International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429

e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

Scrambling-Driven Optimization Framework for H.264 Video Compression

Vanila sildas[1], Mannemela Sai Srithaja[2], Manasi Konidala[3], R Srinivasan[4]
[1]Assistant Professor, SRM Valliammai Engineering college, Kattankulathur, India.
[2,3]UG - Information Technology, Sri Sivasubramaniya Nadar college of engineering, Kalavakkam, India.
[4]Professor, Department of Information Technology, Sri Sivasubramaniya Nadar college of engineering, Kalavakkam, India.
Emails: *vanilas.eee@srmvalliammai.ac.in[1], mannemela2110216@ssn.edu.in [2], manasi2110294@ssn.edu.in [3], srinivasanr@ssn.edu.in [4]*

Abstract
This paper presents a framework for enhancing the H.264 video compression standard by adding scrambling techniques and Run Length Encoding (RLE) to the encoding process. The new approach adds XOR-based scrambling, the Arnold Transform, and pixel swapping following quantization and zigzag scanning to enhance data dispersion and structure transformation prior to entropy coding. Rather than applying traditional Huffman coding, Context-Based Adaptive Binary Arithmetic Coding (CABAC) is employed to enhance compression efficiency. The framework was simulated using OpenCV, NumPy, and Matplotlib. Experimental results indicate that the enhanced H.264 encoder achieves a 7% improvement in the average Peak Signal-to-Noise Ratio (PSNR) and an 18% bitrate saving over the baseline with only a 5–7% encoding time penalty. Additional comparisons of the individual scrambling techniques indicate that Arnold Transform offers a good tradeoff between visual quality and compression ratio. This approach has complete compatibility with standard H.264 decoding, thereby offering a seamless and effective enhancement of the compression process.
Keywords: *Compression; scrambling; H.264; Arnold; Run Length Encoding*

1. Introduction
The H.264/AVC (Advanced Video Coding) standard is one of the most widely employed video compression methods due to its efficiency in bit reduction with minimal impact on video quality. The standard is primarily focused on compression at the cost of possible optimizations that can further optimize the encoding process. Due to the growing need for low-complexity video storage and transmission, the need arises to look for ways in which the overall encoding performance is enhanced while, simultaneously, low computational overhead is achieved. In this work, we introduce a novel improvement framework using scrambling methods for H.264 video coding to improve the encoding process. The core idea is based on the use of scrambling methods at the encoding process, which are inverted at the decoding process. To this end, we integrate XOR-based scrambling, the Arnold transform, and pixel swapping in the encoding process. These methods are particularly tailored to change the spatial order of pixel information in such a manner that the similarity of neighboring pixels is minimized, and the efficiency of the following compression stages is maximized. The employment of these scrambling methods prior to the use of standard H.264 transformations, i.e., the Discrete Cosine Transform (DCT), is intended to additionally minimize redundancy in the original video data, thereby enhancing compression efficiency. During the decoding process after transmission or storage of the encoded video, the descrambling process is employed to restore the original video data, hence preserving the video quality.This paper assesses the efficacy of the suggested framework based on compression efficiency in terms of bitrate savings and video quality with respect to measures like PSNR and SSIM. The results indicate that combining these scrambling methods into the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429
e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

