# Fusion Model of Quantum Wavelet Transform and Neural Network for Video Coding on the Internet of Things Environment

**Iptehaj Alhakam[1], Ali Abdullah Ali[2], Oday Ali Hassen[3,*], Saad M. Darwish[4]**
**Nur Azman Abu[5]**

[1]Department of Computer, College of Education for Pure Sciences Ibn Al-Haitham, University of Baghdad, Iraq
[2]Minister Office of Higher Education and Scientific Research, Iraq
[3] Ministry of Education, Wasit Education Directorate, Iraq
[4] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt
[5]Department of Information Technology, University Technical Malaysia Melaka, Hang Taya, Melaka 76100, Malaysia

Emails: ibtihaj.a.a@ihcoedu.uobaghdad.edu.iq; aaaoea@gmail.com; odayali@uowasit.edu.iq; saad.darwish@alexu.edu.eg; nura@utem.edu.my

## Abstract

Solving the video compression problem requires a multi-faceted approach, balancing quality, efficiency, and computational demands. By leveraging advancements in technology and adapting to the evolving needs of video applications, it is possible to develop compression methods that meet the challenges of the present and future digital landscape. To address these objectives, machine learning and AI approaches can be utilized to predict and remove redundancies more effectively, optimizing compression algorithms dynamically based on content. Still, state-of-the art neural network-based video compression models need large and diverse datasets to generalize well across different types of video content. Wavelets can provide both time (spatial) and frequency localization, making them highly effective for video compression. This dual localization allows wavelet transforms to handle both rapid changes in video content and slow-moving scenes efficiently, leading to better compression ratios. Yet, some wavelet coefficients may be more critical for maintaining visual quality than others. Inaccurate quantization can lead to noticeable degradation. For the first time, the suggested model combine Quantum Wavelet Transform (QWT) and Neural Networks (NN) for video compression. This fusion model aims to achieve higher compression ratios, maintain video quality, and reduce computational complexity by utilizing QWT's efficient data representation and NN's powerful pattern recognition and predictive capabilities. Quantum bits (qubits) can encode large amounts of information in their quantum states, enabling more efficient data representation. This is especially useful for encoding large video files. Furthermore, quantum entanglement allows for correlated data representation across qubits, which can be exploited to capture intricate details and redundancies in video data more effectively than classical methods. The experimental results reveal that QWT achieves a compression ratio of almost twice that of traditional WT for the same video, maintaining superior visual quality due to more efficient redundancy elimination.

**Keywords:** Video compression; Quantum wavelet transform; Neural network; Adaptive coding; Optimization

## 1. Introduction

Video compression is a crucial technology in the modern digital world, enabling the efficient storage and transmission of video data. With the rapid growth of online streaming, video conferencing, and multimedia applications, the demand for effective video compression methods has increased significantly. Video compression techniques reduce the file size of video data by removing redundancies and irrelevant information, making it easier to store and transmit over networks [1]. Despite advances in video compression technologies, several challenges

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

persist: (1) Trade-off between quality and compression ratio: Achieving a high compression ratio often results in a loss of video quality. Finding the optimal balance between video quality and compression efficiency remains a critical issue. (2) Computational complexity: advanced compression algorithms, while effective, often require significant computational resources, making them less suitable for real-time applications and devices with limited processing power. (3) Diverse use cases: different applications have varying requirements. For instance, live streaming needs low latency, while video storage demands high compression ratios. Adapting compression techniques to meet these diverse requirements without compromising on quality or efficiency is challenging. (4) Emerging Technologies: The rise of 4K, 8K, and 360-degree video, as well as virtual and augmented reality, poses new challenges for video compression. These formats demand higher data rates and more efficient compression to maintain quality [2].

Redundancy plays a crucial role in achieving high video compression ratios. By identifying and eliminating spatial, temporal, and spectral redundancies, compression algorithms can significantly reduce the amount of data required to represent video content without sacrificing quality. Effective redundancy reduction ensures that most of the important visual information is retained, even at high compression ratios. This balance is crucial for maintaining perceived video quality while achieving efficient compression. Techniques such as transform coding, predictive coding, quantization, and entropy coding are key to leveraging these redundancies, enabling efficient video storage and transmission [1]. As video resolutions and formats continue to evolve, advanced neural network-based models offer promising avenues for further enhancing compression efficiency by more effectively exploiting redundancies in video data [3]. Vector quantization (VQ) plays a significant role in reducing redundancy in video compression by efficiently encoding similar patterns and reducing the overall data size. Its ability to control bitrates, adaptability to various compression needs, and computational simplicity make it a valuable tool in certain video compression scenarios. However, its effectiveness heavily depends on the quality of the codebook and the nature of the video content. Addressing challenges like block artefacts and ensuring adaptability to dynamic content are crucial for maximizing the benefits of VQ in video compression [4].

Wavelet transform plays a significant role within video coding, particularly in the compression stage, due to its ability to effectively represent video data with fewer coefficients. Wavelet transform decomposes video frames into multiple levels of resolution, separating the frame into different frequency components. This decomposition helps in isolating important features like edges and textures. The multi-resolution analysis allows for efficient encoding of various parts of the video frame (image) with different levels of detail, focusing more bits on important features and fewer on less significant details. Wavelet transform provides better energy compaction compared to traditional transforms like the Discrete Cosine Transform (DCT). This means that a significant portion of the signal's energy is represented by a small number of wavelet coefficients. By efficiently representing the image data with fewer coefficients, wavelet transform reduces the amount of data needed to encode the video, resulting in better compression rates [5]. Yet, Wavelet transforms can require more memory for storing intermediate coefficients and multiple resolution levels. This increased memory usage can be a significant drawback for real-time applications or devices with limited resources. Furthermore, most video codecs use motion compensation to exploit temporal redundancy between frames. Integrating motion compensation with wavelet-based coding can be more complex. Also, efficiently quantizing wavelet coefficients and allocating bits across different resolution levels can be more challenging than with DCT. The varying importance of wavelet coefficients at different scales requires careful bit allocation to maintain visual quality. Addressing these issues requires advances in algorithm design to make wavelet-based video coding practical and efficient for widespread use [6].

Quantum Wavelet Transform (QWT) has the potential to address several challenges faced by traditional wavelet transform (WT) in video coding. By leveraging the principles of quantum computing, QWT can offer significant improvements in computational efficiency, error resilience, and scalability [7]. Quantum computing inherently supports parallelism through superposition, allowing multiple calculations to be performed simultaneously. This can drastically reduce the time required for wavelet transform operations. Furthermore, quantum systems can represent and process large amounts of data more compactly through entanglement and superposition, potentially reducing memory requirements. Besides, quantum algorithms can explore larger solution spaces more efficiently, leading to better optimization of quantization and bit allocation strategies. The practical implementation of QWT will require advances in quantum hardware, error correction, and algorithm development. However, the potential benefits in terms of computational efficiency, memory usage, error resilience, and scalability make QWT a promising area of research for the future of video coding [8].

Video coding for the Internet of Things (IoT) involves the compression and transmission of video data over IoT networks. With the increasing use of video surveillance, remote monitoring, and smart cameras in IoT applications, efficient video coding techniques are essential to manage bandwidth, storage, and energy consumption. IoT devices often have limited resources, so lightweight and efficient video codecs are crucial. VQ is a lossy compression technique used in video coding to reduce the amount of data required to represent video signals. It involves

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

mapping blocks of video data (vectors) to a finite set of representative vectors in a codebook. However, video content can vary significantly between frames and scenes; so static codebooks may not adapt well to these variations, leading to suboptimal compression and quality. Dynamic codebook adaptation can address this but at the cost of increased complexity. Combining wavelet transform with VQ in video coding can potentially leverage the strengths of both techniques: the multi-resolution analysis capability of wavelet transform and the efficient data representation of VQ. However, ensuring that the combined wavelet and VQ approach scales well with increasing video resolutions and frame rates while maintaining efficient handling of spatial redundancy can be difficult.

