters [28]—and integrates novel techniques to address specific limitations or exploit previously untapped redundancies. This iterative refinement allows the industry to leverage accumulated knowledge and existing hardware design paradigms while continuously pushing the boundaries of compression. Truly disruptive shifts, such as a complete departure from the block-based hybrid coding model, are infrequent and face substantial hurdles to adoption due to the established ecosystem.[53]

Finally, the designation of a codec as a "successor" does not guarantee its immediate or universal replacement of the older standard. H.264/AVC remains extensively used despite H.265/HEVC offering superior compression, partly due to HEVC's licensing complexities and the vast installed base of H.264-compatible hardware.[1] Similarly, AV1, despite its technical advantages and royalty-free status, is still in the process of achieving widespread hardware support comparable to HEVC.[51] Codec transitions are inherently slow and multifaceted, influenced by a confluence of factors beyond mere technical superiority. These include hardware refresh cycles, software update mechanisms, licensing costs and terms, existing infrastructure investments, and the specific requirements of diverse applications. Consequently, multiple generations of

codecs often coexist for extended periods, each serving different segments of the market or different points in the quality-bitrate-complexity trade-off space.

VI. Comparative Analysis of Modern Video Codecs

The landscape of modern video codecs is characterized by a continuous push for greater compression efficiency, enabling higher resolutions, improved visual quality, and reduced bandwidth consumption. This section provides a comparative analysis of prominent codecs—H.264/AVC, H.265/HEVC, VP9, AV1, and the emerging VVC/H.266—across several key dimensions: compression efficiency, computational complexity, feature sets, licensing models, and adoption rates.

A. Compression Efficiency and Quality Benchmarks (PSNR, VMAF)

Compression efficiency is a primary metric for evaluating video codecs, typically measured by the bitrate required to achieve a certain level of visual quality, or the quality achieved at a given bitrate. Objective metrics like Peak Signal-to-Noise Ratio (PSNR) and perceptually-oriented metrics like Video Multimethod Assessment Fusion (VMAF) are commonly used in such comparisons.

- **PSNR (Peak Signal-to-Noise Ratio):** An engineering metric that measures the ratio between the maximum possible power of an original signal and the power of the noise (error or distortion) introduced by compression. Higher PSNR values generally indicate better reconstruction quality (less distortion) in decibels (dB).[12]
- **VMAF (Video Multimethod Assessment Fusion):** A perceptual video quality metric developed by Netflix, designed to correlate more closely with human subjective perception of video quality than traditional metrics like PSNR or SSIM (Structural Similarity Index). VMAF scores range from 0 (worst) to 100 (best) and are particularly relevant for assessing the quality of streaming video.[12]

Comparative Performance:

- **H.264/AVC as Baseline:** Due to its long-standing ubiquity, H.264/AVC often serves as the reference point for comparing newer codecs.
- **H.265/HEVC vs. H.264/AVC:** HEVC generally achieves a bitrate reduction of approximately 35-50% compared to H.264 for the same subjective visual quality.[29] This means HEVC can deliver comparable quality at about half the bitrate of H.264.
- **VP9 vs. H.264/AVC & HEVC:** VP9 was designed to offer compression efficiency on par with HEVC. Thus, it provides significant bitrate savings over H.264 (potentially up to 50% or more) and performs similarly to HEVC in many scenarios.[1]