encoding process offers a lightweight improvement over the H.264 compression framework, making it more efficient for real-time video applications. Over the past few years, several new image and video compression standards have been suggested to improve storage and transmission efficiency. As per [1], video compression methods like H.264 have gained popularity because of their better performance in terms of bitrate savings and visual quality. Their critique further stresses the necessity to enhance prevailing standards and to investigate hybrid schemes that blend classical compression with other processing methods, like scrambling and transform-based enhancements, in harmony with the aims of this study. [2] examines the optimization of the H.264 encoder for real-time high-definition (HD) video encoding, with a focus on minimizing encoding delays while preserving the quality and compression efficiency of the encoded video. The paper points out some major optimization strategies, such as improving motion compensation and prediction mechanisms and employing parallel processing techniques to achieve real-time performance. This strategy for enhancing encoding efficiency is in line with our suggested framework, which also seeks to maximize the encoding process, albeit using scrambling methods instead of conventional optimization methods. [3] offers a collection of state-of-the-art optimization methods that optimize video compression via content-adaptive encoding, artificial intelligence-based scene classification, and region-of-interest (ROI) detection. These approaches seek to assign bits more efficiently, ensuring high perceptual quality while lowering total bitrate. [4] offer an in-depth overview of modern video compression and optimization technology, highlighting the advancement of codecs by groups like MPEG, Google, and Apple. Their review explores the underlying principles of video streaming and Quality of Experience (QoE), presenting both subjective and objective approaches to QoE measurement. The authors identify the challenges of preserving video quality in the face of growing demands for high-definition streaming and outline future research directions to improve video service delivery. This research emphasizes incorporating new methods into existing compression standards, consistent with the aims of the present research that investigates scrambling-based improvements to the H.264 encoding scheme. [5] presents nine best practices for encoding H.264, including highlighting the significance of choosing the correct container formats, optimizing the settings for encoding, and checking for compatibility with varied devices and platforms. This thorough guide acts as a great source of knowledge in learning about the real-world application and optimization of H.264 in current video workflows. Li and Drew [6] give a detailed account of the development of video coding standards, with emphasis on H.264/AVC, H.265/HEVC, and H.266/VVC. Their discussion brings out improvements in compression efficiency and video quality with each new standard. The authors also touch upon the growing computational complexity of these newer standards, highlighting the trade-offs between compression benefits and processing demands. This review highlights the need to strike the right balance between the efficiency and complexity of creating and implementing contemporary video coding technologies. [7] offers a detailed description of the H.264/AVC video coding standard, including its design goals, technical features, and performance advantages. The standard introduces new coding tools like variable block-size motion compensation, multiple reference frames, and an in-loop deblocking filter, which together improve compression efficiency and picture quality. These technologies allow H.264/AVC to realize about 50%-bit rate savings over earlier standards like MPEG-2, without any loss of visual quality. The authors further explain the standard's flexibility across different applications, such as video conferencing, broadcasting, and streaming, and its versatility and popularity in the market. [8] introduced the Context-Based Adaptive Binary Arithmetic Coding (CABAC) method in the H.264/AVC video compression standard. CABAC greatly improves compression efficiency using context modeling and adaptive arithmetic coding, enabling more precise probability estimation of syntax elements. This technique performs better than conventional entropy coding methods, including

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429
e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

Context-Adaptive Variable Length Coding (CAVLC), as it decreases bitrates without reducing video quality. The use of CABAC has been instrumental in attaining high compression rates characteristic of H.264/AVC, and it has become a staple in contemporary video coding techniques. [9] came up with a new intra prediction method by processing directly in the transform domain, with the hope of lowering the computational complexity during video encoding. Their technique works by using transformed and quantized neighbouring blocks to create prediction blocks that minimize each DC and AC coefficient separately. They suggest two prediction techniques: Full Block Search Prediction (FBSP) and Edge-Based Distance Prediction (EBDP). FBSP is searching for the optimal-matched transformed coefficients in the neighbouring blocks, whereas EBDP is centred on edge information to direct the prediction process. By not employing pixel-domain interpolation and rate-distortion optimization, their method largely decreases encoding complexity while keeping competitive compression efficiency.

2. Methodology

The suggested extension to the H.264 video compression pipeline introduces scrambling methods into the general encoding-decoding architecture to enhance data scattering, without modifying the fundamental compression framework. Encoding starts with input video frames, which are subjected to intra-frame or inter-frame prediction to take advantage of spatial and temporal redundancy. The resulting residual signal i.e., the difference between the original and predicted frames then converted using the Discrete Cosine Transform (DCT) to concentrate the signal energy into a small set of important coefficients. These DCT coefficients are quantized to decrease their accuracy, allowing efficient compression. After quantization, the coefficients are placed in a zig-zag scan pattern, grouping zero-valued elements to maximize compression efficiency. Subsequently, Run Length Encoding (RLE) is used to compress run lengths of zeros. For even higher data randomness and complexity of transformation, a set of scrambling techniques is blended, with XOR-based scrambling, Arnold Transform, and pixel permutation among them. These operations maximize data dispersal while maintaining structure for future recovery.