Traditional video coding techniques, such as H.264 and HEVC, face limitations in terms of compression efficiency and computational complexity. As video resolutions and frame rates increase, these classical methods struggle to maintain performance without significant increases in computational power and storage requirements. Vector quantization training aims to improve data distribution and reduce quantization errors between data points and codebook components. Various algorithms are used to create codebooks, but their ideal value is not achieved due to varying distortion produced by codebook-generating techniques. Recent methods for coding videos use static codebooks, making them unsuitable for new data. In this work, by combining quantum wavelets and neural networks, it's possible to create an adaptive codebook for video coding system that offers superior compression efficiency and high-quality reconstruction. Quantum wavelets involve the use of quantum computing principles to perform wavelet transformations for decomposing video frames into different frequency components, allowing for efficient data representation and to identify significant features and reduce redundancy. The suggested model uses a neural network (NN) paradigm to extract high-level features from the decomposed components. The NN can be trained on a large dataset of video frames to learn efficient encoding strategies. QWTs can concentrate energy in fewer coefficients, producing a sparse representation of the data. This sparsity is beneficial for vector quantization as it reduces the number of vectors needed in the codebook, leading to more efficient storage and faster processing

The structure of this article is as follows: The relevant literature is presented in Section 2. The suggested quantum video coding system's design is presented in Section 3. Results from experiments and comparisons to relevant literature and the suggested methodology are presented in Section 4. The conclusion and plans for further research are summarized in Section 5.

## 2. Related Work

Image coding is a diverse field encompassing various methods from classical techniques to modern deep learning-based approaches. The development and enhancement of these techniques aim to improve compression efficiency, image quality, and computational performance [9]. The taxonomy of image coding methods include: (1) Lossless image coding (e.g. run-length encoding, entropy coding, and predictive coding);(2) Loss image coding (e.g. transform coding, vector quantization, block truncation coding, and perceptual coding); (3) Deep learning-based compression(e.g. variational auto encoders, and generative adversarial networks); (4) Graph-based methods; (5) Region-based coding. The choice of image coding technique depends on the specific requirements of the application, such as the need for lossless reconstruction, acceptable levels of data loss, computational resources, and the importance of compression efficiency. Each method has its strengths and trade-offs, and hybrid or advanced techniques often combine the best aspects of multiple methods to optimize performance [10].

In order to decrease the computational cost of Versatile Video Coding (VVC), the authors in Ref. [11] offered optimization algorithms for the determination of coding unit (CU) partition and intra-coding modes. First, CUs are classified as basic, fuzzy, or complicated according to their texture features; this is in relation to the high complexity of the CU partitioning procedure. Next, in order to make the (Rate-distortion optimization) RDO-based brute-force recursive search faster, the authors trained two random forest classifiers. In the meanwhile, a rapid hierarchical intra-mode search approach is developed using the texture attributes of CUs, such as texture complexity, texture direction, and texture context information, in order to simplify intra-mode prediction. To carry out interpolation in inter-prediction in High Efficiency Video Coding (HEVC), the authors in Ref. [12] presented an interpolation filter based on deep learning. Using the low-resolution image as a starting point, the network predicts the high-resolution image by extracting patches and features. By training the network to anticipate the HR image for the provided patch, the whole frame in HEVC may be generated by repeating the process. To simplify computing, the system employs a cleave technique.

A compression algorithm for multi-level feature maps taken from the feature pyramid network (FPN) structure was presented in Ref. [13] using principal component analysis (PCA). In contrast to existing PCA-based research, which perform PCA separately on each feature map, their method uses a generalized PCA-derived mean vector and generalized basis matrix to do away with PCA altogether. Using the spatial redundancy within these multi-level feature maps, further compression is accomplished by merging high-dimensional feature maps. The

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

generalized data does not suffer any compression loss as a consequence of the proposed VCM encoder, which forgoes the PCA procedure. For effective depth video compression, the authors in Ref. [14] suggested an inter-coding approach based on deep frame creation. By immediately generating the reconstruction of depth non-key frames and encoding depth key frames, the proposed approach may significantly cut down on temporal redundancies in depth videos. In addition, a high-quality depth non-key frame may be generated at the decoder side using a warping-based frame generation network that is boundary aware (Ba-WFGNet). The warped coarse depth non-key frame is produced by the Ba-WFGNet by making use of the temporal correlations among depth frames. Next, a boundary-aware refinement module is created to enhance the coarse-depth non-key frame for high-quality borders.

To improve compression efficiency and reduce computational complexity while maintaining visual quality, the introduced method in Ref. [15] is based on the deep video compression (DVC) framework. This framework replaces traditional block-based video compression with end-to-end video compression based on deep learning. Using the DVC as a foundation, the authors enhanced optical flow estimation by including hyper prior-based entropy coding into motion compression and by using window attention and rapid residual channel attention for motion vector estimation. To further improve residuals and the quality of reconstructed frames, they included an intermediary module for residual channel attention into both the encoding and decoding processes. To simulate the distribution of features, they used residual compression with hyper prior-based entropy coding. Additionally, reference frames were generated using learnt image compression for intrafirm coding using a rapid residual channel attention network. On the UVG dataset, experimental results shown that their technique outperforms both classic block-based and newer deep learning-based video compression algorithms in terms of PSNR and MS-SSIM.

To obtain a precise target bit rate and great video quality, the authors in Ref. [16] presented a learning-based mapping approach between video contents and rate control parameters. Two primary structural codes, spatial and temporal, are included in their framework. They set out to find the best settings for the coding tree unit (CTU) by using a learning-based particle swarm optimization for spatial and temporal coding. In order to properly control the bit in the real image, they included semantic residual information into the parameter update process for image-level temporal coding. Their method outperforms the state-of-the-art rate control in the HEVC reference software by an average of 0.19 dB and as much as 0.41 dB for low-delay P-coding structures, according to experimental data. A hybrid video compression framework is proposed in Ref. [17] to illustrate how deep learning-based methods may go beyond conventional coding techniques. An improved version of the Versatile Video Coding (VVC) standard, the Enhanced Compression Model (ECM) forms the basis of the proposed hybrid architecture. Using well-designed coding techniques, the authors have improved the coding performance of the latest ECM reference software. These techniques include block partitioning, deep learning-based loop filtering, and activating block importance mapping, which was integrated but previously inactive within ECM.

In Ref. [18], the authors presented a video coding system for machines that utilizes motion aided saliency analysis, which is based on versatile video coding (VVC). Using multi-object tracking as an example job, saliency analysis for important video frames is carried out using Gradient-weighted Class Activation Mapping (Grad-CAM), a deep learning visualization approach. In order to enhance the association performance of multi-object tracking and take into account the complexity of saliency analysis, a motion aided saliency analysis has been developed for non-key frames. This technique predicts saliency maps for non-key frames using motion vectors and saliency maps of key frames. A coding tree unit (CTU)-level Quantization Parameter (QP) adjustment technique has been created using all the frames' saliency maps; this approach uses bitrate allocation to prioritize salient areas related to machine vision. The suggested method saves at least 20.3% bitrate while maintaining the same level of tracking accuracy, according to the results of the study, as compared to VVC.

To repack feature sequences in a way that is more suitable for current video codecs, the authors of Ref. [27] proposed a distance-based patch tiling and intra-block quilting approach based on statistical analysis of feature attributes in the channel dimension. The results of the experiment show that their strategy outperforms benchmark methods by 65.54 % in terms of BD-rate. A novel machine learning-based approach to micro video coding is discussed in Ref. [28]. Intra-prediction, inter-prediction, transformation, quantization, entropy coding, and loop filtering form the basis of the video coding processing architecture. The results of the experiment demonstrated that this strategy substantially improves the processing ability and coding accuracy of microscopic video signal coding.

An upgraded version of the latest Versatile Video Coding (VVC) standard using a new learning adaptive motion search algorithm was proposed as a low-complexity solution for surveillance video coding in Ref. [19]. The method was developed by analysing the spatial texture and temporal motion features of surveillance video. The authors began by researching and defining a set of spatial and temporal elements that represent the texture and motion in video surveillance. To properly allocate a search range for the VVC motion search, these features are

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

used in conjunction with a machine learning method. Second, the authors used an adaptive Test Zone (TZ) search to cut down on search points. This search method involves early termination of TZ phases in response to variations in spatial-temporal variables. Their method outperformed the state-of-the-art VVC solution and relevant benchmarks in performance evaluations carried out on a rich set of surveillance videos and relevant benchmarks, saving approximately 33% of the encoding time while requiring almost no compression loss.

In Ref. [20], the authors introduced LSSVC, an end-to-end learned spatially scalable video coding scheme, inspired by learned video coding. They used motion, texture, and latent information from the base layer (BL) as interlayer information for compressing the enhancement layer (EL). To reduce interlayer redundancy, they designed three modules: a contextual motion vector encoder-decoder, a hybrid temporal-layer context mining module, and an entropy model using up sampled BL latent information. Experimental results showed that LSSVC outperforms H.265/SHVC reference software. The work presented in Ref. [21] aims to reduce the computational complexity of the VVC by establishing a large intra-prediction database and utilizing multitask learning-based intra-mode decision framework. Experimental results showed that this proposal can reduce complexity by up to 30% while slightly increasing the Bjontegaard bit rate.

In Ref. [22], the authors established optimized video coding configurations based on video content to reduce bitrate while maintaining perceptual quality. They extracted spatial-temporal features from original videos and used support vector machines (SVM) for spatial-temporal rescaling, improving efficiency. This approach is compatible with existing encoders and decoders, making it suitable for practical applications. Experimental results showed high accuracy in predicting optimal encoding configurations, resulting in significant bitrate savings for videos of different resolutions and frame rates. The study presented in Ref. [23] offers a new method for predicting semantic decoder parameters using temporal correlation, tested using an auto encoder-based semantic communication system. The approach outperformed the Neural Network Encoder-Decoder (NNCodec) in terms of rate distortion, PSNR gains, and average bitrate saving, with gains ranging from 3 to 25 dB depending on video complexity.

In Ref. [24], the study demonstrated the use of a residual deep convolutional neural network (Res DCNN) in the VVC framework to improve video quality post-compression. The Res DCNN framework, consisting of layers and residual density blocks, effectively extracts features from video data. The study used VVC Test Model 22.2 for encoding and decoding processes and TensorFlow for model training, resulting in an increase in image quality and a PSNR value from 35.36 dB to 37.33 dB. In Ref. [25], the authors suggested a content adaptive down sampling scheme to enhance video coding efficiency at low bitrate. They extracted content-aware spatial-temporal features like normalized spatial information, normalized temporal information, and spatial masking perceptual features, and used support vector machine to predict the optimal coding configuration. Experimental results showed significant bitrate savings for video sequences at different resolutions with low computational complexity.

In Ref. [26], the authors found that standard-coded videos significantly reduce the performance of deep vision models. They proposed the first end-to-end learnable deep video codec control that considers bandwidth constraints and downstream deep vision performance while adhering to standardization, demonstrating that this approach better preserves downstream performance than traditional video coding. In Ref. [27], the authors recommended an efficient coding unit partitioning algorithm using an extreme learning machine (ELM) to reduce coding complexity and ensure efficiency. They model the coding unit size decision as a classification problem and train an ELM classifier to predict the size. The experiments were verified using the VVC reference model, revealing significant coding complexity reduction and improved image quality.

The authors in Ref. [28] have developed a versatile neural video coding (VNVC) framework that uses compact representations for reconstruction and direct enhancement/analysis. This framework is versatile for both human and machine vision, with a feature-based compression loop. The intermediate feature is used for motion compensation and estimation, and is directly fed into video reconstruction, enhancement, and analysis networks. The evaluation shows that the VNVC framework with the intermediate feature achieves high compression efficiency for video reconstruction and satisfactory task performance with lower complexities. In Ref. [29], the authors propose a deep video prediction method using block clustering through a neural network. This approach addresses the issue of blocks with multiple clusters affecting the prediction performance. The neural network consists of a spatial feature prediction network and a clustering network, which use spatial features in both vertical and horizontal directions. The mean square error is used as a loss function between the original and predicted blocks, with a penalty introduced for output values far from both ends. The simulation results showed a 12.45% bit rate savings under the same distortion condition compared to the VVC video coding standard.

In Ref. [30], the authors introduced iHELP, a state-of-the-art adaptive instant learning-based model, to address computational complexity in encoders' adaptive block structures. The model improves coding efficiency and quality while enhancing encoding speed. It uses entropy-based block similarity to predict the splitting decision of the largest coding unit (LCU), based on spatial and temporal correlations. The iHELP method has been rigorously

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

evaluated using the HEVC standard, achieving an 80% reduction in encoding time while maintaining comparable PSNR performance compared to the rate-distortion optimization (RDO) approach. The study presented in Ref. [31] uses the learning-based AV1 complexity controller (LACCO) to dynamically optimize the encoding time of the AV1 encoder for HD 1080 and UHD 4K resolution videos. LACCO predicts future frame encoding times and classifies input videos using machine learning models. When integrated into the AV1 encoder's reference software, LACCO reduces encoding time by 10-70%, with average error results ranging from 0.11-1.88 percentage points for HD 1080 resolution and 0.14-3.33 percentage points for UHD 4K resolution.

Codebook-based vector quantization is often considered a strong choice for video coding due to several key advantages. VQ compresses data by mapping vectors (blocks of pixels) to a limited set of code words in a codebook. This can result in high compression ratios because it reduces the amount of data needed to represent the image or video frame. Furthermore, video frames can be divided into blocks, and similar blocks are encoded using the same code word, leading to efficient compression of redundant information. In addition, videos have significant temporal redundancy, meaning consecutive frames are often very similar. VQ can exploit this redundancy by reusing code words across frames. Finally, once the codebook is generated, the encoding and decoding processes are relatively fast. This is beneficial for real-time video applications where computational efficiency is crucial. Yet, the process of generating an optimal codebook can be computationally intensive and time-consuming especially in the context of increased data volume such as high-definition (HD) or ultra-high-definition (UHD) videos. It requires a significant amount of training data and careful design to ensure the codebook is representative of the video content.

Different from state-of-the-art video coding methods that rely on traditional frequency transformations to represent video' frames that do not adequately deal with large volumes of data, the suggested model utilizes quantum wavelet transforms to handle high-dimensional data more effectively. Quantum wavelet transforms can handle larger datasets more efficiently, leading to better compression ratios. This is because quantum algorithms can exploit the superposition and entanglement properties of quantum states to capture and represent more information with fewer coefficients. This increased efficiency in representation directly translates to more effective vector quantization. When combined with neural networks, this leads to highly compressed yet accurate representations. In general, dimensionality reduction using quantum wavelet transforms combines the benefits of wavelet transforms and quantum computing to efficiently handle high-dimensional data such as HD video frames. This approach involves applying QWT to decompose the data, selecting significant coefficients, and forming reduced-dimension vectors for VQ codebook generation that offers speed, efficiency, high-quality compression, scalability, and noise robustness.

## 3. Methodology

The research suggested a new method that combines intra-frame and inter-frame video coding to reduce spatial, temporal, and statistical redundancy. Intra-frame coding involves de-correlation of input frame pixels using Quantum wavelet transformation and quantization, resulting in more efficient encoding. Quantization information is derived from Differential Pulse Code Modulation (DPCM), a fundamental component of lossless compression algorithms. The model's success relies on the optimal codebook for vector quantization, which is built using the genetic algorithm model. The fitness function is the Euclidean distance between the initial codebook and each video frame. QWTs can potentially provide a more compact representation of video frames. This means better compression ratios can be achieved, which reduces the amount of data needed to store or transmit video frames without significantly sacrificing quality. The ability to represent video frames more efficiently can lead to reduced bandwidth requirements and storage costs. Furthermore, QWTs can scale more efficiently with the increasing resolution and complexity of video frames. As video resolutions continue to grow, the ability of QWTs to handle large amounts of data efficiently becomes increasingly important [32]. The objective herein is to prove that QWT enhances the ability to extract detailed features from signals, leading to more accurate and representative codebooks. The following subsections briefly highlight the main components of the suggested model.

### Phase 1: Encoding

### Step 1: Initial Codebook Generation

An off-line codebook is created for each video by isolating the frames' moving parts (foreground) and their static parts (background), representing them as code words. Background subtraction is used to separate out moving items, a common task in detecting moving objects in videos captured by stationary cameras (see Fig.1). In our case, Linde-Buzo-Gray (LBG) method is employed to construct an initial codebook. The LBG algorithm iteratively refines a codebook by assigning input vectors to the nearest code vectors and updating the code vectors to be the

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

centroids of the assigned vectors. By integrating these steps, the algorithm ensures that the codebook accurately represents the input data, minimizing the quantization error. The main steps of the LBG method include:

1. Initialization

Select an initial codebook, often by randomly selecting a subset of the input vectors or by taking the mean of the entire dataset.

2. Assignment

Assign each input vector to the nearest code vector (centroid) in the codebook.

3. Update

Recalculate each code vector as the centroid (mean) of the vectors assigned to it.

4. Convergence Check

Check if the codebook has converged (i.e., the changes in code vectors are below a certain threshold). If not, repeat steps 2 and 3.

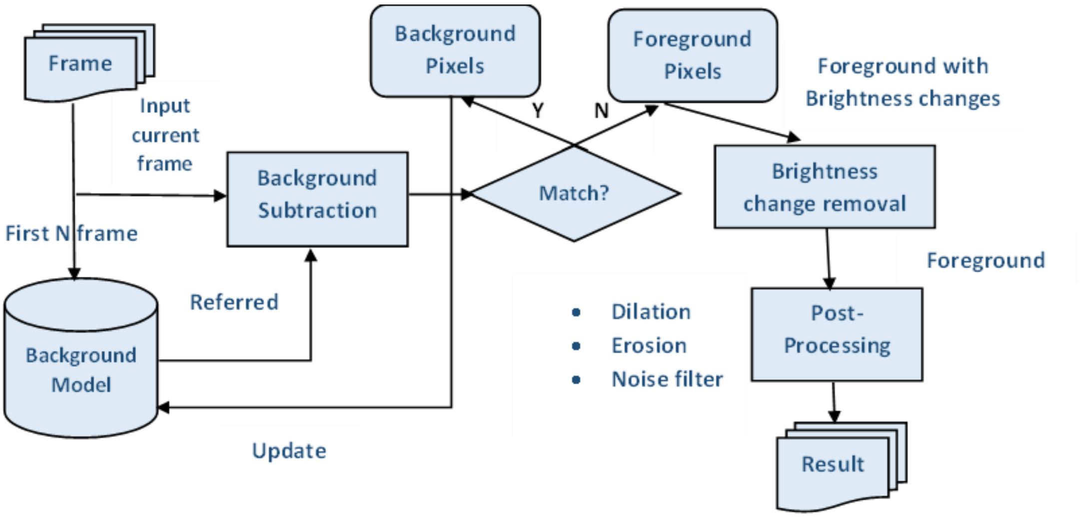

**Figure 1.** Basic steps for background subtraction algorithms.

**Step 2: Codebook Optimization**

Genetic algorithms are a type of evolutionary algorithm that simulate the process of natural selection to find optimal solutions. Using a genetic algorithm for vector quantization codebook optimization involves iteratively improving a population of codebooks through selection, crossover, and mutation. This approach can effectively minimize quantization error, leading to an optimized codebook that accurately represents the input data. The following pseudocode outlines the main steps and functions involved in implementing this optimization process.

- Initialization: The population of codebooks is initialized randomly. Each codebook in the population represents a potential solution.
- Fitness Evaluation: The fitness of each codebook is evaluated based on the total quantization error (distortion) which is the sum of the squared Euclidean distances between each input vector and its nearest code vector. The goal is to minimize this total distortion. Lower distortion means better fitness.
- Selection: Codebooks are selected for reproduction based on their fitness. Codebooks with lower distortion (higher fitness) are more likely to be selected.
- Crossover: Selected codebooks are paired, and crossover operations are performed to create offspring. The crossover rate determines the likelihood of recombination.
- Mutation: Offspring undergo mutation with a certain probability to maintain genetic diversity. Each code vector in the codebook has a chance to be replaced by a random vector from the input vectors.
- Replacement: The old population is replaced with the new population of offspring.
- Convergence Check: The algorithm checks for convergence based on the fitness threshold or the maximum number of generations.

Using genetic algorithms for codebook optimization in vector quantization offers several advantages over swarm optimization techniques. GAs provide a good balance between exploration (searching new areas of the solution

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

space) and exploitation (refining existing solutions). This balance is achieved through selection, crossover, and mutation. While swarm optimization (e.g., PSO) focuses more on exploitation by updating particle positions based on personal and global best positions, it may not explore the solution space as extensively as GAs, particularly in complex problems. Moreover, GAs can be more effective in handling high-dimensional problems due to their population-based approach and genetic diversity. Yet, swarm optimization techniques might face difficulties with high-dimensional problems due to the curse of dimensionality, where the number of dimensions increases the complexity of the search space.

**Step 3: Quantum Haar Wavelet Transform**

Designing an efficient data coding system involves balancing three key factors: compression ratio, lossy compression distortion, and processing power. Lossless compression using DPCM and lossy compression using an optimized codebook-based neural network are the two primary coding phases employed in our suggested model. On the quantum Haar wavelet transform (QHWT) domain of each video's frame, both phases operate for the processing of high volumes of data in large-scale applications. By decomposing a video's frame into QHWT, less significant coefficients are selectively discarded, resulting in a compressed representation of the image while preserving important details. QHWT offers significant advantages for video frame representation in coding applications, including high compression ratios, reduced distortion, computational efficiency, enhanced feature extraction, scalability, and compatibility with future quantum computing advances. These benefits make QHWT a promising approach for next-generation video coding technologies [11-14]. In formal, general wavelet transform can be expressed:

$$F(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\, \psi_{a,b}^{*} \left(\frac{t-b}{a}\right) dt, \qquad (1)$$

$$\psi\left(\frac{t-b}{a}\right) = u\left(\frac{t-b}{a}\right) - 2u\left(\frac{t-b}{a} - \frac{1}{2}\right) + u\left(\frac{t-b}{a} - 1\right), \qquad (2)$$

$$\psi_D\left(\frac{q-j}{K}\right) = \begin{cases} +1, & 0 \le (q-j) < \left(\frac{K}{2}\right), \\ -1, & \left(\frac{K}{2}\right) \le (q-j) < K, \\ 0, & otherwise, \end{cases} \qquad (3)$$

Where $\psi$ is called the mother wavelet function in complex conjugate form, and $a$, $b$ are the time dilation and displacement factors, respectively. The discretized version of the Haar wavelet function is defined in Eq.3, where $t = q.\Delta t$ , $b = j.\Delta t$, $a = K$, $\Delta t$ is the sampling period, and $K$ is the Haar window size in samples. The expression for the discrete Haar wavelet transform can be derived as where $N$ is the number of data samples:

$$F_D(j,K) = \sum_{q=0}^{N-1} f_D(q.\Delta t) \psi_D\left(\frac{q-i}{K}\right), \qquad (4)$$

Classical signal samples can be encoded as the coefficients of a quantum state, which is in superposition of its constituent basis states [11-14][46]

$$|\psi\rangle = \sum_{q=0}^{N-1} f(q.\Delta t)|q\rangle, \;\; \sum_{q=0}^{N-1} |f(q.\Delta t)|^2 = 1, \qquad (5)$$

$$|\psi\rangle_{QHT} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{q=0}^{N-1} f\,(q.\Delta t)\;\; \psi_D\left(\frac{q-i}{K}\right) |j\rangle, \qquad (6)$$

where $n$ is the number of qubits, $N = 2^n$ is the number of basis states of the quantum system, $|\psi\rangle$ is the input quantum state, $|\psi\rangle_{QHT}$ is the output quantum state of quantum Haar Transform $(QHT)$.

The Haar wavelet kernel can be generalized by quantum operations using $n$ qubits and a $d$-dimension kernel, where $\otimes$ is the Kronecker product, $H$ is the Hadamard transform, and $I$ is an identity matrix. The quantum Haar function can be implemented using $d$- entangled $H$ gates and $n-d$ entangled $I$ gates.

$$U_{QHT} = I_{2^{(n-d)}} \otimes H_{2^d}\,, \qquad (7)$$

$$H_{2^d} = \underbrace{H \otimes H \otimes \ldots \otimes H}_{d}\,, \qquad (8)$$

$$I_{2^{(n-d)}} = \underbrace{I \otimes I \otimes \ldots \otimes I}_{(n-d)}\,, \qquad (9)$$

$$H = H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad (10)$$

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

$$I = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} , \qquad (11)$$

$$U^{2D}_{QHT} = I_{2^{(n-2)}} \otimes H_{2^2} = I_{2^{n/4}} \otimes H_4 = I_{N/4} \otimes H_4 , \qquad (12)$$

$$H_4 = H \otimes H = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} , \qquad (13)$$

Herein, we apply shuffle permutation operations, partitioning the vector in half and shuffling the top and bottom partitions of the halves, before and after applying the QHT kernel, respectively. The QHT kernel is performed on a set of $k$ points, where $k = 2^d$. For 2D-QHT operations, the input permutations, $p^{2D}_{in}$ and $p^{2D}_{out}$ are calculated as (see Fig. 2. for the design circuit model for 2D-QHT):

$$P^{2D}_{in} : \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ . \\ . \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ x_{(n_{rows}+2)} \\ X_{(n_{rows}+3)} \\ . \\ . \\ . \\ . \\ . \\ . \\ X_{(N-1)} \end{bmatrix} \longmapsto \begin{bmatrix} x_0 \\ x_1 \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ \vdots \\ . \\ x_2 \\ x_3 \\ x_{(n_{rows}+2)} \\ x_{(n_{rows}+3)} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ X_{(N-1)} \end{bmatrix} , \quad P^{2D}_{out} : \begin{bmatrix} y_0 \\ y_{(n_{rows/2})} \\ y_{(N/2)} \\ y_{((n_{rows/2})+(N/2))} \\ . \\ . \\ . \\ . \\ . \\ . \\ y_{(N-1)} \end{bmatrix} \longmapsto \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ . \\ . \\ . \\ y_{(N-1)} \end{bmatrix} , \qquad (14)$$

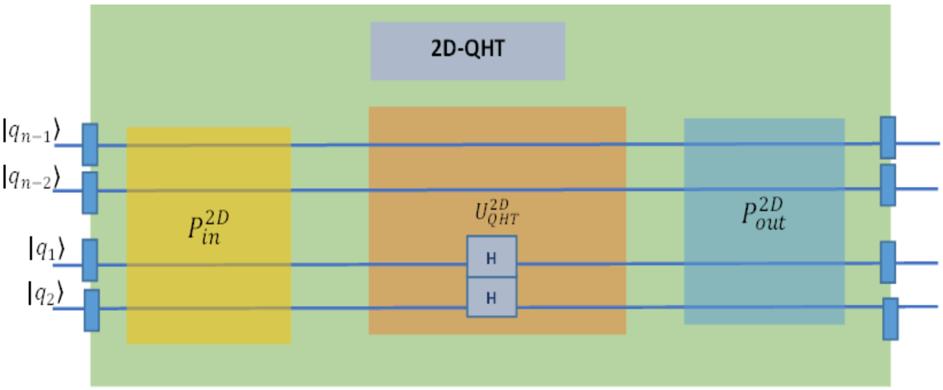

**Figure 2.** 2D-quantum Haar transform circuit.

The 2D-QHT model consists of input permutation models ($P_{in}$), followed by Haar kernel model ($U_{QHT}$) and then output permutation models ($P_{out}$). The first step in the 2D-QHT operation is the input permutation $P^{2D}_{in}$. The permutations can be modeled as quantum circuits with multiple swap gates. The 2D-Haar transformation, $U^{2D}_{QHT}$, is modeled using a pair of Hadamard gates. The Hadamard pair operation reduces to kernel operations on a set of four coefficients. The final step in the 2D-QHT operation is the output permutation $P^{2D}_{out}$.

$$2D - QHT : P^{2D}_{out} . U^{2D}_{QHT} . P^{2D}_{in} , \qquad (15)$$

In our case, Qiskit's is utilized for developing and experimenting with quantum algorithms like the Haar wavelet transform on classical computers [33]. Qiskit includes powerful simulation backend such as Aer, which allow

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

users to simulate quantum circuits on classical hardware efficiently. This is crucial for testing and debugging quantum algorithms like the Haar wavelet transform.

**Step 4: Lossless Coding**

Lossless coding refers to a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data. In our case, every video frame's high-energy low-wavelet-frequency coefficients (QHT) undergo lossless compression to ensure that important details (salient features) are preserved. This application makes use of DPCM, a signal encoder that expands upon PCM by including characteristics predicted from the signal samples as illustrated in Fig. 3. DPCM uses a prediction mechanism where the current frame is predicted based on previous frames. The prediction error (difference) is then encoded, which is typically smaller and more compressible. The difference values (residuals) between successive frames typically have smaller magnitudes and less variance than the original values, leading to better compression ratios.

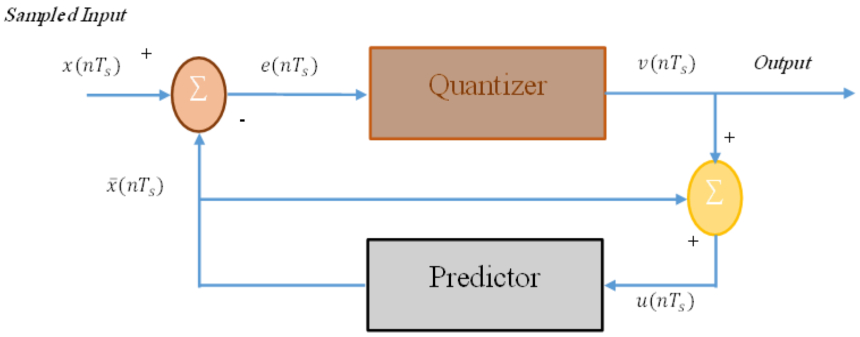

**Figure 3.** DPCM's main steps.

**Step 5: Loss Coding**

In order to achieve a high compression ratio, each frame's high wavelet frequency coefficients, which have little energy and are not salient features, are loss compressed. With the help of an optimized codebook for every video, the neural network compresses these coefficients using dynamic vector quantization, which is inserted into the hidden layer as an activation function. Using a VQ codebook as an activation function in the hidden layers of a neural network for video coding is a new approach that combines the strengths of neural networks with the efficiency of vector quantization. In general, using the codebook as an activation function in neural networks for video coding can provide several benefits:

- By using a codebook, the neural network can enforce sparsity in the representations, which is beneficial for compressing high-dimensional video data into a lower-dimensional space without significant loss of quality.
- The use of a codebook can make the neural network more robust to variations in video data, such as changes in lighting, occlusions, or noise. This is because the codebook captures the essential patterns in the data, filtering out irrelevant variations.
- Codebook-based activation functions can act as a form of regularization, helping to prevent overfitting by constraining the model to use a predefined set of codes.
- The use of a codebook can lead to faster inference times because the network only needs to look up codes in the codebook rather than performing complex computations. This can be particularly advantageous for real-time video applications.
- By using a discrete set of codes, the computational complexity of the network can be reduced, leading to lower power consumption and faster processing times.
- The codes in the codebook can be interpreted as specific patterns or features in the video data, making the neural network's decisions more transparent and understandable.

As an enhancement over current methods that employ the neural network as a black box for loss compression, the proposed methodology incorporates the optimized codebook from step 2 as an activation function into each neuron of the hidden layer. Our vector quantization neural network utilizes supervised learning in conjunction with vector quantization as a compression method. In general, traditional video coding methods often require extensive hand-crafting of algorithms and tuning of parameters. In contrast, neural networks can be trained on large datasets to automatically learn optimal encoding strategies, reducing the time and effort needed for development. Finally, a single vector with well-defined boundaries is created by combining the quantized coefficient vector from the DPCM lossless compression stage with the VQ index vector from the neural network loss compression stage. This allows for their simultaneous de-coding. A single unified vector is present in each frame in this case.

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

### Phase 2: Decoding

Decoding entails the exact opposite steps as encoding. The quantized coefficients and VQ index from each frame are part of this vector, which is used to produce the combined coefficient vector. Inverse DPCM applied to the quantized coefficients yields the uncompressed coefficients (low frequencies). Each frame has one high frequency coefficient vector, and to get them, we use the previously recorded codebook table to convert the index value of the VQ vector to its matching vector. In order to get the decompressed frame, we invert the Quantum Haar Wavelet Transform (IQHT), combine the bands from the previous phase (LL and HL) with the two unmodified bands from the database (LH and HH), and then process the image. Repeat these procedures for every row of the compressed matrix to recover the original video. Herein, multidimensional inverse quantum Haar transform 2D-IQHT is then applied to reconstruct the original video' frames. The 2D-IQHT model in Figure 4 consist of inverse output permutation model$(P_{out})^{-1}$, followed by Haar kernel models $U_{QHT}$ and then inverse input permutation model $(P_{in})^{-1}$. The inverse models are equivalent to the direct models, as the permutation operations are reversible. The IQHT operations for 2D is summarized as unitary transformations as:

$$2D - IQHT: (P_{in}^{2D})^{-1}. U_{QHT}^{2D}. (P_{out}^{2D})^{-1} . \qquad (16)$$

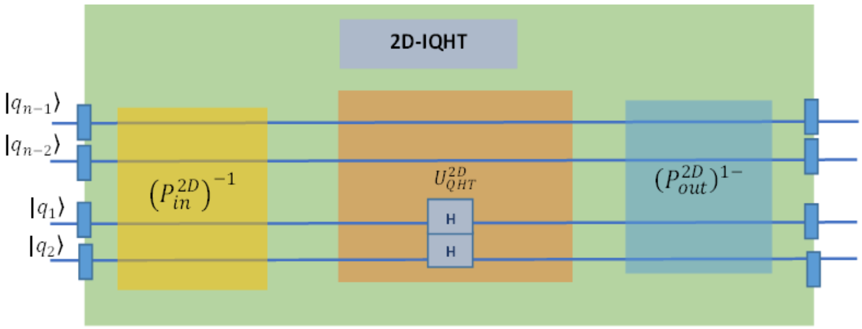

**Figure 4.** 2D inverse -quantum Haar transform circuit.

## 4. Results and discussions

The suggested approach was evaluated using a benchmark video dataset from https://media.xiph.org/video/derf/. The testbed consists of videos in various formats, including avi and mpeg. Additionally, eight video sequences with varying spatial and temporal information were downloaded in the uncompressed format *.YUV, available online at http://medialab.sjtu.edu.cn/web4k/index.html, to validate the model's dependability when dealing with Full HD (1920x1080) and Ultra HD (3840x2160) video sequences. The model was implemented in MATLAB (Release 2022a) and executed on a laptop with an Intel (R) Core (TM) i7 CPU, L640 @ 2.31 GHz 2.31 GHz, 4 GB RAM, 64-bit operating system, Microsoft Windows 8.1 Enterprise, and 500 GB hard disk. The model prototype simulates QHT using Qiskit toolbox in Python programming language. Compression ratio (CR) and signal-to-noise ratio (PSNR) are metrics used to evaluate the effectiveness of a proposed model compared to other strategies. A higher CR indicates a better signal-to-noise ratio, with the original frame as the signal and the reconstruction error as the noise. Using the right crossover and mutation probability values prevents GA from being stuck at local optima, a problem in all GA-based video compression methods. A good GA balances wide exploration and deep exploitation.

The first set of experiments aimed to evaluate the impact of QHT detail bands on the CR and PSNR quality of the proposed system. Results in Table 1 showed that compression ratio is independent of band type, as the proposed approach encodes fewer loss detail coefficients than approximation coefficients. A single index was used to decode these small coefficients from a lookup table, resulting in identical compression ratios across all spectrums. The HL band provided the highest PSNR due to its salient features, particularly in the frame with texturing and smooth areas. The Haar wavelet transform is known for its ability to represent data sparsely by focusing on significant coefficients and discarding insignificant ones. QHT enhances this by efficiently transforming the entire video data set, leading to a more compact representation of the video. By decomposing video frames into different frequency components, QHT can more effectively isolate and retain essential information while discarding redundant data. This process reduces the amount of data needed to represent the video without significant loss in quality. Furthermore, quantum computing is highly scalable and can manage large and complex video datasets more efficiently than classical computing. As video resolutions and frame rates increase, QHT can handle these larger datasets without a proportional increase in computational resources. Quantum algorithms can perform transformations with high precision, which ensures that the important features of the video frames are preserved

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

accurately during compression. This precision minimizes the loss of critical details, resulting in higher PSNR. The low-frequency band retains the most significant features of the video frame, which are crucial for maintaining visual quality. High PSNR is often associated with the preservation of these features during compression and decompression.

**Table 1:** The relationship between system performance evaluation in terms of CR and PSNR regarding the QHT details bands (codebook size=256).

| Video | HH | | HL | | LH | |
|---|---|---|---|---|---|---|
| | CR | PSNR | CR | PSNR | CR | PSNR |
| Aquarium | 35.55 | 39.72 | 35.55 | 40.18 | 35.55 | 38.81 |
| Miss America | 35.67 | 46.51 | 35.67 | 46.97 | 35.67 | 46.67 |
| Runners (UHD) | 35.52 | 44.83 | 35.52 | 45.08 | 35.52 | 44.92 |
| Fountains (FHD) | 35.38 | 44.96 | 35.38 | 45.15 | 35.38 | 44.89 |

**Table 2:** Effect of Codebook Size of Vector Quantization (HL bands).

| **Codebook size** | **Video** | **CR** | **PSNR** |
|---|---|---|---|
| **16** | Akiyo | 30.61 | 46.93 |
| | Miss America | 30.76 | 46.97 |
| | Runners (UHD) | 26.81 | 46.18 |
| | Fountains (FHD) | 26.84 | 46.25 |
| **32** | Akiyo | 32.50 | 46.71 |
| | Miss America | 32.36 | 46.73 |
| | Runners (UHD) | 34.51 | 46.08 |
| | Fountains (FHD) | 34.18 | 46.15 |
| **128** | Akiyo | 33.65 | 46.61 |
| | Miss America | 33.68 | 46.52 |
| | Runners (UHD) | 35.24 | 45.50 |
| | Fountains (FHD) | 35.16 | 45.85 |
| **256** | Akiyo | 35.61 | 46.34 |
| | Miss America | 35.64 | 46.11 |
| | Runners (UHD) | 35.52 | 45.08 |
| | Fountains (FHD) | 35.38 | 45.15 |

The second set of experiments showed how codebook size affects the proposed system's efficacy. Table 2 verified that a larger codebook enhances CR but reduces PSNR. CR stabilizes or even significantly improves as the codebook size is raised to 256 or higher. The results may be due to a larger codebook vector containing more information. Since this information is stored by a single index, compression ratio improves as codebook size increases. The lookup table searches for data during decompression, thus if the index representation is incorrect, a lot of data will be affected, lowering PSNR and visual quality. In summary, the vector quantization codebook and QHT resolutions are linked through their combined application in high resolution video coding. Haar transformation, by breaking down a signal into different resolution components, prepares the data for effective compression through vector quantization, where a codebook represents the transformed data efficiently. After QHT, the transformed data typically has a more structured and predictable form. High-frequency noise and redundancies are minimized, leaving a more concise representation of the essential features. This can make the data more amenable to efficient quantization, as the transformed coefficients may exhibit more sparsity or follow a more predictable distribution. The clearer structure in the transformed data allows for more efficient codebook design. In summary, QHT can adaptively capture features at different resolutions. In the VQ stage, the codebook can be tailored to these resolutions, potentially reducing the number of vectors needed for higher-resolution details.

Our suggested methodology will be compared to other video compression methods in the next set of experiments [34][35]. An ant colony-based motion estimation approach with a modified fast Haar wavelet trans-form remove temporal redundancy in the first comparative approach [34]. The second approach [55] employs a fast curve let transform, run-length encoding, and Huffman coding to remove spatial redundancy. By using quantum wavelet

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

transform as a video' frame representation, optimal vector quantization using GA, and DPCM, the suggested approach removes temporal, spatial, and statistical redundancy. Table 3 shows that the suggested method generates video with higher PSNR than the first and second algorithms. PSNR increases by 8% and 11% with the new technique. The suggested system improves CR by 2% and 7% over other techniques.

QHT can achieve higher compression ratios due to its ability to process data in superposition states. This means that it can represent and manipulate multiple data points simultaneously, leading to more efficient data representation and compression. While effective, traditional transformations often require more coefficients to represent the same amount of information, leading to lower compression ratios. Furthermore, quantum computing allows for handling data with higher precision and resolution (UHD and FHD). This means that QHT can achieve finer granularity in compression, preserving more detail in the video while still reducing data size significantly. Traditional frequency transformations, while effective, might not achieve the same level of precision and resolution, especially for high-dimensional or complex data. QHT preserves critical features of the video data better, ensuring that compressed video retains high quality. This better feature preservation is crucial for maintaining high PSNR. For traditional frequency transformations, preserve features to some extent but might not be as effective at maintaining high quality at higher compression ratios, resulting in lower PSNR.

**Table 3:** Comparative results with related video coding techniques.

| **Video** | **Choudhury, H. et al. [51]** | | **Nithin, S. et al. [52]** | | **Proposed Model** | |
|---|---|---|---|---|---|---|
| | *CR* | *PSNR* | *CR* | *PSNR* | *CR* | *PSNR* |
| Man Running | 32.35 | 36.76 | 25.98 | 33.49 | 33.63 | 45.24 |
| Traffic Road | 31.45 | 30.63 | 26.24 | 32.23 | 33.50 | 36.23 |
| Aquarium | 30.49 | 30.34 | 25.72 | 31.70 | 33.50 | 35.84 |
| Akiyo | 31.83 | 32.09 | 26.41 | 32.28 | 33.64 | 45.93 |
| Miss America | 31.41 | 39.80 | 27.13 | 50.31 | 33.68 | 45.91 |
| Boxing | 32.06 | 38.65 | 28.34 | 34.46 | 33.62 | 43.60 |
| Runners (UHD) | 35.52 | 39.16 | 29.84 | 35.09 | 35.52 | 45.08 |
| Fountains (FHD) | 35.38 | 39.97 | 29.45 | 35.16 | 35.38 | 45.15 |

Regarding complexity analysis in terms of big $O$ notation, for a video with $F$ frames, each containing $P$ pixels, the complexity the classical Haar wavelet transform would be $O(F.P\log P)$. The quantum Haar wavelet transform can theoretically achieve an exponential speedup for certain operations $O(F.\log P)$. The quantum Haar wavelet transform is significantly more efficient for large datasets (e.g., high-resolution videos) due to its logarithmic time complexity. In summary, Quantum Haar wavelet transform-based video coding, particularly when combined with optimal codebook techniques, offers substantial advantages in terms of compression efficiency and computational performance as revealed in Table 4. On average, QHT and optimal codebooks achieve higher compression ratios compared to QHT without optimal codebook. Furthermore, Quality metrics in term of PSNR, Structural Similarity Index (SSIM) show improvement in quantum approaches, indicating better preservation of video quality. Finally, Quantum methods significantly reduce computational time, especially for large datasets.

**Table 4:** Comparative results with and without optimal codebook for benchmark datasets (average)

| **Method** | **CR** | **PSNR** | **SSIM** | **Computational Time** |
|---|---|---|---|---|
| QHWT with Optimal Codebook | 35:1 | 46 | 0.96 | Low (260 ms) |
| QHWT without Optimal Codebook | 25:1 | 39 | 0.92 | Low (200 ms) |
| HWT with Optimal Codebook | 32:1 | 35 | 0.90 | High( 600 ms) |

## 5. Conclusion

Video coding involves compressing video data to reduce the amount of storage space or transmission bandwidth required. Quantum wavelet transforms (QWT) offer a promising avenue for video coding due to their potential for exponential speedup in processing time compared to classical methods. Additionally, using an optimal codebook can further enhance compression efficiency. This approach leverages the QHWT for efficient frame representation and an optimal codebook for effective compression. The QHWT offers significant computational speedup through quantum parallelism, while the optimal codebook enhances compression efficiency by minimizing redundancy. The QHWT leverages quantum parallelism, resulting in logarithmic time complexity compared to classical

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

methods. This leads to significant time savings, particularly for high-resolution videos with large amounts of data. The optimal codebook further enhances compression by reducing the bit rate needed to encode wavelet coefficients. High PSNR and SSIM values indicate that video quality is well preserved, with minimal loss of detail and high similarity to the original frames. These advantages are particularly relevant as video resolutions continue to increase, demanding more efficient compression techniques. Future plan includes conducting scalability studies to assess the performance of QHWT-based video coding on large-scale video datasets. Evaluating the impact of varying video resolutions, frame rates, and compression requirements on the algorithm's performance. Finally, developing quantum-safe encryption techniques to protect compressed video data during storage and transmission.

**Funding**: “This research received no external funding”

**Conflicts of Interest:** “The authors declare no conflict of interest”.

**Data Availability Statement:** Data is available on https://media.xiph.org/video/derf/

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

## References

[1] Zhang, Y., Zhu, L., Jiang, G., Kwong, S., Kuo, C. A survey on perceptually optimized video coding. ACM Computing Surveys, 55(12):1-37, 2023.

[2] Bidwe, R., Mishra, S., Patil, S., Shaw, K., Vora, D., Kotecha, K., Zope, B. Deep learning approaches for video compression: a bibliometric analysis. Big Data and Cognitive Computing, 6(2):44, pp.1-40, 2022.

[4] Yang, J., Yang, C., Zhai, Y., Wang, Q., Pan, X., Wang, R. Improving learned video compression by exploring spatial redundancy. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2860-2864, 2024.

[5] Gowrisetty, V., Fernando, A. Static video compression’s influence on neural network performance. Electronics, 12(1):1-23, 8, 2022.

[6] Gunanandhini, S., Kalamani, M., Bhagavathipriya, M. Wavelet based Video Compression techniques for Industrial monitoring applications. *Journal of Physics: Conference Series*, 2272(1):1-8, 012019, IOP Publishing, 2022.

[7] Nithin, S., Suresh, L., Krishnaveni, S., Muthukumar, P. Developing novel video coding model using modified dual-tree wavelet-based multi-resolution technique. Multimedia Systems, 28(2):643-657, 2022.

[8] Bagherimehrab, M., Aspuru-Guzik, A. Efficient quantum algorithm for all quantum wavelet transforms. Quantum Science and Technology, 9(3):1-16, 035010, 2024.

[9] Mu, X., Wang, H., Bao, R., Wang, S., Ma, H. An improved quantum watermarking using quantum Haar wavelet transform and Qsobel edge detection. Quantum Information Processing, 22(5):1-16, 223, 2023.

[10] Zeng, H., Xu, J., He, S., Deng, Z., Shi, C. Rate Control Technology for Next Generation Video Coding Overview and Future Perspective. Electronics, 11(23):1-25, 4052, 2022.

[11] Choi, K. A study on fast and low-complexity algorithms for versatile video coding. Sensors, 22(22):8990, 2022.

[12] Chen, L., Cheng, B., Zhu, H., Qin, H., Deng, L., Luo, L. Fast Versatile Video Coding (VVC) Intra Coding for Power-Constrained Applications. Electronics, 13(11):2150, 2024.

[13] Joy, H., Kounte, M. Deep CNN Based Interpolation Filter for High Efficiency Video Coding. In Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things, pp. 519-524, 2024.

[14] Lee, M., Park, S., Oh, S., Kim, Y., Jeong, S., Lee, J., Sim, D. Transform-Based Feature Map Compression Method for Video Coding for Machines (VCM). Electronics, 12(19):4042, 2023.

[15] Hu, Y., Jung, C., Qin, Q., Han, J., Liu, Y., Li, M. HDVC: Deep Video Compression with Hyperprior-Based Entropy Coding, IEEE Access, 2024.

[16] Chen, S., Aramvith, S., Miyanaga, Y. Learning-Based Rate Control for High Efficiency Video Coding. Sensors, 23(7):3607, 2023.

[17] Zhao, Y., He, W., Jia, C., Wang, Q., Li, J., Li, Y., Lin, C., Zhang, K., Zhang, L., Ma, S. A Neural-network Enhanced Video Coding Framework beyond ECM. arXiv preprint arXiv:2402.08397, 2024.

[18] Zhang, Y., Gong, X., Yu, H., Wu, Z., Yu, L. Distance-based feature repack algorithm for video coding for machines. Journal of Visual Communication and Image Representation, 16:104150, 2024.

[19] Thanh, H., Quang, S., Huu, T., Hoang, X. Learning adaptive motion search for fast versatile video coding in visual surveillance systems. IET Image Processing, 8(4):981-95, 2024.

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024

[20] Bian, Y., Sheng, X., Li, L., Liu, D. LSSVC: A Learned Spatially Scalable Video Coding Scheme. IEEE Transactions on Image Processing, 33(3): 3314 – 3327, 2024.
[21] Zouidi, N., Kessentini, A., Hamidouche, W., Masmoudi, N., Menard, D. Multitask learning based intra-mode decision framework for versatile video coding. Electronics, 11(23):4001, 2022.
[22] Yang, C., Qin, S., An, P., Huang, X., Shen, L. Content adaptive spatial-temporal rescaling for video coding optimization. Expert Systems with Applications, 124482, 2024.
[23] Samarathunga, B., Ganearachchi, Y., Fernando, T., Alahapperuma, I., Fernando, A. Semantic Communication Based Video Coding Using Temporal Prediction of Deep Neural Network Parameters. In Proceedings of the International Conference on Gaming, Entertainment and Media (GEM) Conference, IEEE, ITA, 2024.
[24] Ibraheem, M., Dvorkovich, A. Enhancing Versatile Video Coding Efficiency via Post-Processing of Decoded Frames Using Residual Network Integration in Deep Convolutional Neural Networks. In Proceedings of the International Conference on Digital Signal Processing and its Applications, pp. 1-9, IEEE, 2024.
[25] Qin, S., Yang, C., An, P. Content adaptive downsampling for low bitrate video coding. Multimedia Tools and Applications, 83(9):26547-63, 2024.
[26] Reich, C., Debnath, B., Patel, D., Prangemeier, T., Cremers, D., Chakradhar, S. Deep Video Codec Control for Vision Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5732-5741, 2024.
[27] Jiang, X., Xiang, M., Jin, J., Song, T. Extreme Learning Machine-Enabled Coding Unit Partitioning Algorithm for Versatile Video Coding. Information, 14(9):494, 2023.
[28] Sheng, X., Li, L., Liu, D., Li, H. VNVC: A Versatile Neural Video Coding Framework for Efficient Human-Machine Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(7): 4579 – 4596, 2024.
[29] Lee, D., Kwon, S. Intra Prediction Method for Depth Video Coding by Block Clustering through Deep Learning. Sensors, 22(24):9656, 2022.
[30] Sharrab, Y., Alsmirat, M., Eljinini, M., Sarhan, N. iHELP: a model for instant learning of video coding in VR/AR real-time applications. Multimedia Tools and Applications, 1-40, 2024.
[31] Bender, I., Rehbein, G., Correa, G., Agostini, L., Porto, M. Adaptive complexity control for AV1 video encoder using machine learning. Journal of Real-Time Image Processing, 21(3):96, 2024.
[32] Delibaşoğlu, İ. Moving object detection method with motion regions tracking in background subtraction. Signal, Image and Video Processing, 7(5):2415-23, 2023.
[33] Ammous, D., Kessentini, A., Khlif, N., Kammoun, F., Masmoudi, N. An Enhancement of Lossless Video Compression Using Two-Layer Approach. In Applications of Encryption and Watermarking for Information Security, pp. 105-135, IGI Global, 2023.
[34] Zhu, L., Zhang, Y., Li, N., Wu, W., Wang, S., Kwong, S. Neural Network Based Multi-Level In-Loop Filtering for Versatile Video Coding. IEEE Transactions on Circuits and Systems for Video Technology. 2024 Jun 28.
[35] Nithin, S., Suresh, L., Krishnaveni, S., Muthukumar, P. Developing novel video coding model using modified dual-tree wavelet-based multi-resolution technique. Multimedia Systems, 28(2):643-657, 2022.

DOI: https://doi.org/10.54216/FPA.170219
Received: February 06, 2024 Revised: May 07, 2024 Accepted: October 03, 2024