TECHNICAL PAPER

KEYWORDS HEVC // METAVERSE // MOTION ESTIMATION // SUSTAINABILITY // VP9

Green Video Compression for Metaverse: Lessons Learned from VP9 and HEVC

By Natalia Molinero Mingorance

Advanced codecs such as HEVC and VP9, integral to facilitating seamless interactions, entail considerable energy consumption, notably in resource-intensive motion estimation processes during encoding and decoding. This energy expenditure predominantly occurs within data centers during video processing. The imperative balance between visual fidelity and processing efficiency underscores the pressing need for green video compression methods.

1545-0279/24©2024SMPTE

Abstract

Over the past decade, video consumption applications have surged, reaching new heights with the metaverse's emergence. This expansion burdens networks, data centers, and devices due to increased data volume and processing, leading to substantial energy consumption and high CO_2 emissions annually. Priority should be given to developing lightweight video compression algorithms to tackle this. Current standards fall short of achieving the desired efficiency. This study conducts a comprehensive analysis of Motion Estimation (ME) in leading metaverse video compression algorithms, VP9 and HEVC. Using Matlab, an exhaustive evaluation focuses on ME, allowing an objective comparison and integrates novel sustainability assessments. The findings highlight areas for future video compression improvements, paving the way for sustainable and optimized video storage and transmission in the metaverse.

The rise of multimedia consumption, particularly video content, has become an integral part of our lives. Videos are widely used in diverse fields, such as online entertainment, education, and remote work. The metaverse, an immersive digital universe spanning virtual worlds and augmented reality experiences, is witnessing exponential growth and demands high-quality digital video to meet the increasing demand for engaging content. In spite of that, many content consumers are unaware of the growing carbon footprint associated with this paradigm shift, where video creation and communication processes contribute significantly to global energy consumption and greenhouse gas (GHG) emissions.

Within the metaverse, video content is characterized by high resolutions, real-time streaming, and interactive features. To efficiently handle large volumes of video data, two of the most commonly used video compression methods are High-Efficiency Video Coding (HEVC) and VP9.

HEVC and VP9 are cutting-edge video codecs known for their ability to significantly reduce video file sizes without compromising visual quality. These codecs enable smoother streaming experiences and faster video transmission, facilitating seamless interactions within the metaverse. However, it is essential to note that the encoding and decoding processes of HEVC and VP9, especially the resource-intensive motion estimation (ME) processes, require substantial computational power and time, leading to increased energy consumption and pollution.

Efforts are now focused on creating more energy-efficient video compression methods to address the environmental impact of video processing in the expanding metaverse. By striking a balance between maintaining video quality and reducing processing times, sustainable and cost-effective solutions can be achieved.

To address this challenge, this paper aims to help in the development of new efficient video compression algorithms prioritizing energy efficiency. Using Matlab, the research provides a standardized and freely accessible implementation for comparing the two video codecs, focusing on the ME component of HEVC and VP9. The incorporation of energy and computational complexity metrics empowers researchers to experiment with novel parameters to advance energy-conscious video compression techniques.

The contributions of this paper extend beyond codec implementation and performance evaluation. It sheds light on the importance of considering energy consumption in the design of video compression methods and its impact on the digital carbon footprint. As the metaverse's popularity grows, the development of energy-efficient video compression algorithms becomes even more critical to minimize environmental impact and reduce carbon emissions.

The structure of the paper is as follows: the next section provides a literature review of the work accomplished to date, highlighting advancements in this field and elucidating how this study can address existing gaps. The Methodology section explains the implementation of the ME processes on HEVC and VP9, provides a description of the metaverse video file used in these tests, and introduces the metrics used to assess performance in terms of energy consumption and computational complexity. The Results section presents the outcomes of predicting the same frame in both codecs, offering valuable insights into their comparative performance. In the Discussion section, a broader perspective is provided on the escalating environmental impact resulting from our collective use of digital

video, especially with the rise of the metaverse. Finally, the paper concludes by summarizing key findings and outlining promising directions for future research.

By addressing the energy efficiency challenges in video compression and providing a systematized implementation, this paper significantly contributes to the advancement of sustainable video transmission in the metaverse, bringing us closer to a greener and more optimized digital future.

