Rate-Distortion Optimized Quantization in Motion JPEG

Tomasz Grajek
Institute of Multimedia Telecommunications
Poznań University of Technology
60-965, Poznań, Poland
tomasz.grajek@put.poznan.pl

Jakub Stankowski
Institute of Multimedia Telecommunications
Poznań University of Technology
60-965, Poznań, Poland
jakub.stankowski@put.poznan.pl

ABSTRACT

Rate-Distortion Optimized Quantization (RDOQ) is an encoding optimization technique that may be applied to any transform-based compression technique preserving bitstream compliance with the standard. In the paper, the application of the RDOQ to Motion JPEG is described and evaluated. The proposed solution includes block-level optimization with picture-level Lagrange multiplier estimation. Performed evaluation results in higher compression ratios as compared to typical Motion JPEG.

Keywords

Rate-Distortion Optimized Quantization (RDOQ), Motion JPEG, video compression, video streaming.

1. INTRODUCTION

In 2023 JPEG [ITU21] will celebrate its 30th anniversary. Despite the fact that many more efficient techniques were developed, just to mention a few: JPEG 2000 [ISO19], JPEG-XR [ITU20], JPEG-XL [ISO22], HEVC Intra (HEIF) [ISO17], AV1 Intra (AVIF) [Con21], it is still very popular and commonly used in many applications and products [Bor21, Hud18, W3T22].

Based on JPEG, Motion JEPG [RFC98] (sometimes abbreviated as MJPEG) was developed to handle video sequences. The general idea is to encode images from the sequence independently using traditional JPEG. The Motion JPEG is very commonly used in non-linear video editing systems allowing native random access to any frame. Motion JPEG is not so efficient as modern video compression techniques, e.g. HEVC [ISO20a] or VVC [ISO20b], however, it is by far a less complex and resources-demanding solution. Therefore, Motion JPEG is still an attractive technique.

The compression efficiency of a given technology strongly depends on the rate control of the encoders [Bea19, Ric02]. Therefore new control techniques may be added to encoders as long as conformance with the standard is preserved.

In literature, many different approaches to optimizing quantization for image or video encoders are described [Cro97, He14, Luo21, Ram94, Saf19, Wan22, Xu18]. Most of the solutions for JPEG assume that new quantization and Huffman tables have to be defined and transmitted to the decoder. However, such solutions cannot be applied to Motion JPEG as according to RFC 2435 [RFC98] does not allow transmitting custom quantization and Huffman tables.

Rate-distortion optimized quantization (RDOQ) [Kar08] is a non-normative technique allowing for compression efficiency increase by additional analysis of quantized transform coefficients before entropy encoding.

The authors of this paper in [Sta15] presented an extensive analysis of RDOQ application to HEVC.

The paper presents the adaptation and implementation of the RDOQ technique from [Sta15] in Motion JPEG.

2. Motion JPEG

Motion JPEG is a technique that uses a subset of JPEG [ITU21] to compress video sequences by independent coding of consecutive frames. JPEG [Pen93] is a very simple and straightforward approach utilizing transform coding (see Fig. 1).

Figure 1. JPEG compression scheme. O - input data, T - transform coefficients, Q - quantized transform coefficients, S - quantized transform coefficients after zig-zag scanning, b - bitstream.

https://www.doi.org/10.24132/CSRN.3201.29

An encoded image is divided into 8x8 blocks. Then each block is independently processed in raster scan order. Firstly discrete cosine transformation is performed. Then quantization to transformed coefficients is applied. The strength of the quantization is adjusted by the scaling factor. The quantized transform coefficients are rearranged from an 8x8 matrix into a 64-element vector using a zig-zag scan algorithm. Finally, entropy encoding is performed. The entropy coding is mostly a Huffman coding, although an arithmetic one is also available. Prediction is applied only to DC coefficients. The AC coefficients contain direct image data. It should be stressed here, that in modern techniques a prediction error is transformed and quantized. Because there is no prediction in JPEG (besides prediction of DC coefficients) a huge number of non-zero quantized transform coefficients have to be entropy encoded. Therefore, bitstream (or file) contains almost only quantized transform coefficients.

Quantization in JPEG is defined as a simple division (uniform, scalar quantization) and rounding [Mia99, Pen93]. It is a very fast solution but may be suboptimal.

