

AN OVERVIEW OF QUANTIZATION IN AV2

Joe Young^{*} Jin Zhao^{*} In Suk Chong^{*} Todd Nguyen^{*} Wan-Teh Chang^{*}
 Madhu P. Krishnan[†] Zhijun Lei[‡] Alican Nalci[‡]
 Xiaoqing Zhu[§] Joel Sole[§] Hilmi E. Egilmez[¶]
 Luhang Xu^{||} Yue Yu^{||} Dong Wang^{||}

^{*}Google, [†]Tencent, [‡]Meta, [§]Netflix, [¶]Apple, ^{||}Oppo

ABSTRACT

This paper presents an overview of the quantization techniques used in the AV2 video coding standard. For scalar quantization, an improved design consolidates six lookup tables into a simplified function where the step size doubles with every 24 quant index increments. User-defined quantization matrices and segment-level matrix selection are supported. Moreover, advanced quantization methods are described, including a parity hiding method that can implicitly signal the parity of the DC (0,0) coefficient based on the sum of the AC coefficients, thereby reducing the number of explicitly coded symbols. Additionally, an eight-state trellis-coded quantization method is introduced as a high-performance alternative to scalar quantization. This paper covers the rationale, implementation details, and the resulting coding gain benefits of these methods.

Index Terms— AV2, AV1, Video, Codec, Compression

1. INTRODUCTION

The Alliance for Open Media (AOMedia) has developed the AV2 video codec [1] as the successor to AV1 [2]. Released in 2018, AV1 delivered an average of 30% bitrate savings over VP9 at equivalent or better quality and has since been deployed widely across web platforms and devices [3] [4] [5]. AV2 extends this trajectory by providing an additional 30% bitrate reduction over AV1. At the same time, the design of AV2 emphasizes reduced hardware cost, implementation feasibility, and consistent decoder throughput, ensuring practicality for both software and hardware deployments. These principles guided the AV2 quantization design, which also contributed to the compression efficiency improvements.

In this paper, the quantization design used in AV2 is introduced. In Section 2, the scalar quantization design in AV1 is first described and a few issues are highlighted. Then the improved quantization design in AV2 is introduced, which resolves the limitation of the AV1 standard. As part of the scalar quantization design, user-defined quantization matrix support in AV2 is also explained. Section 3 covers advanced quantization techniques in AV2, including parity hiding (PH) and trellis-coded quantization (TCQ). Section 4 presents the coding gain achieved through these advanced quantization techniques. Section 5 is the conclusion of the paper.

2. QUANTIZATION DESIGN

2.1. Quantization Design in AV1

The main quantization design in AV1 was inherited from that of VP9 without fundamental changes. In the frame header, a `base_q_idx`

is encoded to specify the base frame quantization index, which is used for luma AC coefficients (i.e., coefficients with index different from 0,0) and also as the base value for other quantizers. The valid range for `base_q_idx` is [0, 255], and it is fixed-length coded with 8 bits. On top of `base_q_idx`, delta values can be coded in the frame header to specify the `q_idx` for other quantizers, including DC coefficients and chroma planes. The final `q_idx` for other quantizers is calculated as:

$$q_idx = \text{clip}(0, 255, \text{base_q_idx} + \delta) \quad (1)$$

The final `q_idx` is an index to a lookup table that defines the actual quantization step size (QStep). In AV1, for different bit depths (8, 10, or 12 bits) and DC/AC coefficient sets, a total of six different quantization step size mapping tables (`Dc_Qlookup` and `Ac_Qlookup`) are defined for AC and DC coefficients. Fig. 1 illustrates the mapping between `q_idx` and $\log_2(\text{QStep})$ values for AC/DC coefficients of 8-bit video.