Literature Review

In a study by M. Uitto,[1] the energy and power consumption of open-source video encoders, including x264 for H.264/AVC, x265 for H.265/HEVC, and VP9, were examined. H.264/AVC, where AVC stands for Advanced Video Coding, is a widely used video compression standard and the predecessor of H.265, also known as HEVC. The x264 encoder had the lowest energy consumption but the lowest compression efficiency, while the x265 encoder had the best efficiency but higher energy consumption. VP9 demonstrated a favorable tradeoff between compression efficiency and energy consumption. It is important to note that these findings are specific to the analyzed encoder implementations and may not be universally applicable to all compression algorithms. The open-source encoders were developed with different programming styles and optimization goals, making generalizations challenging. To bridge this absence, this paper offers a detailed analysis of the complexity of VP9 and HEVC, using an implementation in Matlab that provides valuable insights into the intricacies of the algorithms.

In their work, D. Grois et al.[2] conducted a complexity analysis using the reference software implementations for H.264, HEVC, and VP9. Similar low-delay configurations were employed for all encoders. The results showed that HEVC achieved average bit rate savings of 32.5% and 40.8% compared to VP9 and H.264, respectively, for 1-pass encoding. For 2-pass encoding, HEVC yielded average bitrate savings of 32.6% and 42.2% relative to VP9 and H.264, respectively. However, the VP9 encoder had significantly higher encoding times than the x264 encoder, approximately 2,000 times higher for 1-pass encoding and 400 times higher for 2-pass encoding. Notably, the evaluation was specific to the encoder implementations used, particularly without direct comparability to the previous reference. The present investigation aims to provide reliable conclusions for new algorithm designs by evaluating HEVC and VP9 through a custom Matlab implementation, allowing for the analysis of the current method's complexity bottlenecks, concentrating on ME.

R. Monnier et al.[3] presented power consumption comparisons of different available encoders (x264 for H.264/AVC, VPxenc for VP9 and its previous version, VP8, x265 for H.265/HEVC, and KVazaar for high-performance HEVC encoding), considering the Peak Signal-to-Noise Ratio for the Y luminance component (PSNR-Y) to measure the quality by comparing the original video's maximum power to the distortion or noise affecting it. They evaluated HEVC with two encoders (x265 and Kvazaar) and found that Kvazaar exhibited twice the power consumption compared to x265 when assessing PSNR-Y. While this provides insights into encoder performance, it emphasizes the importance of specific codec implementations, which may differ in programming principles, programming languages, and optimization parameters. Therefore, to accurately evaluate different algorithms, they should be developed using the same programming language and follow consistent principles, such as variable and function structures, as demonstrated in the present paper.

A. Katsenou et al. investigated the energy, quality, and bitrate tradeoffs across the following codecs:[4] Scalable Video Technology for Alliance for Open Media Video 1 (SVT-AV1), VP9 (vpxenc), VVenC (a high-performance, open-source video encoder developed by Netflix, optimized for encoding video content in the HEVC format), and x265. They proposed a new metric for the required bits: the energy cost. Similar to the previous references, their performance results were obtained using third-party implementations, introducing uncertainties. They concluded that x265 appeared to be the best choice for low-energy solutions, albeit with slightly lower average video quality. While this study helps understand the implications of input parameters on different codecs, it does not provide specific reasons for these observations or potential solutions. Moreover, the lack of detail can lead to discrepancies with the conclusions presented,[1] where VP9 is regarded as the most efficient codec. The present work isolates the ME process, the most resource-consuming part of current video compression algorithms. This way, researchers can better understand its inefficiencies and take more concise actions.

To date, there are no energy-efficient video compression publications related to video frames with metaverse characteristics.

Methodology

In this section, the implications of utilizing the same software platform, specifically Matlab, for comparing both ME algorithms (i.e., VP9 and HEVC) will be reviewed. Then, the ME principles will be described, along with the features of the metaverse video sample used for the tests. This section ends with a description of the new metrics introduced.

Using Matlab to Study Video Compression Algorithms

Comparing existing codecs using the available software is challenging and does not yield objective results, potentially leading to inconsistencies.[5] Using Matlab offers several advantages for researchers working with new video codecs. Matlab provides a clear and straightforward comparison platform, allowing for precise control of configuration parameters and features.

As images and videos can be treated as matrices of data in Matlab, inserting, testing, and analyzing new ideas in compression algorithms involving transformations, prediction, and reconstruction of matrix data becomes more intuitive.

1545-0279/24©2024SMPTE

Due to Matlab's resource consumption and execution speed when compared to compiled software in languages such as C++, the computational cost must be carefully considered during performance testing. Matlab also offers a performance testing toolkit for translating code into C and C++.