A huge number of non-zero quantized transform coefficients, especially for weak quantization, leave a space for further optimization, for example by using rate-distortion optimized quantization. As a result compression efficiency may be improved preserving conformance with the standard.

3. PROPOSED TECHNIQUE

In the paper, we propose to apply simplified rate-distortion optimized quantization based on that described in [Sta15] to Motion JPEG. The general scheme of the improved JPEG encoder is presented in Fig. 2.

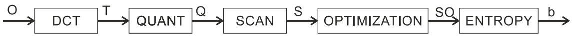

Figure 2. JPEG with RDOQ compression scheme. O - input data, T - transform coefficients, Q - quantized transform coefficients, S - quantized transform coefficients after zig-zag scanning, SO - quantized transform coefficients after RDOQ, b - bitstream.

The details of the proposed RDOQ are the following. After zig-zag scanning of a block of quantized transform coefficients, a vector is derived with DC coefficient at the beginning of the vector and AC coefficients with the highest frequency at the end. Only non-zero transform coefficients are analyzed starting from the last non-zero quantized transform coefficient in a given vector. For each non-zero coefficient, up to four cases are considered, namely: leaving the coefficient without any change, increasing the value of this quantized transform coefficient by 1, decreasing its value by 1, and finally setting it to 0. For each case, the RD cost is calculated using the Lagrange multipliers (λ) approach [Kar08] and the case with the lowest RD cost is chosen as the best one (see Fig. 3).

$$RD\_cost\,(S,\,n) = SSD(S,\,n) + \lambda \cdot R(S,\,n),$$

where:
n – a quantized transform coefficient identifier,
S – a value of the quantized transform coefficient n,
$RD\_cost(S,\,n)$ – the cost of quantization coefficient n to value S,
$SSD(S,\,n)$ – a sum of squared differences between the original and reconstructed block of samples,
$R(S,\,n)$ – number of bits needed to encode block with coefficient n quantized to value S,
λ – Lagrange multiplier.

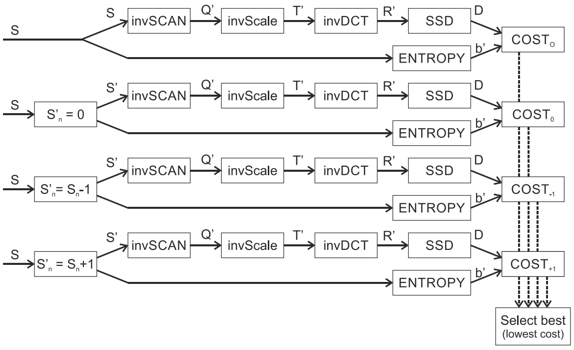

Figure 3. Flowchart of the RDOQ. *S* - quantized transform coefficients after zig-zag scanning, S'_n - modified quantized transform coefficients.

The described procedure is repeated for the rest of the non-zero quantized transform coefficients in the vector towards the beginning of the vector. DC coefficient is excluded from the analysis as it is predictively encoded between blocks. The reason behind this decision is to allow unrestricted parallelization of RDOQ, as no other data is encoded with dependencies between blocks.

Lagrange multiplier calculation

Application of the Lagrange multiplier optimization requires a calculation of mentioned λ multiplier. In the analyzed case, the λ multiplier expresses how to balance distortions introduced by the given technique and the number of bits needed to encode quantized transform coefficients and establishes the operational tradeoff for the encoder rate-distortion optimization stage. Lower λ values encourage the encoder to prefer bitrate reduction over quality and higher λ values strengthen the importance of quality over bitrate.

In typical applications, the JPEG encoder does not use any optimization as it works with constant quality (constant scaling factor), very often set by the user. The authors are not aware of any general formula to calculate the λ multiplier for JPEG similarly as it was derived for HEVC or VVC. Therefore, the λ multiplier is calculated locally by encoding the given picture with Q and Q-1 values. After each encoding information about SSD and bitrate (R) are gathered. Finally, the λ multiplier is derived as:

$$\lambda = -\frac{SSD_Q - SSD_{Q-1}}{R_Q - R_{Q-1}},$$

where:

SSD_x – the sum of squared differences between the original and reconstructed image for a given scale factor setting (Q or Q-1),
R_x – number of bits needed to encode coefficients from the whole image for a given scale factor setting (Q or Q-1).

The local calculation of λ is performed at the picture level. This allows for the adaptation of the λ parameter to the characteristic of every encoded image.

