MOTION-GUIDED QUANTIZATION FOR VIDEO TONE MAPPING

Ronan Boitard, Dominique Thoreau

Technicolor
975 Av. des Champs Blancs
35576 Cesson-Sevigne, France
first.last@technicolor.com

Rémi Cozot, Kadi Bouatouch

IRISA
263 Avenue du Général Leclerc
35000 Rennes, France
first.last@irisa.fr

ABSTRACT

Tone Mapping Operators (TMOs) transform High Dynamic Range (HDR) contents to address Low Dynamic Range (LDR) displays. However, before reaching the end-user, these contents are usually compressed using a codec (coder-decoder) for broadcasting or storage purposes. Achieving the best trade-off between rendering and compression efficiency is of prime importance. Any TMO includes a rounding quantization to convert floating point values to integer ones. In this work, we propose to modify this quantization to increase the compression efficiency of the tone mapped content. By using a motion compensation, our technique preserves the rendering intent of the TMO while maximizing the correlations between successive frames. Experimental results show that we can save up to 12% of the total bit-rate as well as an average bit-rate reduction of 8.5% for all the test sequences. We show that our technique can be applied to other applications such as denoising.

***Index Terms*—** HDR Video, Video Tone Mapping, Video Compression, HEVC, Adaptive Quantization.

1. INTRODUCTION

Tone Mapping Operators (TMOs) transform High Dynamic Range (HDR) images or video sequences to a lower dynamic range. Tone mapping still images has been a field of active research over the last decade and several satisfying solutions exist [1, 2]. With the recent developments in the HDR video acquisition field [3], more and more HDR video contents are available. Consequently, video tone mapping has drawn a lot of attention recently [4, 5, 6].

These techniques usually focus on the subjective quality of the tone mapped video contents. However, before reaching the end-user, these contents need to be compressed using a codec (coder-decoder) for broadcasting or storage purposes. By tuning the tone map curve of a TMO, one can increase the quality of the decoded contents [7, 8].

However, the tone map curve is not the only aspect that can influence the compression efficiency. Indeed, the last operation performed by a TMO consists in quantizing floating point values to integer ones. We believe that by making this quantization temporally coherent a higher compression efficiency can be achieved. To this end, we propose to adapt the quantization to increase the temporal correlations between successive frames.

This paper is organized as follows. In section 2, we provide the necessary background to understand the proposed technique. Then, we present our method which adapts the quantization based on a motion estimation/compensation. In section 4, we compare, in terms of compression efficiency, our method with a rounding-based quantization. Then we show how we extend our approach to handle denoising. Before concluding, we provide a detailed explanation on how our technique can be improved to achieve an even higher compression efficiency.

2. BACKGROUND

We present in this paper a technique which consists in adapting the quantization of any TMOs to increase the video compression efficiency. To this end, we first define what we mean by quantization before presenting tone mapping. Then we briefly introduce some key aspects of the ITU-T H.265 / MPEG-H Part 2 'High Efficiency Video Codec' (HEVC) [9].

2.1. Quantization

In this paper, the term quantization refers to converting a floating point value to an integer one. When quantizing a value, one has only three choices: floor ($\lfloor\cdot\rfloor$), ceil ($\lceil\cdot\rceil$) or round ($\lfloor\cdot + 0.5\rfloor$). This quantization is different from the adaptive quantization, usually considered in imagery, that consists in optimizing the distribution of the quantization bins in regard to an image's cumulative distributive function [10]. To the best of our knowledge, no work has been performed on adapting the quantization (the conversion) to an application.

2.2. Tone Mapping

TMOs convert HDR contents to Low Dynamic Range (LDR) ones. In HDR imaging, the pixels represent the physical scene

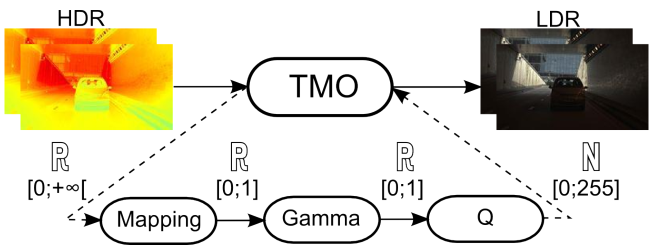

Fig. 1. Workflow of the three steps needed to perform a tone mapping operation.

luminance (expressed in cd/m^2) stored as floating point values. In the case of LDR imaging, the pixels are assigned code values corresponding to a color space standard representation, the most common one being the BT.709.

