iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68746

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

3D-Based Compression Framework for High Quality Video Streaming

Mr. D. Veeraswamy[1], P.V.N. Sriram[2], S. Viharith Goud[3], Y. Yashwanth Kumar[4]
[1]*Asst. Professor, Dept. Electronics and Communication EngineeringInstitute of Aeronautical Engineering, Hyderabad, India*
[2, 3, 4]*Dept. Electronics and Communication EngineeringInstitute of Aeronautical Engineering, Hyderabad, India*

Abstract: ***Video compression plays a pivotal role in managing the storage and transmission of multimedia content, especially in bandwidth-constrained environments. Nowadays, volumetric video has emerged as an attractive multimedia application, which provides highly immersive watching experiences. How- ever, streaming the volumetric video demands prohibitively high bandwidth. Thus, effectively compressing its underlying pointcloud frames is essential to deploying the volumetric videos. The existing compression techniques are either 3D-based or 2D-based, but they still have drawbacks when being deployed in practice. The 2D-based methods compress the videos in an effective but slow manner, while the 3D-based methods feature high coding speeds but low compression ratios. In this paper, we propose a 3D-based compression framework that reaches both a high compression ratio and a real-time decoding speed. This project presents an integrated hybrid video compression framework that combines traditional techniques such as motion estimation and Discrete Cosine Transform (DCT) coding with machine learning methodologies. The framework aims to improve compression efficiency and maintain high-quality video output through intelligent prediction and optimization.***

Index Terms: ***Volumetric, Real-time decoding, Integrated hy- brid video compression framework, motion estimation, DCT coding, efficient video processing***

I. INTRODUCTION

In the era of immersive media and advanced digital expe- riences, volumetric videos have emerged as a groundbreaking medium, enabling the capture and rendering of 3D spaces with unprecedented detail and realism. These videos, which encompass a full spatial representation of scenes, offer viewers an interactive and dynamic experience, revolutionizing field such as virtual reality (VR), augmented reality (AR), and mixed reality (MR). However, the sheer volume of data generated by volumetric videos presents significant challenges in terms of storage, transmission, and real-time streaming. A real-time compression framework specifically designed to address the unique demands of volumetric video streaming. This framework not only enhances the feasibility of deploying volumetric videos across different platforms but also pushes the boundaries of what is possible in immersive media. By optimizing the balance between compression efficiency and computational complexity, the framework enables real- time processing and playback, making high-quality volumetricvideo experiences accessible to a broader audience.

II. RELATED WORK

A. Video Streaming

Video streaming has become a huge part of how we consume media today. Whether you're watching movies, TV shows, live events, or even playing video games, streaming offers a convenient way to access content on demand. The Services like Netflix, Disney+, Amazon Prime Video, and HBO Max offer vast libraries of movies and TV shows. For live events, platforms like Twitch or YouTube Live might be more relevant. Streaming quality can vary based on your internet connection and the service you're using. Many platforms offer different resolutions, including HD, 4K, and sometimes even HDR. Streaming can be done on various devices including smart TVs, laptops, tablets, and smartphones. Some devices, like Roku or Amazon Fire TV, are designed specifically for streaming.

B. Cloud Compression

Cloud compression refers to the process of compressing data before storing it in the cloud or transmitting it over the internet. This can apply to a variety of data types, including video, images, documents, and more. By compressing data, you can reduce the amount of storage space needed in the cloud. This can significantly lower storage costs, especially for large volumes of data. Compressed files are smaller insize, which means they can be uploaded or downloaded faster.

This is particularly useful for transferring large files or datasets. Cloud compression is a vital tool for optimizing storage and transfer efficiency in cloud environments. By choosing the right compression methods and understanding the implications for data quality and processing, you can effectively manage costs and improve performance.

III. METHODOLOGY

The electronics industry has achieved a phenomenal growth over the last two decades, mainly due to the rapid advances in integration technologies, large-scale systems design - in short, due to the advent of VLSI. The number of applications of integrated circuits in high-performance computing, telecommunications, and consumer electronics has been rising steadily, and at a very fast pace. Typically, the required computational power (or, in other words, the intelligence) of these applications is the driving force for the fast development of this field. Figure 1.1 gives an overview of the prominent trends in information technologies over the next few decades. The current leading- edge technologies (such as low bit-rate video and cellular communications) already provide the end-users a certain amount of processing power and portability. As more and more complex functions are required in various data processing and telecommunications devices, the need to integrate these functions in a small system/package is also increasing. The level of integration as measured by the number of logic gates in a monolithic chip has been steadily rising for almost three decades, mainly due to the rapid progress in processing technology and interconnect technology.