4. IMPLEMENTATION

In order to evaluate the proposed technique, it was necessary to prepare and test its implementation. Authors considered using one of the existing JPEG/MJPEG implementations, for example, libjpeg-turbo [LJT22] which can be considered the fastest software-based implementation of JPEG encoder and decoder. Unfortunately, the complexity of highly optimized implementations and some design choices made by its authors made the usage of existing JPEG encoders very burdensome. Therefore, the authors decided to develop their own implementation of the encoder.

The implementation created for RDOQ-related experiments is restricted to a subset of features used in Motion JPEG and is highly modular with each processing stage (i.e. transform, quantization, scan, entropy) clearly separated from each other. Some algorithmic techniques used in modern JPEG implementations (like fast integer-based DCT transform and fast reciprocal-based integer quantization) were included to match the behaviour of production-quality JPEG encoders. In order to speed up experiments, some of the encoder stages (transform and quantization) were implemented using vector instructions (SSE4.1, AVX2) [Lom11].

As described in Section 3, the RDOQ optimization step requires the calculation of block distortion (SSD) and the number of bits required to represent the currently processed block. SSD calculation is quite straightforward as it corresponds to decoding of JPEG compressed block of pixels, but with entropy decoding omitted. Therefore, the SSD calculation steps (inverse scan, inverse scaling, and inverse transform) are inherited from the JPEG decoder and followed by distortion calculation.

In the case of calculation of the number of bits required to represent the currently processed block, one can use an already existing Huffman encoder. Nonetheless, this is a very inefficient approach. Since the Huffman encoder in JPEG is designed to create a valid bitstream, most of the work done by the Huffman encoder is useless when this block is used to calculate the number of bits only. To avoid unnecessary computations, the authors developed the so-called Huffman counter module. The Huffman counter is a simplified version of the Huffman encoder and it is responsible for fast and accurate calculation of the number of bits required to represent a block of transform coefficients.

The developed implementation is designed to be easily parallelized, although in this paper authors concentrated on the proposed algorithm and compression efficiency to avoid distracting the reader with parallelization-related details.

As mentioned in section 1, the authors concentrated on the Motion JPEG use case, therefore some of the coding techniques and coding tools are out of the scope of this paper. The optimization of Huffman tables is not studied nor implemented since RFC2435 [RFC98] forces the encoder to use default Huffman tables. Similarly, the JPEG standard [ITU21] includes the possibility to use an arithmetic encoder instead of a Huffman one. However, due to patent issues, the adoption of arithmetic encoding in JPEG is negligible and it is almost impossible to find supporting implementation.

It should be emphasized that all developed implementations of MJPEG encoder (both with and without RDOQ) are fully conformant to JPEG standard [ITU21] which means they produce correct and decodable bitstreams. This bitstream can be transmitted as described in RFC2435 [RFC98] but also could be embedded into JFIF files [ITU11].

5. EVALUATION

Methodology

As mentioned before, the authors of the paper concentrate on the Motion JPEG use case. Therefore, to evaluate the compression efficiency of the proposed solution a wide range of video sequences recommended by MPEG Committee experts of the International Organization for Standardization was used. These video test sequences are commonly used for video compression techniques development and evaluation as they cover a wide range of content characteristics. Experiments were conducted on the following 16 sequences with 1920x1080 resolution: *BQTerrace*, *BasketballDrive*, *Cactus*, *Kimono1*, *ParkScene*, *blue_sky*, *pedestrian_area*, *riverbed*, *rush_hour*, *station2*, *sunflower*, *tennis*, *toys_and_calendar*, *tractor*, *vintage_car*, *walking_couple* (see Fig.4). For evaluation, the first 100 frames from each sequence were used.

Figure 4. Single images selected from video test sequences used in the experiments in order from top-left: *tractor*, *sunflower*, *station2*, *soccer*, *rushhour*, *riverbed*, *pedestrian_area*, *ice*, *harbour*, *crew*, *city*, *bluesky*.

The experimental evaluation was performed for all quality point values (*Q*) in the range 1-100. It is worth noting that in JPEG nomenclature the *Q* parameter means “quality”, while in modern techniques (e.g. AVC, HEVC, or VVC) the *q* or rather *qp* parameter corresponds to the quantization step size. Therefore, *Q*=1 means the strongest quantization, so the lowest possible quality and the highest compression ratio; while *Q*=100 means the weakest quantization thus the highest possible quality and the lowest compression ratio.

