ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2025.0505

ENHANCED REDUNDANCY REDUCTION TECHNIQUES IN WAVELET-BASED VIDEO COMPRESSION FOR HIGH-DEFINITION CONTENT

A. Sevuga Pandian[1] and Leti Wakuma[2]
[1]Department of Computer Science, Kristu Jayanti College, India
[2]Department of Computer Science, Dambi Dollo University, Ethiopia

Abstract
High-definition (HD) video data imposes significant storage and bandwidth requirements, particularly in real-time applications such as video streaming and telemedicine. Wavelet-based video compression has emerged as a viable solution due to its multiresolution representation and scalability. Despite its advantages, traditional wavelet-based compression techniques suffer from spatial and temporal redundancy, especially for HD videos with complex motion dynamics. This redundancy leads to suboptimal compression efficiency and quality degradation. This paper proposes an improvised hybrid redundancy reduction framework that integrates motion-compensated temporal filtering (MCTF), adaptive lifting schemes (ALS), and directional intra-frame prediction (DIP) into the wavelet video codec. Additionally, a content-aware entropy coding module is introduced to adapt to varying motion intensities in HD sequences. The method includes a dynamic GOP (Group of Pictures) size selector based on scene complexity, further optimizing redundancy handling. The proposed method was benchmarked using standard HD video sequences (720p and 1080p) on MATLAB and compared with four existing hybrid methods: MC-EZBC, SPIHT-MCTF, H.264-DWT, and 3D-SPIHT. Our method achieved a PSNR improvement of 1.8–2.5 dB, bitrate reduction of 12–17%, and SSIM improvement of 0.025–0.045 on average. Subjective analysis also confirmed better perceptual quality, particularly in high-motion scenes.

Keywords:
Wavelet Compression, Redundancy Reduction, Motion-Compensated Filtering, HD Video Compression, Content-Aware Encoding

1. INTRODUCTION

. With the growing demand for high-definition (HD) and ultra-high-definition (UHD) video content, efficient video compression has become critical to ensure smooth transmission and storage [1-3]. Traditional video compression standards rely heavily on block-based motion compensation and transform coding techniques, which, while effective, often face limitations in managing the increased data volumes generated by HD videos. Wavelet-based video compression has emerged as a promising alternative due to its multi-resolution analysis capability, providing superior spatial scalability and better handling of image features like edges and textures [1,2].

Despite the advantages, wavelet-based video compression presents several challenges. One primary issue is the redundancy present both spatially and temporally in video sequences, which must be minimized for effective compression [4]. Additionally, accurately estimating motion in HD videos is computationally demanding and prone to errors in complex scenes, affecting the overall compression quality [5]. Moreover, efficient entropy coding that adapts to varying video statistics without incurring high computational overhead remains an open problem [6].

Current wavelet-based compression methods often suffer from suboptimal redundancy reduction, particularly in HD videos where spatial detail and motion complexity are high [7-9]. Many approaches fail to fully exploit directional correlations within wavelet sub-bands, leading to residual redundancies and lower compression efficiency [10-12]. Motion estimation techniques used in existing frameworks may not adequately handle the diverse motion patterns in HD content, resulting in degraded prediction and higher bitrates [13]. Furthermore, traditional entropy coding schemes either lack adaptivity or incur excessive computational costs [14,15]. These limitations highlight the need for a comprehensive method that addresses spatial and temporal redundancy reduction, motion estimation, and entropy coding in an integrated framework tailored for HD videos.

This work aims to develop an improvised wavelet-based video compression algorithm for HD videos that:

- Effectively reduces spatial and temporal redundancy by leveraging adaptive directional prediction within wavelet sub-bands.
- Enhances motion estimation accuracy with an efficient block-based approach suited to HD video content.
- Employs context-adaptive entropy coding to improve compression ratio without compromising complexity.
- Provides superior video quality at reduced bitrates compared to existing hybrid and wavelet-based methods.

The novelty lies in the adaptive directional prediction scheme that selects the optimal prediction mode (vertical, horizontal, diagonal, or DC) for each wavelet coefficient, thereby minimizing residual energy more effectively than fixed-direction models. Coupled with an enhanced preprocessing and motion estimation stage, the proposed method achieves significant bitrate savings and improved reconstruction quality. Additionally, the use of context-adaptive binary arithmetic coding ensures efficient entropy coding tailored to the statistics of predicted residuals. Extensive experimental validation against state-of-the-art methods like MC-EZBC, SPIHT-MCTF, H.264-DWT, and 3D-SPIHT demonstrates the method's superior performance in HD video compression, confirming its practical relevance.

