Adaptive Re-Quantization For High Dynamic Range Video Compression

Mikael Le Pendu, Christine Guillemot, Dominique Thoreau

ADAPTIVE RE-QUANTIZATION FOR HIGH DYNAMIC RANGE VIDEO COMPRESSION

Mikaël Le Pendu [1,2]*, Christine Guillemot* [1] *and Dominique Thoreau* [2]

[1] *INRIA*
Campus de Beaulieu, 35042 Rennes Cedex France
firstname.lastname@irisa.fr

[2] *Technicolor*
Av. des Champs Blancs, 35576 Cesson-Sévigné France
firstname.lastname@technicolor.com

ABSTRACT

High Dynamic Range (HDR) images contain more intensity levels than traditional image formats. Instead of 8 or 10 bit integers, floating point values are generally used to represent the pixel data. To extend the use of existing video codecs such as HEVC to HDR floating point video sequences, we propose a method that converts the floating point data and reduces the bit depth of input images with minimal loss. Several variants of the method are proposed. They are adapted to different quality requirements. In particular, near lossless compression is addressed.

Index Terms— High Dynamic Range (HDR), HEVC, OpenExr, Floating point, Re-quantization

1. INTRODUCTION

In HDR images, the pixel values are often stored as floating point numbers that have more precision and can cover a much higher dynamic range than the 8 or 10 bit integers used in traditional images. For instance, OpenExr [1] defines the half float data format with a bit depth of 16 bits including one sign bit, five exponent bits and 10 mantissa bits. This format is widely used in the visual effects industry. It has received even more interest recently since the development of the ACES workflow for cinema production [2]. Another popular floating point representation is the RGBE Radiance format [3], where 8 bit mantissas are stored for R, G and B values and an additional 8 bit channel represents a common exponent.

Although some simple compression algorithms such as PIZ [4], implemented in OpenExr, can encode directly the floating point data, the most advanced video coding standards are designed for integer-valued images. Another limitation of the existing standards is the relatively low bit depth they support. A conversion resulting in loss of precision is thus necessary before compression.

However, in their extended versions, the coding standards H264/AVC [5] and more recently HEVC [6] can take up to 14 bit per channel input images. Several attempts have been made to encode HDR content using the high bit depth versions of the standard after a minor reduction of the data precision. For instance, [7] presents a modified LogLuv transform where the minimum and maximum luminances of the frames are used to map the floating point numbers to 14 bit integers for the luma channel. The resulting frame adapted values are then encoded using H264/AVC compression. In [8], the performance gain of HEVC over its predecessor H264/AVC is studied for sequences of floating point images previously converted to 10 and 12 bits per component.

Several other HDR compression methods such as [9], [10] and [11] consist in encoding a base LDR layer and combining it with a residual HDR enhancement layer. With this approach, each layer can be encoded at low bit depth using a classical compression scheme.

In this article, we study the impact on compression of several bit depth reduction techniques used before HEVC encoding. Our approach is based on OpenExr's half float format that has the advantage of being much lighter than single or double precision floating point numbers while remaining sufficient for storing HDR images. A significant improvement was observed compared to the Adaptive LogLuv transform. For high levels of PSNR, 52% bitrate saving was obtained.

2. COMPRESSION METHOD

Our coding method is based on the OpenExr image format. It is a very flexible format which supports 32 bit integer, 32 bit floating point as well as 16 bit floating point (i.e. half float) data representations. In this article, we suppose input images are represented as half float RGB channels which is the most common case in practice for HDR imagery. A half float number f is defined from an exponent e, a mantissa m and a sign s according to equation (1).

$$f = \begin{cases} (-1)^s \cdot 2^{e-25} \cdot (1024 + m), & \text{if } e > 1 \\ (-1)^s \cdot 2^{-24} \cdot m, & \text{otherwise} \end{cases} \quad (1)$$

Where s is 0 or 1, e is a 5 bit integer between 0 and 30 and m is a 10 bit integer ranging from 0 to 1023. The value 31 for the exponent exists but is reserved for special values such as Not a Number (NaN) or Infinities.

The encoding of the sign bit is beyond the scope of this article. Hence, the input images are supposed to contain only positive values.