The inverse quantization process is formulated as:

$$\text{Denom} = \begin{cases} 2 & \text{if } \text{txSize} \in \{32 \times 32, 16 \times 32, 32 \times 16, \\ & 16 \times 64, 64 \times 16\} \\ 4 & \text{if } \text{txSize} \in \{64 \times 64, 32 \times 64, 64 \times 32\} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

$$\text{coeff} = \frac{\text{sign} \times ((\text{level} \times \text{QStep}) \& 0\text{FFFFFF})}{\text{Denom}} \quad (3)$$

$$\text{coeff} = \text{clip}(-2^{7+\text{BitDepth}}, 2^{7+\text{BitDepth}} - 1, \text{coeff}) \quad (4)$$

In the AV1 reference encoder implementation, constant QP (CQP) mode is supported through the `--cq-level` parameter, which specifies the target quality level. The valid range for `cq-level` (also referred to as QP) is [0, 63], with conversion to `q_idx` defined as:

$$q_idx = 4 \times \text{cq-level} \quad (5)$$

After the AV1 specification was finalized, the following limitations have been observed:

- The calculation of the QStep value from `q_idx` is not defined by a clear equation and lacks an analytical justification for the mapping.
- Storing these lookup tables requires a significant memory footprint in AV1 decoder and encoder implementations.
- The maximum QStep values, especially for DC coefficients, are insufficient to achieve very low bitrates, potentially hindering low-delay applications that require rapid bitrate reduction to meet network constraints.

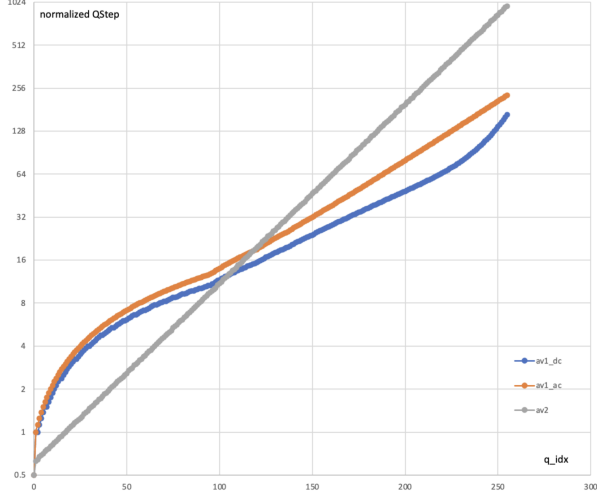


Fig. 1: Mapping between q_idx and quantization step size (QStep) for AV2 (in gray) compared with AV1 (in blue and orange). Separate mappings are defined in AV1 for AC and DC coefficients, whereas AV2 unifies such mapping via a common equation. The y-axis shows the values of QStep in log scale.

2.2. Quantization Design in AV2

As described in Section 2.1, AV1 employs six lookup tables to define the q_idx to QStep mapping for different bit depths (8, 10, or 12 bits) and coefficient types (AC/DC). This approach results in the non-analytical mapping between the QStep and q_idx shown in Fig. 1. AV2 replaces this table-based lookup approach with a closed-form exponential equation, providing a clear mathematical formulation between q_idx and QStep and reducing memory requirements in both encoder and decoder implementations, as also shown in Fig. 1. This makes the relationship between q_idx and bitrate more predictable across different quantization levels, simplifying the design of rate-control algorithms built on top of AV2.

AV2 further extends the maximum quantization step size beyond AV1's range, broadening the operational bitrate regime. This enhances the codec's flexibility for low-latency applications, which often require large bitrate reductions in bandwidth-constrained scenarios.

Unified equation for q_idx to QStep mapping. AV2 replaces the six separate QStep lookup tables for AC and DC coefficients in AV1 with a single unified exponential function. For 8-bit video, the following piecewise equation defines the mapping between q_idx and QStep:

$$QStep(q) = \begin{cases} 32 & q = 0 \\ \text{round}(2^{(q+127)/24}) & q \in [1, 24] \\ \left[QStep((q-1) \bmod 24) + 1 \right] \times 2^{\lfloor (q-1)/24 \rfloor} & q \geq 25 \end{cases} \quad (6)$$

This formulation ensures that the quantization step size doubles when q_idx is incremented by 24, providing consistent bitrate behavior across the valid q_idx range. To support lossless coding as in AV1, the QStep value at $q = 0$ is set to 32, accounting for the gain of 4 from the 4×4 Hadamard transform used in lossless mode.