A. Patch-Based Encoding

Each patch is encoded using the VVC standard, with specific adaptations to optimize performance for volumetric content.
Localized Compression: By focusing on smaller patches, the encoder can apply more localized compression techniques, reducing the overall data rate.
Adaptive Techniques: Different patches may require different levels of compression based on their content and importance within the scene. For example, patches representing the foreground might receive higher quality encoding compared to those in the background.

B. Adaptive Compression Techniques

The framework incorporates adaptive compression techniques that adjust based on the viewer's perspective and interaction. Region of Interest (ROI) Encoding: Areas of the scene that are more relevant to the viewer (e.g., objects in focus or areas of interaction) receive higher-quality encoding. Less important regions are compressed more aggressively to save bandwidth.
Dynamic Adjustment: The framework dynamically adjusts compression parameters in real time based on the viewer's position, ensuring optimal quality where it is most needed.

C. Real-Time Encoding and Decoding

Parallel Processing: Real-time performance is achieved through parallel-processing techniques.
Multi-Core Processing: The encoding and decoding processes are distributed across multiple processor cores to accelerate computation.
Pipeline Architecture: A pipeline approach is used to manage different stages of encoding and decoding concurrently, reducing latency.

D. Low-Latency Technique:

Technique implemented to minimize delays.
Real-Time Buffer Management: Efficient buffer management strategies are employed to handle incoming and outgoing data with minimal delay.

E. Video Processing

1) Encoder

Global Motion Estimation. In a volumetric video, the neighbouring PC frames are expected to be highly similar. The global motion estimation helps extract a compact transformation matrix Mg that describes the major difference between an I-frame and a P-frame. We start by filling the encoding buffer with a GOP of N frames, and always set up the first one as the I-frame and the others as the P-frames. The global motion estimation is carried out for each P-frame. We first align the centroids of the I-frame and the P-frame, introducing a better initial status to the ICP process. The centroid alignment results in a translation vector R qg. After ICP is performed on the translated I-frame and the P-frame, a transformation matrix $M'g$ is calculated.

2) Patch Deformation

In the patch deformation stage, we generate a common patch for a patch group to align the points from different frames. After deforming a patch into its common patch, we color the points via a distance-weighted interpolation method. In the end, each patch is represented by an octree of the common patch and the colors (the I-patch) or the color residuals (the P-patches). The deviate patches have their octree and colors.

Color Interpolation and Compensation: Though the common patch represents the unifying geometric structure of a patch group, the color information of each patch might be different because of the changing lighting conditions. Therefore, we introduce a distance weighted interpolation method to color the common patch for *N* the transformation matrices for each P-patch in addition to the compressed color information.

3) Decoder

When the decoder receives the bitstream, we follow three phases to decode the PC frames in a GOP:

1) decoding the common patch
2) decoding the I-frame
3) decoding the P-frames.

Common Patch Decoding: According to the GOP header, the decoder understands the GOP size *N* and the number of patch groups *L*. With the parameters, the decoder reads *L* compressed octrees and feeds them to the specified entropy decoder. With these octrees, we are able to recover all points in the common patches by calculating the coordinates of the leaf nodes. The coordinate residuals are added back if the encoder chooses to preserve them. It is worth noting that since the common patches are discovered independently, it is trivial to decode the octrees and recover the coordinates in parallel.

I-frame Decoding: The decoder recovers the I-frame by associating the colors with the common patches. According to the patch index in an I-patch header, the compressed colors are paired with the corresponding common patch. After decoding the colors with the color decoder, the common patch could be straightforwardly colored because the colors are organized in the order of extracting the points from the octree. All the recovered I-patches then form the I-frame to be rendered.

P-frame Decoding: Decoding a P-frame demands recovering all P-patches belonging to it. Recovering a P-patch is similar to recovering an I-patch except that two additional steps must be involved:

1) We should use the transformed common patch as the skeleton instead
2) the colors are recovered by adding the decoded color residuals to the colors of the corresponding I-patch. For those deviate patches, they are decoded with their own octrees, and the remaining steps are the same as the I-patches.