2.1. General coding scheme

The general compression scheme is shown in figure 1. Since HEVC can only encode integer values, a conversion from floating point to integer is performed first. As mentioned in [12],

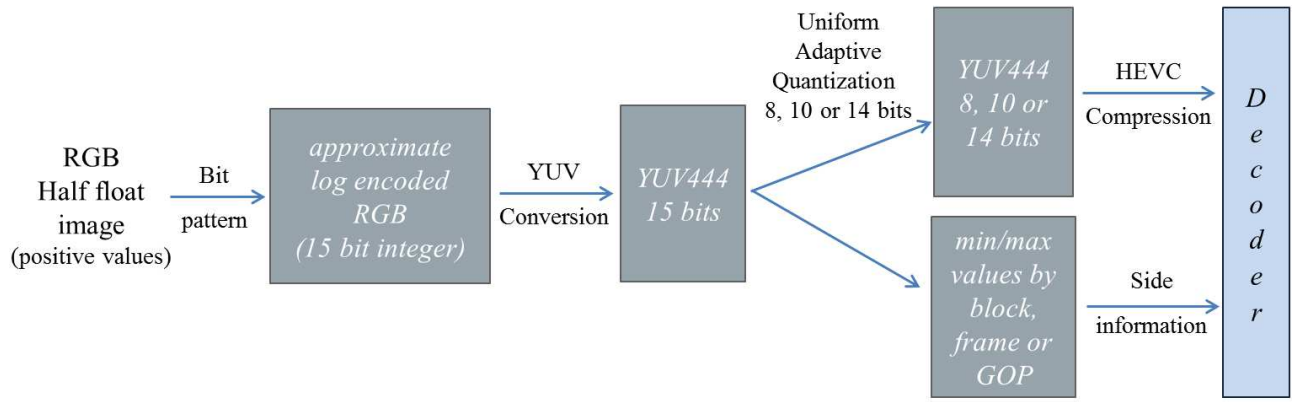


Fig. 1. Our general coding scheme for floating point HDR image encoding.

taking the bit pattern of a positive floating point value gives a piecewise linear approximation of the logarithm function. In the case of positive half float numbers, the integer representation of the bit pattern is given by $i = 2^{10}.e + m$, thus, $e = (i - m).2^{-10}$. According to equation (1) in the case $e > 0$, we have :

$$f = 2^{(i-m)\cdot 2^{-10}-25} \cdot (1024 + m),$$

$$f = 2^{i\cdot 2^{-10}-25} \cdot \frac{1024 + m}{2^{m\cdot 2^{-10}}},$$

$$f = 2^{i\cdot 2^{-10}-25} \cdot 1024 \cdot \frac{1 + m \cdot 2^{-10}}{2^{m\cdot 2^{-10}}}.$$

Knowing that for $0 \leq m < 2^{10}$:

$$1 \leq \frac{1 + m \cdot 2^{-10}}{2^{m\cdot 2^{-10}}} \leq 1.0615,$$

the following approximation can be derived :

$$f \approx 2^{i\cdot 2^{-10}-15}$$

It follows :

$$i \approx 2^{10} \cdot (log_2(f) + 15) \tag{2}$$

This approximate logarithmic encoding has the advantage that it does not require any computation since it is the way OpenExr's half float values are stored internally. It is also advantageous in lossless or near lossless compression and has been used in this context, for example, in [11] or in the PIZ algorithm from OpenExr. Moreover, a logarithmic encoding of luminance values is in accordance to Weber law of perceptual uniformity.

The resulting 15 bit integer RGB values are then converted to YCbCr as described in the next subsection. Then, the re-quantization method presented in subsection 2.3 is applied to reduce the bit depth before HEVC compression. This method is declined in three versions detailed in subsection 2.4. The method requires an additional side information on local minimum and maximum values to be transmitted to the decoder. Subsection 2.5 describes the encoding of those additional values.

2.2. YCbCr conversion

Equations (3) to (5) show the forward transformation from RGB to YCbCr. sRGB chromaticities are used to compute the Y channel. As mentioned earlier, the exponent value 31 is used to store Infinities or NaN. The color images to encode are not supposed to contain such values. We then consider that the maximum value for the 15 bit integer i is reached for an exponent equal to 30 and a mantissa equal to 1023. Thus, i has a maximum value of 31743, instead of $2^{15} - 1$. In order to use the whole range of the 15 bits and reduce rounding errors caused by the conversion, a scale factor w is applied to the original RGB values. Floating point calculations are used in the equations but after the conversion, the Y, Cb and Cr values are rounded to keep 15 bit integers.

$$Y = w(0.2126R + 0.7152G + 0.0722B), \tag{3}$$

$$Cb = \frac{wB - Y}{1.8556} + \frac{2^{15} - 1}{2}, \tag{4}$$

$$Cr = \frac{wR - Y}{1.5748} + \frac{2^{15} - 1}{2}, \tag{5}$$

with $w = \frac{2^{15}-1}{31743}$.

