MASTERING QUANTIZATION IS KEY FOR VIDEO COMPRESSION

M. Ropert, J. Le Tanou and M. Blestel

Mediakind, France

ABSTRACT

This paper discusses some key elements and solutions for efficient video compression. More precisely, we introduce key differentiating techniques for video signal quantization within modern hybrid video codecs. First, we recall the fundamentals about scalar quantization and Rate-Distortion (R-D) theory, then describe the different control points and levels of granularity for optimizing signal quantization into modern video compression standards. From this common knowledge, we discuss approaches in how to jointly optimize those various levels of quantization refinements for improved video encoding quality. Most notably, we explain how to take advantage of a look-ahead module (available in most of industrial encoder implementations) to model spatio-temporal coding dependencies, and how to further compute optimal quantization information from an R-D standpoint for a group-of-pictures (GOP). Complementary to this first approach, we share some insights about a Local Quantization Refinement (LQR) algorithm. Such algorithm is often ignored in practical encoder implementation due to its apparent complexity; we develop how and why it can efficiently work in real-time software encoding.

INTRODUCTION

Video delivery from live event capture or content production to the final customer undergoes several stages of transformation along the chain. The initial compression applies at the contribution side to convey the video to a studio or production facility. Then, several compression and decompression steps occur in the ecosystem. At the distribution side, the video was historically broadcasted on traditional networks (terrestrial, cable, or satellite) as linear TV channels, and more recently IPTV services. Today, thanks to OTT services, new habits of video consumption have been created around the nonlinear TV combining live, on-demand and various possible use cases (e.g. replay, start-over, follow-me, download and go, etc.). For all the market segments, video compression is the crux of the matter for bandwidth or storage savings, and for implementation cost reduction of the overall delivery chain.

The motivation of this paper is to share some technical insights for efficient video compression. More precisely, we discuss some advanced features for video signal quantization, as one of the most fundamental processing steps in any video delivery solution. Complementary to this aspect, additional techniques related to pre- and post-processing [1], that could be used for further optimization, are not addressed in this paper.

Video Compression in a Nutshell

Lossless compression is conceptually simple. It consists in exploiting signal information redundancy to reduce the data rate while no signal loss is introduced, such that the decompressed signal is strictly identical to the source signal. A lossless encoder tries to avoid sending what can be predicted by any means. It is commonly based on the principle of differential predictive coding (DPC). Any signal information to code is predicted from signal information already coded or known at the decoder. Then, only the difference between the source and prediction, i.e. the residual signal information, needs to be coded and sent to the decoder. Usually, DPC is complemented with transformation and entropy coding steps for further signal redundancy reduction. Prediction, transformation, or entropy coding capabilities are specified by the standards (e.g. H.264/AVC, H265/HEVC, H.266/VVC or VP8/9, AV1, etc.). In addition to the residual information, the prediction (eventually the transformation or entropy coding, etc.) model parameters to apply at the decoder side are also sent. The overall data flow to send to the decoder is responsible for the data rate (i.e. bitrate in case of binary signal coding).

Lossy compression adds a lossy quantization step on-top of a lossless compression scheme. Residual signal information is quantized (integer division) to further reduce the data rate. This operation is irreversible, some data are lost and the decompressed signal is not strictly identical to the source signal. It introduces the notion of distortion which measures the distance or alteration between the original and the decoded signals.

Hybrid Video Compression Standards

Hybrid video coding is defined as the combination of a differential prediction (e.g. Motion Compensated Prediction (MCP)) stage and a transformation stage (e.g. 2D Discrete Cosine Transform (DCT)) of the residual signal. It can be lossless or lossy, i.e. with or without quantization stage. Most if not all the video codecs used for video delivery are based on a lossy hybrid video coding scheme.

Figure 1 describes a generic lossy hybrid encoder as a block diagram. There are some subtilties depending on the standards, but basically, this generic architecture holds. As shown in Figure 1, the decoder reconstruction process is closed-loop into the encoder, such previous reconstructed samples can be used for prediction of the incoming video signal samples. The difference between the video input samples and its prediction produces residues that are transformed and quantized. The transformation is an operation that decorrelates and compacts residual signal information on fewer samples or coefficients. The transformed residues are then further quantized. The quantization is the only lossy element that this paper will focus on. Inverse quantization and inverse transform are part of the decoder reconstruction loop producing the distorted residues. They are added back to the prediction and (optionally) filtered to reconstruct the final video signal samples, identical to the one the decoder will obtain. The entropy coding block is a lossless compression engine.