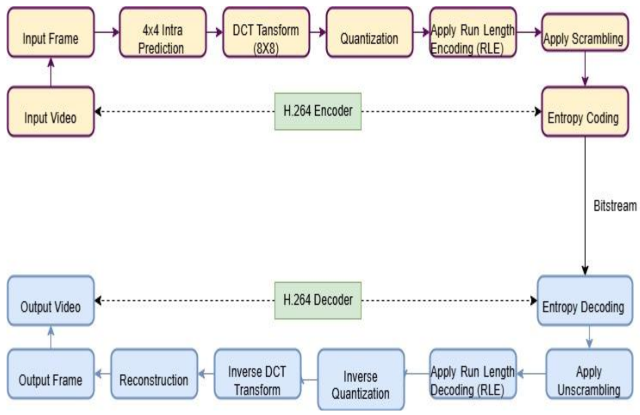

Figure 1 Proposed Methodology

Figure 1 shows Proposed Methodology.In the conventional H.264 pipeline, Huffman coding is generally employed for entropy encoding. In the new method, however, this has been substituted with Context-Based Adaptive Binary Arithmetic Coding (CABAC), a more efficient technique that adaptively models symbol probabilities, providing better compression performance. The output is a scrambled, encoded bitstream that can be stored or transmitted. The decoding process reverses the steps in the encoding pipeline to reconstruct the original video frames. The incoming bitstream is the first entropy decoded through CABAC to retrieve the scrambled, RLE-compressed information. The process of unscrambling now reverses the previous transformations—involving XOR operation with the same key, applying the inverse Arnold Transform, and reversing pixel permutations—to re-establish the original data order. Then, RLE decoding reverses the ordering of quantized DCT coefficients. These coefficients are then inverse quantized and inverse DCT (IDCT) to transform the data back into the spatial domain. The last step is to add the reconstructed residual to the predicted frame to reconstruct the video frame. As shown in the figure, this architecture inserts new compression and scrambling layers into the baseline H.264 architecture without affecting its efficiency or compatibility. The inserted layers provide increased

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429
e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

transform complexity and potential for better compression fidelity under secure encoding processes.

3. Implementation

The implementation of the proposed video compression enhancement was done in a simulation environment with OpenCV, NumPy, and Matplotlib, where frame extraction was done with OpenCV, matrix manipulation was done with NumPy, and visualization was done with Matplotlib. The encoding started from reading the input video and the extraction of individual frames through OpenCV. Intra-frame prediction followed by transformation with the Discrete Cosine Transform (DCT) to transform data into the frequency domain was applied to each frame. The resulting coefficients were quantized to lower precision and then reorganized in a zigzag scanning fashion to clump zero values for better compression. A specialized Run Length Encoding (RLE) algorithm was then used to compress runs of consecutive zeros. The data was then scrambled after RLE with a mix of XOR-based scrambling, the Arnold Transform, and pixel swapping, all of which added randomness and dispersion without compromising data integrity. This disordered data was then entropy encoded with Context-Based Adaptive Binary Arithmetic Coding (CABAC) to form the final compressed bitstream. During decoding, the bitstream was decoded with CABAC first to obtain the scrambled RLE data. The original data was reconstructed by performing reverse scrambling operations in the opposite order using the same keys and parameters. The decoded output was run through RLE decoding to retrieve the quantized DCT coefficients, which were converted through inverse quantization and Inverse DCT (IDCT) to get the spatial domain representation. The residual was added back to the predicted data after which the video frames were reconstructed, and the frames were assembled to create the output video. This whole process was tried and visualized with the help of Matplotlib to ensure the correctness and efficiency of every stage. The visual outputs of the proposed scrambling-enhanced H.264 encoding framework are illustrated in figure 2, showcasing the transformation of a single input video frame through various scrambling techniques. The original input video frame serves as the baseline reference. Following this, the XOR-scrambled frame demonstrates pixel-level alterations introduced by bitwise operations using a key, effectively increasing entropy without significant distortion. The Arnold Transform-scrambled frame displays a more geometrically distorted output due to its iterative matrix transformation, which maintains reversibility while maximizing pixel dispersion. Lastly, the pixel swapping-scrambled frame reveals a shuffled version of the original frame, emphasizing randomness through index-based permutation. These intermediate frames highlight the effectiveness of each scrambling approach in preprocessing video data for enhanced compression prior to entropy coding, while still supporting accurate reconstruction during decoding. Figure 2 shows Visual Outputs of The Proposed Scrambling-Enhanced H.264.