To prevent the exponential function from producing duplicate QStep values at low q_idx ranges, the precision of QStep is increased by 8 times compared to AV1. The enhanced-precision QStep values can still be represented as 16-bit unsigned integers, and the actual quantization operation remains unchanged from AV1.

Delta q_idx Offset for DC and AC Coefficients. In AV1, the same q_idx maps to different QStep values for DC and AC coefficients. AV2 provides greater flexibility by allowing applications to adaptively select different QStep values for luma, chroma, and their DC/AC components. This is achieved by signaling delta offsets at the sequence and frame levels. Specifically, a 5-bit $base_y_dc_delta_q$ and a 5-bit $base_uv_dc_delta_q$, both in the range of $[-8, 23]$, can be coded. These values represent the offset of the DC q_idx for luma (Y) and chroma (UV) from the $base_q_idx$ used for luma AC coefficients.

$$\begin{aligned} q_idx_{Y_DC} &= \text{clip}(0, 255, base_q_idx \\ &\quad + base_y_dc_delta_q) \\ q_idx_{UV_DC} &= \text{clip}(0, 255, base_q_idx \\ &\quad + base_uv_dc_delta_q) \end{aligned} \quad (7)$$

For more precise color channel adjustments, separate delta values can optionally be signaled for the U and V chroma components, giving more control over visual quality.

Quantization for High Bit-Depth Video. For videos with bit depths greater than 8, AV2 adopts a unified offset-based approach to match bitrates across different internal bit depths (IBDs) while enabling access to the smallest QStep values for higher bit depths. This method introduces the following normative changes:

- The q_idx syntax is extended from 8-bit to 9-bit for IBDs greater than 8. For the main profile, the operational range becomes $[0, 255]$ for 8-bit and $[0, 303]$ for 10-bit video.
- When q_idx exceeds 255, the value is offset by -48 for 10-bit video, and the QStep is scaled by a factor of four.
- Similar changes are anticipated for IBDs greater than 10 bits to be defined in future profiles.

At the encoder, to match bitrates across different IBDs, an offset is added to the q_idx , and the modified q_idx is signaled in the bitstream. This unified approach eliminates the need for storing additional lookup tables while maintaining consistent quantization behavior across bit depths.

2.3. Quantization Matrices

Quantization matrices (QMs) are used to scale transform coefficients prior to quantization, reducing visible artifacts by preserving low-frequency energy while allowing higher frequencies to be quantized more aggressively.

In AV1, QMs were limited as they were predefined and user-defined QMs were not allowed. AV2 introduces a more flexible framework. Unlike AV1, AV2 allows user-defined QMs for transform sizes of 8×8 and smaller (including 8×4 , 4×8 , and 4×4). AV2 further supports segment-adaptive QMs, allowing application-specific tuning, such as more aggressive quantization at low bitrates. Signaling for user-defined QM parameters is further optimized by transmitting only half of the symmetric 8×8 matrices and collapsing long sequences of repeated values using a stop symbol. Additionally, a new QM open bitstream unit (OBU) is introduced, which allows

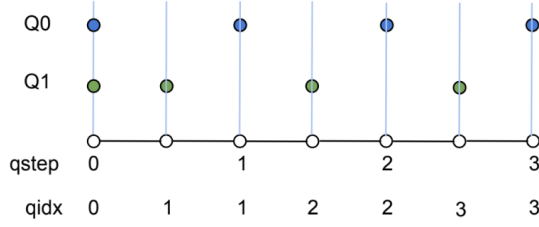


Fig. 2: TCQ with dual sub-quantizers Q0 and Q1

per-frame updates for both predefined and user-defined QMs. Collectively, these improvements make QMs in AV2 more flexible and perceptually effective, improving quality at low bitrates with negligible coding loss.

3. ADVANCED QUANTIZATION METHODS IN AV2

3.1. Trellis Coded Quantization

Trellis coded quantization (TCQ) is a well-studied advanced quantization method that can improve coding efficiency with a modest increase in complexity [6]. AV2 supports TCQ with flexible signaling at both the sequence and frame levels. When TCQ is enabled, it is applied to luma transform blocks (TBs) with 2D scan, whereas TBs with 1D scan, chroma blocks, and blocks with Forward Skip Coding (FSC) all fall back to SQ.