By implementing key features of video codecs, VP9 and HEVC, in Matlab, this paper provides researchers with an educational tool to easily run and understand the ME module, a crucial and time-consuming task in current video encoders with a significant impact on energy efficiency.

This unified approach allows for an objective and efficient comparison of the two encoders, enabling researchers to gain valuable insights into video compression algorithms and their implications for energy consumption.

Motion Estimation Based on Block Match Algorithms

Although VP9 and HEVC algorithms differ, both codecs employ the Block Matching Algorithm (BMA) during ME at the encoder to discover motion vectors (MVs). Moreover, ME in both codecs supports variable block sizes (i.e., 4 x 4, 8 x 8, 16 x 16, 32 x 32, and 64 x 64 pixels). **Figure 1** illustrates the BMA, demonstrating a current block from a frame compared to blocks within a specified search area in a reference frame. The algorithm determines the best match by assessing the similarity between the current block and those in the search area, identifying the MV representing the necessary displacement to align the blocks. This MV denotes the optimal position in the reference frame, facilitating motion estimation for effective video compression. The specific block of pixels selected from the reference frame, aligning with the corresponding block in the current frame as the optimal match, is termed the prediction block (PB).

As will be examined in the Results section, the resolution of MVs plays a significant role in efficiency but introduces additional complexity. Finer MV resolutions lead to increased complexity in subpixel interpolations. Additionally, representing higher-resolution MVs requires more bits. Thus, there is a tradeoff in MV resolution to balance ME accuracy for improved coding gains, the bit budget allocated for signaling MVs, and encoding complexity.

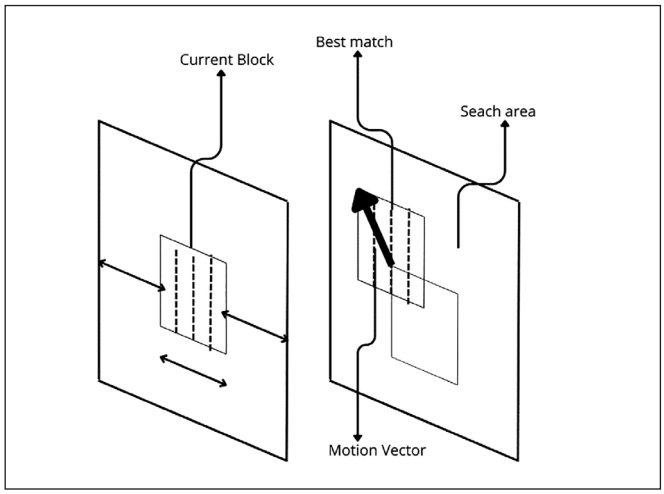

FIGURE 1. Block Matching Algorithm.

MATLAB PROVIDES A **CLEAR AND STRAIGHTFORWARD COMPARISON PLATFORM,** ALLOWING FOR PRECISE CONTROL OF CONFIGURATION PARAMETERS AND FEATURES.

Moreover, the length of the chosen interpolation filter impacts the amount of data fetched from memory and the number of operations performed. The number of multiplications and additions per sample required for interpolating the block is very significant when compared to other steps in the ME process, especially for smaller blocks of size NxM when MVs represent fractional displacements in both horizontal and vertical directions.

Both encoders employed used a 256 x 256 pixel reference and current coding unit (CU) sizes and a 32 x 32 PB size to aid testing. This PB size strikes an intermediate performance balance, as reducing it is known to increase the bit rate and decoding time, but can compromise quality.[6]

HEVC: ME Matlab Implementation

In HEVC, two search methods for ME aim to find the best-matched predicted block: Full Search and Fast Motion Search.

The Full Search method checks all points within the search window, which can be blocks of pixels or subpixels, depending on the resolution. While simple, it is time-consuming. On the other hand, Fast Motion Search checks a subset of points in multiple iterations. This method is faster than Full Search but sacrifices accuracy, making it a more common choice for software implementations.

The Test Zone (TZ) Search scheme (a Fast Motion Search method), implemented in the Matlab simulator, follows these steps:

- Square Search: it calculates the best distance (*bestDistance*) as the minimum cost among all the blocks scanned.
- If *bestDistance* > *iRaster* (typically set as 4), do Raster Search.
- If 0 < *bestDistance* < *iRaster*, do a Raster Refinement Search.