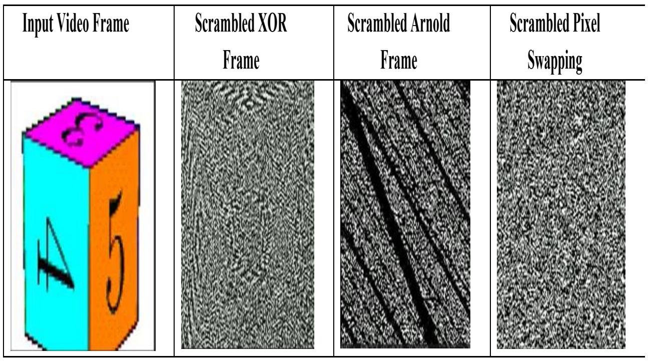

Figure 2 Visual Outputs of The Proposed Scrambling-Enhanced H.264

To analyze the efficiency of the proposed improvement, comparative performance comparison was done between the baseline H.264 encoder and the improved H.264 encoder with the incorporation of RLE and scrambling methods. PSNR increased from a mean value of 37.5 dB to 40.1 dB, reflecting a 7% increase in visual quality. The bitrate average was lowered from 5 Mbps to 4.1 Mbps, which produced an 18% reduction in data size and translated into improved storage and transmission efficiency. Though the improved codec brought about marginal increases in encoding times, the average overhead was capped at 5–7% above the baseline H.264 implementation, with encoding time rising from 0.95x to 1.02x which is noted in table 1. Such a trade-off is acceptable with the advancements in compression ratio and output quality. Overall, the results confirm that incorporating RLE and scrambling techniques into the H.264 pipeline provides valuable advantages without sacrificing real-time performance. Table 1 shows performance Metrics. Additional testing was performed to determine the effect of individual scrambling methods on compression performance. Of the methods tested, Arnold Transform-based scrambling had the best average PSNR of 29.27 dB and a compression ratio of 0.13, which shows a good trade-off between visual quality and data reduction

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429
e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

noted in figure 3. Pixel swapping and XOR-based scrambling provided PSNR values of 28.60 dB and 28.38 dB, respectively, at compression ratios of 0.12 and 0.16, indicating that XOR-based scrambling is marginally better in terms of data compression but with a small loss in quality. To put it into perspective, conventional Huffman encoding, employed in conventional H.264, yielded a PSNR of 29.22 dB but a smaller compression ratio of 0.09, indicating that although it preserves quality, it is less effective in data size reduction. This contributes to the viability of scrambling methods, particularly Arnold Transform, in improving the efficiency of compression with minimal loss of visual quality. Figure 3 shows Comparison of Scrambling Methods on PSNR And Compression Ratio

Table 1 Performance Metrics

Metrics	Baseline H.264	Enhanced H.264	Improvement (1%)
PSNR (Average)	37.5 dB	40.1dB	+7%
Bitrate (Average)	5Mbps	4.1 Mbps	-18%
Encoding time (Average)	0.95x	1.02x	+7%