Similar to existing methods—such as the version adopted by JPEG 2000 extensions [7]—the TCQ mode in AV2 defines two underlying sub-quantizers, labeled as Q0 and Q1. As Fig. 2 illustrates, the non-zero reconstruction levels of Q1 are offset by half of the quantization step size relative to Q0. The sub-quantizer selection is governed by a finite-state machine where each state is associated with either Q0 or Q1 (see Fig. 3).

The initial state is set to zero at the start of decoding each TB. As coefficients are processed in scan order, their magnitudes and parities are decoded using novel syntax that ensures consistent parity across multiple syntax passes. The parity bit of the decoded coefficient (i.e., even or odd) dictates the transition to one of two possible next states, which in turn determines which sub-quantizer will be applied for the next coefficient.

Syntax Adjustments for Coefficient Coding. In AV2, the quantiza-

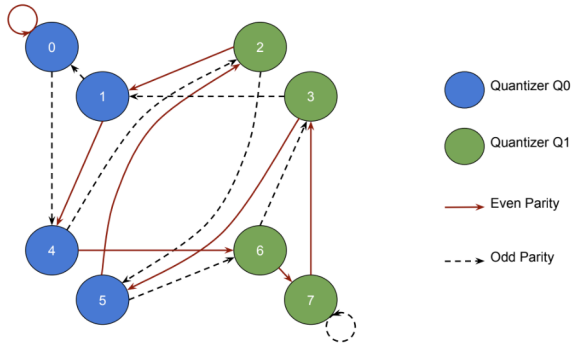


Fig. 3: TCQ state machine.

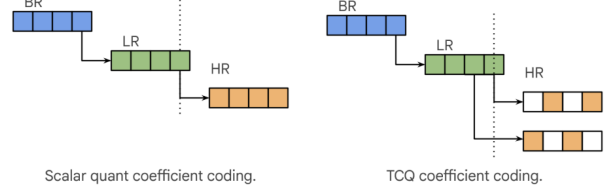


Fig. 4: Coefficient coding in TCQ ensures consistent parity between the high-range (HR) pass and the base-range (BR) and low-range (LR) pass.

tion index of a given coefficient is coded using several syntax symbols—base range (BR), low range (LR), and high range (HR) respectively—each representing a portion of the overall index value. The highest-valued symbol in both the BR and LR syntax serves as an escape code, signaling the use of the subsequent syntax level. The coefficients in a block are coded in two passes. The first pass conveys the BR and LR values using multi-symbol arithmetic coding; the second pass is dedicated to coding HR symbols and signs in bypass bins. Fig. 4 shows the changes in TCQ mode to maintain consistent parity across both passes. The two highest values of the combined BR and LR ranges are designated as escape codes. This necessitates scaling the HR syntax by a factor of two. In addition, TCQ mode uses separate BR probability models for Q0 and Q1, resulting in more efficient arithmetic coding of these symbols.

Encoder Algorithm. The AV2 reference encoder uses the Viterbi algorithm [8] [9] to jointly optimize both the quantization decisions and the end-of-block (EOB) position. As illustrated in Fig. 5, the encoder maintains nine states: one indicating all-zero coefficients and eight coded states corresponding to different choices of sub-quantizers (Q0/Q1) and parity bits (even/odd). The state transition rules follow the TCQ state diagram in Fig. 3. Note that no other states can transition into the all-zero state.

The encoder algorithm is executed in two passes. In the first pass, the encoder starts from the last non-zero coded coefficient (identified through a simple pre-truncation step) and iterates through each coefficient position in reverse scanning order. At each position, quantization decisions (represented by the transition arrows in Fig. 5) are evaluated to update the joint rate-distortion (RD) costs for each of the potential next states. When multiple decisions lead to the same next state, only the decision with the lowest RD cost is retained. This iterative process continues until the final coefficient