2.3. Re-quantization algorithm

After the color conversion, an adaptive uniform re-quantization is applied independently to the 15 bit Y, Cb and Cr channels to reduce their bit depth. Given a target bit depth n, the formulas (6) and (7) apply respectively for the forward and backward operation :

$$x' = x - x_{min}, \qquad \text{if } x_{max} - x_{min} \leq 2^n - 1 \tag{6a}$$

$$x' = \frac{(x - x_{min}) \cdot (2^n - 1)}{x_{max} - x_{min}}, \qquad \text{otherwise} \tag{6b}$$

$$x_{dec} = x' + x_{min}, \qquad \text{if } x_{max} - x_{min} \leq 2^n - 1 \tag{7a}$$

$$x_{dec} = \frac{x' \cdot (x_{max} - x_{min})}{2^n - 1} + x_{min}, \qquad \text{otherwise} \tag{7b}$$

Where x_{min} and x_{max} are respectively the minimum and maximum values in the region containing the pixel. The encoded and decoded values are denoted x' and x_{dec}.

Similarly to [7], the minimum and maximum values are used to adapt the mapping to the data. However, we distinguish two cases. When the dynamic of the data is superior to that of the target bit depth, it is necessary to downscale the values. Otherwise,

when the data has a sufficiently low dynamic to fit into the target bit depth, no rescaling is applied. If the same formula were used in this case, the data would be unnecessarily upscaled.

2.4. Block, frame and GOP wise variants

Three variants of the method are proposed using either a block 16x16, a frame, or a Group Of Pictures (GOP) as the basic unit. As shown in figure 1, minimum and maximum values of all the sequence's regions (i.e. blocks, frames or GOPs) have to be transmitted to the decoder to be able to perform the inverse quantization.

The advantage of the block-wise method is that the dynamic is more likely to be small in a block than in the whole image or in a GOP. As a result, the re-quantization is less severe for low target bit depths. Moreover, if the input image contains some pixels with extremely high or low values, only the blocks containing those pixels will be affected. In counterpart, the coherence between blocks is not preserved. This property is not well suited to the HEVC encoder which tries to estimate blocks of pixels by a prediction based on spatial or temporal neighboring pixels. The discontinuities between the re-quantized blocks will bias those predictions and thus degrade the compression performance. In addition, more minimum and maximum data must be encoded along with the HEVC bit stream. The frame-wise variant is better to keep the effectiveness of intra predictions, but does not preserve temporal coherence.

In order to improve the performance of temporal predictions, a GOP-wise method is also proposed. In this method, the first and last frames of the GOP are intra coded and are used as reference pictures in a hierarchical coding scheme for the prediction of the frames in-between. The last frame of a GOP corresponds to the first frame of the next one so that only one I pictures is needed at the transition. However, as the re-quantization applied to each GOP may be different, using directly the last frame of a GOP as a reference frame for the next one would break the temporal coherence. Instead, the first GOP is coded, decoded and rescaled to its original dynamic. The last decoded frame is then re-quantized with the minimum and maximum values of the next GOP before being used as a reference. This process is illustrated in figure 2. Note that for the experiment, the coding cost of the first frame in the second GOP is not taken into account for the calculation of the total bit rate because it was already encoded in the previous one and did not need to be re-encoded in our scheme.

2.5. Encoding of the quantization parameters

The information of minimum and maximum values is necessary to perform the inverse quantization and must be transmitted to the decoder. In the frame and GOP-wise methods, the number of bits needed to store those two values is negligible compared to the total coding cost. But it is not the case for the block-wise method. A simple way to reduce the bit length of the quantization parameters was used here. For the minimum value, the 15 bits are directly transmitted for each block. Then the difference $delta$ between the maximum and minimum is computed. According to

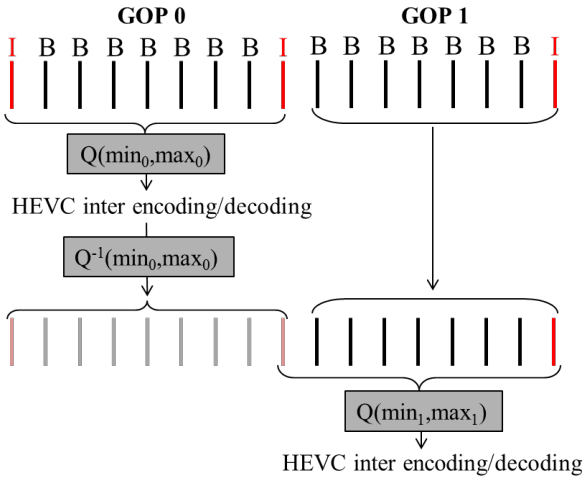


Fig. 2. The process used for the GOP-wise method. This method keeps the temporal coherence inside a GOP and between two GOPs to better exploit temporal predictions. The quantization and inverse quantization steps are referred to as Q and Q-1 respectively.

equation (7), if $delta < 2^n - 1$ (where n is the target bitdepth), the exact value of delta is not needed. In this case, only the $15-n$ most significant bits of delta are transmitted. They are all equal to zero. Otherwise at least one of these bits is non-zero and we transmit all the bits. On the decoder side, we first read the $15-n$ most significant bits, if they are all zeros, equation (7a) applies, otherwise the remaining n bits are read and equation (7b) applies.