During this refinement process, subpixel interpolation of the last selected block occurs, necessitating comparisons of all pixels and subpixels from the reference frame with the current frame. This additional level of precision in the ME comes at the cost of increased computational complexity and higher energy consumption.

The complexity of an encoder is also influenced by the choice of metric used to express the similarity between a current and a reference image during ME. The most commonly used metric in HEVC is Sum of Absolute Differences (SAD). For subpixel accuracy (½- and ¼-pel), Sum of Absolute Transform Differences (SATD) is employed. SATD is more complex as it involves computing the transform of a block. Equations 1 and 2 define SAD and SATD, respectively:

$$SAD(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i,j) - R(x+i, y+j)| \quad (1)$$

$$SATD(x,y) = \frac{1}{2} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |H \cdot (C(i,j) - R(x+i, y+j)) \cdot H^T| \quad (2)$$

In both equations, C(i, j) and R(x + i, y + j) represent pixel intensities of two images, and M and N are the dimensions of a block within the images. The coordinates (x, y) indicate the MV coordinates of the reference block, and H denotes the Hadamard transform.[7]

The Hadamard transform is a type of linear transformation with elements of +1 and -1. In the context of SATD, it is applied as a means of decorrelating the pixel values or capturing the spatial frequency information in an image or block of pixels. H is used to transform the pixel differences between two corresponding blocks before calculating the sum of their absolute values. This transformation helps capture the spatial frequency characteristics of the differences.

SAD is a simpler metric that calculates the sum of absolute differences between corresponding pixel values in two image blocks.

Both functions are fundamental operations in the block-matching ME subsystem.

VP9: ME Matlab Implementation

The implementation of the VP9 inter-prediction process in Matlab is an adaptation of the pseudo-code detailed in Ref. 8.

- Motion vector selection: finding the MV for the current block.
- Motion vector clamping: changing the MV into the appropriate precision and clamping MVs that go too far off the edge of the frame, i.e., beyond the boundaries of the frame or outside the permissible search range defined by the encoder settings. Choosing the motion vector clamping limit significantly impacts encoding efficiency, balancing accurate representation and resource allocation. The correct limit optimizes the process by excluding irrelevant data without compromising video quality.
- Motion vector scaling: computing the sampling locations in the reference frame based on the MV. The sampling locations are also adjusted to compensate for any difference in the size of the reference frame compared to the current frame.
- Block inter-prediction: obtaining the 2D array containing inter-predicted samples. The sub-sample interpolation is obtained using two one-dimensional convolutions. First, a horizontal filter is used to build-up a temporary array, and then, this array is vertically filtered to obtain the final prediction.

Among these, motion vector scaling and block inter-prediction stand out as the most demanding operations. Motion vector scaling involves computing sampling locations in the reference frame based on MVs and adjusting them to account for any discrepancies in frame size between the current and reference frames. This process requires numerous arithmetic operations and addressing calculations, significantly contributing to computational overhead. On the other hand, block inter-prediction involves sub-sample interpolation using two one-dimensional convolutions to generate the 2D array of inter-predicted samples. These convolution operations are computationally intensive, particularly for larger block sizes and high-resolution videos.

Key Features of Metaverse Videos

A typical metaverse video is a dynamic and immersive medium that plays a crucial role in shaping the user experience within the virtual environment. Several key features define the quality and realism of metaverse videos:

- Resolution: it refers to the number of pixels used to display the video image. Higher- resolution videos provide sharper and more detailed visuals, enhancing the overall immersive experience. Common resolutions include HD (720p), Full HD (1080p), 2K (1440p), and 4K Ultra HD (2160p). As the metaverse aims for realism, higher resolutions are often preferred to create lifelike and detailed virtual worlds.
- Frames Per Second (FPS): it denotes the number of individual frames displayed per second in the video. Higher FPS values result in smoother motion and reduced motion blur, which is crucial for interactive experiences within the metaverse. Standard frame per sec values are 24, 30, and 60, but for an optimal virtual reality (VR) experience, 60 frames/s or higher is recommended to ensure fluid and comfortable interactions.
- Dynamic Lighting and Shading: Metaverse videos incorporate dynamic lighting and shading techniques to simulate realistic lighting conditions. Real-time rendering of shadows, reflections, and global illumination improves the visual fidelity and adds depth to virtual scenes.
- Interactive Elements: Metaverse videos often feature interactive elements, allowing users to participate and influence the virtual environment. This could include real-time interactions with objects, characters, and other users, enabling a sense of agency and immersion within the virtual world.
- Compression and Streaming: Efficient video compression is essential for streaming metaverse content smoothly over the internet. High-quality video codecs, such as HEVC or VP9, are commonly used to reduce file sizes without significant loss in visual fidelity.