During evaluation three coding scenarios were considered:

- 1st – using a base MJPEG encoder without RDOQ;
- 2nd – using an MJPEG encoder with RDOQ enabled only for luminance component;
- 3rd – using an MJPEG encoder with RDOQ enabled for all available components (luminance and chrominances).

Both 2nd and 3rd scenarios were compared against 1st (base). According to [ISO20c], results are presented as Bjøntegaard-Delta bitrate and Bjøntegaard-Delta PSNR. Results are presented for luminance component only (ΔPSNR-Y and ΔBitrate-Y) as well as for all components averaged (ΔPSNR-YCbCr and ΔBitrate-YCbCr) with 6:1:1 weights for Y:Cb:Cr respectively.

Each test sequence was encoded, decoded and PSNR values were calculated. Taking into account that experiments were performed for 16 test sequences, 100 quality points, and 3 scenarios, the total number of test points is equal to 4800 which corresponds to 480000 processed pictures.

Results

In Figures 5 and 6 examples of two test sequences were presented for a useful range of bitrates that guarantee very good quality (above 40dB) of reconstructed data.

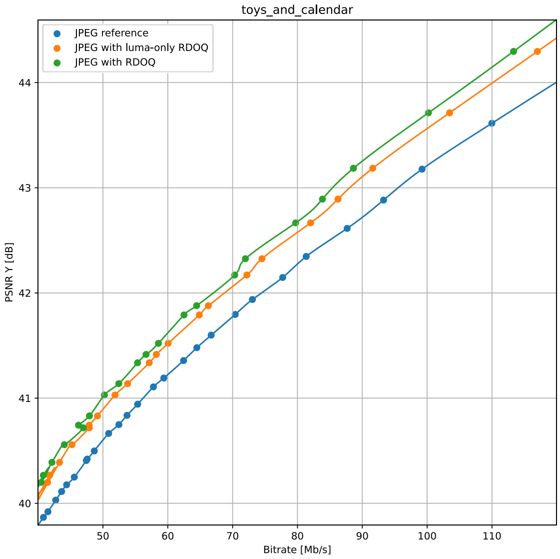

Figure 5. Examples of Rate-distortion curves for the *toys_and_calendar* sequence.

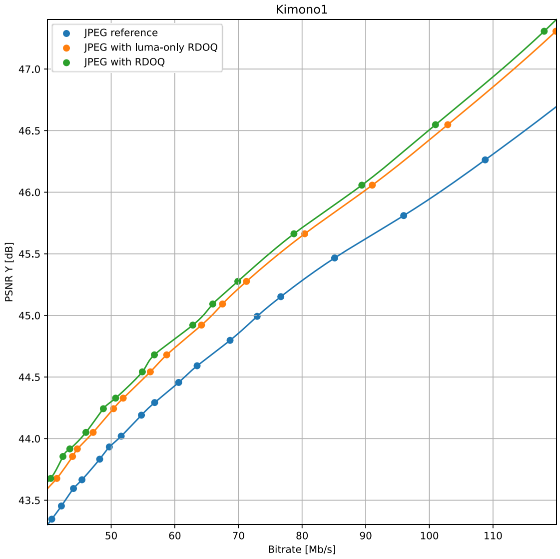

Figure 6. Examples of Rate-distortion curves for the *Kimono1* sequence.

In mentioned figures three R-D curves are presented corresponding to three considered scenarios: using a base MJPEG encoder without RDOQ (blue line), using an MJPEG encoder with RDOQ enabled only for the luminance component (orange line), and using an MJPEG encoder with RDOQ enabled for all available components (green line). It is clear that enabling RDOQ results in compression efficiency. The R-D curves for other test sequences are very similar.

More detailed data are gathered in Tables 1 and 2 where Bjøntegaard-Delta bitrate and Bjøntegaard-Delta PSNR measures are presented for all test sequences. To calculate them four *Q* values were selected i.e. 70, 75, 80, and 85 that correspond to a useful range of bitrates. In Table 1 results for the 2nd scenario are presented whereas in Table 2 results for the 3rd scenario are gathered. Depending on the characteristic of video content 2nd scenario offers from 0.34 to 0.86 dB gain in quality, 0.42dB on average for the same bitrate. On the other hand, scenario 2 results in 6.2 to 12.1% (8.3% on average) bitrate reduction preserving the same quality of reconstructed data. Scenario 3 results in quality gain from 0.40 to 1.10dB, 0.54dB on average for the same bitrate, and from 8.4 to 13.0% (10.38% on average) bitrate reduction.