F. Challenges in Streaming Volumetric Videos

Streaming volumetric videos presents several unique challenges:

Data Volume: The sheer amount of data needed to represent a 3D scene in real time can be overwhelming.

Bandwidth Constraints: Limited network bandwidth can restrict the quality of streamed content.

Latency: Delays in processing and transmission can disrupt the user experience, particularly in interactive applications.

Quality Preservation: Maintaining high visual quality while reducing data size is critical to ensure an immersive experience.

G. Quality Assessment Metrics

Quality assessment metrics evaluate the fidelity and perceptual quality of compressed videos:

1) PSNR (Peak Signal-to-Noise Ratio) measures fidelity by quantifying the difference between original and compressed frames in terms of noise introduced during compression. Higher PSNR values indicate less perceptible distortion.

2) SSIM (Structural Similarity Index) assesses structural similarity between images, considering luminance, contrast, and structure. It correlates well with human perception and is sensitive to artifacts introduced during compression.

3) VMAF (Video Multi-method Assessment Fusion) integrates multiple quality metrics into a unified score, considering perceptual aspects of video quality. It takes into account human visual sensitivity to artifacts and provides a comprehensive evaluation of video quality.

IV. IMPLEMENTATION

The workflow for the given code snippet revolves around video compression using techniques like quadtree decomposition, DCT coding, motion estimation, and ultimately, quality enhancement and audio compression. Here's a detailed workflow breakdown based on the provided code:

A. *Workflow Overview*

Initialization and Parameters Setup: Define parameters such as blockSize, searchWindow, Thresh, and Qscale for video processing and compression.
Read Video File: Load the input video file (fileName) using VideoReader.
Setup Output Video: Define an output video file (outputVideoFile) and configure a VideoWriter object (outputVideo1) to write processed frames.

B. *Frame Processing Loop:*

Iterate through each frame of the video (for k=frameStart:frameEnd):
Read the current frame (curFrame).
Resize the frame to fit the specified blockSize.
Perform quadtree decomposition and DCT coding using varSizeDCTcoder.
Store processed frames in quadtreeFrames.
Write processed frames to the output video file (outputVideo1).

C. *Motion Estimation and Compression*

Implement full search motion estimation (fullSearchME) to find optimal motion vectors and generate difference frames.
Apply DCT coding and quantization to compress frames efficiently.

D. *Compression Quality Enhancement*

Apply non-local means denoising or other preprocessing techniques to enhance frame quality.
Compress frames aggressively by resizing or adjusting encoding parameters.

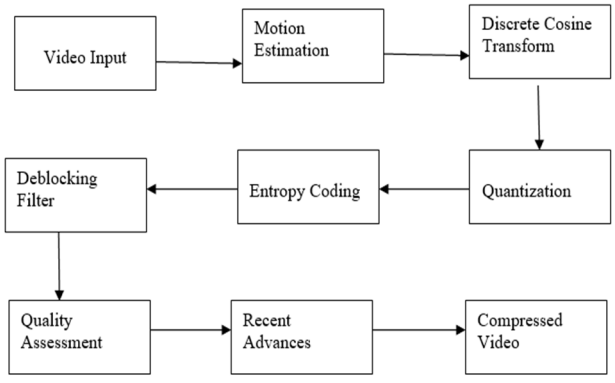


Figure 1: Block Diagram

1) Video Input: Raw video frames are fed into the compression pipeline.
2) Motion Estimation (Block Matching): Determines motion vectors between consecutive frames to predict motion.
3) Discrete Cosine Transform (DCT): Converts spatial data into frequency coefficients.
4) Quantization: Reduces precision of DCT coefficients to achieve compression.
5) Entropy Coding: Applies context-adaptive binary arithmetic coding or context-adaptive variable-length coding to further reduce data size.
6) Deblocking Filter: Smooths block artifacts at compression boundaries.