1545-0279/24©2024SMPTE

- Stereoscopic Rendering: This technique can be employed in metaverse videos designed for VR experiences. It creates a sense of depth and three-dimensionality, providing an immersive and realistic visual perception when viewed through VR headsets.

In this study, a 4K (UHD-1) raw video with a frame rate of 60 frames/sec and a duration of 120 sec served as the source material for studying ME in VP9 and HEVC compression. For this research, a single frame from the video (i.e., the same for both codecs) was selected to perform the ME analysis. The chosen frame provided a representative snapshot of the dynamic and interactive content present in metaverse environments. **Figure 2** depicts the metaverse frame utilized for these tests.

By focusing on a single frame, the study aimed to isolate and evaluate the performance of the ME algorithms employed by VP9 and HEVC in capturing the motion and temporal redundancies within the metaverse video. This approach allowed for an accurate comparison and a thorough investigation of the efficiency and accuracy of the ME process.

The frame's UHD-1 resolution ensured high detail and smooth motion representation, essential for accurately assessing the ME algorithms' capabilities. Through this targeted analysis, the study sought to shed light on the strengths and weaknesses of VP9 and HEVC's ME techniques.

Proposed Metrics Included in Motion Estimation

A combination of three metrics was used to evaluate each codec.

When evaluating video quality, metrics such as the Structural Similarity Index Measure (SSIM) play a crucial role. SSIM is a perception-based index that captures changes in structural information within an image or between different images. It takes into account important perceptual phenomena like *luminance masking* and *contrast masking*. *Luminance masking* refers to the phenomenon where image distortions are less visible in bright areas, while *contrast masking* refers to the phenomenon where distortions are less noticeable in regions with significant activity or texture. Equation 3 shows the calculation of SSIM:

$$\mathrm{SSIM(x,y)} = [I(x,y) \cdot c(x,y) \cdot s(x,y)]^{\alpha} \quad (3)$$

where x and y are the two input images compared, $l(x,y)$ represents the luminance comparison between the images and captures the differences in brightness, $c(x,y)$ represents the contrast comparison and takes into account differences in contrast, $s(x, y)$ represents the structure comparison, capturing differences in structural information, and α is a parameter that controls the influence of each component and is typically set to 1. Each component is calculated as the average of local measurements obtained by dividing the images into smaller windows (i.e., blocks). Luminance comparison is based on mean intensity values; contrast comparison considers standard deviations and structure comparison evaluates the covariance of intensity values. Combining these components and raising the result to the power of α provides an SSIM value that measures similarity between the images, accounting for differences in luminance, contrast, and structure. Higher SSIM values suggest a higher level of structural similarity between the two images. In the test, SSIM is utilized to assess the reference PB and the current PB to determine whether ME should be computed, conserving energy when the frames exhibit significant similarities. Therefore, ME is performed when the SSIM is less than the selected threshold of 0.90.

The second metric incorporated into the VP9 and HEVC Matlab implementations is the number of computations performed in each step of the algorithms. Quantifying the complexity of the software through counting the executed operations gains a comprehensive understanding of the computational demands. It is essential to carefully assess the computational complexity of mathematical operations, particularly within programming loops, and strive to minimize resource-intensive calculations such as multiplications. Reducing this metric directly contributes to energy savings and subsequently reduces GHG emissions. Matlab's Profiler[9] proves to be a valuable tool in determining the complexity of different parts of the code, aiding in optimizing the software's efficiency.

The analysis follows by deriving operational emissions (O),[10] with a particular emphasis on the laptop's power consumption while executing each ME algorithm in Matlab. The computer utilized to do the tests was an ACER Aspire F5-573G, with the characteristics shown in **Table 1**.

FIGURE 2. Selected metaverse frame for this study.

TABLE 1. Laptop specifications.