Sequence	ΔPSNR Y [dB]	ΔPSNR YCbCr [dB]	ΔBitrate Y [%]	ΔBitrate YCbCr [%]
BQTerrace	0.859	0.735	-6.52	-6.80
BasketballDrive	0.480	0.446	-8.52	-8.69
Cactus	0.397	0.368	-7.26	-7.40
Kimono1	0.341	0.307	-10.46	-9.70
ParkScene	0.421	0.393	-6.71	-6.76
blue_sky	0.536	0.561	-9.39	-9.15
pedestrian_area	0.412	0.391	-9.22	-8.97
riverbed	0.348	0.315	-6.54	-6.22
rush_hour	0.390	0.361	-12.16	-11.59
station2	0.383	0.357	-8.55	-8.21
sunflower	0.411	0.393	-10.97	-9.94
tennis	0.384	0.354	-9.38	-9.22
toys_and_calendar	0.344	0.318	-7.85	-7.69
tractor	0.366	0.347	-6.17	-5.82
vintage_car	0.376	0.318	-6.58	-6.62
walking_couple	0.352	0.305	-7.08	-7.07
Average	**0.425**	**0.392**	**-8.337**	**-8.116**

Table 1. Experimental results for luma-only RDOQ (2nd scenario) presented as Bjøntegaard Delta (BD) for bitrate and PSNR.

Sequence	ΔPSNR Y [dB]	ΔPSNR YCbCr [dB]	ΔBitrate Y [%]	ΔBitrate YCbCr [%]
BQTerrace	1.101	0.914	-8.32	-8.37
BasketballDrive	0.604	0.543	-10.53	-10.36
Cactus	0.534	0.471	-9.64	-9.36
Kimono1	0.405	0.371	-12.42	-11.56
ParkScene	0.582	0.510	-9.14	-8.68
blue_sky	0.683	0.656	-11.82	-10.67
pedestrian_area	0.491	0.463	-10.88	-10.51
riverbed	0.441	0.387	-8.25	-7.61
rush_hour	0.443	0.428	-13.72	-13.53
station2	0.473	0.427	-10.50	-9.76
sunflower	0.488	0.462	-12.98	-11.61
tennis	0.476	0.435	-11.48	-11.15
toys_and_calendar	0.467	0.418	-10.50	-9.95
tractor	0.503	0.447	-8.40	-7.46
vintage_car	0.488	0.413	-8.47	-8.49
walking_couple	0.459	0.396	-9.16	-9.07
Average	**0.540**	**0.484**	**-10.387**	**-9.883**

Table 2. Experimental results for RDOQ performed on both luma and chroma (3rd scenario) presented as Bjøntegaard Delta (BD) for bitrate and PSNR.

6. CONCLUSIONS

In the paper adaptation and implementation of the RDOQ technique to Motion JPEG was presented. The proposed solution includes block-level optimization with picture-level Lagrange multiplier estimation. Moreover, some possible ways to parallelize the RDOQ in Motion JPEG were highlighted. Extensive experiments with a wide range of test video sequences proved that this simple technique offers about 10% bitrate reduction when preserving the same quality of reconstructed videos.

7. ACKNOWLEDGMENTS

The research was supported by the Ministry of Education and Science of the Republic of Poland under the subvention for research SBAD.

8. REFERENCES

[Bea18] Beach, A., Owen, A., “Video Compression Handbook”, 2nd Edition, Peachpit Press, 2018.

[Bor21] Borkowski, D., Janczak-Borkowska, K., “Reduction of JPEG Artifacts using BSDEs”, Computer Science Research Notes, vol. 3101, 29th WSCG, Plzen, Czech Republic, 2021

[Con21] Concolato, C., Klemets, A., (eds.), AV1 Image File Format (AVIF), Alliance for Open

Media, https://aomediacodec.github.io/av1-avif/, 2021.

[Cro97] Crouse, M., Ramchandran, K., “Joint Thresholding and Quantizer Selection for Transform Image Coding: Entropy-Constrained Analysis and Applications to Baseline JPEG”, IEEE Transactions on Image Processing, vol. 6, no. 2, pp. 285-297, 1997.