Processing blocks in red are fixed. They must be considered as automatons. Any change in their specified process at the encoder would produce a mismatch with the decoding process. In such scenario, the encoder would be non-conformant with the decoder as specified by a standard, and in the worst-case, the compressed signal would not be decodable.

Conversely, processing blocks in green or dashed green are relatively free regarding what they output. The only constraint they have is to comply with the syntax (i.e. the processing model parameters) specified by the considered standard for being properly interpreted by the decoder. Solid green blocks, i.e. transform and quantization processing steps, have even

more freedom, they could to its limit output null sample values for example. Finally, for a given standard, optimizing the video compression efficiency of an encoder is all about optimizing these processing blocks.

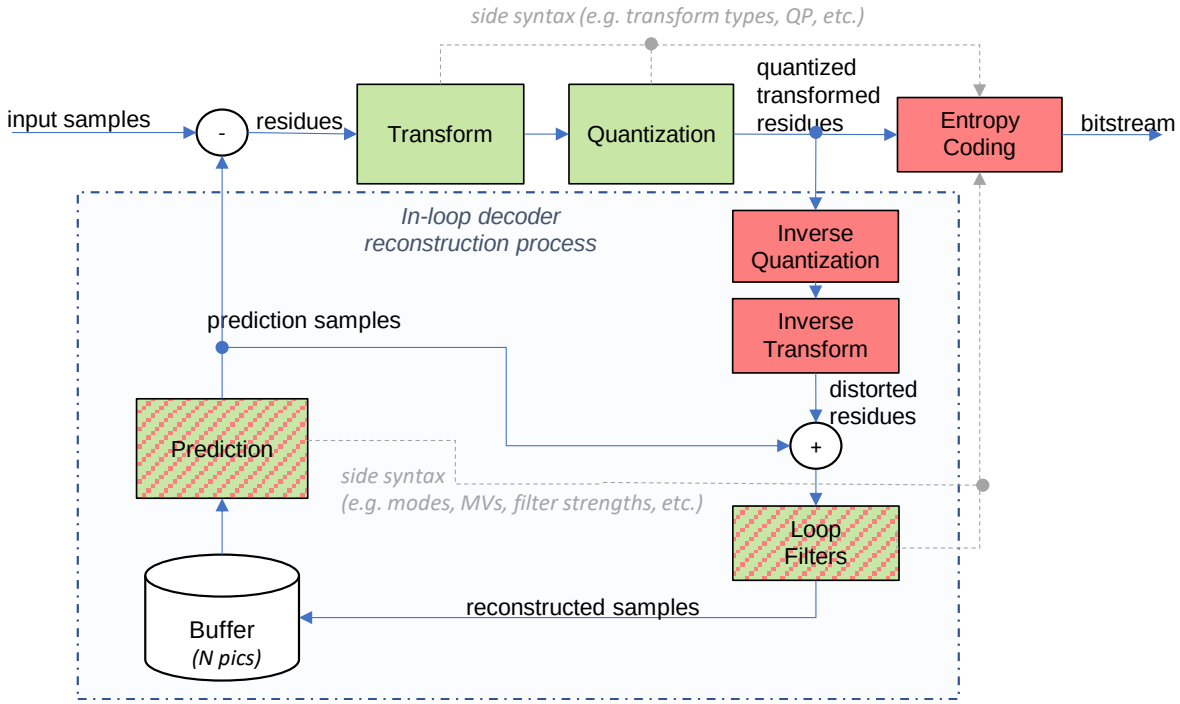

Figure 1: Generic Hybrid Encoder Scheme

BASICS OF QUANTIZATION IN HYBRID VIDEO COMPRESSION STANDARDS

Principle

Once again, the concept of quantization is simple. The purpose is to map a set of values to a smaller number of values. Quantization is an irreversible process since it introduces data loss. This is where lossy compression stands: select a single representative for several values.

Vector quantization is not covered here, only scalar quantization is considered in usual video coding schemes (e.g. Figure 1) where the transform process is already assumed to output highly decorrelated samples.

In common encoders, the scalar quantization operation falls down to an integer division (with rounding) where the quantization step (QStep) is the denominator. For example, as shown Figure 2, dividing by 10 and rounding to the closest integer would quantize the range {0,1,2,3,4} to the value 0, and {5,6…., 13,14} to value 1, etc. The higher the quantization step, the stronger the quantization and the signal loss.

The dequantization is specified and locked by the standard. Basically, it is a multiplication by the quantization step. If we come back to previous example, 0 is dequantized to 0 x 10 = 0, 1 dequantized to 10, 2 to 20, etc. Overall, if we combine quantization and inverse quantization, the resulting effect is simply a rounding.