Parameter	Value
RAM	16 GB
CPU	Intel(R) Core(TM) i7-7500U
CPU Frequency	2.7 GHz
OS	Windows 10
Number of cores	2

A power meter was utilized to measure the power consumption accurately. Consequently, to assess the energy consumed by the software for a specific task, Equation 4 is used:

$$O = E \cdot I \tag{4}$$

where E represents the energy consumed by the software for the task, measured in kWh. In this study, the task was defined as the execution of inter-prediction for one frame. E was determined by measuring the laptop's power consumption with a power meter, during idle state, and while running each algorithm in Matlab. The difference between these two values was then converted to kWh using the total processing time for each algorithm obtained from Matlab. The parameter I, denoting location-based marginal carbon intensity, was obtained and defined as 275 gCO_2Eq/kWh.[11] It embodies the carbon emissions associated with generating an extra unit of electricity at a specific location on the grid. This measure exemplifies the environmental impact of electricity generation in the selected region, Europe, to enhance result precision, as the tests were performed in Spain.

Results

Before conducting ME, the SSIM is obtained, yielding a value of 0.86. Since it falls below the chosen threshold (i.e., 0.90), the ME computation was carried out for the two codecs.

For HEVC-ME, detailed performance measurements were obtained for each step, including the number of computations performed and the corresponding processing time. The results are summarized in **Table 2**.

TABLE 2. HEVC-ME execution performance in Matlab.

Step	Computations	Processing Time (seconds)
Initial Search	2,753	0.068
Raster Search	548	0.029
Refinement Search	2,644	0.062

The analysis of the HEVC-ME computations yielded the following results:

- The SAD function was called 2,305 times during the Initial Search, accounting for 17% of the computing time of this search algorithm. It was called 484 times in the Raster Search, accounting for 14.3% of this step's processing time. In the Refinement Search, SAD was called 2,004 times, taking up 4.4% of the execution time for this part. SATD and subpixel interpolation were each called 64 times during the Refinement Search, constituting 33.6% and 32.9% of the total function time, respectively. Therefore, overall, SAD, SATD, and subpixel interpolation operations were the most demanding steps.
- The average power consumption was 8.1 W without the refinement step and increases to 14 W when the refinement search is included.
- The estimated operational emissions were 0.0006 gCO_2Eq without the refinement step and rise to 0.0009 gCO_2Eq when the refinement search is executed.

Furthermore, the computations and processing time for each of the VP9-ME steps were measured in Matlab, and the results are presented in **Table 3**.

TABLE 3. VP9-ME execution performance in Matlab.

Step	Computations	Processing Time (seconds)
Motion vector selection	28	0.0026
Motion vector clamping	1	0.0043
Motion vector scaling	1	0.0064
Block inter-prediction	18,471	0.072

In this case, the following results were obtained:

- The most time-consuming step is the "block inter-prediction" function, which accounted for 84.4% of the total execution time. Within this function, 47.2% of the time was dedicated to the sub-sample interpolation process.
- The average power consumption observed during the measurements was 5.1 W.
- The estimated operational emissions associated with the process were found to be 0.0004 gCO_2Eq.

The performance analysis of HEVC-ME and VP9-ME algorithms provides valuable insights into their computational requirements and energy consumption.

In HEVC-ME, the most time-consuming steps are the SAD, SATD, and subpixel interpolation functions. Particularly, the SATD function's substantial time consumption is due to the linearithmic time complexity *(nlogn)* of the absolute value of the Walsh-Hadamard transform, where n represents the size of the subpixel interpolated reference and current CUs. This significantly contributes to the overall execution time, resulting in higher power consumption and operational emissions.

Conversely, VP9-ME dedicates a considerable amount of time to sub-sample interpolation. This is due to VP9's utilization of an 8-tap fractional pixel interpolation filter, which enhances accuracy, although more computationally intensive than the 6-tap interpolation filter used in HEVC. Overall, VP9-ME demonstrates superior energy efficiency, leading to lower power consumption and estimated operational emissions compared to HEVC-ME.

These findings underscore the importance of understanding computational complexities and energy metrics when evaluating and optimizing video compression algorithms for improved efficiency and reduced environmental impact.

Discussion

In this study, we compared the environmental impact of video compression algorithms, specifically focusing on VP9 and HEVC. Our results revealed that VP9 emitted lower carbon emissions and consumed less power during execution than HEVC. This suggests that VP9 performs faster predicting sample frames, while HEVC operations involve higher complexity and more iterations in the search process.