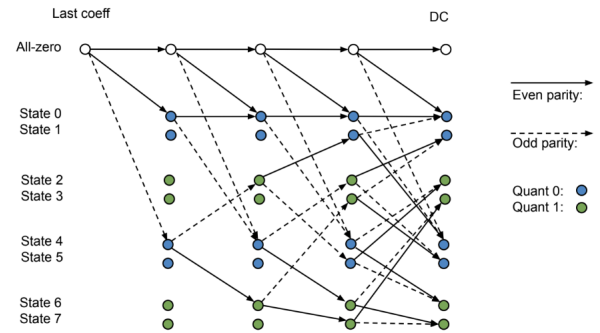


Fig. 5: TCQ encoder algorithm.

position is reached. The final state with the lowest cumulative joint RD cost is selected. The encoder then performs a second backtracking pass through the trellis, yielding the quantization decisions of all previous coefficients as well as the optimal EOB position.

Encoder Runtime Reduction. The Viterbi algorithm used in the encoder, while efficient, can still lead to a significant increase in encoder runtime. The AV2 reference encoder employs the following techniques to mitigate this complexity.

First, Single Instruction Multiple Data (SIMD) optimizations are leveraged to parallelize the trellis state evaluations. This approach yields significant reductions in encoding cycles, with a 52% decrease observed for All-Intra (AI) configurations and a 40% decrease for Random-Access (RA) configurations. Further vectorization is achieved by processing consecutive coefficients within the same diagonal coordinate in parallel, exploiting the absence of entropy context dependencies between them. Updates across trellis states are also efficiently parallelized with the execution of a small number of SIMD instructions.

Second, the complexity of the trellis search is inherently linked to the number of coefficients evaluated, which is proportional to the initial tentative EOB position. To expedite the encoding process, an EOB pre-truncation technique can be applied. This involves using a simpler, faster search method, such as rate-distortion optimized (RDO) scalar quantization, to determine a reduced EOB position. This pre-truncation effectively limits the search space for the more complex Viterbi algorithm, leading to a reduction in encoding time.

Quantization Step Size Adjustment. For a given content and quantization step size QStep, TCQ systematically yields lower quantization distortion than its SQ counterpart. To better accommodate TCQ within the existing rate-distortion optimization (RDO) framework of AV2, the encoder adds an internal offset to q_idx before deriving the value of QStep following the same mapping as in SQ. This offset is empirically chosen as +2 for 8-bit sequences and +4 for 10-bit sequences, so as to approximately align the expected distortion between SQ and TCQ at the same input q_idx .

3.2. Parity Hiding (PH)

Parity hiding (PH) [10] is a novel entropy coding tool in AV2 that infers the parity of the DC coefficient level in a transform block based on the magnitudes of the AC coefficients. This enables implicit signaling of the DC parity, thereby reducing the coefficient level value range by half and lowering the number of explicitly coded bits.

PH is applied to transform blocks that use a trigonometric transform in at least one direction (i.e., not used for 2D IDTX). PH is disabled for lossless coding and for chroma blocks. When a transform block has at least four non-zero AC coefficients, the decoder computes the parity of their sum and uses it to implicitly determine the parity of the DC coefficient. In addition, PH is implicitly disabled in blocks where TCQ is enabled.

If the preceding AC coefficients indicate that the DC term should be parity-hidden, the decoder reconstructs the DC coefficient level value as:

$$qcoeff_{DC} = ([BR_{DC} + LR_{DC} + HR_{DC}] \times 2 + \text{parity}) \times \text{sign} \quad (8)$$

where BR, LR, and HR indicate the total reconstructed level sum based on Base, Low, and High range coding passes. Otherwise, if the number of non-zero AC coefficients is fewer than four, the DC

coefficient is reconstructed conventionally:

$$qcoeff_{DC} = [BR_{DC} + LR_{DC} + HR_{DC}] \times \text{sign} \quad (9)$$

The parity term is computed over the non-zero AC terms as:

$$\text{parity} = \left(\sum_i [BR_{AC_i} + LR_{AC_i}] \right) \bmod 2 \quad (10)$$

Encoder Behavior. After quantization, the encoder evaluates the cost of possible parity adjustments to match the desired parity without significantly increasing the bitrate. When needed, it selectively perturbs one or more coefficients as an encoder optimization (e.g., converting a zero to one, or adjusting an existing value by ± 1) to align the summed AC parity with the desired hidden parity of the DC coefficient.