Fig. 1 illustrates the HDR to LDR conversion. First, the mapping operation, which is the core of a TMO, compresses HDR values to fit in the range [0-1]. As there is a multitude of TMOs and their description is not relevant to this paper, we refer the reader to [1, 2] for further details. Secondly, the gamma encoding redistributes the tonal level closer to how our eyes perceive them (usually $\gamma = 1/2.2$). Finally, the quantization step converts floating point values to integer code values corresponding to the used bit-depth (i.e. $[0; 2^n - 1]$ for n bits). This operation consists in scaling the gamma encoded values to the maximum value desired (i.e. $2^n - 1$) and then rounding them to the nearest integer:

$$\mathbf{I} = Q(\mathbf{I^u}) = \lfloor (2^n - 1)\mathbf{I^u} + 0.5 \rfloor \tag{1}$$

where $Q(\cdot)$ represents the quantization operation, n the used bit-depth and $\lfloor \cdot \rfloor$ the rounding to the nearest lower integer. $\mathbf{I^u}$ (respectively $\mathbf{I}$) represents the unquantized (respectively quantized) gamma encoded image. For the sake of clarity, scaling by a factor of $(2^n - 1)$ will be omitted in the quantization equations. The quantization is performed for each channel of the image $\mathbf{I^u}$ separately, regardless of its representation, i.e. RGB, YUV etc.

When dealing with video sequences, TMOs apply the described operations to each frame separately. As for video TMOs, they deal with temporal coherency only by modifying the mapping operation [4] or by post-processing $\mathbf{I^u}$ [11].

2.3. High Efficiency Video Codec

HEVC is the successor of the ITU-T H.264 / MPEG-4 Part 10 'Advanced Video Coding' (AVC) codec. Developed by the Joint Collaborative Team on Video Coding (JCT-VC) group, it was released in January 2013 and is reported to double AVC compression ratio. The HEVC test Model (HM) is currently in its version 12.

HEVC is a block-based codec that exploits both spatial and temporal correlations between the code values of the pixels to achieve a high compression ratio. To exploit these correlations, blocks are predicted using two processes. Intra-prediction relies on spatial correlation to predict the current block using blocks already decoded in the current frame. Inter-prediction exploits the temporal correlation by predicting the current block using blocks from a set of previous/subsequent decoded frames. The predicted block is then subtracted from the original block leaving only the residuals to be encoded.

As this paper focuses on temporal coherency, a detailed explanation of the inter-prediction is given hereafter. To predict the current block, a block-based motion estimation is performed to find the best temporal prediction. It consists in finding a block that minimizes the distortion with the current block to be encoded. An example of distortion metrics is the Sum of Absolute Differences (SAD) or the Mean Square Error (MSE). This motion estimation is performed for different block-sizes (from 64 to 4 organized in a quad-tree in the HM). The selected motion vector is the one that minimizes a rate-distortion function. The inter-prediction results from performing a motion compensation on a reference frame using the selected motion vectors.

In video compression, it is generally considered that the closest the prediction, the more efficient the compression. In the next section, we present a technique to adapt the quantization at the tone mapping stage so as to improve a codec's inter-prediction.

3. MOTION-GUIDED QUANTIZATION (MGQ)

3.1. Motivations

Before broadcasting it to the end-user, any video content needs to be encoded using a codec. Tone mapped HDR video contents cannot skip this process. By optimizing the tone mapping/compression combination, one could increase the quality of the rendered contents while reducing the broadcaster's bandwidth use. In section 2.2, we mentioned that the quantization performed by a TMO consists in a rounding to the nearest integer (Eq. 1). In the meanwhile, we stated that increasing the quality of a codec's inter-prediction results in higher compression ratios (because the resulting residuals are smaller). From the two above observations, we believe that by adapting the quantization to the prediction process of a codec, its compression efficiency can be increased.