To evaluate the carbon footprint of metaverse videos, we analyzed a 120-sec UHD-1 video with a frame rate of 60

1545-0279/24©2024SMPTE

frames/s. In the best-case scenario (i.e., VP9-ME), compressing each frame resulted in 0.0004 gCO_2Eq/frame emission. If this is extrapolated to the entire 120-sec video with 7,200 frames, the total estimated carbon emissions would be 2.88 gCO_2Eq. With a data center using 50% renewable energy, overall carbon emissions could be reduced to 1.44 gCO_2Eq.

Extending these findings to the vast number of metaverse users worldwide, there are approximately 400 million active users each month,[12] and assuming each user generates or interacts with an average of ten videos monthly, we can anticipate an annual carbon emission of approximately 69,120 metric tons of CO_2Eq. This carbon output is comparable to the annual emissions of hundreds of small to medium-sized power plants or tens of thousands of cars.[13] This highlights the significant environmental impact of metaverse videos on a global scale, comparable to specific industrial sectors in terms of carbon emissions. The magnitude of metaverse video emissions emphasizes the urgency for adopting sustainable practices in the rapidly expanding digital landscape.

Notably, the estimated 69,120 metric tons of CO_2Eq closely align in magnitude with the results published by Meta,[14] which state that 57,000 metric tons of CO_2Eq primarily originate from data centers processing videos and metaverse applications due to the nature of the company. This reinforces the need for increased attention to energy-efficient practices in video processing to mitigate the environmental impact of the metaverse.

It is essential to acknowledge the intricate nature of ME steps in HEVC and VP9, which leads to a non-linear relationship between the number of iterations and processing time. The complexity of the operations within each iteration is the primary factor responsible for the increased consumption of time and energy, as both processing time and energy usage exhibit a proportional connection. This emphasizes the importance of optimizing these algorithms to achieve a balance between computational efficiency and energy consumption in practical video coding applications.

Regarding the comparison of VP9 and HEVC in Matlab, although the processing time may be longer compared to other software platforms, the number of computations remains the same as it solely depends on the algorithm implementation. Thus, Matlab allows for a fair comparison of both codecs.

Given the high volume of videos today, even if the pollution were cut in half by using faster software, the associated emissions would still be a concern. This highlights the need for more energy-efficient video compression algorithms that use real power consumption data and metrics like Operational Emissions (O). These metrics can be employed during algorithm development stages, where real measurements can be taken and specific hardware data can be obtained

from suppliers for accurate calculations. Furthermore, these metrics can be incorporated into the algorithms as decision values to optimize efficiency.

In line with the current investigation, perception-based metrics such as SSIM can be employed to assess the resulting compressed video and be integrated into the compression algorithm. This integration optimizes the balance between perceived quality and computational considerations, including energy consumption. In regions characterized by high SSIM values, indicating substantial structural similarity, it becomes possible to skip the ME process, thereby reducing algorithmic complexity. Furthermore, in scenarios where CUs exhibit significant similarities, strategies like aggressive quantization, which reduces bit rate allocation without compromising perceived quality, could come into play.

Analyzing the efficiency of current algorithms, as demonstrated in the Matlab code developed for this work, is an essential initial step towards a more sustainable metaverse.

Conclusion

The rapid growth of mobile devices, internet accessibility, video-on-demand services, social media, and the emergence of the metaverse, have led to a significant surge in digital video traffic. However, this growth has also increased processing complexity in current compression methods, resulting in higher energy demand and associated pollution.

Addressing this issue requires lighter compression methods that can efficiently handle video encoding. This study compared the performance of VP9 and HEVC, focusing on their ME processes, using a custom implementation in Matlab for a metaverse video sample frame. The findings revealed that both codecs employ thousands of computations to predict a single video frame. While both codecs possess remarkable features and designs, the results underscore the urgency of developing more efficient and sustainable video compression techniques. Reduced video processing times and smaller file sizes enable quicker upload, download, and streaming, enhancing user experience and reducing overall energy demand on servers, data centers, and network infrastructure.

Future research of this work will address inefficiencies in the full compression algorithms, optimize the entire compression pipeline, and promote eco-conscious behavior among users. The overarching objective is to establish criteria for novel video compression algorithms that combine ecological sustainability with superior experiential quality. To facilitate this pursuit, the Matlab code for VP9 and HEVC is accessible through the Harmony Valley research project's website.[15]

The code for two Android apps dedicated to video compression and "eco-cam" functionality has also been shared. This initiative invites enthusiastic researchers to partake in the collective endeavor of advancing video compression standards with heightened efficiency. Energy consumption will be a primary design constraint in technological advancements, ensuring that people can enjoy digital technology without causing harm to the planet. Prioritizing energy efficiency in video processing for the metaverse and other digital applications is crucial. By optimizing video compression techniques and reducing computational complexity, we forge the way for a greener digital landscape.