E. Advanced Video Coding Standards

1) *H.264/AVC:* H.264/AVC (Advanced Video Coding) introduced several key features that revolutionized video compression. It implemented variable block-size motion compensation, allowing different block sizes (e.g., 16x16, 8x8) for more precise motion prediction. This flexibility improved compression efficiency by adapting block sizes to different types of motion within frames. Context-adaptive binary arithmetic coding (CABAC) enhanced entropy coding efficiency by dynamically adjusting coding contexts based on local information, improving compression ratios. The deblocking filter in H.264/AVC reduced artifacts at block boundaries, enhancing visual quality by smoothing out discontinuities between adjacent blocks.
2) *H.265/HEVC:* H.265/HEVC (High Efficiency Video Coding) builds upon H.264 by focusing on further enhancing coding efficiency. It achieves better compression ratios without sacrificing quality, making it suitable for high-resolution video a applications up to 8K. HEVC introduces improved intra prediction modes to handle spatial redundancy more effectively, reducing bitrate requirements for static scenes. This standard optimizes encoding complexity and bitrate allocation, providing significant improvements in video quality at lower bitrates compared to H.264.

F. Encoding Process

encode_video function takes an input video file (input_file), applies compression with a specified bitrate (default is '2000k'), and outputs the compressed video to output_file.It checks if the input file exists before starting the encoding process.Uses subprocess.run() to execute the ffmpeg command and checks for errors during execution.
Bitrate may be varied according to the requirement the lesser the bitrate more the compression and quality of the video may be reduced for lower bitrate such as 1000k or less. If the bitrate is higher the lesser will be the compression but gives much better quality video.

G. Decoding Process

decode_video_with_ffmpeg function takes an encoded video file (input_file), decodes it using ffmpeg, and saves the decoded video to output_file. It specifies -c:v copy to copy the video stream directly (since we're not re-encoding), and -c:a aac to encode the audio with AAC codec.Uses subprocess.run() to execute the ffmpeg command and handles any errors that occur during decoding.

H. Results

The input contains a video file which is a bunch of frames which is further encoded and compressed to a reasonable size while maintaining higher quality.
Our Framework helps lossless compression techniques to ensure the video quality without compromising while being highly compressed.
The File size will be reduced to more than a half of the original file size and contains great quality which is absolutely meant for Video Streaming.
The decoding process takes the encoded video or the compressed video and decodes it for further process. In this way the method works for high quality volumetric videos and also helps in compression of conventional videos.