3. RESULTS

3.1. Testing conditions

The tests were performed on 17 frames of the 1920x1080 sequence "Tunnel video clip" shown in figure 3. This sequence was produced by Binocle and Technicolor within the framework of the french collaborative project NEVEx.

For HEVC encoding, we used the HM range extension in [13] which enables YUV 4:4:4 input because no subsampling was performed on the chroma channels. Two GOPs of size 8 were used for the tests with inter predictions. All the tests were performed with Rate Distortion Optimized Quantization (RDOQ) enabled but without Sample Adaptive Offset (SAO) and Deblocking filter. Each curve is constructed by encoding the

Fig. 3. HDR sequence "Tunnel video clip".

Method	8 bits	10 bits	12 bits	14 bits
GOP-wise	65.32	77.27	87.68	95.38
Frame-wise	66.87	78.75	89.98	95.38
Block-wise	85.21	93.31	95.32	95.38

Table 1. PSNR levels (dB) on HDR reconstruction (15 bits) without HEVC compression. Only YUV conversion and adaptive re-quantization to 8, 10, 12, or 14 bits are applied.

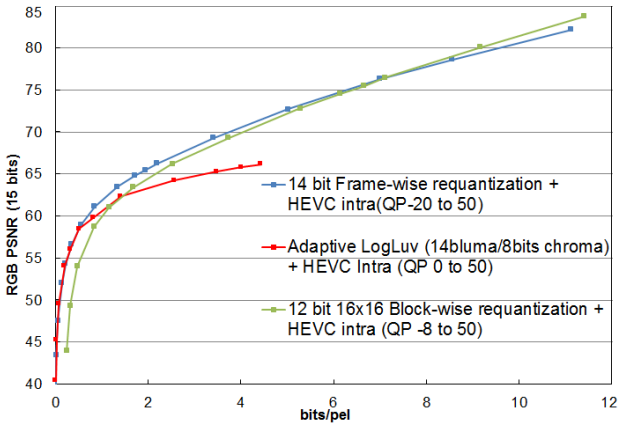


Fig. 4. Comparison between 14 bit frame-wise, 10 bit block-wise and frame adaptive LogLuv in [7].

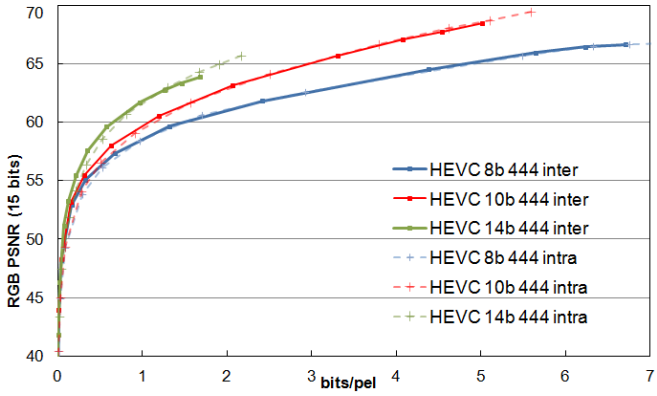
sequence under varying Quantization Parameters (QP) ranging from 0 to 50. Negative QP values (i.e. very small quantization steps) have also been used in figure 4 to compare 14 bit frame-wise and 12 bit block-wise methods in the context of near lossless compression. The PSNR levels obtained on the HDR (15 bit) reconstruction without HEVC compression are also presented in table 1. It represents the maximum quality reachable by each variant of our coding scheme.

We also implemented the frame adaptive LogLuv transform from [7] with 14 bits for luma and 8 bits for chroma. In the article, the transform was applied before H264/AVC encoding. Instead, we used HEVC intra so as to be able to compare it with our re-quantization method. The results are presented in the form of bit rate distortion curves where the bit rate is computed in average number of bits per pixel and the quality metric is the PSNR computed on the 15 bits integer RGB components.