- **AV1 vs. H.265/HEVC, VP9, & H.264/AVC:**
  - AV1 aims for an additional ~30% improvement in compression efficiency over HEVC and VP9.[2] This translates to potential savings of over 50% compared to H.264.[50]
  - Objective tests by NVIDIA using their NVENC hardware encoders showed AV1 achieving ~1.5-2 dB higher PSNR than H.264 at the same bitrate. This corresponded to approximately 40% bitrate savings for AV1 over H.264 at 1080p60 for equivalent PSNR. VMAF scores also favored AV1, particularly at lower bitrates where AV1 maintained better perceptual quality.[12]
  - Independent studies, such as those by Moscow State University, have indicated AV1 can outperform HEVC by around 28% in encoding efficiency. In one comparison, AV1 achieved the same quality as the x264 (H.264) encoder at 55% of the average bitrate, while the x265 (HEVC) encoder (in a high-quality placebo mode) required 67% of the x264 bitrate.[78]
  - However, real-world performance can vary. Some user-conducted tests have shown AV1's performance to be very close to that of mature HEVC encoders like x265, with AV1 sometimes excelling at lower quality/bitrate points and HEVC at very high quality points.[80] This suggests that encoder implementations, specific encoding settings, and content characteristics heavily influence the observed efficiency.[79] The general consensus remains that AV1 offers its most significant advantages at lower bitrates, crucial for streaming over constrained networks.[78]
- **VVC (H.266) vs. HEVC & AV1:**
  - VVC is designed to offer another substantial leap in compression, targeting a 30-50% bitrate reduction over HEVC for the same perceptual quality.[42]
  - Early comparative studies indicate VVC significantly outperforms both HEVC and AV1. For 8K resolution content, one study reported VVC achieving average bitrate savings of approximately 59% relative to HEVC and about 46% relative to AV1.[82] The efficiency gains of newer codecs like VVC and AV1 tend to be more pronounced at higher resolutions (4K, 8K).[82]

It is crucial to recognize that "efficiency" is a multifaceted concept extending beyond mere bitrate for a given quality. For live streaming, attributes like encoding speed and low latency might take precedence over achieving the absolute minimum bitrate.[77] In archival scenarios, long-term stability and perfect quality retention are paramount. For mobile devices operating on battery power, decoding efficiency (which correlates with power consumption) becomes

Works cited

1. Video codec: here's what you need to know | Kaltura, accessed June 8, 2025, https://corp.kaltura.com/blog/video-codec/
2. The Popular Video Codes, Their Pros and Cons, and Related File Formats - Cloudinary, accessed June 8, 2025, https://cloudinary.com/guides/video-formats/a-primer-on-video-codecs
3. Understanding Codecs: A Beginner's Guide to Audio & Video Conversion - Media Mojo, accessed June 8, 2025, https://themediamojo.com/index.php/codecs/
4. What is a Codec? | Livery's Beginner's Guide - Livery Video, accessed June 8, 2025, https://www.liveryvideo.com/explanation/what-is-a-codec-your-beginners-guide-to-codecs-and-how-they-work-2/
5. Web video codec guide - Media | MDN, accessed June 8, 2025, https://developer.mozilla.org/en-US/docs/Web/Media/Guides/Formats/Video_codecs
6. Video Codecs and Encoding: Everything You Should Know (Update) - Wowza, accessed June 8, 2025, https://www.wowza.com/blog/video-codecs-encoding
7. Video File Size Calculator (by format) - Omni Calculator, accessed June 8, 2025, https://www.omnicalculator.com/other/video-size
8. How to Accurately Calculate Video File Size (Plus: Bonus Glossary) - CircleHD, accessed June 8, 2025, https://www.circlehd.com/blog/how-to-calculate-video-file-size
9. A Detailed Overview Of Popular Video Compression Techniques, accessed June 8, 2025, https://imagekit.io/blog/video-compression-techniques/
10. Video interframe compression algorithms, spacial and temporal redundancy, accessed June 8, 2025, https://theteacher.info/index.php/fundamentals-section-1/1-1-information-representation/1-1-4-video/2567-video-interframe-compression-algorithms-spacial-and-temporal-redundancy
11. The H.264 Video Coding Standard - Department of Electrical Engineering and Computer Science, accessed June 8, 2025, https://www.cse.fau.edu/~hari/files/1743/Kalva_2006_The%20H.pdf
12. Improving Video Quality and Performance with AV1 and NVIDIA Ada ..., accessed June 8, 2025, https://developer.nvidia.com/blog/improving-video-quality-and-performance-with-av1-and-nvidia-ada-lovelace-architecture/
13. Emerging Advances in Learned Video Compression: Models, Systems and Beyond - arXiv, accessed June 8, 2025, https://arxiv.org/html/2504.21445v1
14. The Ultimate Guide to Video Encoding: Everything You Need to ..., accessed June 8, 2025, https://optiview.dolby.com/resources/blog/playback/the-ultimate-guide-to-video-encoding-everything-you-need-to-know/
15. YCbCr - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/YCbCr
16. YCbCr Color Space: Understanding Its Role in Digital Photography - PRO EDU, accessed June 8, 2025, https://proedu.com/blogs/photography-fundamentals/ycbcr-color-space-unders