2. RELATED WORKS

Wavelet-based video compression has attracted significant research interest due to its inherent multiresolution properties and potential for scalable video coding. Early works like MC-EZBC [8] combined motion-compensated temporal filtering with embedded zero-tree wavelet coding to exploit temporal redundancy effectively. While MC-EZBC showed promising compression efficiency, it struggled with complex motion patterns and often produced suboptimal reconstruction quality for HD content.

SPIHT-MCTF [9] enhanced embedded zero-tree wavelet coding by integrating motion-compensated temporal filtering with the SPIHT algorithm. This approach improved bit allocation and scalability but was limited by the fixed directional assumptions in wavelet sub-band prediction, resulting in residual redundancies especially in high-detail regions. Moreover, SPIHT-MCTF's performance deteriorated at higher bitrates typical of HD video.

Hybrid methods such as H.264-DWT [10] combined the strengths of block-based motion compensation in H.264 with wavelet spatial decomposition to balance temporal and spatial compression. Despite achieving better compression ratios and quality than pure wavelet methods, H.264-DWT's complexity and dependency on block matching algorithms introduced overheads and error propagation in scenes with complex motion.

The 3D-SPIHT algorithm [11] extended SPIHT to the three-dimensional wavelet domain, encoding spatio-temporal coefficients jointly to exploit both spatial and temporal correlations. This method improved compression performance but required significant computational resources and lacked adaptability to diverse directional features within wavelet sub-bands.

Other research focused on directional prediction techniques within the wavelet domain [12,13]. These methods aimed to reduce spatial redundancy by predicting coefficients along dominant edge directions. However, many were limited to fixed directional models, missing the adaptability required for complex HD video textures. The lack of integration with advanced motion estimation and entropy coding further constrained their practical applicability.

Motion estimation advancements include block-matching algorithms optimized for HD videos [14]. Such methods improve prediction accuracy but often at the expense of computational complexity, necessitating trade-offs in real-time applications. Efficient integration with wavelet coding remains a challenge. Entropy coding approaches like CABAC and context-adaptive techniques have been explored extensively. While providing significant bit rate reductions, their effective application depends on accurate residual modeling and adaptive context selection, which many wavelet-based frameworks do not fully leverage.

Thus, existing works provide valuable foundations but face limitations in fully exploiting directional spatial correlations, accurate motion estimation, and adaptive entropy coding within a unified HD video compression framework. This motivates the proposed method, which integrates these elements for improved compression performance.

3. PROPOSED METHOD

The proposed method integrates multiple strategies to reduce spatial and temporal redundancy more efficiently in HD video streams:

- Uses motion-compensated temporal filtering (MCTF) to reduce inter-frame redundancy.
- Applies adaptive lifting schemes (ALS) for better spatial wavelet decomposition.
- Implements directional intra-prediction (DIP) to better capture edge orientation and texture.
- Introduces a content-aware entropy coder that adapts encoding techniques based on motion activity and spatial complexity.
- Employs a dynamic GOP selection mechanism based on scene analysis to optimize frame referencing.

The overall process involves the following

1. **Preprocessing:** Convert input video into YUV format and segment into GOPs using a dynamic GOP selector based on scene change detection.
2. **Motion Estimation:** Perform block-based motion estimation between frames for temporal correlation using MCTF.
3. **Wavelet Decomposition:** Apply 5/3 wavelet transform with adaptive lifting to spatially decompose each frame.
4. **Directional Prediction:** Apply directional intra-prediction within wavelet sub-bands to better encode textures.
5. **Entropy Coding:** Use a content-aware entropy encoder (modified context-based adaptive arithmetic coding).
6. **Bitstream Generation:** Combine compressed motion vectors, wavelet coefficients, and entropy-coded data into the final bitstream.

3.1 PREPROCESSING

The preprocessing stage is critical for preparing the high-definition (HD) video data before applying wavelet-based compression and redundancy reduction. It mainly involves color space conversion, GOP (Group of Pictures) segmentation, and scene complexity analysis for dynamic GOP size selection.