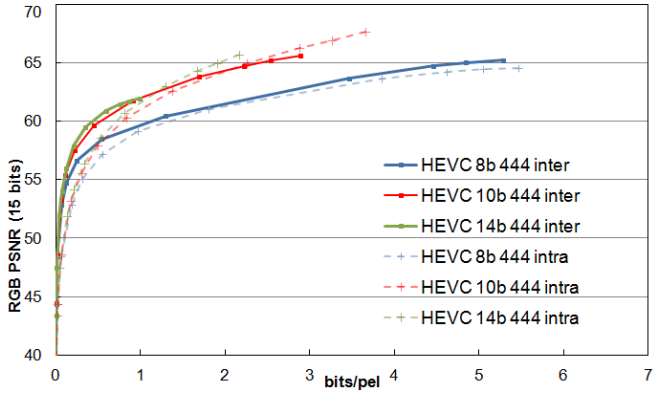
3.2. Discussion

In figure 4, the 14 bit frame-wise method is compared to the adaptive LogLuv transform which also modifies the bit depth of the data on a per frame basis. A significant improvement over the latter method is observed, especially at high bit rates and PSNR. For low QP values ranging from 0 to 4, a 52% average rate improvement is observed using the Bjontegaard metric [14].

In the same figure, we plot the rate-distortion curve of the 12 bit block-wise method. It outperforms the 14 bit frame-wise method at very high bit rates and PSNR. However the loss of coherence between blocks and the coding cost of the quantization parameters significantly degrades its performance at lower bit rates.



(a) Frame-wise re-quantization



(b) GOP-wise re-quantization

Fig. 5. Rate-distortion curves for HEVC intra and inter (QP 0-50) after : frame-wise method (a) and GOP-wise method (b).

In figure 5, inter and intra coding are compared at different bit depths for GOP-wise and frame-wise variants. For both of these variants, the best results are obtained with a target bit depth of 14, which is the maximum supported by HEVC.

For the frame-wise method, almost no difference is observed between inter and intra in figure 5 (a). This is due to the loss of temporal coherence caused by this method.

When the GOP-wise method is used, the temporal coherence is kept. As a result, inter frame predictions perform better than intra frame predictions. This is visible in figure 5 (b) especially in the 14 bit version. However, when high PSNR levels are reached, inter and intra encoding have similar rate-distortion performance.

4. CONCLUSION

We have proposed several bit depth reduction techniques based on an adaptive uniform re-quantization that can be applied prior to HEVC encoding. The method makes it possible to compress HDR content and benefit from the advanced tools of HEVC. A GOP-wise adaptation was proposed to keep temporal coherence for inter predictions. We have shown that our frame-wise approach notably improves the adaptive LogLuv transform which is based on the same principle. We have also shown the potential of performing an adaptive block-wise re-quantization in the context of near lossless HDR compression.

5. REFERENCES

[1] R. Bogart, F. Kainz, and D. Hess, "OpenEXR image file format," *ACM Siggraph, Sketches & Applications*, 2003.

[2] "Academy Color Encoding System (ACES)," `http://www.oscars.org/science-technology/council/projects/aces.html`.

[3] G. Ward, *Graphics Gems II*, chapter Real pixels, pp. 80–83, Academic Press, 1991.

[4] "Technical introduction to OpenExr," `http://www.openexr.com/TechnicalIntroduction.pdf`.

[5] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[6] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[7] A. Motra and H. Thoma, "An adaptive LogLuv transform for high dynamic range video compression," *IEEE Int. Conf. Image Process. (ICIP)*, pp. 2061–2064, Sept. 2010.

[8] Y. Dong, P. Nasiopoulosa, and Mahsa T. Pourazad, "HDR video compression using high efficiency video coding (HEVC)," *Ubicomm*, pp. 205–209, Sep. 2012.

[9] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel, "Backward compatible high dynamic range MPEG video compression," *ACM Trans. Graph*, vol. 25, no. 3, pp. 713–723, Jul. 2006.

[10] J.-U. Garbas and H. Thoma, "Inter-layer prediction for backwards compatible high dynamic range video coding with SVC," *Picture Coding Symposium (PCS)*, pp. 285–288, May 2012.

[11] M. Iwahashi and H. Kiya, "Two layer lossless coding of HDR images," *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 1340–1344, May 2013.

[12] J.F. Blinn, "Floating-point tricks," *IEEE Computer Graphics and Applications*, vol. 17, no. 4, pp. 80–84, Jul. 1997.

[13] "HM range extension, version 12.0 RExt 4.0 rc2," `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0+RExt-4.0rc2/`.

[14] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," in *document VCEG-M33, ITU-T VCEG Meeting*, Austin, Texas, USA, Apr. 2001.