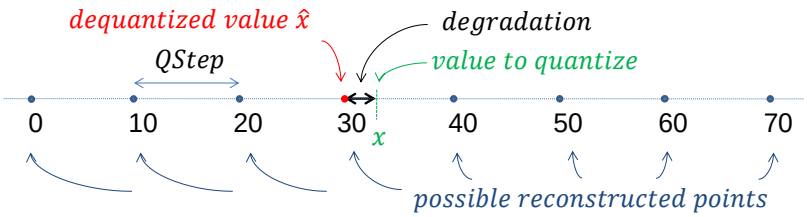

Figure 2: Example of rounding operation out of the scalar quantization/dequantization

Recent advances in the quantization syntax coding are brought by VVC [2]. Indeed, by adding memory in the coefficient coding syntax, “dependent quantization” [3] achieves the coding of two possible quantized values with a single syntax element. The dequantization to apply at the decoder side depends on the path of previously decoded coefficients using a dedicated state machine. Nevertheless, the underlying mechanism remains scalar quantization.

Rate-Distortion Optimization

Consequently, quantization generates distortion: reconstructed signal samples are not identical to original signal samples. The distortion D, as a distance between the original and the reconstructed signals, is often measured using objective scores (e.g. PSNR, SSIM [4], VMAF [5], etc.) or based on subjective criteria and protocol (e.g. MOS, [17]).

Overall, encoders try to maximize the video quality (i.e. minimize the distortion), while constraining to a target bitrate. Naively, this could be achieved by just lowering the quantization step to minimize the distortion, at the expense of a dramatic increase in the bitrate. Modern encoders are based on a trade-off of the two: called Rate-Distortion Optimization (RDO) [6].

The rate R is the bitrate, it is just a matter of counting bits. At the opposite, the distortion D is not necessarily the standard Mean-Squared Error (MSE), as explained before it could be any other criterion. Most of commercial encoder providers in the Industry have researched and designed their own computable (i.e. real-time) D embedded into the encoding loop and accounting for Human Visual System (HVS) quality perception. In the hybrid video coding scheme of Figure 1, it is worth mentioning that distortions propagate, by design, from image to image through a Group-Of-Pictures (GOP). We built the framework, briefly described afterward, to account for the impact of this propagation on the video quality.

QUANTIZATION IN MODERN VIDEO STANDARDS

The quantization process in modern video standards is controlled at different levels by a Quantization Parameter (QP) and several optional refinement steps. In AVC/H.264 or HEVC/H.265, the QP ranges from 0 to 51. In VVC/H.266 the range has been extended to [0, 63]. For all these standards, the QP is used as an index to derive the quantization step (QStep), that doubles each time the QP increases by 6.

The QP can be adapted either at the picture level, or using a finer granularity at the block (or coding unit (CU)) level, as shown in Figure 3. The picture QP adaptation is generally used to fulfil the global rate constraint (i.e. target bitrate) over a GOP, i.e. trading bits within the GOP to optimize the R-D criterion. For example, it may be appropriate to spend more bits on a frame used as a reference into a given GOP structure, since its errors will be propagated onwards by prediction.

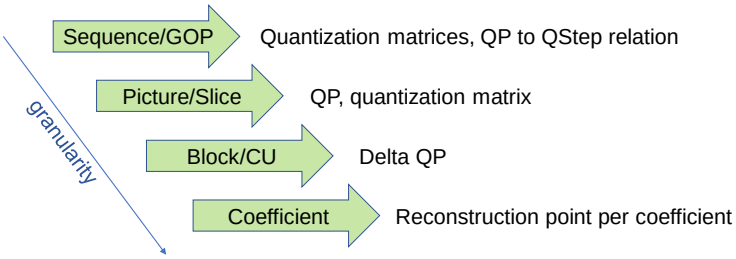

Figure 3: Quantization control point hierarchy

In addition to these two levels of QP adaptation, the quantization process can be refined down to the transformed coefficients (i.e. transformed residues), optimizing the quantized coefficient value selection.

Picture/slice level

The picture/slice QP is then used to compute the QP for the first block of a slice. If not further refined, this QP value stands for the whole slice (either a whole picture or a part of a picture).

Complementary to the QP, and usually signaled at the sequence and/or picture-level, quantization matrices allow supporting frequency-dependent quantization. For each transform block size, the quantizer step can be adapted per frequential coefficient position. For instance, and in order to better match an HVS perception, low-frequency coefficients can be quantized with a finer quantization step compared to high-frequency coefficients. Within MPEG video standards, this is managed with scaling matrices, which are optionally transmitted in the sequence and picture parameter sets (SPS and PPS), and referred for use (or not) at the picture level.