I. Input/Output Video Properties

1) *Sample 1*

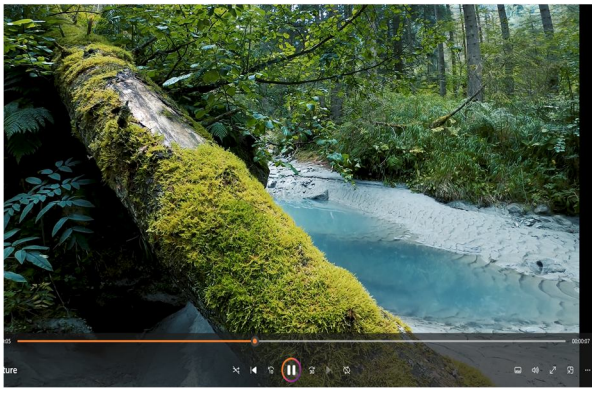


Figure 2: Original Video

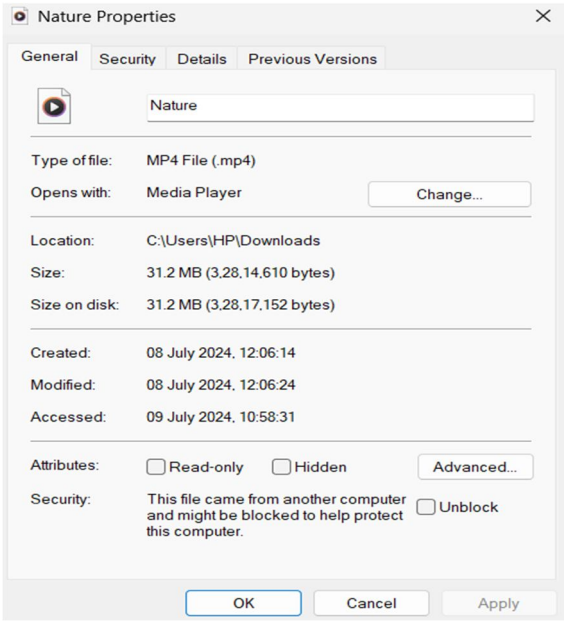


Figure 3: Original Video Properties

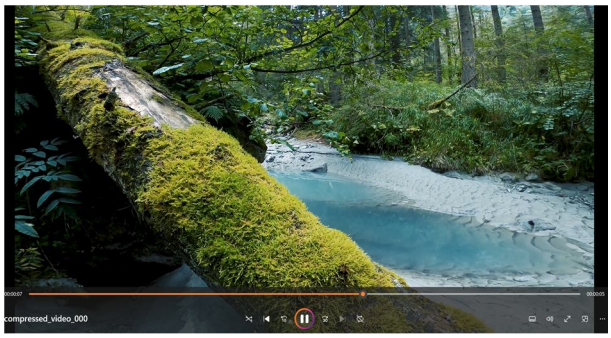


Figure 4: Compressed Video

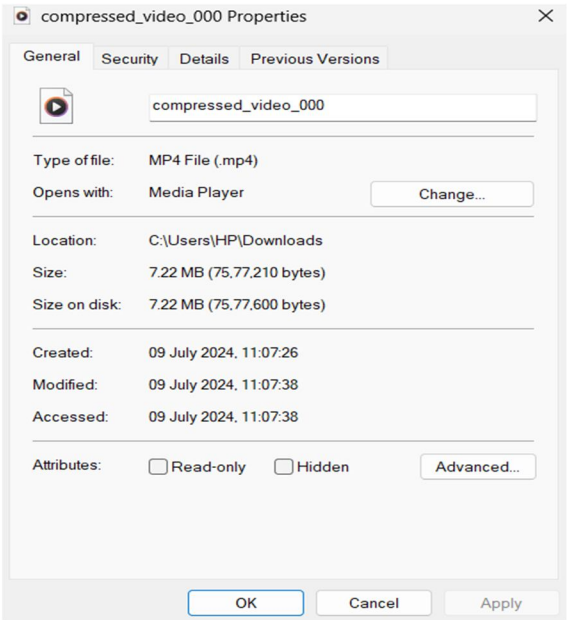


Figure 5: Compressed Video Properties

2) Sample 2

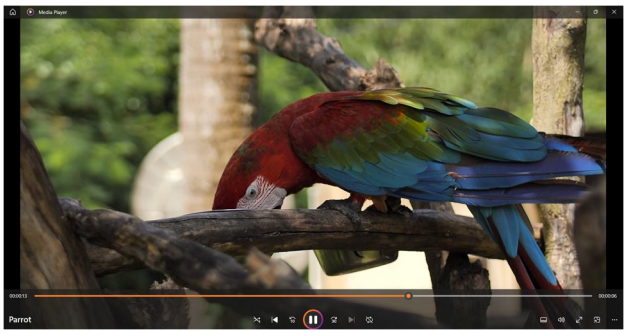


Figure 6: Original Video

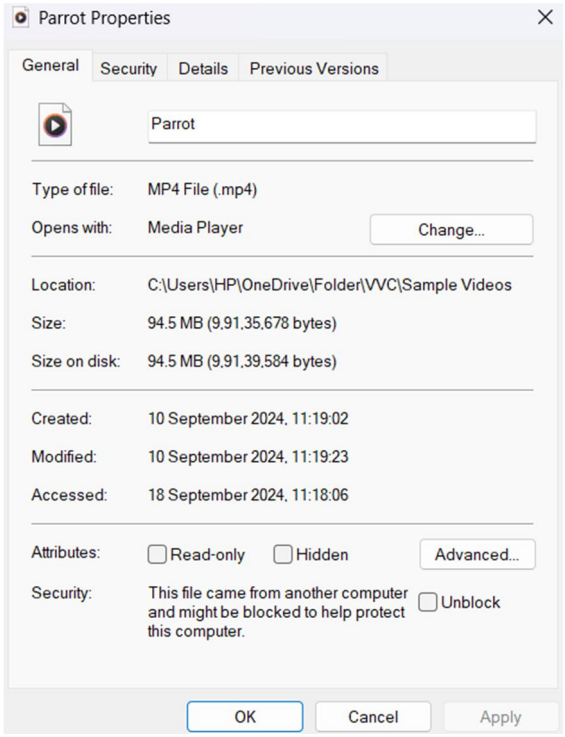


Figure 7: Original Video Properties

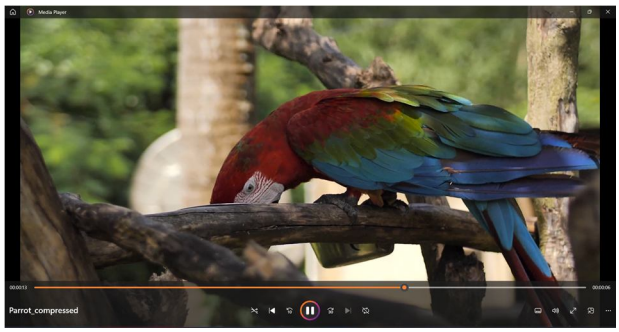


Figure 8: Compressed Video

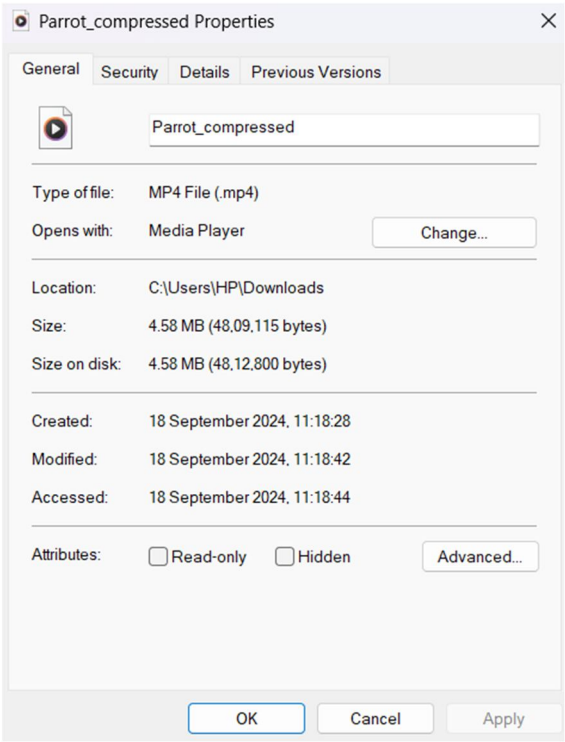


Figure 9: Compressed Video Properties

V. CONCLUSION

In the proposed methodology presented for video compression in the provided code snippet showcases a comprehensive approach to enhancing storage efficiency and transmission quality of digital video content. By leveraging techniques such as quadtree decomposition, discrete cosine transform (DCT) coding, motion estimation, and advanced compression parameters, the workflow effectively reduces redundancy while preserving visual fidelity.

Video compression frameworks play a crucial role in managing and optimizing the storage and transmission of video content. By employing various compression techniques and algorithms, these frameworks significantly reduce file sizes while maintaining acceptable quality levels, enabling efficient streaming and playback across diverse devices and network conditions.

A. Application Scope

Video compression frameworks are indispensable in numerous applications, from online video streaming services and video conferencing to multimedia storage and broadcasting. Their role is fundamental in enabling seamless and high-quality video experiences for users worldwide.

Key Contributions and Findings

1) Efficient Compression Techniques: The integration of quadtree decomposition allows adaptive partitioning of frames into variable-sized blocks, optimizing the encoding process based on spatial complexity. This method not only reduces bitrate but also enhances compression efficiency by focusing computational resources on significant image regions.