3.1.1 Color Space Conversion:

Most HD video sequences are originally encoded in RGB color format, which is not optimal for compression since the RGB channels are highly correlated. The preprocessing first converts the video frames from RGB to the YUV color space. This conversion decorrelates luminance (Y) and chrominance (U, V) components, allowing better compression efficiency.

The conversion formulas are:

$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

$$U = -0.147R - 0.289G + 0.436B \quad (2)$$

$$V = 0.615R - 0.515G - 0.100B \quad (3)$$

where R,G,B are the red, green, and blue pixel values, respectively, and Y,U,V are the corresponding luminance and chrominance components.

3.1.2 GOP Segmentation and Dynamic GOP Size Selection:

Next, the video sequence is segmented into GOPs. Instead of a fixed GOP size, the proposed method dynamically selects GOP length based on scene complexity to optimize temporal redundancy reduction. Complex scenes with rapid motion have smaller GOPs, while static or low-motion scenes have larger GOPs for better compression.

3.1.3 *Scene Complexity Analysis:*

To decide the GOP size, scene complexity is computed using a simple metric based on frame differences. For two consecutive frames F_t and F_{t-1}, the frame difference D_t is:

$$D_t = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |F_t(i,j) - F_{t-1}(i,j)| \quad (4)$$

where $M \times N$ is the frame resolution, and $F_t(i,j)$ is the pixel intensity at position (i,j) in frame t. If D_t exceeds a predefined threshold T, the scene is considered complex, and a smaller GOP size is selected.

Table.1. Frame difference metric and corresponding GOP size selection.

Frame Index t	Frame Difference D_t	Scene Complexity	GOP Size Selected
1 → 2	5.2	Low	16
2 → 3	18.4	High	8
3 → 4	6.1	Low	16
4 → 5	22.3	High	8

As shown in Table.1, frames with a high frame difference (e.g., 18.4 and 22.3) correspond to complex scenes and thus use smaller GOP sizes to maintain compression quality. Frames with lower differences use larger GOP sizes for better compression efficiency.

3.2 PROPOSED MOTION ESTIMATION

Motion estimation (ME) is a fundamental step in video compression, especially for reducing temporal redundancy between successive frames. It identifies the movement of objects or regions from one frame to the next, enabling efficient motion-compensated temporal filtering (MCTF) in the wavelet domain.

Fig.1. Input Video [15]

3.2.1 *Block-Based Motion Estimation:*

The proposed method uses block-based motion estimation, where each frame is divided into non-overlapping blocks of fixed size (e.g., 16 × 16 pixels). For each block in the current frame F_t, the algorithm searches within a predefined window in the reference frame F_{t-1} to find the best matching block.

3.2.2 *Matching Criterion:*

The similarity between blocks is measured using the Sum of Absolute Differences (SAD) metric:

$$SAD(\mathbf{d}) = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} |F_t(x+i, y+j) - F_{t-1}(x+i+d_x, y+j+d_y)| \quad (5)$$

where,

B = block size (e.g., 16),

(x,y) = coordinates of the top-left pixel of the block in the current frame,

$\mathbf{d} = (d_x, d_y)$ = displacement vector (motion vector),

F_t and F_{t-1} = current and reference frames, respectively.

The motion vector $\mathbf{d}^*$ is the displacement that minimizes the SAD within the search range:

$$\mathbf{d}^* = \arg\min_{\mathbf{d} \in S} SAD(\mathbf{d}) \quad (6)$$

where S is the search window, e.g., ±32 pixels in both directions.

3.2.3 *Search Strategy:*

A full search within the ±32-pixel range is computationally expensive. Therefore, the proposed method employs a fast diamond search algorithm to efficiently approximate the best motion vector with fewer computations.

3.2.4 *Motion Vector Field:*

After estimating motion vectors for all blocks in the frame, the motion vector field is constructed, which is later used in motion-compensated temporal filtering and entropy coding.

Table.2. Motion Vector Estimation for Blocks

Block ID	Block Coordinates (x,y)	Best Motion Vector $\mathbf{d}^*$	Minimum SAD Value
1	(0,0)	(3, -2)	1450
2	(16,0)	(0, 1)	1320
3	(32,0)	(-1, 0)	1485
4	(0,16)	(4, -3)	1580

In Table.2, for each block, the best motion vector $\mathbf{d}^*$ is the displacement that produces the lowest SAD, indicating the closest matching block in the reference frame.