Acknowledgments

The author wishes to thank the European University of Madrid and Banco Santander Universidades. This work was supported in part by a grant from the Singular Alumni Award.

References

1. M. Uitto, "Energy consumption evaluation of H.264 and HEVC video encoders in high-resolution live streaming," Proc. *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1-7, New York, NY, 2016, doi: 10.1109/WiMOB.2016.7763234.
2. Dan Grois, Detlev Marpe, Tung Nguyen, Ofer Hadar, "Comparative Assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders for Low-delay Video Applications," *Proc. SPIE 9217, Applications of Digital Image Processing XXXVII*, 92170Q, 23 Sep. 2014; [Online]. Available: https://doi.org/10.1117/12.2073323
3. R. Monnier, K. Jerbi, and M. Uitto, "Specification of Power Efficient Encoder-Transcoder," Sep. 2017. Accessed Dec. 23, 2022. [Online]. Available: https://convince.wp.imtbs-tsp.eu/files/2017/09/CONVINcE-D2.2.2-Updated-specification-of-power-efficient-encoder-V1.01.pdf
4. A. Katsenou, J. Mao, and I. Mavromatis, "Energy-Rate-Quality Tradeoffs of State-of-the-Art Video Codecs," *Electr. Eng. and Syst. Sci., Image and Video Processing*, Oct. 2022.
5. T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh, and J. Ostermann, "A Comprehensive Video Codec Comparison," *APSIPA Transactions on Signal and Information Processing*, Vol. 8, Nov. 2019, doi: 10.1017/atsip.2019.23.
6. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE Trans. on Circ. and Syst. for Vid. Tech.* 22 (12):1669–1684, Dec. 2012, doi: 10.1109/tcsvt.2012.2221192.
7. A. M. Joshi, M. S. Ansari and C. Sahu, "VLSI Architecture of High Speed SAD for High Efficiency Video Coding (HEVC) Encoder," *Proc. 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, pp. 1-4, Italy, 2018, doi: 10.1109/ISCAS.2018.8351271.
8. J. Hunt and A. Design, "VP9 Bitstream & Decoding Process Specification," Mar. 2016. Accessed: Nov. 15, 2022. [Online]. Available: https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf
9. Mathworks, "Run code and measure execution time to improve performance - MATLAB - MathWorks España," mathworks.com. Accessed Jan. 05, 2023. [Online]. Available: https://mathworks.com/help/matlab/ref/profiler-app.html
10. G. S. Foundation, "Software Carbon Intensity Standard," GitHub, Nov. 01, 2021. Accessed Dec. 16, 2023. [Online]. Available: https://github.com/Green-Software-Foundation/sci
11. "Greenhouse gas emission intensity of electricity generation — European Environment Agency," Jan. 26, 2023, www.eea.europa.eu. Accessed Dec. 20, 2022. [Online]. Available: https://www.eea.europa.eu/data-and-maps/daviz/co2-emission-intensity-12#tab-googlechartid_chart_11
12. Metaverse of Things, "Metaverse Users Worldwide – Metaverse Statistics to Prepare for the Future," April. 1, 2023. Accessed June 1, 2023. [Online]. Available: https://metaverseofthing.com/metaverse/metaverse-users-worldwide/
13. U.S. Environmental Protection Agency (EPA), "Greenhouse Gas Equivalencies Calculator." www.epa.gov, Aug. 28, 2015. Accessed July 1, 2023. [Online]. Available: https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator#results
14. Meta, "2021 Sustainability report." Accessed Jan. 15, 2023. [Online]. Available: https://sustainability.fb.com/wp-content/uploads/2022/06/Meta-2021-Sustainability-Report.pdf
15. N. Molinero, "Harmony Valley – Research Project Resources, [Online]. Available: https://linktr.ee/harmonyvalley

About the Author

Natalia Molinero Mingorance, a telecommunications engineer, is balancing her research and development commitments in Harmony Valley with her new role at Airbus Defense and Space, where she has the opportunity to explore her other interests, such as wireless communications and cybersecurity.

Presented at the 2023 Media Technology Summit. Hollywood, CA, 16-19 October 2023.

DOI: 10.5594/JMI.2024/EBVF7405
Date of publication: 24 January 2024