Block/CU Level

Adapting the QP at the block or coding unit (CU) level is the cornerstone of compression efficiency.

In most of the video codecs/standards, block QP value adaptation is allowed. Usually, to reduce the amount of data to transmit for each coding block, the local QP value is DPC coded. A QP prediction is made from neighboring blocks and previous QP in decoding order, such that only the delta QP, difference between prediction and current QP values, must be signaled.

Again, only the QP prediction and signaling mechanism are standardized, that leaves us free to develop Adaptive QP (AQP) algorithms with various optimization criteria, various scopes (i.e. block level, frame level, GOP level), and various complexities. Typically, the block QP value could optimize the quantization level to the local characteristics of the signal and/or to the block pertinence in the prediction scheme, both to provide better visual quality.

Adapting the local QP to local characteristics using RDO is an option for maximizing compression efficiency. As mentioned before, the purpose is to limit the distortion with respect to the rate. The distortion is linked to the QStep i.e. as QStep increases, the distortion increases. The rate evolution according to the quantization step is slightly trickier, as the rate depends on delta QP cost, the number of quantized coefficients, and their magnitude. The higher the QStep, the lower the number of transformed coefficients and their magnitude, but higher could be the delta QP cost. The ideal solution would be to identify for each block in a GOP the sweet set of QP values that provides the best global R-D trade-off.

AQP algorithms usually aim at determining a priori for each block the best QP that would provide the best global subjective or objective quality while fulfilling the rate constraint. These algorithms can be designed to estimate the QP map for a unique frame considering only spatial information (i.e. statistics intra-frame or block). Better algorithms would typically account for temporal information (i.e. statistics inter-frame), e.g. trying to measure the block persistence into the GOP. These kinds of algorithms, e.g. Spatio-Temporal AQP, succeed in better optimizing the global R-D trade-off by estimating all temporal and spatial dependencies between blocks into a GOP.

Complementary to a priori AQP algorithms (i.e. based on estimations), the local QP can be a posteriori refined in order to adjust the R-D trade-off. Such a posteriori algorithms, e.g. Local QP Refinement (LQR) or “Multiple QP optimization” [7], adjust a set of QP candidates for a given block by minimizing a local R-D criterion. If carefully implemented, LQR algorithms bring significant gains in coding efficiency without compromising a global RDO. The add-on in coding efficiency comes from not being based on estimations but on the real distortion and rate measurements, considering accurately all dependencies between blocks.

Coefficient level

Final quantization adjustment per transformed coefficient is also possible. It can help at improving objective scores based on a given R-D criteria minimization. But additional perceptual criteria (e.g. noise shaping [8], coefficient filtering/discarding, etc.) can also be used to reduce specific visual artifacts (e.g. banding, ringing, etc.). The main advantage of coefficient quantization optimization resides in not introducing any additional syntax bit-cost overhead; only quantized values are tuned while keeping the quantization parameter unchanged.

For each coefficient, the rounding (into the integer division introduced by the quantization) sets the threshold for mapping a set of values to a unique value. Coming back on the previous example of a quantization step equal to 10, we could have shifted the rounding threshold such that {0,1,2,3,4,5,6,7,8} quantizes to 0, {9,10,….,17,18} to 1, {19,20,….,27,28} to 2, etc. Adjusting rounding offers great freedom in the quantization process. For the discussed example, it was just a modification of the *dead zone* [9], but smarter strategies can be designed.

Trellis quantization, e.g. RDO-Q [10], is an option for smart quantization strategy at the coefficient level. In a typical configuration, for each coefficient, two possible reconstructed values (lower and upper rounding) should be tested and the best one should be retained based on a given R-D criterion. For example, given a coefficient of 57 and quantization step equal to 10, possible quantized coefficients surrounding 5.7 are 5 and 6 with the possible reconstructed values of 50 and 60. The same two options exist for every coefficient in the block resulting in a trellis architecture. It defines a minimal path problem optimization to be solved using a Viterbi algorithm, for identification of the optimal combination of rounding.

We will not develop further this part; however, one important comment is that the coefficient level quantization optimization does not impact the optimization of the quantization step/parameter which is the purpose of this paper.

Summary

Quantization process can be optimized at various levels of granularity. Most of the techniques can be combined, and once a D is defined, the optimization problem to solve is about minimizing D subject to the R(ate) constraint (i.e. a target bitrate to fulfil).