To this end, we propose a quantization method that adapts to any TMO and increases the correlation between successive frames. With this aim in view, the quantization of the gamma encoded current frame to be tone mapped ($\mathbf{F_c^u}$, represented by floating point values) should adapt to the previously tone mapped frame ($\mathbf{F_r}$ which values have been already quantized). Fig. 2 illustrates the tone mapping of two successive frames of an HDR video sequence with and without using our technique. The following section details how our technique,

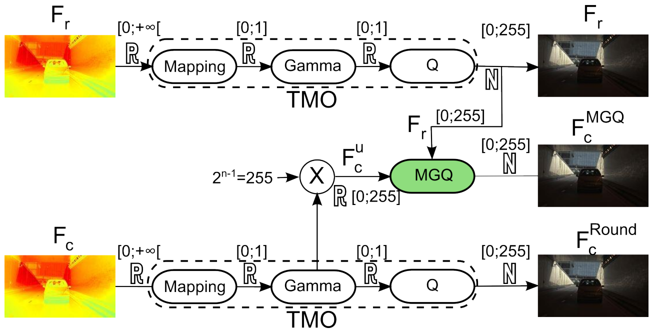

Fig. 2. Tone mapping two consecutive frames with and without applying the MGQ. The range of each input/output as well as the type of data is indicated ($\mathbb{N}$ = uint, $\mathbb{R}$ = float).

called Motion-Guided Quantization (MGQ), adapts the quantization.

3.2. Our Approach

Recall that the aim of our technique is to increase the compression efficiency of a tone mapped content by adapting the quantization operation. As a codec greatly relies on the inter-prediction to remove redundant data, increasing the quality of this prediction should provide a higher compression efficiency. As mentioned in section 2.3, the inter-prediction relies on a motion estimation/compensation operation to remove redundant data between frames of a video sequence. That is why we first perform a Motion Estimation (ME) between $\mathbf{F_r}$ and $\mathbf{F_c^u}$ to obtain, for each pixel location (x, y), a motion vector $(\delta x, \delta y)$. We then compute the Motion Compensation (MC) which provides the inter-predicted frame $\mathbf{F_p}$:

$$\mathbf{F_p}(x, y) = \mathbf{F_r}(x + \delta x, y + \delta y) \tag{2}$$

To be consistent with the prediction process used in HEVC, the motion estimation is only performed on the luma channel and the resulting motion vectors are used for each channel of a YUV frame. Our technique uses the predicted frame $\mathbf{F_p}$ to adapt the quantization of the current frame $\mathbf{F_c^u}$:

$$\mathbf{F_c^{MGQ}} = MGQ(\mathbf{F_c^u}) = \begin{cases} \lfloor \mathbf{F_c^u} \rfloor & \text{if } \mathbf{F_c^u} - \mathbf{F_p} \geq 0 \\ \lceil \mathbf{F_c^u} \rceil & \text{if } \mathbf{F_c^u} - \mathbf{F_p} < 0 \end{cases} \tag{3}$$

where $MGQ(\cdot)$ represents the Motion-Guided Quantization operation while $\lfloor \cdot \rfloor$ (respectively $\lceil \cdot \rceil$) represents the rounding to the nearest lower (respectively higher) integer. Recall that $\mathbf{F_c^u}$ is expressed with floating point values while $\mathbf{F_p}$ with integer ones. Both frame's values range from 0 to $2^n - 1$. As in section 2.2, the MGQ is applied to each channel of the frame $\mathbf{F_c^u}$ separately. The workflow of the MGQ technique is illustrated in Fig. 3. Our method efficiently increases the quality of the inter-prediction by reducing the distortion between the predicted frame $\mathbf{F_p}$ and the current frame $\mathbf{F_c^u}$.

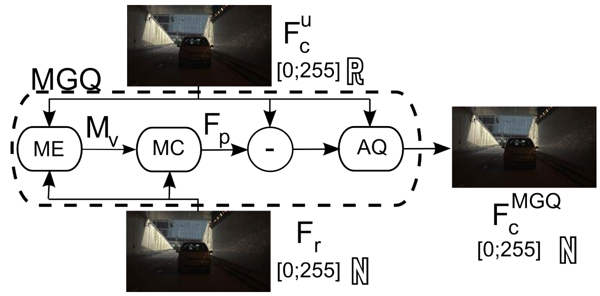

Fig. 3. Details on the MGQ. AQ stands for Adaptive Quantization (Eq. 3 or Eq. 4). $\mathbf{F_r}$ is the tone mapped reference frame, $\mathbf{F_c^u}$ the tone mapped current frame before quantization. The MGQ provides the quantized tone mapped frame $\mathbf{F_c^{MGQ}}$.