tanding-its-role-in-digital-photography
17. Introduction to H.264 Video Compression Standard - EnGenius, accessed June 8, 2025, https://www.engeniustech.com/technical-papers/H.264-video-compression.pdf
18. What Is H.264/AVC (Advanced Video Coding) - VideoProc, accessed June 8, 2025, https://www.videoproc.com/resource/h264-codec.htm
19. Coding tree unit - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/Coding_tree_unit
20. HEVC: An introduction to high efficiency coding — Vcodex BV, accessed June 8, 2025, https://www.vcodex.com/hevc-an-introduction-to-high-efficiency-coding
21. US20200314432A1 - Intra-frame and Inter-frame Combined ..., accessed June 8, 2025, https://patents.google.com/patent/US20200314432A1/en
22. www.tek.com, accessed June 8, 2025, https://www.tek.com/en/support/faqs/what-discrete-cosine-transform-dct-mpeg#:~:text=The%20DCT%20converts%20the%20video,signal%20amplitudes%2C%20called%20DCT%20coefficients.
23. What is the discrete cosine transform (DCT) in MPEG? | Tektronix, accessed June 8, 2025, https://www.tek.com/en/support/faqs/what-discrete-cosine-transform-dct-mpeg
24. How Video Codecs Works - Lumenci, accessed June 8, 2025, https://lumenci.com/blogs/how-video-codecs-works/
25. Video Compression Techniques: Enhancing Quality and Performance | Cloudinary, accessed June 8, 2025, https://cloudinary.com/guides/video-effects/video-compression-techniques
26. Context-adaptive binary arithmetic coding - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/Context-adaptive_binary_arithmetic_coding
27. Entropy Coding - Cloudinary, accessed June 8, 2025, https://cloudinary.com/glossary/entropy-coding
28. H.265 (HEVC) | High Efficiency Video Coding | Next-Generation Video Compression, accessed June 8, 2025, https://flussonic.com/glossary/h.265
29. Navigating the codec landscape for 2025: AV1, H.264, H.265, VP8 and VP9 | Uploadcare, accessed June 8, 2025, https://uploadcare.com/blog/navigating-codec-landscapes/
30. lwks.com, accessed June 8, 2025, https://lwks.com/blog/understanding-video-formats-and-codecs-a-beginners-guide#:~:text=Codecs%2C%20on%20the%20other%20hand,265.
31. Understanding Video Formats and Codecs: A Beginner's Guide - Lightworks, accessed June 8, 2025, https://lwks.com/blog/understanding-video-formats-and-codecs-a-beginners-guide
32. Video Formats vs Video Codecs vs Video Containers - Gumlet, accessed June 8, 2025, https://www.gumlet.com/learn/understanding-video-formats-codecs-containers/
33. What Are Container File Formats (Media Containers)? | Cloudinary, accessed June 8, 2025,

https://cloudinary.com/guides/video-formats/what-are-container-file-formats-media-containers
34. The Differences Between Video Codecs and Containers: A ..., accessed June 8, 2025, https://callaba.io/difference-between-video-codecs-and-containers
35. Understanding the Role of Codec and Container in Live Video Streaming - Muvi, accessed June 8, 2025, https://www.muvi.com/blogs/video-streaming-codecs-container/
36. Media container formats (file types) - Media | MDN, accessed June 8, 2025, https://developer.mozilla.org/en-US/docs/Web/Media/Guides/Formats/Containers
37. MPEG-2 - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/MPEG-2
38. Container File Formats: Definitive Guide (2023) | Bitmovin, accessed June 8, 2025, https://bitmovin.com/blog/container-formats-fun-1/
39. What is metadata and image formats in videos? - Tdarr - Reddit, accessed June 8, 2025, https://www.reddit.com/r/Tdarr/comments/10dvwo5/what_is_metadata_and_image_formats_in_videos/
40. www.fastpix.io, accessed June 8, 2025, https://www.fastpix.io/blog/guide-to-container-file-formats-in-video#:~:text=The%20container%20holds%20the%20video,265%20codec.
41. Video codec - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/Video_codec
42. Evolution of Video Compression - TVyVideo + Radio, accessed June 8, 2025, https://www.tvyvideo.com/en/more-in-depth/technology/22347-evolution-of-video-compression.html
43. AV1 vs H265 vs VP9: Best Video Codec For Streaming in 2025 - Muvi, accessed June 8, 2025, https://www.muvi.com/blogs/best-video-codec-for-streaming/
44. Codec Licensing and Web Video Streaming, accessed June 8, 2025, https://www.streamingmedia.com/Articles/Post/Blog/Codec-Licensing-and-Web-Video-Streaming-161116.aspx
45. HEVC Codec Explained: Performance, Compatibility & Benefits, accessed June 8, 2025, https://www.wowza.com/blog/hevc
46. High Efficiency Video Coding - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding
47. HEVC Licensing: Misunderstood, Maligned, and Surprisingly ..., accessed June 8, 2025, https://streaminglearningcenter.com/codecs/hevc-licensing-misunderstood-maligned-and-surprisingly-successful.html
48. VP9 - Red5 Pro, accessed June 8, 2025, https://www.red5.net/vp9-codec/
49. Everything you need to know about VP9 codec - ImageKit, accessed June 8, 2025, https://imagekit.io/blog/vp9-codec/
50. AV1 - The Powerful Next Generation Video Codec - Bitmovin, accessed June 8, 2025, https://bitmovin.com/av1/
51. AV1 Codec Hardware Decode Adoption – ScientiaMobile, accessed June 8, 2025, https://scientiamobile.com/av1-codec-hardware-decode-adoption/
52. H.266 Codec: What is Versatile Video Coding (VVC) - Castr, accessed June 8,

2025, https://castr.com/blog/h-266-codec-vvc/
53. Advances in video compression: a glimpse of the long ... - SET, accessed June 8, 2025, https://set.org.br/setep/ed8_pt/v8-painel44-3.pdf
54. AV2 vs AV1: Next-Gen Video Codec Comparison - FastPix, accessed June 8, 2025, https://www.fastpix.io/blog/av2-vs-av1-a-comprehensive-comparison-of-next-gen-video-codecs
55. Reveal AV2 Codec: Future Video Streaming Next Gen | Coconut©, accessed June 8, 2025, https://www.coconut.co/articles/unveil-av2-codec-nextgen-video-streaming
56. Video Format Timeline of Services Provided by Oxford Duplication Centre - A history of video recording and playback. From 1950's onward, accessed June 8, 2025, https://www.oxfordduplicationcentre.com/Video_Format_Timeline_Oxfordshire_UK.htm
57. Timeline of video formats - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/Timeline_of_video_formats
58. What is MPEG-2 Codec? Here is the Answer - VideoProc, accessed June 8, 2025, https://www.videoproc.com/resource/mpeg-2-codec.htm
59. Blu-ray FAQ - Blu-ray.com, accessed June 8, 2025, https://www.blu-ray.com/faq/
60. SD Blu-ray - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/SD_Blu-ray
61. Advanced Video Coding - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/Advanced_Video_Coding
62. What is H.264? How it Works, Applications & More - Gumlet, accessed June 8, 2025, https://www.gumlet.com/learn/what-is-h264/
63. Understanding WebRTC Codecs, accessed June 8, 2025, https://webrtc.ventures/2025/02/understanding-webrtc-codecs/
64. Encoding Definition - YouTube Explained - Tella, accessed June 8, 2025, https://www.tella.com/definition/encoding
65. Mezzanine requirements - Support – Prime Video Direct - Amazon ..., accessed June 8, 2025, https://videodirect.amazon.com/home/help?topicId=G202129880
66. Best Video Codec for Streaming & Quality in 2025: Top Options Compared - Dacast, accessed June 8, 2025, https://www.dacast.com/blog/best-video-codec/
67. Decoding the Video Codec Wars: H.264, HEVC, and AV1 Compared ..., accessed June 8, 2025, https://flussonic.com/blog/news/evolution-of-video-codecs
68. Mastering Video Codecs: H.264, H.265, and AV1 Compared, accessed June 8, 2025, https://medialooks.com/articles/mastering-video-codecs-h-264-h-265-and-av1-compared/
69. HEVC/VVC License Fees - ViaLa, accessed June 8, 2025, https://www.via-la.com/licensing-2/hevc-vvc/hevc-vvc-license-fees/
70. Independent economic study suggests HEVC royalties should be comparable to or less than rates for AVC - Unified Patents, accessed June 8, 2025, https://www.unifiedpatents.com/insights/2019/1/9/independent-economic-study-

suggests-hevc-royalties-should-be-comparable-to-or-less-than-rates-for-avc
71. It's Time to Move Forward with HEVC - Streaming Media, accessed June 8, 2025, https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Its-Time-to-Move-Forward-with-HEVC-113278.aspx
72. The State of the Video Codec Market 2025 - Streaming Media Europe, accessed June 8, 2025, https://www.streamingmediaglobal.com/Articles/Editorial/Featured-Articles/The-State-of-the-Video-Codec-Market-2025-168627.aspx
73. What is ATSC 3.0? - Comcast Technology Solutions, accessed June 8, 2025, https://www.comcasttechnologysolutions.com/what-is-atsc-3.0
74. ATSC 3.0 - Wikipedia, accessed June 8, 2025, https://en.wikipedia.org/wiki/ATSC_3.0
75. AV1 vs HEVC: Video Codec Guide 2024 - Video Tap, accessed June 8, 2025, https://videotap.com/blog/av1-vs-hevc-video-codec-guide-2024
76. Understanding Video Codecs: H.264, H.265 (HEVC), VP9 & AV1 ..., accessed June 8, 2025, https://pixflow.net/blog/understanding-video-codecs-h-264-hevc-h-265-vp9-and-av1-explained/
77. AV1 vs H.264 vs H.265: Video Codec Comparison Guide - FastPix, accessed June 8, 2025, https://www.fastpix.io/blog/av1-vs-h-264-vs-h-265-best-codec-for-video-streaming
78. AV1 vs HEVC: Is AV1 a Better Codec than HEVC for the Future?, accessed June 8, 2025, https://www.winxdvd.com/video-transcoder/av1-vs-hevc.htm
79. Is H.265 preferred over AV1 when the source material is 1080p? - Reddit, accessed June 8, 2025, https://www.reddit.com/r/AV1/comments/1b7rk3j/is_h265_preferred_over_av1_when_the_source/
80. Personal comparison dataset of x264, x265, and AV1 using Constant Quality & VMAF : r/handbrake - Reddit, accessed June 8, 2025, https://www.reddit.com/r/handbrake/comments/19b49b7/personal_comparison_dataset_of_x264_x265_and_av1/
81. The Future of Video Compression | Nokia, accessed June 8, 2025, https://www.nokia.com/blog/the-future-of-video-compression-is-vvc-ready-for-prime-time/
82. Performance Comparison of VVC, AV1, HEVC and AVC for High Resolutions - Preprints.org, accessed June 8, 2025, https://www.preprints.org/manuscript/202402.0869/v1