Unfortunately, when dealing with real-world implementations, computational complexity and resources consumption are additional constraints to trade against compression efficiency. Among the various algorithms discussed so far, some are less computationally intensive and more interesting in terms of coding efficiency versus CPU usage. As an example, two advanced quantization algorithms used in MediaKind commercial encoders are introduced and discussed in the next section.

TWO KEY DIFFERENTIATING QUANTIZATION FEATURES

Spatio-Temporal Adaptive Quantization (STAQ)

STAQ is a global R-D optimization algorithm minimizing D subject to R over the entire GOP and providing an optimal local quantizer for each block. In practice, this algorithm is a deep evolution of the macroblock-tree algorithm [11], where all the mechanisms have been revisited with better modelling of the R-D criterion. Most notably, the distortion modelling into STAQ allows easy introduction of perceptual criteria, that helps to significantly improve the subjective quality results in comparison to simpler model based on the MSE.

STAQ is based on a single principle: distortion propagates along time. Quantization process applied on each block generates distortion. By design of the prediction scheme (i.e. motion compensated prediction), a part of the distortions produced on each reference block is propagated on next blocks to code by motion compensation. Thus, image by image, compensation after compensation, block distortions accumulate over time. Typically, the temporal distortion propagation (from one image to another) is maximal with a Skip-coded block, while the propagation is stopped with an Intra-coded block (i.e. no motion compensation).

The essence of the algorithm is then to identify the sample areas that are the most referenced by prediction, encode these areas as good as possible (low distortion/low QP), and copy these areas as much as possible (bitrate almost zero).

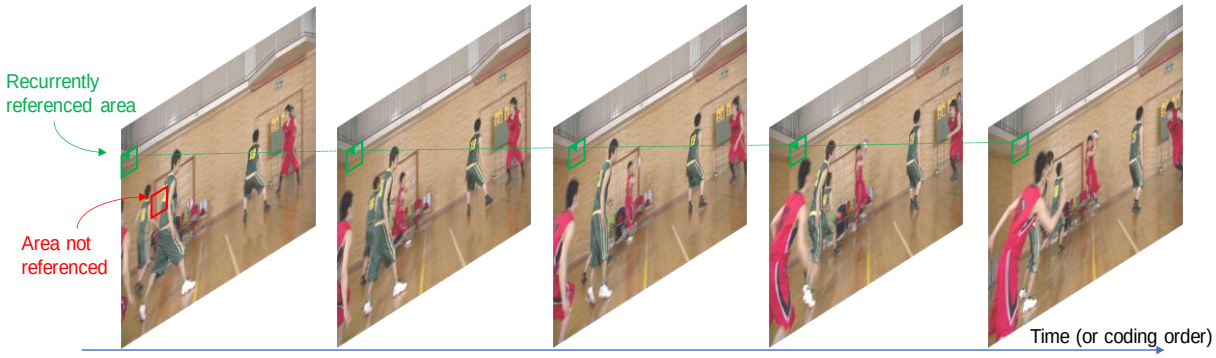

Figure 4: block unused in the future and top-left block recurrently reused

As shown in Figure 4, at the top-left of the first image, the green area (or block) persists in the next images of the sequence and would be often referenced for prediction. By design, this area is then favored on the first image with a lower quantization step (relatively to other sample areas into the GOP). In this example, the considered area is relatively still across time, hence successive motion compensations would tend toward the Skip mode (i.e. a copy of the sample area), and the encoder would generate almost no bits with minimal distortion. The same principle also holds for any motion area that is well predicted. Consequently, a desirable side effect occurs: copies don't generate video quality fluctuations, and the video quality is stabilized across time. Conversely, when an occlusion occurs in future images (red

area in Figure 4), it is highly probable that the next block will be Intra-coded, thus breaking the temporal block dependency. Hence, for an area with a low probability to be referenced for prediction, there is no need to spend too many bits on coding.

Of course, STAQ is more subtle. It builds a weighted dependency network connecting all blocks of the same GOP together, accounting for motion estimation, coding mode probabilities, other information estimated from a Look-ahead module [12][13], and the target bitrate for the GOP. The spatial (i.e. intra-frame) distortion also propagates, usually from the top-left corner of the image down to the bottom right of the image (standard dependent). STAQ integrates both the spatial and temporal distortion propagations into its R-D optimization.

STAQ provides impressive objective gains. We extensively detailed and analysed into [13] a simplified model of STAQ, named Rate-Distortion based Spatio-Temporal Quantization (RDSTQ) algorithm. By implementing RDSTQ algorithm into the HEVC reference Model (HM) [14], we report up to −26.9% and −15.6% average bitrate savings compared to no adaptive quantization for the same SSIM-based and PNSR-based quality, respectively. In the context of the HM, the proposed algorithm significantly outperforms related methods from state-of-the-art. Coding efficiency results using JCT-VC test conditions [15] are summarized in Table 1. We invite interested readers to refer to [13] Section VI for further details of the test conditions.

Classes (Resolutions)	**BD-BR PSNR**	**BD-BR SSIM**
Class A (2160p)	-12.53%	-27.98%
Class B (1080p)	-9.35%	-20.82%
Class C (480p)	-17.56%	-29.78%
Class D (240p)	-15.82%	-30.86%
Class E (720p)	-25.09%	-27.37%
Average	**-15.59%**	**-26.93%**

Table 1: Average bitrate savings of STAQ (RDSTQ) against no adaptive quantization using JCT-VC test conditions [15] and HEVC reference Model (HM).

It is worth mentioning that the same bitrate-saving range is observed if implementing RDSTQ algorithm into x265 open-source encoder [16]. Coding efficiency results into x265 under JCT-VC test conditions are reported in Table 2. The differences in compression efficiency between both implementations into HM and x265, are mainly explained by the differences in the Look-ahead implementation for the two contexts (e.g. difference in coding dependency estimation accuracy).

Classes (Resolutions)	**BD-BR PSNR**	**BD-BR SSIM**
Class A (2160p)	-9.92%	-23.87%
Class B (1080p)	-8.48%	-20.96%
Class C (480p)	-15.12%	-27.88%
Class D (240p)	-11.77%	-27.57%
Class E (720)	-14.28%	-16.43%
Average	**-11.81%**	**-23.53%**

Table 2: Average bitrate savings of STAQ (RDSTQ) against no adaptive quantization using JCT-VC test conditions [15] and x265 encoder.

In addition to the objective metric score comparisons, several subjective quality assessment sessions were run among non-expert MediaKind's employees, and based on a paired

comparison methodology derived from [17]. Analysis of the results demonstrated consistent spatial and temporal quality improvements thanks to the STAQ algorithm. One very important and intrinsic benefit of STAQ is the improvement of the video quality stability along time, which is a characteristic not measured by either SSIM or PSNR. Furthermore, the definition and use of a perceptually weighted distortion into the STAQ optimization, e.g. accounting for spatial masking effect, makes the video quality much more perceptually compelling, picture to picture.

Finally, and as introduced earlier, STAQ relies on a pre-analysis module for various signal statistics estimations, known as Look-ahead. The Look-ahead module is a sub-process available in most (if not all) of efficient commercial encoders. For instance, the computation overhead added for STAQ modelling into MediaKind optimized SW encoder impacts the overall encoding runtime by less than 3% (with optimization and multi-threading) compared to no adaptive quantization. The significant video quality gain relative to the small run-time increase makes STAQ one of the most powerful adaptive quantization algorithm.

Local QP Refinement

LQR stands for Local QP Refinement. Coarsely, it consists, for each block or CU to code, to exhaustively compete a set of local quantization parameters by measuring the resulting Distortion (out of a reconstruction loop) and Rate (out of an entropy coding estimation) trade-off. Such brute-force algorithm or concept is not new [7]; but requires a lot of know-how to be efficiently implemented for real-time software encoding and combined with a global R-D optimization. The motivation of LQR is that by refining or adjusting a posteriori a set of local quantizer candidates would help tracking two beneficial situations (described later on): i.e. either a local “distortion drop” (for almost the same rate) or a local “rate drop” (for almost the same distortion). In this paper, we make the evidence that exploiting (a posteriori) those local drops in Distortion or Rate brings additional compression efficiency without compromising any (a priori) global R-D optimization. Indeed, the LQR add-on in coding efficiency can be complementary to an adaptive QP algorithm with global RDO motivation, such as STAQ. One explanation is that it helps to compensate estimation errors of a priori model such as STAQ, by real posteriori measurements (verifications) of the distortion and rate; for instance, better optimizing the delta QP syntax costs. Besides, as in STAQ, the R-D optimization performed by LQR can be driven by various distortion criteria, e.g. MSE or any other HVS-based metric.

The quantization followed by the dequantization produces a distortion. Slightly changing the quantization step also slightly changes the possible reconstruction values scale: possible reconstructed values slide toward or in the opposite direction with respect to the value to quantize (i.e. transformed coefficients noted x in Figure 5).