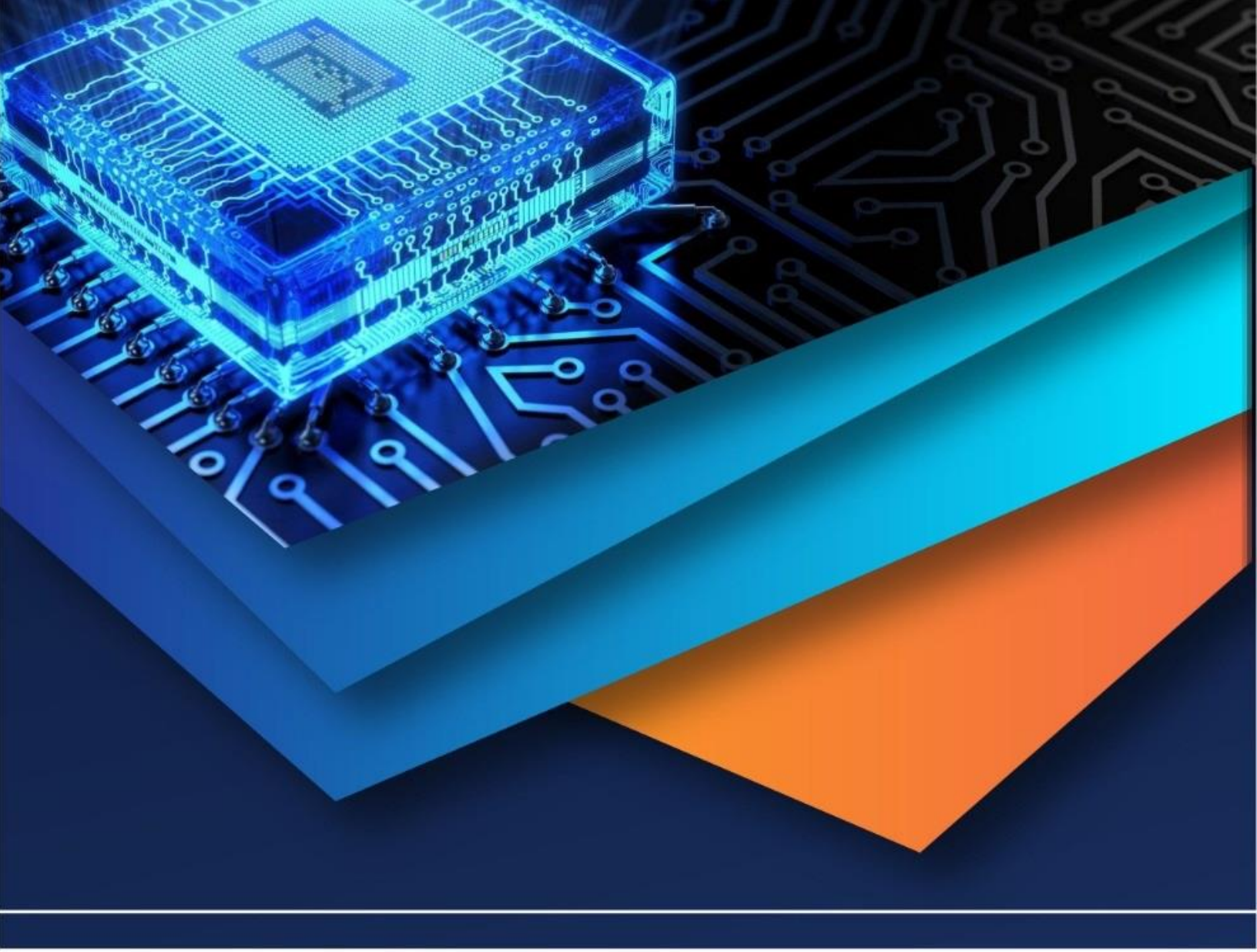
2) Motion Estimation and Compensation: Full search motion estimation combined with DCT-based coding facilitates accurate prediction of inter-frame motion, reducing temporal redundancy. This approach improves compression ratios without compromising perceptual quality, crucial for maintaining smooth video playback and minimizing storage requirements.

3) Quality Assessment and Enhancement: The inclusion of quality assessment metrics such as PSNR provides quantitative measures of compression effectiveness. Techniques like non-local means denoising and aggressive resizing further refine video quality, ensuring that compressed outputs meet perceptual expectations.

4) Practical Implementation: The utilization of external tools like ffmpeg for audio handling and final video compilation enhances the workflow's robustness and scalability. This integration streamlines the process of combining compressed video with optimized audio, facilitating seamless multimedia content delivery.

REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," IEEE Trans. Consum. Electron., vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.
[2] T. Wiegand,G. J. Sullivan,G. Bjontegaard, and A. Luthra, "Overview ofthe H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, 2003.
[3] S. Ma, T. Huang, C. Reader, andW. Gao, "AVS2? Making video coding smarter [standardsin a nutshell]," IEEE Signal Process. Mag., vol. 32, no. 2, pp. 172–183, Mar. 2015.
[4] G. J. Sullivan,J.-R. Ohm,W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC)standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[5] A. Norkin et al., "HEVC deblocking filter," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1746–1754, Dec. 2012.
[6] P. List, A. Joch, J. Lainema,G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 614–619, Jul. 2003.
[7] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding atlow bitrates," IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 7, pp. 849–866, Nov. 1998.
[8] S.-M. Lei, T.-C. Chen, and M.-T. Sun, "Video bridging based on H.261 standard," IEEE Trans. Circuits Syst. Video Technol., vol. 4, no. 4, pp. 425–437, Aug. 1994.
[9] L. Fan, S. Ma, and F. Wu, "Overview of AVS video standard," in Proc. IEEE Int. Conf. Multimedia Expo (ICME), vol. 1, Jun. 2004, pp. 423–426.
[10] C.-M. Fu et al., "Sample adaptive offsetin the HEVC standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1755–1764, Dec. 2012.
[11] C.-Y. Tsai et al., "Adaptive loop filtering for video coding," IEEE J. Sel. Topics Signal Process., vol. 7, no. 6, pp. 934–945, Dec. 2013.
[12] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," IEEE Trans. Image Process., vol. 23, no. 8, pp. 3336–3351, Aug. 2014.
[13] S. Ma, X. Zhang, J. Zhang, C. Jia, S. Wang, and W. Gao, "Nonlocal in-loop filter: The way toward nextgeneration video coding?" IEEE MultiMedia, vol. 23, no. 2, pp. 16–26, Apr./Jun. 2016.
[14] X. Zhang et al., "Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 10, pp. 2177–2188,Oct. 2017.
[15] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifactsreduction by overlapped-block transform coefficient estimation with block similarity," IEEE Trans. Image Process., vol. 22, no. 12, pp. 4613–4626, Dec. 2013 17.
[16] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Lowrank decomposition-based restoration of compressed images via adaptive noise estimation, " IEEE Trans. Image Process., vol. 25, no. 9, pp. 4158–4171, Sep. 2016.
[17] X. Zhang, R. Xiong, W. Lin, S. Ma, J. Liu, and W. Gao, "Video compression artifact reduction via spatio-temporal multi-hypothesis prediction," IEEE Trans. Image Process., vol. 24, no. 12, pp. 6048–6061, Dec. 2015.
[18] Y. LeCun, Y. Bengio, andG. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
[19] K. Zhang,W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond aGaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[20] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifactsreduction by a deep convolutional network," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 576–584.
[21] Rufael Mekuria, Kees Blom, and Pablo Cesar. 2016. Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. IEEE Transactions on Circuits and Systems for Video Technology 27, 4 (2016), 828–842.
[22] Rufael Mekuria and Pablo Cesar. 2016. MP3DG-PCC, Open Source Software Framework for Implementation and Evaluation of Point Cloud Compression. In Proceedings of the 24th ACM International Conference on Multimedia. 1222–1226.
[23] Jounsup Park, PhilipAChou, and Jenq-Neng Hwang. 2019. Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9, 1 (2019), 149–162.
[24] Feng Qian, Bo Han, Jarrell Pair, and Vijay Gopalakrishnan. 2019. Toward Practical Volumetric Video Streaming on Commodity Smartphones. In Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications. 135–140.
[25] ITUR Rec. 1995. BT 601: Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios. ITU-R Rec. BT 656 (1995).
[26] Ruwen Schnabel and Reinhard Klein. 2006. Octree-based Point-Cloud Compression. In Proceedings of the 3rd Eurographics. 111–120.
[27] Sebastian Schwarz, Gaëlle Martin-Cocher, David Flynn, and Madhukar Budagavi. 2018. Common Test Conditions for Point Cloud Compression. Document ISO/IECJTC1/SC29/WG11 w17766, Ljubljana, Slovenia (2018).
[28] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. 2018. Emerging MPEG standards for Point Cloud Compression. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9, 1 (2018), 133–148.
[29] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. 2009. Generalized-icp. 435.
[30] Jacopo Serafin and Giorgio Grisetti. 2015. NICP: Dense Normal based Point Cloud Registration. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems. 742–749.
[31] Yiting Shao, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li. 2017. Attribute Compression of 3D Point Clouds Using Laplacian Sparsity Optimized Graph Transform. In 2017 IEEE Visual Communications and Image Processing. 1–4.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY