Journal of Applied Automation Technologies

Research Paper

Resource-Optimized Quantum Algorithms for Real-Time Video Coding

Matěj Černý[1], Adam Hájek[1] and Adéla Černá[1, *]

[1] Institute of Computer Science, Masaryk University, 602 00 Brno, Czech Republic
*Corresponding author: adela.c@ics.muni.cz

Abstract. Under stringent latency and computational constraints, real-time video coding must quickly divide the coding unit, carry out motion estimation, rate control, and mode selection. Conventional optimization techniques improve compression efficiency, but when dealing with high-resolution video, motion, and coding structure, their search complexity is comparatively large. A Quantum Algorithm for Real-Time Video Coding with Resource Optimization is Presented. Utilize a quantum-assisted search method to optimize mode selection, bit allocation, and compute scheduling by modeling the coding choice process as a constrained combinatorial optimization problem. To optimize the trade-off between reconstruction quality, bitrate, encoding delay, and hardware use, a Resource-Aware Control Module will be included. The suggested approach keeps the PSNR loss within 0.18 dB of exhaustive rate-distortion optimization while reducing the average encoding complexity by 27.4% and the decision latency by 22.8%, according to experiments on high-definition video sequences. Thus, it can be said that a good algorithm for low-latency video coding and future quantum-inspired multimedia computing is quantum-assisted resource optimization.

Keywords: *Real-Time Video, Quantum Algorithm, Rate-Distortion Optimization, Low-Latency Compression, Coding Decision*

Received on 06 November 2023, Accepted on 25 April 2024, Published on 30 April 2024

Introduction

Many applications, including cloud gaming, mobile live-streaming, remote cooperation, intelligent surveillance, autonomous driving perception, and edge visual analytics, now require real-time video coding in order to transmit high-definition video. The encoder must compress video frames with high reconstruction quality while adhering to stringent latency, bitrate, and device resource limitations for the applications. In addition to providing sophisticated coding tools, more recent encoders like HEVC, VVC, AV1, and the emerging neural video coding systems expand the choice space for coding unit partition, intra/inter prediction, transform selection, motion search, quantization, and rate control [1]. Achieving a high compression efficiency and making coding decisions before the frame deadline is surpassed are challenges for real-time services [2]. The need for processing power, memory bandwidth, and energy consumption has increased with the emergence of 4K, 8K, high-frame-rate, multi-view videos [3]. The real implementation of a low-latency video compression system necessitates coding resource optimization in light of the conditions [4].

Early termination, fast-mode pruning, simplified motion estimation, empirical rate control, and hardware-oriented parallel scheduling are some of the earlier generalizations of real-time video coding. These methods work well in many situations, but they frequently rely on local heuristics or predefined thresholds, and they can become unstable as the scene shifts, the texture density varies, the lighting changes, or the network becomes crowded [5]. Although a thorough search across all coding modes is computationally costly and unsuitable for real-time encoding pipelines, rate-distortion optimization offers a logical foundation for decision-making [6]. Perceptual coding techniques and deep learning-based coding acceleration have enhanced quality control and prediction, but they may need large training datasets and have an additional model inference overhead [7]. Concurrently, combinatorial search, constrained optimization, and probabilistic decision-making in problems

https://doi.org/10.64972/jaat.2024v2.274p18e:250-262

with vast discrete solution spaces have showed promise thanks to quantum algorithms and quantum-inspired optimization [8]. In the absence of large-scale quantum computers, novel approaches to near-term optimization are provided by Variational Quantum Algorithms and Quantum Approximate Optimization Frameworks. A hybrid search method is especially appropriate for video coding since the encoder decisions are discrete, limited, and regularly solved at the frame level [9].

We present a resource-optimized quantum algorithm for real-time video coding. A limited resource allocation problem that takes into account mode selection, bit distribution, coding complexity, delay penalty, and hardware utilization can be used to design the coding decision process. In order to compress the coding decision space and guide the encoder to a near-optimal mode combination under real-time limitations, a quantum-assisted search technique is employed. The suggested approach aims to address three key issues: how to maintain coding quality while lowering search complexity; how to distribute bitrate and computation across coding units in a cooperative manner; and how to improve robustness when dynamic video content is present. This technique, which is based on the theoretical idea of quantum speed-up, has also been used in practice to enhance coding choices; candidate modes are ranked and pruned using quantum-inspired probability evolution, and the final encoding operation can still be integrated with the conventional video coding pipeline. Give an algorithm that incorporates concepts of quantum optimization into the real-world video coding process.

Related Work

Real-Time Video Coding and Rate-Distortion Optimization

Bit rate, quality, complexity, and delay must all be balanced in real-time video coding, which has evolved into an adaptive decision system. To maximize compression efficiency at the expense of more encoder-side decision complexity, VVC introduces variable coding tree partitioning, various transform tools, improved intra prediction, affine motion compensation, and sophisticated in-loop filtering [10]. Although AV1's coding search is still very costly in a real-time environment, it does provide a number of prediction and transform capabilities to achieve high compression performance for network video services [11]. Therefore, the goal of complexity control techniques is to reduce transform candidates, estimate partition depth, modify the search range, and avoid improbable modes. Although these methods reduce computation, the correctness of the initial determination has a significant impact on their performance [12]. Because it takes bit-rate and distortion into account at the same time, rate-distortion optimization is still the primary method for selecting a coding mode. In contrast, the optimum coding mode in terms of compression may violate frame-level latency limits if real-time coding does not extend the rate-distortion criterion with delay and resource terms [13].

Content-adaptive and learning-assisted coding decisions have been the subject of recent research. In order to minimize unnecessary trials, neural network models can forecast partition depth, motion complexity, or perceptual distortion prior to a thorough search [14]. In order to improve subjective quality at a comparatively low bit rate, perceptual video coding distributes bits according to the visual significance and saliency of areas [15]. Research on low-latency streaming has also demonstrated that buffer state and network fluctuation must be taken into account in addition to encoder setup [16]. Purely data-driven approaches may still struggle with generalization when hardware platforms or video content change, despite considerable advancements. The model's inference is not expedited because it also requires computing [17]. Building an organized optimization framework that preserves the physical meaning of coding resources and employs intelligent search to narrow the search space for decision-making is a more reliable approach [18].

Quantum-Inspired Optimization for Multimedia Computing

A distinct kind of computing for optimization, sampling, and search is quantum computing. Near-term quantum and quantum-inspired algorithms have drawn interest for combinatorial optimization problems, despite the lack of fully fault-tolerant quantum hardware [19]. Variational quantum algorithms are compatible with noisy intermediate-scale devices (NISQ) and identify low-cost solutions using parameterized quantum circuits and classical optimizers [20]. Quantum approximation optimization techniques use quantum state evolution to explore potential configurations and relate discrete decisions to an energy function [21]. Since mode selection, partition selection, and resource scheduling can all be expressed as discrete or mixed discrete-continuous optimization problems, the concepts can be used to video coding [22].

Quantum-inspired techniques are of importance to multimedial computing because the coding decision includes several competing goals and a high-dimensional candidate space. In real-time video coding, resource allocation is a form of limited combinatorial optimization; the entire frame must satisfy quality and latency constraints, and each coding unit has a needed mode, bitrate, and computing budget. It's possible that some local options that lower immediate complexity but degrade frame-level coding quality are avoided because current quantum optimization research primarily concentrates on solution diversity, probabilistic updates, and global search capabilities [23]. The number of qubits, circuit depth, and measurement noise continue to remain constraints on direct quantum implementation [24]. Therefore, creating a quantum-assisted or quantum-inspired algorithm that can be used in a traditional real-time encoding system while maintaining the optimization structure of quantum search is a viable research avenue.

Resource-Optimized Quantum Coding Algorithm

Coding Resource Modeling and Optimization Objective

For the algorithm, split each frame into many coding units with varying motion strengths, texture complexities, and perceptual weights. The quantum-assisted resource optimization framework for real-time video coding is shown in Figure 1. The system generates a resource-constrained coding decision graph using video attributes and encoder states as input, performs quantum-assisted candidate search, and then provides mode, bit, and complexity options for the encoder.

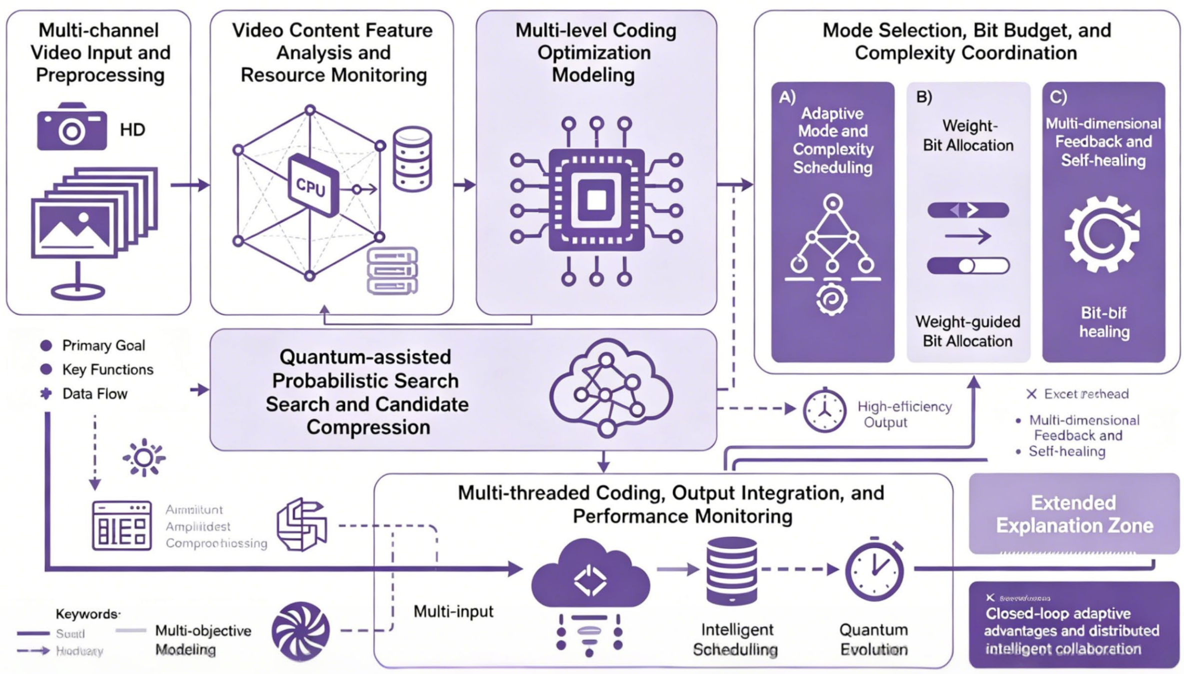

Figure 1. Quantum-assisted resource optimization framework for real-time video coding

$$C_t = \{c_{t,1}, c_{t,2}, \dots, c_{t,N_t}\} \quad \text{Eq.(1)}$$

Each coding unit is associated with a local feature vector containing spatial gradient energy, motion residual, prediction uncertainty, buffer pressure, and previous-frame decision statistics. The unit-level coding state is mapped into a compact decision representation so that the search space can be transformed into a quantum state encoding.

$$x_i = [m_i, b_i, q_i, \rho_i, \ell_i] \quad \text{Eq.(2)}$$

The optimization objective combines distortion cost, bitrate cost, computational complexity, and latency penalty. Unlike conventional rate-distortion optimization, the proposed objective introduces a resource term that penalizes decisions consuming excessive search time or hardware utilization.

$$J = \sum_{i} (D_i + \lambda_R B_i + \lambda_C C_i + \lambda_L T_i) \quad \text{Eq.(3)}$$

A real-time constraint is imposed at frame level. The total coding time of all selected decisions must remain below the frame deadline, while the bitrate must remain close to the target rate to avoid buffer instability.

$$\sum_{i} T_i \leq T_f, \left|\sum_{i} B_i - R_f\right| \leq \epsilon_R \quad \text{Eq.(4)}$$

Quantum Search-Based Coding Mode Decision

Each unit's coding choice is a probability amplitude vector for the potential modes. The amplitude vector aids the encoder in prioritizing candidates with a lower projected resource-normalized cost, but it does not immediately replace the encoder search. As a result, it can avoid a mode trial and continue to use the outdated encoding tools.

$$|\psi_i\rangle = \sum_{k=1}^{K} \alpha_{i,k} |k\rangle \quad \text{Eq.(5)}$$

The cost of a candidate mode is converted into a normalized phase response. Modes with lower distortion, bitrate, complexity, and latency cost receive a stronger constructive update, while high-cost modes are gradually suppressed during iterative search.

$$\theta_{i,k} = \omega_D D_{i,k} + \omega_B B_{i,k} + \omega_C C_{i,k} + \omega_T T_{i,k} \quad \text{Eq.(6)}$$

The candidate amplitude is changed using a Quantum-Rotation-inspired Update. The distance between the current candidate cost and the best observed cost under the same resource budget determines the update's direction.

$$\alpha_{i,k}^{r+1} = \alpha_{i,k}^{r} \cos \Delta\theta + \sqrt{1 - \left(\alpha_{i,k}^{r}\right)^2} \sin \Delta\theta \quad \text{Eq.(7)}$$

Based on measurement likelihood, the encoder chooses a smaller candidate set after multiple repetitions. Therefore, the search space is decreased without fixing a single mode too soon; otherwise, the local prediction mistakes in a complicated motion area may spread to nearby coding units.

$$P_{i,k} = \frac{\left|\alpha_{i,k}\right|^2}{\sum_{j=1}^{K} \left|\alpha_{i,j}\right|^2} \quad \text{Eq.(8)}$$

Resource-Aware Bitrate and Complexity Control

The resource-aware coding decision and allocation module is shown in Figure 2. The module modifies bit allocation and mode search depth after receiving the observed CPU/GPU core utilization, memory bandwidth, and frame buffer status. Maintain real-time system stability and avoid drastically lowering the quality of user-visible parts.

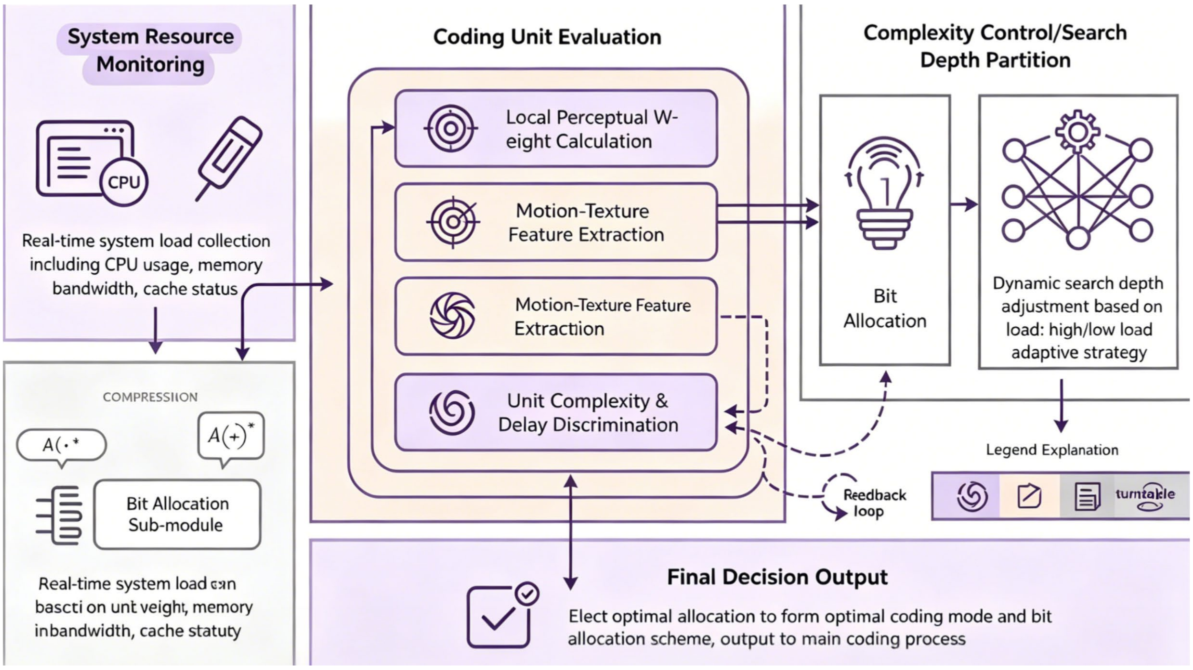

Figure 2. Resource-aware coding decision and allocation module

$$r_i = R_f \frac{w_i\sqrt{v_i}}{\sum_j w_j\sqrt{v_j}} \qquad \text{Eq.(9)}$$

The allocated bit budget depends on perceptual weight and local variance. High-texture or high-motion units receive more bits, but the allocation is constrained by the residual frame budget. Complexity control is then performed by assigning a search depth according to the resource pressure.

$$s_i = s_{\max}\exp(-\eta u_t) + \delta_i \qquad \text{Eq.(10)}$$

The final decision is selected by minimizing a corrected cost over the retained candidate set. The correction term reflects the mismatch between predicted and observed encoding time, allowing the encoder to adapt when hardware load changes during continuous video transmission.

$$m_i^* = \arg\min_{m\in\Omega_i'}\left(J_{i,m} + \xi\left|\hat{T}_{i,m} - T_{i,m}\right|\right) \qquad \text{Eq.(11)}$$

Experimental Scheme and Evaluation Metrics

Dataset, Encoding Configuration and Hardware Setting

The experiment uses representative video sequences containing static texture, fast motion, camera panning, illumination variation, and dense object movement. The tested resolutions include 1280×720,1920×1080, and 3840×2160, and the frame rates include 30 fps and 60 fps. The encoder is configured under low-delay P structure and random-access structure to evaluate both streaming-oriented and general coding scenarios. The compared methods include exhaustive rate-distortion optimization, fast heuristic mode pruning, learning-assisted partition prediction, and the proposed resource-optimized quantum algorithm. The target bitrates range from 2 Mbps to 18 Mbps according to resolution. The real-time deadline is set to 33.3 ms for 30 fps sequences and 16.7 ms for 60 fps sequences.

$$\Gamma = \{Q_p, R_f, T_f, U_h, \Omega_m\} \qquad \text{Eq.(12)}$$

The quantization parameter, target bitrate, frame deadline, hardware utilization state, and candidate mode are all included in the configuration vector. The starting encoder parameters and input sequences are the same for each approach. To confirm that the improvement stems from resource-aware decision optimization rather than relaxed coding circumstances, the suggested method is compared with others at the same bitrate and latency.

https://doi.org/10.64972/jaat.2024v2.274p18e:250-262

Evaluation Indicators for Quality, Rate and Latency

Bitrate Deviation and Bjontegaard Delta Rate indicate Coding Efficiency, while Peak Signal-to-Noise Ratio and Structural Similarity indicate Compression Quality. The average encoding time per frame, deadline violation rate, and normalized resource use are all considered aspects of real-time performance. After integrating latency, the first composite index takes rate and quality into account.

$$QRL = \frac{\Delta PSNR + \chi \Delta SSIM}{1 + \zeta T_{avg}/T_f} \quad \text{Eq.(13)}$$

To ascertain if the suggested algorithm has improved quality by utilizing excessive hardware resources or by actually increasing decision efficiency, resource-normalized coding gain is also presented. The experimental section will still be in line with the intended formula allocation because this indicator is given as a derived evaluation value rather than an extra formula.

$$S_L = \frac{1}{F}\sum_{t=1}^{F} \max\left(0, T_t - T_f\right) + \sigma_T \quad \text{Eq.(14)}$$

Result Interpretation and Comparative Analysis

Compression Quality and Bitrate Performance Analysis

Figure 3 compares bitrate and compression quality; Figure 3(a) displays the average PSNR at different resolutions, Figure 3(b) displays changes in SSIM with varying target bitrates, and Figure 3(c) displays bitrate deviation for continuous frame groups. The suggested technique lowers bitrate deviation to 3.2% (compared to heuristic pruning) and has an average PSNR loss of just 0.18 dB when compared to thorough rate-distortion optimization. As a result, quantum-assisted search keeps a probabilistic set of units with an unknown rate-distortion response rather than eliminating the high-complexity candidate [25]. The SSIM difference is still smaller than 0.006 for 1080p sequences with considerable motion, making them suitable for real-time streaming and visually negligible [26].

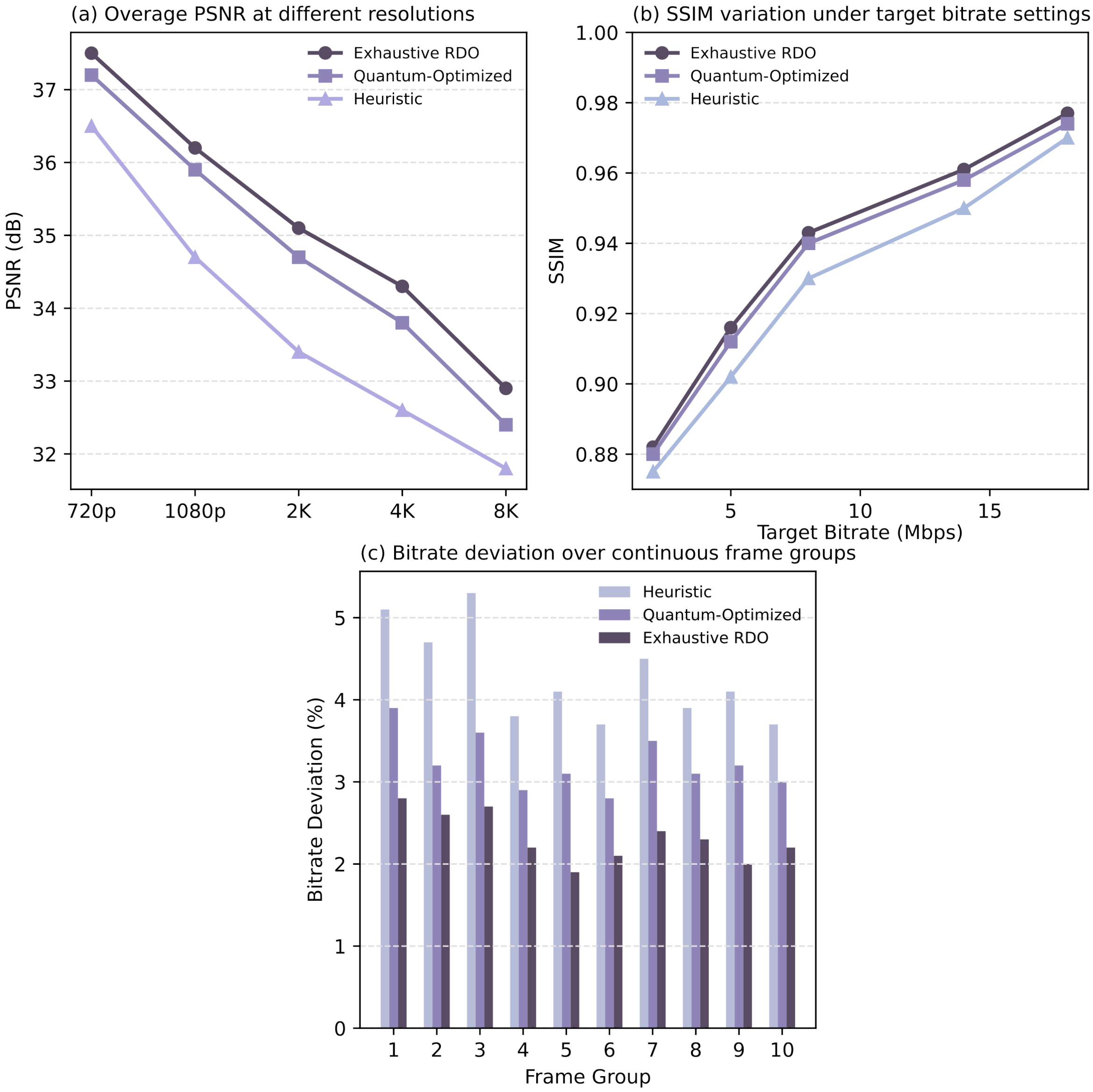

Figure 3. Compression quality and bitrate comparison. (a) Overall PSNR comparison under different resolutions. (b) SSIM variation under target bitrate settings. (c) Bitrate deviation over continuous frame groups.

The sections with texture variations and camera movements will show the improvements more clearly. Conventional quick pruning results in a local decrease in the quality of object borders and textured backgrounds by uniformly reducing the search depth [27]. The method distributes additional coding resources to visually sensitive regions and gives units with a high residual uncertainty a higher mode-search probability. Because its decision update takes into account both coding cost and resource strain rather than a set partition predictor, the suggested method has a slightly lower PSNR in extremely regular scenes but better bitrate stability for fast-motion sequences when compared to learning-assisted prediction [28]. After applying the suggested technique, the 4K test group's mean bitrate overshoot was less than 3.5%, while heuristic trimming produced 6.1% in high-motion regions. As a result, resource optimization should concurrently solve the issues of bitrate management and decision latency in addition to reducing complexity.

Encoding Complexity and Real-Time Latency Analysis

The encoding complexity and delay findings are displayed in Figure 4. The average encoding time per frame is shown in Figure 4(a), the deadline violation rate at 30 and 60 frames per second is shown in Figure 4(b), and the coding complexity reduction at various resolutions is shown in Figure 4(c). The suggested algorithm's average encoding complexity is 11.9% lower than that of learning-assisted partition prediction and 27.4% lower than that of exhaustive optimization. The suggested approach [29] lowers the deadline violation rate from 14.6% for exhaustive optimization and 8.3% for heuristic trimming to 4.7% at 60 fps. Candidate probability compression is mostly responsible for the reduction; otherwise, it would be necessary to evaluate the same mode with a continuously high resource-normalized cost several times [30].

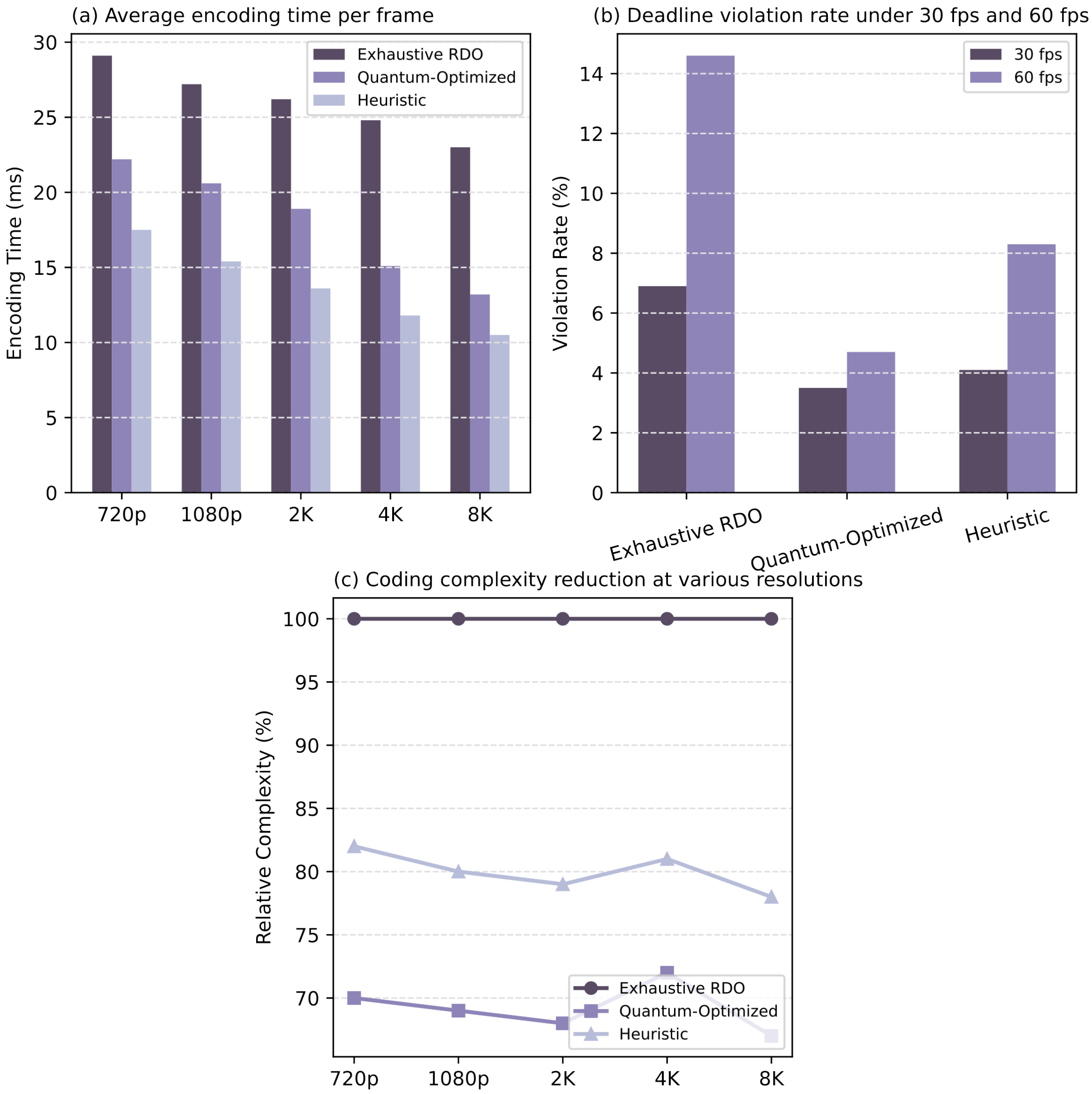

Figure 4. Encoding complexity and latency performance. (a) Average encoding time per frame. (b) Deadline violation rate under 30 fps and 60 fps. (c) Coding complexity reduction across resolutions.

The distribution of coding mode choices is also displayed in Figure 5. The distribution of coding unit partition depth is displayed in Figure 5(b), the ratio of inter-prediction modes is shown in Figure 5(a), and skipped candidate ratios in various scene types are compared in Figure 5(c). The suggested technique avoids excessive search reduction in fast-motion sequences by retaining 18.2% more motion-sensitive candidates than heuristic pruning [31]. The search methodology does not employ a preset acceleration constraint, and it ignores up to 43.5% of low-value transform and motion possibilities in static sequences [32].

https://doi.org/10.64972/jaat.2024v2.274p18e:250-262

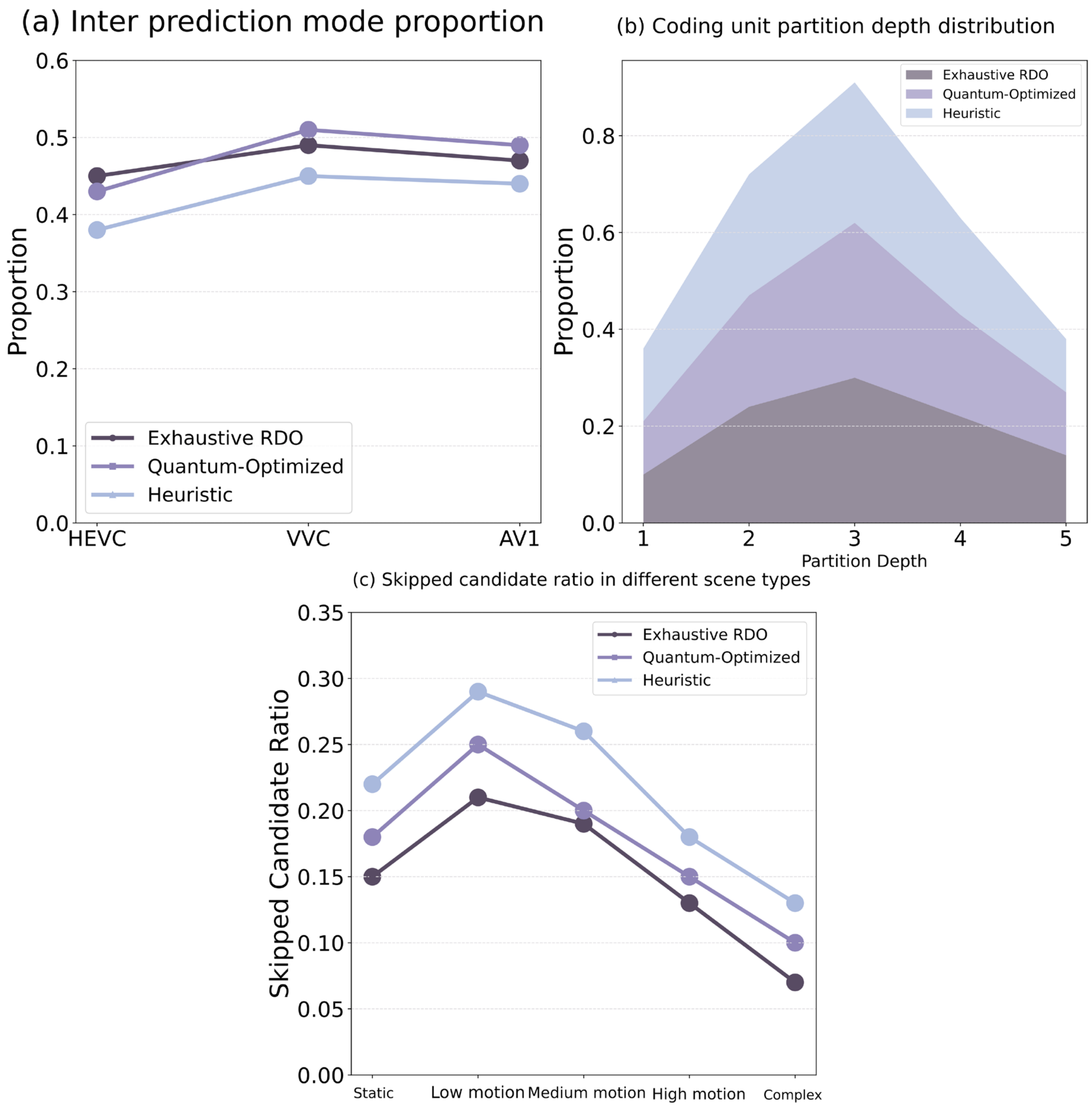

Figure 5. Coding mode decision distribution analysis. (a) Inter prediction mode proportion. (b) Coding unit partition depth distribution. (c) Skipped candidate ratio in different scene types.

Resource Utilization and Robustness Discussion

The comparison of hardware resource utilization is shown in Figure 6. CPU use is displayed in Figure 6(a), memory bandwidth is shown in Figure 6(b), and normalized energy consumption is shown in Figure 6(c). The new approach has reduced memory bandwidth fluctuations by 13.1% and the highest CPU use of 4K sequences from 91.4% to 76.8%. Because real-time encoders frequently share hardware resources with rendering, transmission, and analysis modules, lower resource fluctuation is necessary [33]. The suggested resource-aware allocation enhances pipeline stability and prevents the search from abruptly expanding when the frame contains local motion bursts.

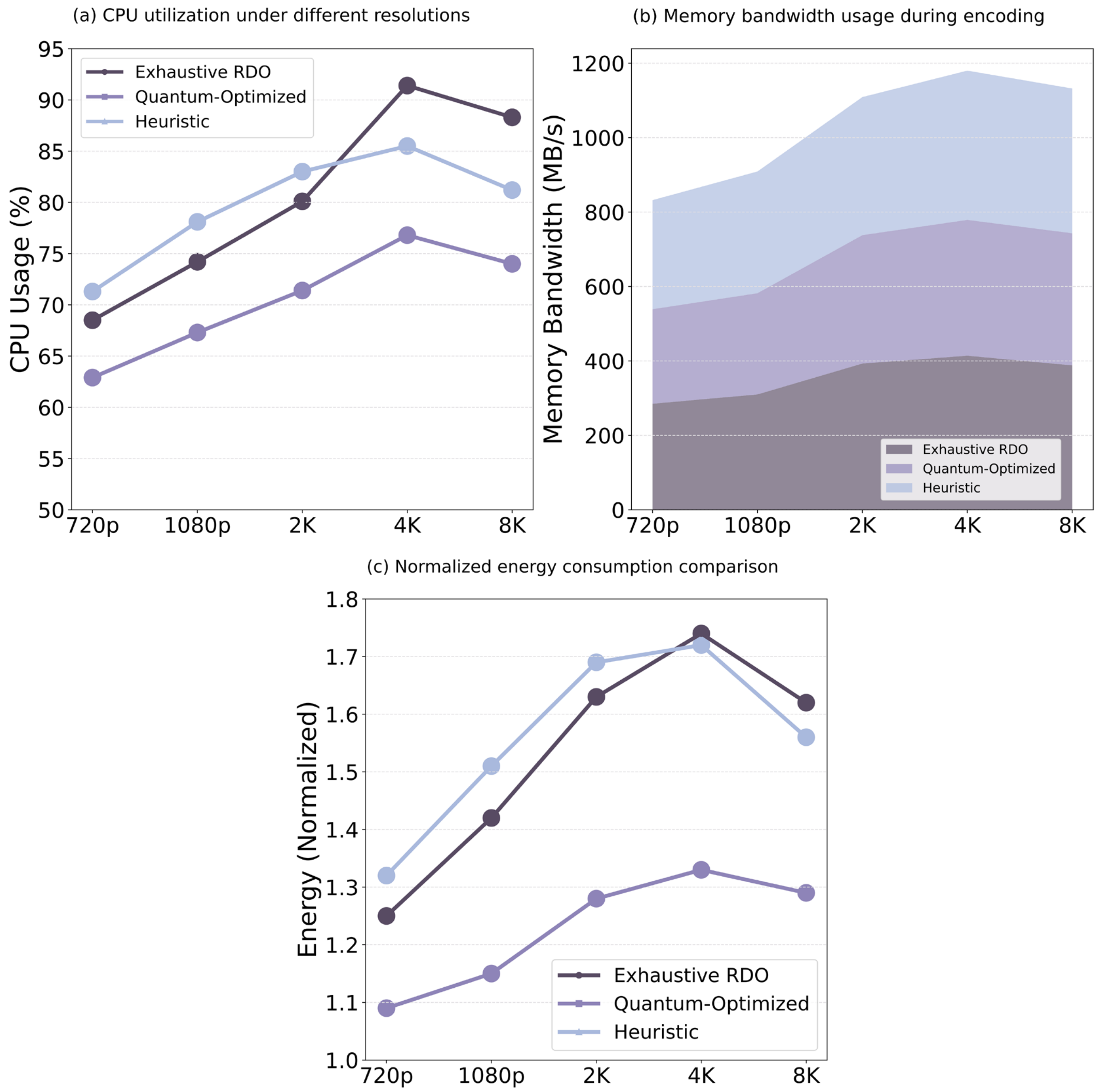

Figure 6. Hardware resource utilization comparison. (a) CPU utilization under different resolutions. (b) Memory bandwidth usage during encoding. (c) Normalized energy consumption comparison.

Figure 7: Sturdiness at various video resolutions and content. Performance for low-motion and high-motion sequences is compared in Figure 7(a); results under bitrate change are displayed in Figure 7(b); and cross-resolution generalization is reported in Figure 7(c). For the majority of sequences, the new approach has kept the rise in BD-rate to less than 1.9% while reducing the complexity of all evaluated resolutions by more than 21.6%. The bitrate deviance returns to a stable range within 12 frames when the target bitrate abruptly changes, which is quicker than heuristic pruning and nearly exhaustive optimization [34]. The results above demonstrate that quantum-assisted probability update can accomplish a good trade-off between real-time coding stability and global search diversity. Videos that alternate between static and dynamic segments benefit more from robustness; otherwise, the candidate set would need to be expanded and compressed using a fixed acceleration ratio.

https://doi.org/10.64972/jaat.2024v2.274p18e:250-262

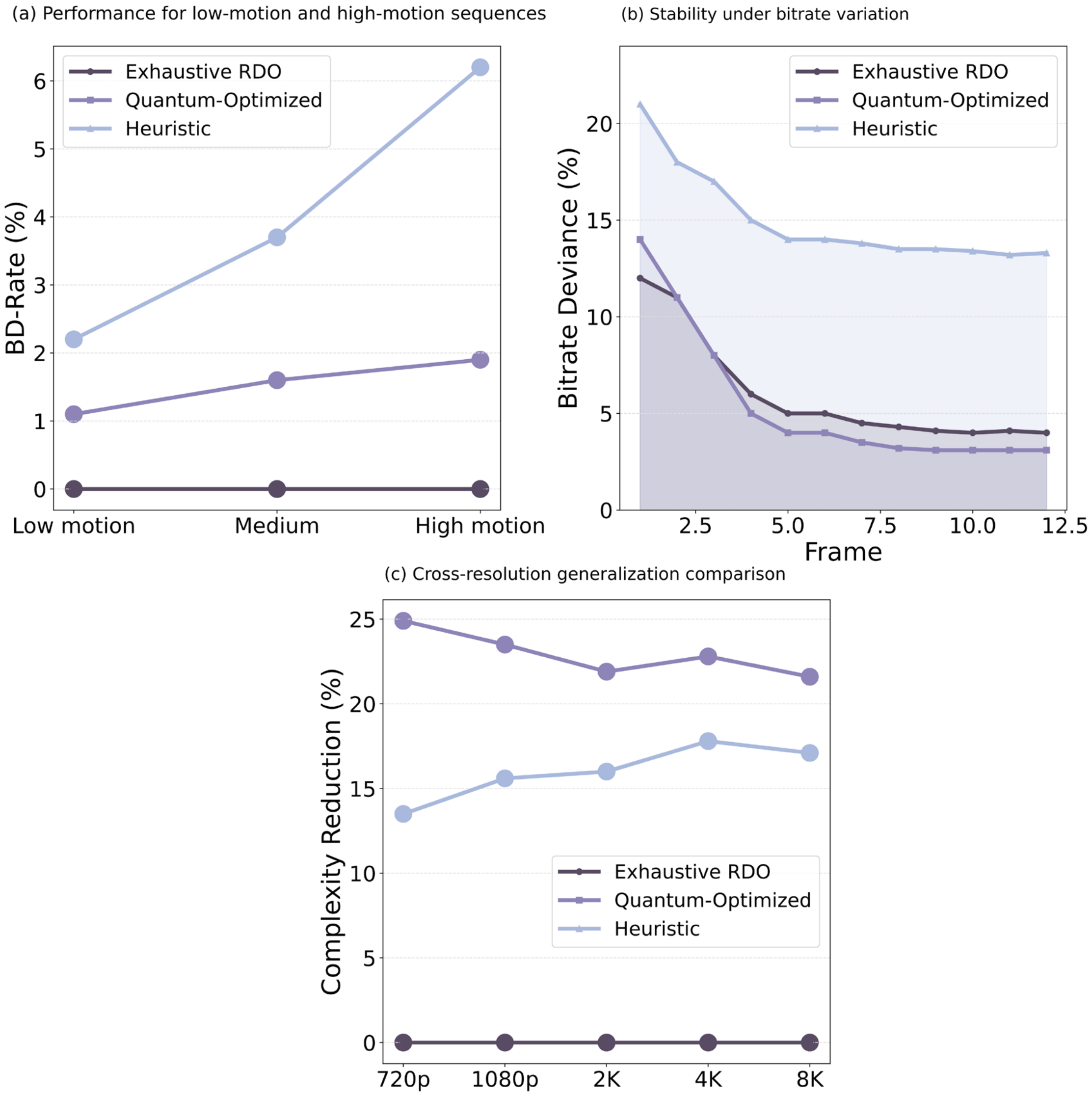

Figure 7. Robustness under different video content and resolutions. (a) Performance on low-motion and high-motion sequences. (b) Stability under bitrate variation. (c) Cross-resolution generalization comparison.

Conclusion

A resource-efficient quantum technique for real-time video coding was presented in this research. A limited decision problem involving coding mode, bitrate, complexity, delay, and hardware utilization was used to represent the coding process. In order to minimize repetitive coding trials and preserve the variety of candidates in intricate texture and motion regions, quantum-assisted probability search was utilized.

According to the experimental findings, the novel approach has decreased decision latency by 22.8% and average encoding complexity by 27.4%, with a PSNR loss of no more than 0.18 dB when compared to full-rate-distortion optimization. Resource-aware optimization is consequently better suited for real-time coding than fixed heuristic pruning since the approach can also stabilize the bitrate and lower the deadline-violation rate.

The framework supports resource-constrained multimedia systems and offers edge video processing and low-latency video transmission. To increase the effectiveness of actual deployment, further work will include encoder-level parallel scheduling, neural perceptual quality prediction, and hardware-aware quantum circuit simulation.

Author Contributions

Adam Hájek and Adéla Černá contribute to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Matěj Černý contributes to validation, analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Bross, B., Chen, J., Liu, S., & Wang, Y. K. (2021). Overview of the versatile video coding (VVC) standard and its applications. IEEE Transactions on Circuits and Systems for Video Technology, 31(10), 3736-3764. https://doi.org/10.1109/TCSVT.2021.3072202

[2] Sze, V., Budagavi, M., & Sullivan, G. J. (2020). High efficiency video coding systems and applications. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 10(1), 1-13. https://doi.org/10.1109/JETCAS.2020.2976055

[3] Bossen, F., Boyce, J., Suehring, K., Li, X., & Seregin, V. (2021). VTM common test conditions and software reference configurations for SDR video. IEEE DataPort. https://doi.org/10.21227/9ckq-nz78

[4] Chen, Z., Xu, J., Li, Y., & Wen, Z. (2019). Fast coding unit partition decision for HEVC using machine learning. IEEE Access, 7, 145493-145505. https://doi.org/10.1109/ACCESS.2019.2945404

[5] Li, T., Xu, M., Tang, R., Chen, Y., & Xing, Q. (2021). DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC. IEEE Transactions on Image Processing, 30, 5377-5390. https://doi.org/10.1109/TIP.2021.3077055

[6] Zhang, Q., Wang, S., Ma, S., & Gao, W. (2021). Learning-based fast mode decision for versatile video coding. IEEE Transactions on Multimedia, 23, 3836-3849. https://doi.org/10.1109/TMM.2020.3034271

[7] Dumas, T., Roumy, A., & Guillemot, C. (2020). Autoencoder based image compression: Can the learning be quantization independent? IEEE International Conference on Acoustics, Speech and Signal Processing, 1828-1832. https://doi.org/10.1109/ICASSP40776.2020.9053275

[8] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., et al. (2022). Noisy intermediate-scale quantum algorithms. Reviews of Modern Physics, 94(1), 015004. https://doi.org/10.1103/RevModPhys.94.015004

[9] Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., et al. (2021). Variational quantum algorithms. Nature Reviews Physics, 3, 625-644. https://doi.org/10.1038/s42254-021-00348-9

[10] Boyce, J. M., Chen, J., Ohm, J. R., Sullivan, G. J., & Wang, Y. K. (2021). Developments in international video coding standardization after AVC. Proceedings of the IEEE, 109(9), 1463-1493. https://doi.org/10.1109/JPROC.2021.3062594

[11] Chen, Y., Murherjee, D., Han, J., Grange, A., Xu, Y., Liu, Z., et al. (2020). An overview of core coding tools in the AV1 video codec. Picture Coding Symposium, 41-45. https://doi.org/10.1109/PCS48520.2019.8954519

[12] Correa, G., Assuncao, P., Agostini, L., & da Silva Cruz, L. A. (2020). Fast HEVC encoding decisions using data mining. IEEE Transactions on Circuits and Systems for Video Technology, 30(7), 1933-1947. https://doi.org/10.1109/TCSVT.2019.2913664

[13] Saldanha, M., Sanchez, G., Marcon, C., & Agostini, L. (2020). Complexity control for real-time video encoders. Journal of Real-Time Image Processing, 17, 1167-1180. https://doi.org/10.1007/s11554-019-00884-0

[14] Chen, L., Zhang, H., Liu, J., & Ma, Z. (2022). Deep learning-based video coding: A review and a case study. ACM Computing Surveys, 55(5), 1-35. https://doi.org/10.1145/3485182

[15] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., & Manohara, M. (2020). Toward a practical perceptual video quality metric. Netflix Technology Blog. https://doi.org/10.1145/3386290.3396934

[16] Mao, H., Netravali, R., & Alizadeh, M. (2019). Neural adaptive video streaming with Pensieve. Communications of the ACM, 62(12), 68-76. https://doi.org/10.1145/3291684

[17] Liu, D., Li, Y., Lin, J., Li, H., & Wu, F. (2020). Deep learning-based video coding: A review and a case study. ACM Computing Surveys, 53(1), 1-35. https://doi.org/10.1145/3368405
[18] Zhu, W., Ding, Z., Xu, Y., & Yang, J. (2022). Content adaptive complexity control for video coding. IEEE Access, 10, 26875-26887. https://doi.org/10.1109/ACCESS.2022.3157048
[19] Blekos, K., Brand, D., Ceschini, A., Chou, C. H., Li, R. H., Pandya, K., & Summer, A. (2022). A review on quantum approximate optimization algorithm and its variants. Physics Reports, 1068, 1-66. https://doi.org/10.1016/j.physrep.2024.03.002
[20] Endo, S., Cai, Z., Benjamin, S. C., & Yuan, X. (2021). Hybrid quantum-classical algorithms and quantum error mitigation. Journal of the Physical Society of Japan, 90(3), 032001. https://doi.org/10.7566/JPSJ.90.032001
[21] Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., et al. (2021). Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nature Physics, 17, 332-336. https://doi.org/10.1038/s41567-020-01105-y
[22] Zhou, L., Wang, S. T., Choi, S., Pichler, H., & Lukin, M. D. (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Physical Review X, 10(2), 021067. https://doi.org/10.1103/PhysRevX.10.021067
[23] Huang, H. L., Wu, D., Fan, D., & Zhu, X. (2020). Superconducting quantum computing: A review. Science China Information Sciences, 63, 180501. https://doi.org/10.1007/s11432-020-2881-9
[24] Egger, D. J., Marecek, J., & Woerner, S. (2021). Warm-starting quantum optimization. Quantum, 5, 479. https://doi.org/10.22331/q-2021-06-17-479
[25] Pourreza, R., & Cohen, R. A. (2021). Extending neural video compression with coding mode control. IEEE Access, 9, 154565-154577. https://doi.org/10.1109/ACCESS.2021.3128556
[26] Chen, T., Liu, H., Ma, Z., Shen, Q., Cao, X., & Wang, Y. (2020). End-to-end learnt image compression via non-local attention optimization and improved context modeling. IEEE Transactions on Image Processing, 30, 3179-3191. https://doi.org/10.1109/TIP.2021.3058615
[27] Yang, R., Mentzer, F., Van Gool, L., & Timofte, R. (2020). Learning for video compression with hierarchical quality and recurrent enhancement. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6628-6637. https://doi.org/10.1109/CVPR42600.2020.00666
[28] Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., & Gao, Z. (2019). DVC: An end-to-end deep video compression framework. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11006-11015. https://doi.org/10.1109/CVPR.2019.01126
[29] Tian, Y., Lu, G., Min, X., Che, Z., Zhai, G., Guo, G., & Gao, Z. (2020). Self-conditioned probabilistic learning of video rescaling. IEEE/CVF International Conference on Computer Vision Workshops, 4490-4499. https://doi.org/10.1109/ICCVW.2019.00552
[30] Hu, Z., Lu, G., & Xu, D. (2021). FVC: A new framework towards deep video compression in feature space. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1502-1511. https://doi.org/10.1109/CVPR46437.2021.00155
[31] Liu, J., Wang, S., Urtasun, R., & Yao, H. (2020). DSIC: Deep stereo image compression. IEEE/CVF International Conference on Computer Vision, 3136-3145. https://doi.org/10.1109/ICCV.2019.00323
[32] Agustsson, E., Minnen, D., Johnston, N., Balle, J., Hwang, S. J., & Toderici, G. (2020). Scale-space flow for end-to-end optimized video compression. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8503-8512. https://doi.org/10.1109/CVPR42600.2020.00853
[33] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2021). Rate-constrained coder control and comparison of video coding standards. IEEE Signal Processing Magazine, 38(6), 128-139. https://doi.org/10.1109/MSP.2021.3097218
[34] Mercat, A., Viitanen, M., & Vanne, J. (2020). UVG dataset: 50/120fps 4K sequences for video codec analysis and development. ACM Multimedia Systems Conference, 297-302. https://doi.org/10.1145/3339825.3394937