Figure 5 illustrates the displacement (shift) of the possible reconstruction values when playing with the HEVC quantization parameter QP. Iso-reconstruction curves for a given quantized value are drawn to exhibit the log form. Quantized values are reconstructed on a reconstruction value grid defined by the QP to QStep scale. As shown in Figure 5, if we consider a given coefficient (x), selecting the appropriate QP, i.e. aligning the reconstruction grid with the transformed coefficient value, can potentially lower or nullify the distortion (which, counter-intuitively, doesn’t mean decreasing the QP value). This beneficial situation is what we defined the *“distortion-drop”*. Despite the probability of such a situation is reducing with the number of non-zero transformed coefficients in a block, the “distortion-drop” effect is still possible for a significant proportion of non-zero transformed coefficients in multiple blocks.

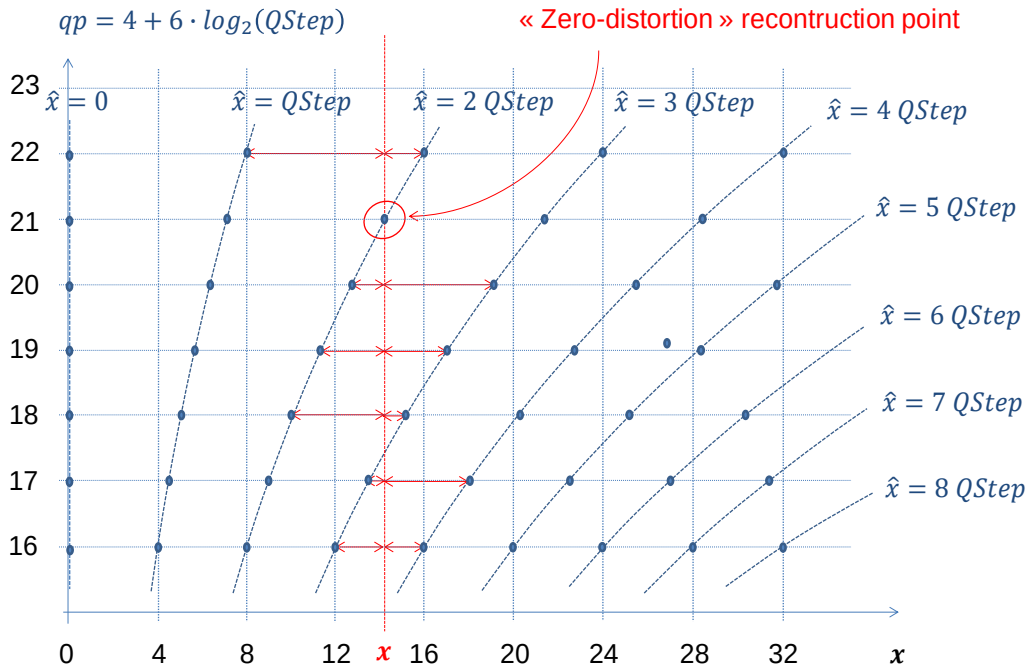

Figure 5: Distortions according to possible reconstruction grids

Conversely, for almost no distortion increase, the rate for a block can favorably decrease for some selected QStep values. By design when increasing the QP, the quantized values magnitude decreases, and the rate decreases as expected; in most of the situations the quantized error and distortion will then increase accordingly. Interestingly, it can be observed that for some transformed coefficient distributions the rate decrease (or "*drop*") can be significant relative to the local distortion increase, resulting in sweet local R-D trade-offs. Additionally, we observe that for some transformed coefficient distributions, and given the CABAC context, a small QP decrease may result in almost no rate increment. It can be explained by two facts. First, the CABAC context may be better suited when changing QP, making the entropy coding of quantized coefficients more efficient. Secondly, a slight quantized coefficients bitrate increase can be compensated by the differential QP syntax bitrate decrease. These two cases generate a lower rate than foreseen that we name "*rate-drop*".

Overall, refining locally a set of QP candidates can at the same time be beneficial for both rate and distortion, without compromising any global RDO, and that is what LQR does. Finally, thanks to additional heuristics and optimizations (e.g. such as distortion estimation in Transform domain), the LQR implementation can be kept reasonable in terms of computation overhead, with overall an encoding runtime increase measured below 10% in a MediaKind optimized software HEVC encoder.

Compression efficiency performance of the LQR algorithm combined with STAQ (RDSTQ), implemented into x265 and using JCT-VC common test conditions, is summarized in Table 3. Complementary to STAQ algorithm, the coding efficiency add-on is about 6% bitrate saving for the same PSNR or SSIM-based quality.

Classes (Resolutions)	BD-BR PSNR	BD-BR SSIM
Class A (2160p)	-5.6%	-4.4%
Class B (1080p)	-6.3%	-7.4%
Class C (480p)	-6.2%	-6.1%

Class D (260p)	-5.4%	-5.4%
Class E (720p)	-8.1%	-7.2%
Average	**-6.3%**	**-6.1%**

Table 3: Additional average bitrate savings of LQR on top of STAQ (RDSTQ) using JCT-VC test conditions [15] and x265 encoder.

CONCLUSION

By sharing an overview of the hybrid video coding scheme, that holds for most of modern video compression standards, we highlight the key role of the quantization in optimizing the video quality-bitrate trade-off, as the (almost) only adjustable lossy processing step in any encoding system. We consequently detail the various levels of granularity and control points available for quantization optimization, and most notably the block or CU-level QP adaptation.

As practical examples, we introduce and share some insights on two differentiating quantization algorithms: namely STAQ and LQR. We show that a careful implementation of these two complementary algorithms can upgrade, by more than -25% in bitrate saving for the same SSIM-based quality, a real-time software encoder based on HEVC. Those algorithms would benefit any standard supporting local QP adaptation (e.g. MPEG-2, H264/AVC, H266/VVC, etc.).

Usually, software reference encoder models used for video standard development, typically from MPEG ISO/IEC or VCEG ITU-T, do not implement look-ahead and/or advanced encoder-only quantization techniques. It somewhat underestimates the performance in compression efficiency offered by a given standard, and it is finally into that missing optimizations where most of commercial encoder providers will compete/differentiate. Data results reported in this paper can help to give an order of the additional gain in compression efficiency resulting from the implementation of such encoder-only optimizations.

REFERENCES

1. Segall, C. and Katsaggelos, A., 2000. Pre- and post-processing algorithms for compressed video enhancement, Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, pp. 1369 to 1373, October. 2000.
2. Information technology — Coded representation of immersive media — Part 3: Versatile video coding. International Organization for Standardization, February 2021.
3. Schwarz, H. Nguyen, T. Marpe, D. and Wiegand. 2018. T. CE7: Transform Coefficient Coding and Dependent Quantization (Tests 7.1.2, 7.2.1), JVET-K0071, Ljubljana, July 2018.
4. Wang, Zhou, Simoncelli, Eero and Bovik, Alan. 2003. Multi-Scale Structural Similarity for Image Quality Assessment. Conference on Signals, Systems & Computers, November 2003.
5. Zhi Li, Anne Aaron et al. 2016. Toward A Practical Perceptual Video Quality Metric, Netflix TechBlog, June 2016.
6. G.-J. Sullivan and T. Wiegand. Rate-Distortion Optimization for Video Compression. 1998. IEEE Signal Processing Magazine, pages 1755–1764, November 1998

7. Li, B. Zhang, D. Li, H. and Xu, J. 2012. QP determination by lambda value, JCTVC-I0426, Geneva, May 2012.
8. Westen, S.J.P. Lagendijk, R.L. and Biemond, J. 1996. Adaptive spatial noise shaping for DCT based image compression, Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 4, pp. 2124 to 2127, May 1996.
9. Sun, J. et al. 2013. Rate-Distortion Analysis of Dead-Zone Plus Uniform Threshold Scalar Quantization and its Application Part II: Two-Pass VBR Coding for H.264/AVC, IEEE Transactions on Image Processing, vol. 22, No. 1, pp. 215-228, Jan. 2013.
10. Karczewicz, M., Ye, Y. and Chong I. 2008. Rate distortion optimized quantization, ITU-T Q.6/SG16 VCEG, VQEG-AH21, Antalya, Turkey, 2008.
11. Garrett-Glaser, J. 2009. A novel macroblock-tree algorithm for high performance optimization of dependent video coding in h.264/avc, Tech. Rep., 2009.
12. Henot, J. P. Ropert, M. Le Tanou, J. Kypréos, J. and Guionnet, T. 2013. High efficiency video coding (HEVC): Replacing or complementing existing compression standards?, IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), London, pp. 1 to 6, 2013.
13. Bichon, M. Le Tanou, J. Ropert, M. Hamidouche, W. and Morin, L. 2019. Optimal Adaptive Quantization based on Temporal Distortion Propagation model for HEVC, IEEE Transactions on Image Processing. Vol. 28, Issue 11, pp. 5419 to 5434, November 2019.
14. HEVC HM reference software, [online] Available at: https://vcgit.hhi.fraunhofer.de/jvet/HM.
15. Bossen, F. 2013. Common test conditions and software reference configurations. 2013. Tech. Rep. JCTVC-L1100, January. 2013.
16. x265, [Online]. Available at: https://bitbucket.org/multicoreware/x265
17. ITU-T Rec. P.910. 2008. Subjective Video Quality Assessment Methods for Multimedia, April 2008.