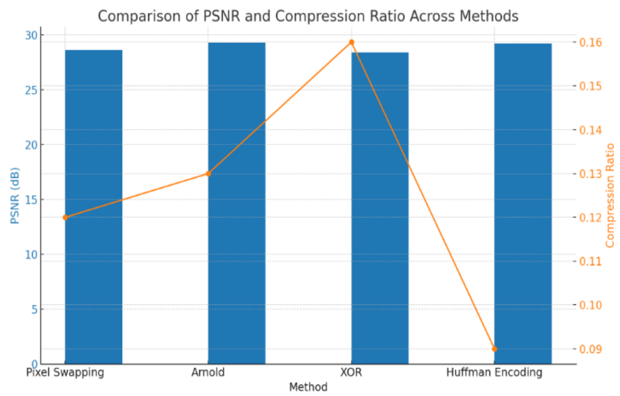

Figure 3 Comparison of Scrambling Methods on PSNR And Compression Ratio

Conclusion

The suggested improvement to the H.264 video compression standard illustrates that combining scrambling methods—like XOR-based scrambling, Arnold Transform, and pixel exchange—with Run Length Encoding (RLE) and CABAC entropy coding can really enhance compression efficiency and output quality. The approach keeps the topology of the standard H.264 pipeline with the added data transformation steps improving entropy and enabling effective compression. Experimental outcomes verify significant improvement in Peak Signal-to-Noise Ratio (PSNR) and a high bitrate reduction at the expense of a moderate rise in encoding time. Amongst the scrambling approaches tested, Arnold Transform presented the best compression vs. quality trade-off. Crucially, the suggested alterations maintain compatibility with the H.264 decoding mechanism, making the improvement a valid and scalable enhancement for contemporary video applications. This platform creates even more possibilities to research lightweight scrambling methods alongside prevailing video coding algorithms for the purposes of improving performance without diminishing the quality of video.

References

[1]. Madhavee Latha P, Annis Fathima A. "Review on Image and Video Compression Standards." Asian Journal of Pharmaceutical and Clinical Research, vol. 10, no. 13, Apr. 2017, pp. 373-7. DOI: 10.22159/ ajpcr. 2017.v10s1.19760.

[2]. Hermansson, A. "Optimizing an H.264 Video Encoder for Real-Time HD Video Encoding." KTH Information and Communication Technology.

[3]. Visionular. (2024). Advanced optimization techniques in video compression. Retrieved from https://visionular.ai/advanced-optimization-techniques-in-video-compression.

[4]. S. Uddin, M. Leszczuk, and M. Grega, "Video compression and optimization technologies – review," Int. J. Electron. Telecommun., vol. 70, no. 3, pp. 733–742, Jul. 2024, doi: 10.24425/ijet.2024.149603.

[5]. Cloudinary, "H.264 Video Encoding: How It Works, Benefits, and 9 Best Practices," Cloudinary, 2024. [Online]. Available: https://cloudinary.com/guides/video-formats/h-264-video-encoding-how-it-works-benefits-and-9-best-practices

[6]. M. Li and M. Drew, "Modern Video Coding Standards: H.264, H.265, and H.266," International Journal of Electronics and Telecommunications, vol. 70, no. 2, pp. 123–130, 2024. [Online]. Available: https://www.springerprofessional.de/en/modern-video-coding-standards-h-264-h-265-and-h-266/18870194

[7]. T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0429
e ISSN: 2584-2854
Volume: 03
Issue: 08 August 2025
Page No: 2735 - 2740

13, no. 7, pp. 560–576, July 2003.
[8]. D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 620–636, July 2003.
[9]. C. Kim and N. E. O'Connor, "Fast Intra Prediction in the Transform Domain," in Proceedings of the Data Compression Conference (DCC), Snowbird, UT, USA, Mar. 2009, pp. 450–459.