This adjustment is guided by a decision table based on the number of non-zero AC coefficients, as shown in Table 1.

Table 1: Encoder Strategy for Parity Hiding Based on Non-Zero AC Coefficients

Non-Zero AC Coefficients	Search Minimal Cost Among	PH State
3	Adjust one zero to non-zero	Enabled
	Disable PH for the block	Disabled
4	Adjust one zero to non-zero	Enabled
	Tweak non-zero by ± 1	Enabled
	Adjust parity to be hidden	Enabled
	Adjust non-zero to zero	Disabled
> 4	Adjust parity of one coefficient	Enabled
	Adjust parity to be hidden	Enabled

4. EVALUATION RESULTS

We evaluate the performance of the advanced quantization techniques in AV2 – TCQ and PH – against the baseline SQ in the AVM research-v13.0.0 reference encoder [11]. Following the Common Test Conditions v8 (CTC) [12], the experiments cover all three configurations: All-Intra, Random-Access, and Low-Delay across multiple CTC content classes, as summarized in Table 2. We report on coding gains in terms of BD-rate [13] for both PSNR-YUV (using weights of 14:1:1 across PSNR-Y, PSNR-U, and PSNR-V) and VMAF [14].

Table 3 shows the coding gains achieved by TCQ and PH over SQ across all configurations per content class. Table 4 further summarizes the overall coding gains per configuration, along with corresponding encoding and decoding run times and additional comparison between TCQ and PH. Both TCQ and PH lead to substantial BD-rate gains over SQ on the CTC sequences in PSNR-YUV. The average coding gains are between -1.09% to -1.33% for TCQ, and between -0.21% to -0.26% for PH across the three configurations, and even higher for still images. When measured in VMAF, coding gains from TCQ remain high (-1.29% – -1.69%) while PH yields a small degradation (+0.01% – +0.13%). It can also be noted that both schemes introduce negligible increase in decoder run time, with fairly modest encoder run time increase from TCQ.

Table 2: Test Sets

Content Type	Class	Typical Resolution
Natural	A1	3840 x 2160 (2160p)
	A2	1920 x 1080 (1080p)
	A3	1280 x 720
	A4	640 x 360
	A5	480 x 270
Synthetic	B1	1920 x 1080
Screen Content	B2	1920 x 1080
Still Image	F1	4032 x 3024
	F2	1208 x 948

Table 3: BD-rate results for PSNR-YUV and VMAF across All-Intra, Random-Access, and Low-Delay configurations.

Configuration	Class	TCQ (vs. scalar)		PH (vs scalar)	
		PSNR-YUV	VMAF	PSNR-YUV	VMAF
All-Intra	A1	−1.29	−1.44	−0.21	+0.05
	A2	−1.54	−1.86	−0.28	+0.11
	A3	−1.24	−1.47	−0.28	+0.06
	A4	−1.37	−2.17	−0.16	+0.29
	A5	−1.14	−1.84	−0.24	+0.30
	B1	−1.09	−1.41	−0.21	+0.14
	B2	−1.33	−1.00	+0.01	+0.10
	Average (without B2)	−1.33	−1.69	−0.24	+0.13
Random-Access	A1	−0.91	−1.10	−0.24	+0.07
	A2	−1.30	−1.40	−0.20	−0.17
	A3	−1.14	−0.97	−0.30	+0.13
	A4	−1.21	−1.93	−0.27	−0.44
	A5	−0.62	−1.10	−0.23	+1.30
	B1	−0.93	−1.23	−0.24	−0.01
	B2	−0.42	−0.50	−0.05	−0.08
	Average (without B2)	−1.09	−1.30	−0.26	+0.01
Low-delay	A2	−1.36	−1.24	−0.22	+0.16
	A3	−0.99	−0.68	−0.20	+0.09
	A4	−1.19	−2.17	−0.15	+0.00
	A5	−0.70	−1.39	−0.32	+0.17
	B1	−0.98	−1.31	−0.19	+0.19
	B2	−1.14	−1.29	+0.29	+0.13
	Average (without B2)	−1.14	−1.29	−0.21	+0.13
Still Image	F1	−2.58	−2.54	−0.55	+0.34
	F2	−1.64	−2.72	−0.29	+0.37

Table 4: Coding gain summary, including relative encode and decode run time.