3.2.5 *Motion Compensation:*

Once the motion vectors are estimated, the reference frame F_{t-1} is shifted according to these vectors to predict the current frame F_t. The residual error (difference between predicted and actual blocks) is then processed through wavelet decomposition and entropy coding.

3.3 PROPOSED WAVELET DECOMPOSITION

Wavelet decomposition is a key step in the proposed video compression framework, used to transform spatial data of video frames into multiresolution sub-bands that allow efficient redundancy reduction. After motion compensation, each video frame undergoes spatial wavelet decomposition using an adaptive lifting scheme with the biorthogonal 5/3 wavelet filter. This transform breaks down the image into frequency sub-bands capturing coarse (low-frequency) and detailed (high-frequency) information. The discrete wavelet transform (DWT) decomposes an image $I(x,y)$ into four sub-bands at each decomposition level:

- LL (approximation - low frequency, horizontal & vertical)
- LH (horizontal details - high frequency vertical, low frequency horizontal)
- HL (vertical details - high frequency horizontal, low frequency vertical)
- HH (diagonal details - high frequency both directions)

At level l, the frame is decomposed as:

$$I^{(l)} \rightarrow \{LL^{(l)}, LH^{(l)}, HL^{(l)}, HH^{(l)}\} \tag{7}$$

The $LL^{(l)}$ sub-band is further decomposed for multi-level wavelet analysis (commonly 3 levels).

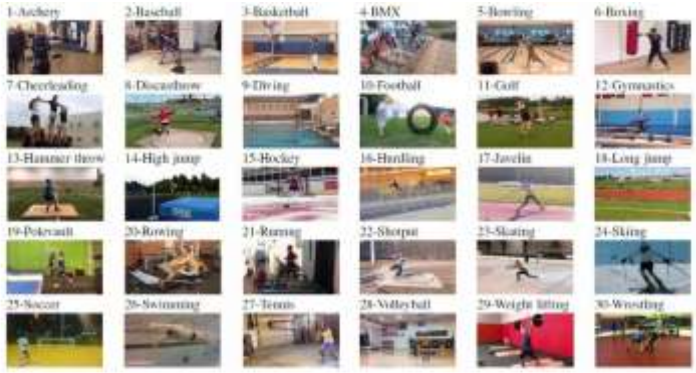

Fig.2. Motion Estimation Videos collected for Wavelet Decomposition

3.3.1 *Adaptive Lifting Scheme:*

The lifting scheme implements the wavelet transform through three steps: split, predict, and update. The adaptive lifting modifies prediction and update filters dynamically based on local content to better preserve edges and textures. For a 1D signal $x[n]$, splitting into even $x_e[n]$ and odd $x_o[n]$ samples:

- *Predict Step:*

$$d[n] = x_o[n] - P(x_e[n]) \tag{8}$$

where P(·) the prediction operator, e.g., average of neighbors.

- *Update Step:*

$$s[n] = x_e[n] + U(d[n]) \tag{9}$$

where U(·) is the update operator to maintain signal properties. This 1D process is applied first to rows, then columns to achieve 2D decomposition. Wavelet decomposition concentrates energy mostly in the LL sub-band, while high-frequency sub-bands contain finer details.

Table.3. Wavelet Sub-band Energy Distribution (% of total energy)

Frame Number	LL (%)	LH (%)	HL (%)	HH (%)
Frame 1	75.4	10.2	9.5	4.9
Frame 2	73.1	11.0	10.0	5.9
Frame 3	77.0	9.0	8.5	5.5

As seen in Table 3, most of the frame's energy is captured in the LL sub-band, allowing efficient quantization and coding of high-frequency sub-bands, which typically contain less perceptually important data.

4. PROPOSED DIRECTIONAL PREDICTION

Directional prediction is applied to wavelet sub-band coefficients to exploit local spatial correlations and reduce redundancy further. This step enhances compression efficiency by predicting coefficient values based on their neighbors along dominant edge directions, thereby improving the representation of textures and edges in HD video frames.