[He14] He, D., Wang, J., “Rate distortion optimized quantization based on weighted mean squared error for lossy image coding”, 2014 IEEE International Conference on Image Processing (ICIP), pp. 5606-5610, Paris, France, 2014.

[Hud18] Hudson, G., Léger, A., Niss, B., Sebestyén, I., Vaaben, J., “JPEG-1 standard 25 years: past, present, and future reasons for a success”, Journal of Electronic Imaging, vol. 27, no. 4, 2018.

[ISO17] ISO/IEC 23008-12, “Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format”, 2017.

[ISO19] ISO/IEC IS 15444, “Information technology - JPEG 2000 image coding system”, Edition 4, 2019 also ITU-T Rec. T.800, Edition 06/2019. 2019

[ISO20a] ISO/IEC 23008-2, “MPEG-H Part 2, High-Efficiency Video Coding”, Edition 4, 2020, also: ITU-T Rec. H.265 Edition 7.0, 2019

[ISO20b] ISO/IEC 23090-3, “MPEG-I Part 3, Versatile Video Coding”, Edition 1, 2020 also: ITU-T Rec. H.266 Edition 1.0, 2020

[ISO20c] ISO/IEC TR 23002-8, “Working Practices Using Objective Metrics for Evaluation of Video Coding Efficiency Experiments”, 2020.

[ISO22] ISO/IEC 18181:2022, “Information technology - JPEG XL image coding system”, 2022.

[ITU11] Recommendation ITU-T T.871, “Information technology – Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)”, International Telecommunication Union - Standardization Sector (ITU-T), 2011.

[ITU20] Recommendation T.832, “Information technology - JPEG XR image coding system”, International Telecommunication Union - Standardization Sector (ITU-T), 2020.

[ITU21] Recommendation H.222.0, “Information technology - Generic coding of moving pictures and associated audio information: Systems”, International Telecommunication Union - Standardization Sector (ITU-T), 2021

[Kar08] Karczewicz, M., Ye, Y., Chong, I., “Rate Distortion Optimized Quantization”, document ITU-T SG16 Q.6, VCEG-AH21, 2008

[LJT22] libjpeg-turbo software webpage, https://libjpeg-turbo.org/, retrieved 2022.

[Lom11] Lomont, C., “Introduction to Intel:registered: Advanced Vector Extensions”, Intel White Paper, 2011.

[Luo21] Luo, X., Talebi, H., Yang, F., Elad, M., Milanfar, P., “The Rate-Distortion-Accuracy Tradeoff: JPEG Case Study”, 2021 Data Compression Conference (DCC), Snowbird, USA, 2021

[Mia99] Miano, J., “Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP”, Addison-Wesley Professional, 1999.

[Pen93] Pennebaker, W.B., Mitchell, J.L., ”JPEG: Still Image Data Compression Standard (Digital Multimedia Standards S), Springer, 1993.

[Ram94] Ramchandran, K., Vetterli, M., “Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility”, IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 700-704, 1994.

[RFC98] RFC 2435, RTP Payload Format for JPEG-compressed video, 1998.

[Ric02] Richardson, I., “Video Codec Design: Developing Image and Video Compression Systems”, Wiley, 2002

[Saf19] Safonov, I.V., Kurilin, I.V., Rychagov, M.N., Tolstaya, E.V., “Fast Control of JPEG Compression Rate”, In: Document Image Processing for Scanning and Printing. Signals and Communication Technology. Springer, 2019

[Sta15] Stankowski, J., Korzeniewski, C., Domański, M., Grajek, T., „Rate-distortion optimized quantization in HEVC: performance limitations”, 31st Picture Coding Symposium, PCS 2015, pp. 85-89, Cairns, Australia, 2015.

[W3T22] W3Techs, Usage statistics of JPEG for websites, https://w3techs.com/technologies/details/im-jpeg, 2022

[Wan22] Wang, Q., Liu, P., Zhang, L., Cheng, F., Qiu, J., Zhang, X., “Rate–distortion optimal evolutionary algorithm for JPEG quantization with multiple rates”, Knowledge-Based Systems, vol. 244, 108500, 2022.

[Xu18] Xu, M., Canh, T.N., Jeon, B., “Rate-Distortion Optimized Quantization: A Deep Learning Approach”, IEEE High Performance Extreme Computing Conference, Waltham, USA, 2018.