Configuration	PSNR-YUV	VMAF	EncTime	DecTime
TCQ vs. baseline				
All-Intra	−1.33	−1.69	122.1%	101.3%
Random-Access	−1.09	−1.30	113.7%	101.1%
Low-Delay	−1.14	−1.29	113.2%	100.1%
PH vs. baseline				
All-Intra	−0.24	+0.13	103.8%	100.6%
Random-Access	−0.26	+0.01	102.0%	100.6%
Low-Delay	−0.21	+0.13	102.4%	100.6%
TCQ vs. PH				
All-Intra	−1.09	−1.82	117.6%	100.1%
Random-Access	−0.84	−1.31	111.5%	100.5%
Low-Delay	−0.93	−1.42	110.6%	100.2%

5. CONCLUSIONS

The quantization design in AV2 includes multiple improvements, ranging from a unified mapping function from quantization index to step size and more flexible quantization matrices (QMs) to more advanced quantization methods such as trellis-coded quantization (TCQ) and Parity Hiding (PH). Collectively, these advancements contribute to significant compression gains and offer more predictable rate control. They are designed with hardware feasibility in mind and incorporate complexity-reduction techniques in the reference implementation, aiming for both improved efficiency and perceptual quality in the next generation of video coding.

6. REFERENCES

- [1] Alliance for Open Media, “AV2 specification draft,” [Online] Available: <https://av2.aomedia.org/>, 2026.
- [2] “Alliance for open media,” [Online]. Available: <https://aomedia.org/>.
- [3] Jingning Han, Bohan Li, Debargha Mukherjee, Ching-Han Chiang, Adrian Grange, Cheng Chen, Hui Su, Sarah Parker, Sai Deng, Urvang Joshi, Yue Chen, Yunqing Wang, Paul Wilkins, Yaowu Xu, and James Bankoski, “A technical overview of av1,” *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, 2021.
- [4] Ryan Lei, Haixia Shi, Haoteng Chen, Ali Monfared, and Cheng Shi, “How meta brought av1 to reels,” [Online] Available: <https://engineering.fb.com/2023/02/21/video-engineering/av1-codec-facebook-instagram-reels/>.
- [5] Liwei Guo, Zhi Li, Sheldon Radford, and Jeff Watts, “Av1 — now powering 30 [Online] Available: <https://netflixtechblog.com/av1-now-powering-30-of-netflix-streaming-02f592242d80>.
- [6] M. W. Marcellin and T. Fischer, “Trellis Coded Quantization (TCQ) of Memoryless and Gauss-Markov Sources,” *IEEE Transactions on Communications*, vol. 38, no. 1, pp. 82–93, Jan. 1990.
- [7] International Telecommunication Union (ITU-T), “Information technology – JPEG 2000 image coding system: Extensions,” *ITU-T Rec. T.801 (08/02)*, pp. 37–48, Aug. 2002.
- [8] Andrew J Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [9] G David Forney, “The viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [10] Luhang Xu and Yue Yu, “Parity hiding for coefficients coding,” *Proc. SPIE 12674, Applications of Digital Image Processing XLVI*, 2023.
- [11] Alliance for Open Media, “Avm reference software,” [Online] Available: <https://gitlab.com/AOMediaCodec/avm>, 2025.
- [12] Xin Zhao, Zhijun Lei, Andrey Norkin, Thomas Daede, Alexis Tourapis, Vibhoothi Vibhoothi, Van Luong Pham, Dzung Hoang, Aki Kuusela, and Mohammed Golam Sarwer, “Aom common test conditions v8.0,” [Online]. Available: https://aomedia.org/docs/CWG-F384o_AV2-CTC_v8.pdf, 2025.
- [13] Gisle Bjontegaard, “Calculation of average psnr differences between rd-curves,” *ITU-T SG16, Doc. VCEG-M33*, 2001.
- [14] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara, “Toward a practical perceptual video quality metric,” *The Netflix Tech Blog*, [Online] Available: <https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652>, vol. 6, no. 2, 2016.