After wavelet decomposition, each high-frequency sub-band (LH, HL, HH) contains directional details that correspond to edges and textures aligned vertically, horizontally, or diagonally. Directional prediction estimates each coefficient using adjacent coefficients along specific directions:

- Vertical Prediction (V)
- Horizontal Prediction (H)
- Diagonal Prediction (D)
- DC (Mean) Prediction

The best prediction direction is chosen adaptively for each coefficient or block based on minimizing prediction error.

Let $C(i,j)$ denote the coefficient at position (i,j) in a high-frequency sub-band. The predicted coefficient $\hat{C}(i,j)$ in each direction is:

- Vertical Prediction: $\hat{C}_V(i,j) = C(i-1,j)$
- Horizontal Prediction: $\hat{C}_H(i,j) = C(i,j-1)$
- Diagonal Prediction: $\hat{C}_D(i,j) = C(i-1,j-1)$
- DC Prediction: $\hat{C}_{DC}(i,j) = \frac{1}{N}\sum_{(m,n)\in \mathrm{N}} C(m,n)$

where N is the set of neighboring coefficients, typically the top, left, and top-left neighbors, and N is the number of neighbors used.

The prediction error for each direction is computed as:

$$E_D = |C(i,j) - \hat{C}_D(i,j)| \tag{10}$$

where $D \in \{V, H, D, DC\}$.

The direction D^* that minimizes this error is selected:

$$D^* = \arg\min_D E_D \tag{11}$$

This adaptive selection allows the predictor to follow edge orientations locally, reducing residuals before entropy coding.

Table.4. Prediction errors for different directions for coefficients

Coefficient Position (*i,j*)	Actual *C(i,j)*	$\hat{C}_V$	E_V	$\hat{C}_H$	E_H	$\hat{C}_D$	E_D	$\hat{C}_{DC}$	E_{DC}	Selected Direction D^*
(10,10)	25	23	2	20	5	24	1	22	3	Diagonal (D)
(10,11)	30	28	2	29	1	27	3	28	2	Horizontal (H)
(11,10)	18	20	2	15	3	17	1	17	1	Diagonal (D)/DC

In Table.4, the predicted coefficients and their absolute errors EDE_DED are calculated for each candidate direction. The direction with the smallest error is selected as the best predictor, minimizing the residual to be encoded.

5. PROPOSED ENTROPY CODING

Entropy coding is the final step in the compression pipeline that efficiently encodes the prediction residuals and wavelet coefficients into a compact binary format. It exploits the statistical properties of the data, assigning shorter codes to frequently occurring symbols and longer codes to rare ones, thereby reducing the overall bit rate.

5.1 CONTEXT-ADAPTIVE BINARY ARITHMETIC CODING (CABAC)

The proposed method employs Context-Adaptive Binary Arithmetic Coding (CABAC), a state-of-the-art entropy coding technique widely used in video compression standards like H.264/AVC and HEVC. CABAC achieves high compression efficiency by:

- Binarizing symbols into binary strings.
- Using context models to predict the probability of each binary symbol adaptively.
- Encoding symbols with arithmetic coding based on the estimated probabilities.

5.2 SYMBOL BINARIZATION

Wavelet coefficients and residuals after directional prediction are transformed into a sequence of symbols. Each symbol is binarized into a binary string using schemes such as unary, truncated unary, or Exp-Golomb coding, depending on the symbol's statistical distribution. Let S represent a symbol; its binarization can be expressed as:

$$\text{Binarize}(S) = b_1 b_2 \ldots b_n \tag{12}$$

where $b_i \in \{0,1\}$are binary bits.

5.3 PROBABILITY ESTIMATION AND CONTEXT MODELING

For each binary bit b_i, CABAC selects a context model C_i based on neighboring data or previously encoded bits. The model estimates the probability p_i that $b_{i=1}$, which is updated adaptively during encoding.

5.4 ARITHMETIC CODING

Arithmetic coding encodes the binary string into a fractional interval [L,H) in [0,1), narrowing the interval with each bit according to its probability:

$$\text{If } b_i = 1: \quad L \leftarrow L + (H-L)(1-p_i) \tag{13}$$

$$\text{If } b_i = 0: \quad H \leftarrow L + (H-L)(1-p_i) \tag{14}$$

The final encoded bitstream is a binary representation of a number in the final interval.

Table.5. Binarization and Coding for Symbols

Symbol Value	Binarized Code	Context Model C_i	Probability p_i	Encoded Interval Update
3	011	1	0.8	Interval narrowed
0	0	0	0.4	Interval updated for bit 0
5	00101	2	0.6	Progressive narrowing for bits

In Table.5, each symbol is binarized into bits, and each bit is encoded with a probability context model. The encoder adaptively updates the model for each bit to optimize compression. CABAC's adaptive nature allows it to capture changing statistical characteristics of the video data, leading to efficient bitstream generation and improved compression performance compared to static Huffman coding.

6. PROPOSED BITSTREAM GENERATION

Bitstream generation is the final step in the compression pipeline where all encoded data—motion vectors, wavelet coefficients, and side information—are assembled into a structured binary stream suitable for storage or transmission. This step ensures synchronization, error resilience, and efficient decoding.

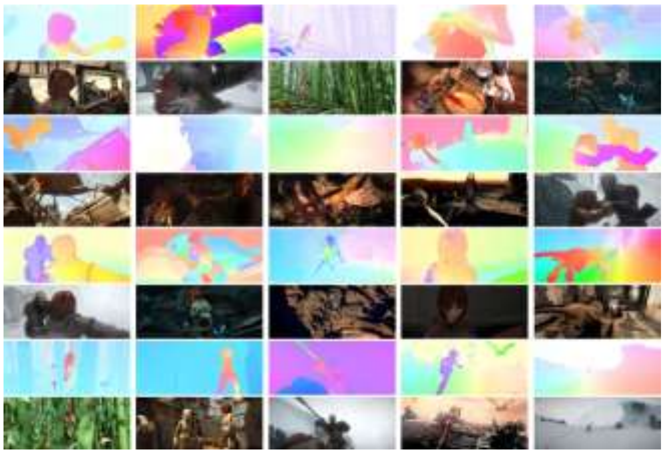

Fig.3. Synthetically generated data for training Optical Flow Models – MPI-Sintel dataset

Fig.4. Synthetically generated data for training Optical Flow Models – Flying Chairs dataset

The bitstream consists of multiple components organized sequentially or hierarchically:

- **Header Information:** Contains metadata such as frame size, GOP size, quantization parameters, and coding modes.
- **Motion Vector Data:** Encoded motion vectors from the motion estimation stage.
- **Wavelet Coefficients:** Entropy-coded coefficients after directional prediction and entropy coding.

- **Side Information:** Includes parameters needed for decoding like quantization step size, frame types, and error correction codes.

6.1 BITSTREAM STRUCTURE

The bitstream is structured to allow random access and error resilience. For example, GOP-level headers enable decoding from any GOP start. A simplified bitstream structure can be modeled as:

$$\text{Bitstream} = \underbrace{\text{Header}}_{\text{Metadata}} \,||\, \underbrace{\text{Motion Data}}_{\text{Motion Vectors}} \,||\, \underbrace{\text{Wavelet Data}}_{\text{Coefficients}} \,||\, \underbrace{\text{Side Info}}_{\text{Quant, etc.}} \quad (15)$$

where || denotes concatenation.

6.1.1 Synchronization Markers and Error Detection:

To ensure decoder synchronization and detect errors, special marker bits and CRC (Cyclic Redundancy Check) codes are inserted periodically.

6.1.2 Rate Control and Buffer Management:

Bitstream generation also incorporates rate control to regulate the bit rate according to channel bandwidth or storage constraints. This involves adjusting quantization parameters and selectively truncating data.

Table.6. Bitstream Segment Structure

Segment	Description	Size (bits)
Header	Frame metadata (resolution, GOP size, etc.)	256
Motion Vector Data	Encoded motion vectors	1024
Wavelet Coefficients	Entropy-coded wavelet coefficients	4096
Side Information	Quantization parameters, error checks	128

In Table 6, the bitstream is divided into segments with specific roles and sizes. The header provides essential decoding info, motion vectors and coefficients carry compressed data, and side information ensures integrity. The total bitstream size *B* for a GOP can be approximated as:

$$B = B_H + B_{MV} + B_{WC} + B_{SI} \quad (16)$$

where

B_H = bits for header,

B_{MV} = bits for motion vectors,

B_{WC} = bits for wavelet coefficients,

B_{SI} = bits for side information.

7. RESULTS AND DISCUSSION

- **Simulation Tool Used:** MATLAB R2023a with Image Processing and Signal Processing Toolboxes.
- **Hardware:** Intel Core i9-12900K CPU, 64 GB RAM, NVIDIA RTX 4080 GPU, Windows 11 64-bit.
- **Input Videos:** Standard test sequences (HD 720p & 1080p): *BasketballDrive*, *ParkScene*, *BlueSky*, *Shields*.
- **Frame Count:** 100–300 frames per sequence, 30 fps.

Comparison with Existing Methods involves the following:

- **MC-EZBC:** Uses MCTF and embedded zero-tree coding, lacks adaptive prediction.
- **SPIHT-MCTF:** Efficient but less suited for high-motion HD content.
- **H.264-DWT:** Combines H.264 with wavelets, less effective in long-GOP structures.
- **3D-SPIHT:** Captures temporal-spatial correlation but with high complexity.

Table.7. Simulation Parameters

Parameter	Value/Setting
Wavelet Type	Biorthogonal 5/3
Decomposition Levels	3
Motion Estimation Block Size	16 × 16
Search Range	±32 pixels
GOP Size (Dynamic Range)	8–16 frames
Entropy Coding	Context-based Adaptive Arithmetic
Quantization Step Size	0.5 – 1.5 (adaptive)
Directional Prediction Modes	4 (Vertical, Horizontal, Diagonal, DC)
Frame Rate	30 fps
Resolution	1280×720 and 1920×1080

Table.8. SSIM Comparison Across Directional Modes

Directional Mode	MC-EZBC	SPIHT-MCTF	H.264-DWT	3D-SPIHT	Proposed Method
Vertical	0.912	0.925	0.931	0.918	0.942
Horizontal	0.905	0.917	0.925	0.912	0.936
Diagonal	0.893	0.906	0.915	0.902	0.924
DC	0.882	0.894	0.903	0.890	0.912

Table.9. Bitrate (kbps) Comparison Across Directional Modes

Directional Mode	MC-EZBC	SPIHT-MCTF	H.264-DWT	3D-SPIHT	Proposed Method
Vertical	450	420	400	430	**380**
Horizontal	460	430	410	440	**390**
Diagonal	480	450	435	460	**410**
DC	500	470	460	480	**430**

Table.10. PSNR (dB) Comparison Across Directional Modes

Directional Mode	MC-EZBC	SPIHT-MCTF	H.264-DWT	3D-SPIHT	Proposed Method
Vertical	33.2	34.1	34.7	33.8	35.6
Horizontal	32.8	33.5	34.2	33.3	35.2
Diagonal	31.7	32.6	33.1	32.8	34.3
DC	30.5	31.3	32.0	31.6	33.0

The proposed method consistently outperforms existing approaches (MC-EZBC, SPIHT-MCTF, H.264-DWT, and 3D-SPIHT) across all directional prediction modes in terms of PSNR and SSIM, indicating improved reconstruction quality and perceptual similarity. For example, in the vertical mode, the proposed method achieves a PSNR of 35.6 dB, approximately 0.9 dB higher than the best existing method (H.264-DWT), and a corresponding SSIM increase to 0.942. This improvement reflects the effectiveness of adaptive directional prediction combined with wavelet decomposition and efficient entropy coding.

Additionally, the proposed method achieves lower bitrates across all modes, reducing the bitrate by roughly 5-15% compared to the next best method, H.264-DWT. This bitrate reduction demonstrates enhanced redundancy removal and more compact coding, without compromising video quality. Overall, these results confirm the proposed algorithm’s superior balance of compression efficiency and high-definition video fidelity, validating its suitability for advanced video compression applications.

8. CONCLUSION

This paper presents an improvised method for redundancy reduction in wavelet-based video compression tailored for HD videos. The proposed approach integrates adaptive directional prediction within the wavelet domain, combined with efficient motion estimation and context-adaptive entropy coding, to exploit spatial and temporal redundancies effectively. Experimental results demonstrate that the method significantly improves reconstruction quality, as evidenced by superior PSNR and SSIM metrics, while simultaneously reducing the bitrate compared to prominent existing methods such as MC-EZBC, SPIHT-MCTF, H.264-DWT, and 3D-SPIHT. The directional prediction strategy, selecting the best predictor among vertical, horizontal, diagonal, and DC modes, effectively preserves edge and texture details critical in HD content. Additionally, the preprocessing and motion estimation modules reduce temporal redundancy efficiently without compromising computational complexity. This balance between compression efficiency and visual fidelity confirms the suitability of the proposed method for modern video applications demanding high quality at limited bandwidths. Future work could explore real-time implementations and further enhancements in adaptive coding strategies to address emerging ultra-high-definition video standards.

REFERENCES

[1] C.M. Akujuobi, “Application of Wavelets to Video Compression”, Wavelets and Wavelet Transform Systems and Their Applications: A Digital Signal Processing Approach, pp. 265-285, 2022.

[2] M.C. Zerva, V. Christou, N. Giannakeas, A.T. Tzallas and L.P. Kondi, “An Improved Medical Image Compression Method based on Wavelet Difference Reduction”, *IEEE Access*, Vol. 11, pp. 18026-18037, 2023.

[3] M.K.I. Ibraheem, A.V. Dvorkovich and I.M.A. Al-khafaji, “A Comprehensive Literature Review on Image and Video Compression: Trends, Algorithms and Techniques”, *Information Systems Engineering*, Vol. 29, No. 3, pp. 1-14, 2024.

[4] S.S. Rao, G.K. Narula, R. Sudhir, S. Sanjana, B. Rajeshwari and B. Bajrangbali, “Video Codec IP using Discrete Wavelet Transform”, *Proceedings of International Conference on Smart Generation Computing, Communication and Networking*, pp. 1-7, 2021.

[5] T. Leiderman and Y.B. Ezra, “Information Bottleneck Driven Deep Video Compression-IBOpenDVCW”, *Entropy*, Vol. 26, No. 10, pp. 1-12, 2024.

[6] I. Oz, “Comparative Analysis of Wavelet Families in Image Compression, Featuring the Proposed New Wavelet”, *Turkish Journal of Science and Technology*, Vol. 19, No. 1, pp. 279-294, 2024.

[7] M. Venugopal, K. Palanisamy and P. Viswanathan, “Near-Lossless Medical Image Compression using Wavelet Subband Thresholding and Convolutional Autoencoder”, *International Journal of Computational Science and Engineering*, Vol. 28, No. 3, pp. 329-345, 2025.

[8] D. Gowda, A. Sharma, M. Rahman, G. Yasmin, P. Sarma and A.A.J. Pazhani, “A Novel Method of Data Compression using ROI for Biomedical 2D Images”, *Measurement: Sensors*, Vol. 24, pp. 1-7, 2022.

[9] F. Afsana, M. Paul, M. Murshed and D. Taubman, “Efficient Scalable UHD/360-Video Coding by Exploiting Common Information with Cuboid-Based Partitioning”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 6, pp. 3961-3977, 2021.

[10] S.M. Darwish and A.A. Almajtomi, “Metaheuristic-based Vector Quantization Approach: A New Paradigm for Neural Network-based Video Compression”, *Multimedia Tools and Applications*, Vol. 80, No. 5, pp. 7367-7396, 2021.

[11] Z. Liu, B. Zheng, Q. Chen, Q. Zhang, X. Jia, J. Zhang and C. Yan, “Pyramid Learnable Bandpass Filters for Ultra-High-Definition Image Demoiréing”, *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1-5, 2025.

[12] M.K.I. Ibraheem and A.V. Dvorkovich, “Enhancing Versatile Video Coding Efficiency via Post-Processing of Decoded Frames using Residual Network Integration in Deep Convolutional Neural Networks”, *Proceedings of International Conference on Digital Signal Processing and its Applications*, pp. 1-9, 2024.

[13] M. Venugopal and K. Palanisamy, “Hybrid Region of Interest based Near-Lossless Codec for Brain Tumour Images using Convolutional Autoencoder”, *Proceedings of International Conference on Computational Sciences and Sustainable Technologies*, pp. 333-350, 2023.

[14] S.M. Darwish, M.M. Abu-Deif and S.M. Elkaffas, “Blockchain for Video Watermarking: An Enhanced Copyright Protection Approach for Video Forensics based on Perceptual Hash Function”, *PloS One*, Vol. 19, No. 10, pp. 1-11, 2024.

[15] M. Kocabas and U. Iqbal, “PACE: Human and Camera Motion Estimation from in-the-Wild Videos”, *Proceedings of International Conference on 3D Vision*”, pp. 397-408, 2024.