However, with our technique the distortion between $\mathbf{F_c^u}$ and $\mathbf{F_c^{MCQ}}$ is always higher than or equal to the rounding quantization. To tune the trade-off between the quantization distortion and inter-prediction efficiency, we add a parameter δ that enables our technique to adapt to the difference between $\mathbf{F_p}$ and $\mathbf{F_c^u}$:

$$\mathbf{F_c^{MGQ}} = \begin{cases} \lfloor \mathbf{F_c^u} \rfloor & \text{if} \quad 0 \leq \mathbf{F_c^u} - \mathbf{F_p} < \delta \\ \lceil \mathbf{F_c^u} \rceil & \text{if } -\delta < \mathbf{F_c^u} - \mathbf{F_p} < 0 \\ \lfloor \mathbf{F_c^u} + 0.5 \rfloor & \text{otherwise} \end{cases} \tag{4}$$

To better understand the way this trade-off behaves, let us consider three cases: $\delta = 0$, $\delta = 1$ and $\delta = \infty$. When $\delta = 0$, the MGQ behaves as a rounding quantization while for $\delta = \infty$ it corresponds to Eq. 3. For $\delta = 1$, we define Ω as the set of pixels to which the MGQ has been applied, say those that satisfy $\|\mathbf{F_p} - \mathbf{F_c^u}\| < \delta$. After applying the quantization, we obtain $\mathbf{F_c^{MGQ}}(\boldsymbol{\Omega}) = \mathbf{F_p}(\boldsymbol{\Omega})$ since the distortion was lower than δ (i.e. 1). Consequently, when predicting $\mathbf{F_c^{MGQ}}(\boldsymbol{\Omega})$ using $\mathbf{F_p}(\boldsymbol{\Omega})$ the resulting residuals are equal to 0. All the other pixels are quantized using the rounding operation. Table 1 illustrates different quantizations corresponding to different pixels conditions. Table 2 summarizes the trade-off between the distortion to the original unquantized values $\mathbf{F_c^u}$ and that of the predicted values $\mathbf{F_p}$.

To sum up, fixing δ allows the user to balance the number of pixels quantized using the MGQ or the rounding, based on the distortion between the unquantized values and the prediction. A higher δ means a higher distortion between $\mathbf{F_c^{MGQ}}$ and $\mathbf{F_c^u}$ as well as a higher quality of the prediction $\mathbf{F_p}$, thereby reducing the amount of residuals to encode.

4. RESULTS

Our technique aims at increasing the compression efficiency while adapting to any TMO without altering its intent. In this section, we show that the distortion obtained with our technique and the rounding quantization are very close. Finally,

Table 1. Example of the different quantization techniques. $\mathbf{F_c^u}$ is the unquatized tone mapped frame, $\mathbf{F_p}$ the predicted frame and $\mathbf{F_{MGQ}^c}$ the current tone mapped frame quantized using different values of δ (cf. Eq. 4).

$\mathbf{F_c^u}$	7.2	30.2	67.8	130.7	236.3
$\mathbf{F_p}$	8	28	67	127	238
$\mathbf{F_c^u} - \mathbf{F_p}$	-0.8	2.2	0.8	3.7	-1.7
$\mathbf{F_c^{MGQ}}, \delta = 0$	7	30	68	131	236
$\mathbf{F_c^{MGQ}}, \delta = 1$	8	30	67	131	236
$\mathbf{F_c^{MGQ}}, \delta = \infty$	8	30	67	130	237

Table 2. Sum of distortion resulting from the different quantization of table 1.

	$\Vert\mathbf{F_c^u} - \mathbf{F_c^{MGQ}}\Vert$	$\Vert\mathbf{F_c^{MGQ}} - \mathbf{F_p}\Vert$
$\mathbf{F_c^{MGQ}}, \delta = 0$	1.2	10
$\mathbf{F_c^{MGQ}}, \delta = 1$	2.4	8
$\mathbf{F_c^{MGQ}}, \delta = \infty$	3.2	6

Table 3. PSNR in dB by quantizing with and w/o using our technique (59 dB correspond to a MSE of 0.081).

Quantization	Sun	Tunnel	Students	TunnelHD
Rounding	58.93	58.91	58.92	58.92
MGQ_1	56.65	56.74	57.13	56.75
MGQ_{Inf}	55.68	55.53	56.51	56.15

we report the compression efficiency of tone mapped contents with and without using our technique.

4.1. Quantization Loss

Integer quantization assigns several floating point values to the same integer. This process obviously results in a loss of information in the quantized signal. We assess the loss due to the quantization by computing the Peak Signal to Noise Ratio (PSNR) between the unquantized current frame $\mathbf{F_c^u}$ and the quantized one $\mathbf{F_c}$. Table 3 reports the PSNR using three different quantizations: Rounding, MGQ with $\delta = 1$ (noted MGQ_1) and MGQ with $\delta = \infty$ (named MGQ_{Inf}). As mentioned before, our quantization technique provides a slightly more distorted sequence than the rounding quantization. This distortion is no greater than one code value for all the quantized pixels (contrary to the rounding quantization that entails a maximum distortion of half a code value). For comparison, the distortion due to a lossy codec is always greater than or equal to one code value.

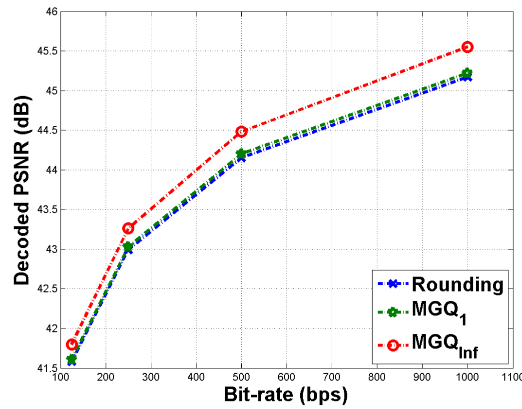

Fig. 4. Rate-distortion results for the Sun sequence.

4.2. Compression Efficiency

For our experiments on compression efficiency, we used the HM 12.0 with the Random Access Main Profile. To assess the compression efficiency, one usually compares the PSNR between the input video and its decoded counterpart. This comparison can be performed in two different ways.

First, each input video is encoded at targeted bit-rates. A direct comparison of the PSNR allows to assess the increased quality of the content for these bit-rates. Fig. 4 and 5 plot the results with and without using the MGQ quantization. The two sequences used (Sun and Tunnel [12]) are of VGA resolution (640x480) while the targeted bit-rates are 125, 250, 500 and 1000 kbps. We used Ramsey et al. TMO [13] to tone map both HDR sequences. Results show that we achieve a higher quality of reconstruction (between 0.15 dB and 0.4 dB gain) at the decoding stage using the MGQ_{Inf}. We can also notice that the higher the bit-rate, the higher the gain. The case MGQ_1 provides only a small improvement over the rounding operation. The trade-off between distortion and compression efficiency is illustrated through Table 3 and Fig. 4 and 5. Note that by tuning the δ parameter, one can shift the MGQ_δ curve from the Rounding to the MGQ_{Inf} curve.

The second technique computes the average percentile bit-rate reduction under the same PSNR. Table 4 reports the Bjontegaard Distortion rate (BD-rate) [14] for the tested video sequences. The sequence TunnelHD is of HD resolution (1920x1080) and has been also tone mapped using Ramsey et al. TMO [13]. The Students sequence [6] however is of resolution 1280x720 and has been tone mapped using Farbman et al. TMO [15]. The results show that for the same quality, the MGQ_{Inf} provides an average bit-rate reduction of 8.5% for all the test sequences. Note that the two VGA sequences perform better than the two other ones. This is due to the fact that these sequences are relatively noisy and our quantization technique reduces some of the temporal noise. In the next section, we show that our technique can be extended to other applications such as denoising.

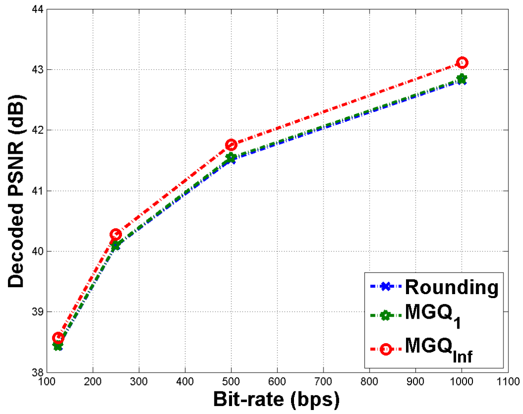

Fig. 5. Rate-distortion results for the Tunnel sequence.

Table 4. Average percentile bit-rate reduction under the same PSNR when comparing the Rounding and the MGQ_{Inf} quantization techniques. The BD-rate is computed using piece-wise cubic interpolation.

Sequence	Y	U	V
Sun	-12.8%	-40.1%	-40.6%
Tunnel	-10.4%	-31.7%	-32.7%
Students	-5.6%	-18.8%	-17.5%
TunnelHD	-5.4%	-21.7%	-24.5%
Average	-8.5%	-28.1%	-28.8%

5. DENOISING APPLICATION

As mentioned above, our method performs better for the Sun and Tunnel sequences because it reduces the temporal noise. When compression is not the targeted applications, our method can reduce the noise present in a tone mapped video sequences. Indeed, in the previous section, the MGQ was guided by the value of the difference between $\mathbf{F_c^u} - \mathbf{F_p}$. Instead of adapting to the inter-predicted frame $\mathbf{F_p}$, we adapt our quantization to a denoised frame $\mathbf{F_d}$. The way $\mathbf{F_d}$ is computed is not relevant to this paper and any existing denoising techniques can be used [16]. For our experiments, we will consider a simple temporal-filtering with motion compensation:

$$\mathbf{F_d^k}(x,y) = \sum_{l=-N}^{M} \frac{\mathbf{F^{k-l}}(x - \delta x^{k,k-l}, y - \delta y^{k,k-l})}{w(l)} \quad (5)$$

where $(\delta x^{k,k-l}, \delta y^{k,k-l})$ is a motion vector obtained through a motion estimation between frames $\mathbf{F^{k-l}}$ and $\mathbf{F^k}$. N (respectively M) represents the number of non-causal (respectively causal) extents of the averaging window and $w(l)$ are the weights or the filter coefficients. Note that causal frames are expressed with integer values while non-causal ones with floating point values (including the current one which is in our case $\mathbf{F_c^u}$).

Table 5. Average PSNR between a denoised version of a tone mapped video sequence and this sequence quantized either with a rounding quantization or the MGQ.

Quantization	Sun	Tunnel	Students	TunnelHD
Rounding	48.52	45.58	47.78	49.15
MGQ_{Inf}	49.59	46.56	48.98	50.58

For our experiments we used only two frames in the filter bank: the previous one $\mathbf{F_r}$ and the current one $\mathbf{F_c^u}$. We tested our method on the same set of sequences and TMOs as in section 4.2. To assess the performance of our method when compared to the rounding quantization, we compute the PSNR between the quantized frame (either $\mathbf{F_c}$ or $\mathbf{F_c^{MGQ}}$) and the desired denoised frame $\mathbf{F_d}$. We report those PSNR in Table 5. For all test sequences, we achieve at least 1 dB of gain using the MGQ technique when compared to the rounding technique. The main advantage of using our technique rather than performing a denoising after the tone mapping resides in the fact that our technique do not introduce additional artifacts to the sequence. Indeed, denoising usually results in a smoothing which is source of problems when performed on edges. However, our technique fails for really noisy sequences.

In a more generalized manner, we believe that our method can adapt to any application where tone mapping is required provided that the right test value is chosen (e.g. $\mathbf{F_p}$ for compression and $\mathbf{F_d}$ for denoising).

6. FUTURE WORK

Our method has some limitations, the main one being its computational complexity due to the motion estimation. Furthermore, the proposed implementation is sub-optimum with respect to the prediction process performed in HEVC. Because, first our technique does not follow the Group Of Pictures (GOP) hierarchical pattern that is used in a codec. Indeed, it only uses motion compensation between successive frames. Second, the intra-prediction process should also benefit from our quantization. Third, a block-based codec uses a rate-distortion cost function to select the best predictor for each block while our solution only relies on a distortion metrics.

To sum up, the MGQ technique, if implemented in the coding loop instead of being a pre-processing, should provide an even higher compression efficiency. It would allow to tune the quantization separately for each of the available prediction mode. The selected mode and its associated quantization would depend on the codec's rate-distortion function rather than solely on the distortion. Regarding the trade-off parameter δ, it could be linked to the rate-distortion function to achieve a higher compression efficiency while reducing the quantization distortion. Furthermore, the computational complexity would no longer be an issue as the motion estimation is already performed for the inter-frame prediction. However, to implement our approach inside the coding loop of a block-based encoder (say HEVC), the computations within the codec should be performed with floating point values rather than integer. Finally, addressing more applications such as color reproduction or tracking should be investigated.

7. CONCLUSION

In this paper, we pointed out that, when performing tone mapping, rounding quantization is not efficient. We chose a quantization that aimed at improving the compression efficiency of tone mapped video contents. Our technique relies on the motion compensation between two successive frames of a sequence to adapt the quantization during the tone mapping. Results showed an average bit-rate reduction under the same PSNR ranging from 5.4% to 12.8%. The proposed method allows a trade-off between compression efficiency and quantization distortion of the original video.

We also applied our technique to denoising to show that our method can be generalized to deal with any applications where tone mapping is needed.

Acknowledgment

The TunnelHD sequence was produced by Binocle and Technicolor within the framework of the french collaborative project NEVEx.

8. REFERENCES

[1] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High Dynamic Range Imaging, 2nd Edition: Acquisition, Display, and Image-Based Lighting*, 2010.

[2] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging: Theory and Practice*, AK Peters (CRC Press), Natick, MA, USA, 2011.

[3] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A versatile HDR video production system," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 1, 2011.

[4] R. Boitard, D. Thoreau, K. Bouatouch, and R. Cozot, "Temporal Coherency in Video Tone Mapping , a Survey," in *HDRi2013 - First International Conference and SME Workshop on HDR imaging*, 2013.

[5] R. Boitard, R. Cozot, D. Thoreau, and K. Bouatouch, "Survey of temporal brightness artifacts in video tone mapping," in *HDRi2014 - Second International Conference and SME Workshop on HDR imaging*, 2014.

[6] G. Eilertsen, R. Wanat, R. Mantiuk, and J. Unger, "Evaluation of Tone Mapping Operators for HDR-Video," *Computer Grahpics Forum*, vol. 32, no. 7, pp. 275–284, 2013, Students sequence available at http://www.hdrv.org.

[7] Z. Mai, H. Mansour, R. Mantiuk, P. Nasiopoulos, R. Ward, and W. Heidrich, "Optimizing a tone curve for backward-compatible high dynamic range image and video compression.," *IEEE Transactions on Image Processing (TIP)*, vol. 20, no. 6, pp. 1558–1571, 2011.

[8] R. Boitard, D. Thoreau, R. Cozot, and K. Bouatouch, "Impact of Temporal Coherence-Based Tone Mapping on Video Compression," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, 2013.

[9] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[10] L. Yang and D. Wu, "Adaptive quantization using piecewise companding and scaling for Gaussian mixture," *J. Vis. Comun. Image Represent.*, vol. 23, no. 7, pp. 959–971, 2012.

[11] R. Boitard, R. Cozot, D. Thoreau, and K. Bouatouch, "Zonal brightness coherency for video tone mapping," *Signal Processing: Image Communication*, vol. 29, no. 2, pp. 229–246, 2014.

[12] G. Krawczyk, ," Sun and Tunnel sequences available at http://www.mpi-inf.mpg.de/resources/hdr/video/.

[13] S. Ramsey, J. J. III, and C. Hansen, "Adaptive temporal tone mapping," *Computer Graphics and Imaging - 2004*, , no. 3, pp. 3–7, 2004.

[14] G. Bjontegaard, "Improvement of the bd-psnr model," ITU-T SG16, VCEG-AI11, 2008.

[15] Z. Farbman and D. Lischinski, "Tonal stabilization of video," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 1, 2011.

[16] J. Brailean, R. Kleihorst, S. Efstratiadis, A. Katsaggelos, and R. Lagendijk, "Noise reduction filters for dynamic image sequences: a review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, 1995.