TEMPORALLY COHERENT LUMINANCE-TO-LUMA MAPPING FOR HIGH DYNAMIC RANGE VIDEO CODING WITH H.264/AVC

Jens-Uwe Garbas, Herbert Thoma
Fraunhofer IIS, Erlangen, Germany
jens.garbas@iis.fraunhofer.de, herbert.thoma@iis.fraunhofer.de

ABSTRACT
This paper presents a technique for the efficient compression of high dynamic range video (HDR) sequences. Such video sequences usually represent several orders of magnitude of real-world luminance intensity levels. Therefore, they are mostly stored in a floating-point representation. In order to obtain a coded representation that is bit stream compatible with the H.264/AVC video coding standard, the float-valued HDR values have to be mapped to a suitable integer representation first. The mapping proposed in this paper is adapted to the dynamic range of each video frame. Furthermore, to compensate for the associated dynamic contrast variation across frames, a weighted prediction method and quantization adaptation are introduced. The experiments show that the proposed method offers highly efficient HDR video compression. Only a fraction of the bit rate of a non-adaptive reference method is required to represent an HDR video sequence at the same quality.

Index Terms— high dynamic range, video coding, LogLuv

1. INTRODUCTION
So far, most image and video coding applications can cover only a luminance range of about 2 orders of magnitude (low dynamic range (LDR)) [1]. However, the human visual system (HVS) allows us to adapt to light conditions that can cover a range of more than 10 orders of magnitude and to perceive about 5 orders of magnitude simultaneously [2]. With an increasing number of applications that can profit from a representation of the full HDR luminance (e.g., computer generated imagery, special effects productions, HDR displays), there will be an increasing demand in HDR video coding. Using a standard coding method, like H.264/AVC, will allow for a seamless transition from LDR towards HDR video coding without much additional effort. Note that the term HDR refers to the representation of real luminance values throughout this work and not to a tone-mapped LDR representation, what is sometimes called HDRI.

Since the most natural representation of HDR data, floating-point numbers, does not result in a good compression and is also costly to handle, several authors proposed a suitable mapping from floating-point luminance values to integer luma values [3, 4, 5, 6]. These luminance-to-luma mappings have in common that the associated loss in precision is below the tolerance of the HVS and no distortion is therefore perceived. They further have in common, that they apply a conversion of the HDR image data to the CIELUV color space [1] before further processing. That is, the data is represented by a luminance component Y and the chromacity components (u', v'). The advantage of the (u', v') color representation is that it is perceptually uniform. That is, equal offsets in this representation represent equal perceptual color differences and therefore they can be linearly mapped to integer values with a bit depth of, e.g., 8 bit. Such a mapping from the perceivable (u', v') interval [0, 0.020] to integer values in the range [0, 255] introduces a maximum absolute quantization error of 0.00172 which is well below the visible threshold.

Since the HVS obeys to Weber’s law, for a large luminance range, in most works a logarithmic mapping of the luminance Y to luma code values is performed [3, 5, 6]. This results in a constant relative quantization error leading to a perceptually uniform representation of the luminance. E.g., in [3] Larson proposed the following luminance-to-luma mapping (LogLuv transform):

\[
L_{15} = \left\lfloor \frac{256(\log_2(Y) + 64)}{256} - 64 \right\rfloor \quad (1)
\]

It maps the real-valued luminances in the interval \([5.44 \times 10^{-20}, 1.84 \times 10^{19}]\) to 15 bit integer luma values in the range \([0, 2^{15} - 1]\) and vice versa. That is, about 38 orders of luminance magnitude are represented with a relative step size of 0.277. This is well below the visible quantization threshold of about 1% [1].

However, the dynamic range covered by such a mapping is far beyond the range of what the HVS can simultaneously perceive and there exists no natural image data that spans such high dynamic ranges. Whereas for lossless image compression of data that can undergo further image processing steps this extremely high range and fidelity might be useful, for lossy video encoding that is intended for being watched by human observers, it is not. There is no need to reserve bits to represent luminance values that are not perceivable or that do not occur in the source image or video frame. Since this would degrade the compression efficiency, e.g., in HDR still image coding with the TIFF library [3], a scaling factor can be used to scale the source image to an appropriate range before the LogLuv transform. In a similar LogLuv approach [6], scaling has been applied to each individual frame of a video sequence in order to exploit the full range of possible luma code values for a given bit depth.

However, like many HDR video coding methods, the latter is just a straightforward extension of HDR image coding to individual video frames. Therefore, the approach lacks some video specific aspects what significantly degrades the compression efficiency. Most notably, mapping the luminance values of successive frames to different code values with an individual scaling for each frame significantly harms the temporal coherence of the sequence. Consequently, the temporal motion compensated prediction in the H.264/AVC video coder mostly fails. Moreover, even though the encoder might use a constant quantization, the effective quantization will largely vary across time, leading to strong variations in quality and bit rate. Further unsolved issues of this approach are that the float-valued scaling information has to be transmitted as side information for each frame, thus complicating standard conformant coding and increasing bit rate.

Therefore, in the following we develop a frame-wise adaptive luminance-to-luma mapping and show how we can make use of the
H.264/AVC weighted prediction tools to maintain temporal coherence and, at the same time, to transmit the adaptive mapping parameters. Furthermore, we show how we have to adapt the quantization parameter (QP) for each frame dependent on the adaptive mapping.

2. PROPOSED METHOD

2.1. Dynamic Range Adaptive Luminance Mapping

In the following we re-visit the luminance-to-luma mapping for video coding applications. The trade-off between the representable luminance range \([Y_{\text{min}}, Y_{\text{max}}]\), the luma bit depth \(n\) and the associated relative precision can be seen in the following more general formulations of the luminance-to-luma mapping functions:

\[
L_n = \left[\frac{2^n - 1}{\log_2(Y_{\text{max}}/Y_{\text{min}})} \cdot (\log_2(Y) - \log_2(Y_{\text{min}}))\right],
\]

\[
Y = 2^{(L_n + 0.5) \log_2(Y_{\text{max}}/Y_{\text{min}})} + \log_2(Y_{\text{min}}) \cdot \frac{2^n - 1}{2^n - 1}.
\]

This linear relationship between the logarithm of the luminance \(Y\) and the luma space \(L\) is also depicted Fig. 1. Obviously, the mapping achieves the highest fidelity when \(L_n = 0\) and the luma space \(L = 0\) respectively. That is, if the existing luminance values in a video frame are mapped to the full luma range by the mapping function with the respective. That is, if the same luminance value \(\hat{Y}\) and the maximum luminance \(Y_{\text{max}}\) equals the minimum and maximum luminance of the current video frame, respectively. That is, if the existing luminance values in a video frame are mapped to the full range luma by the mapping function with the steepest possible slope. However, since the dynamic ranges can vary from one frame to the next (even in a static scene, due to noise), such a straightforward adaptation would break the temporal coherence of the video sequence and prevent an efficient temporal prediction. The next section will present an adaptive mapping that takes such effects into account.

2.2. Temporally Coherent Adaptive Luminance Mapping

Consider that two consecutive frames \(k\) and \(l = k + 1\) of an HDR video sequence exhibit different luminance ranges \([Y_{\text{min},k}, Y_{\text{max},k}]\) and \([Y_{\text{min},l}, Y_{\text{max},l}]\), respectively. Obviously, using these extrema of each frame in (2) will result in a different mapping for each frame. That is, the same luminance value \(\hat{Y} = Y_k = Y_l\) in frame \(k\) and \(l\) will be mapped to different luma values \(L_{n,k}\) and \(L_{n,l}\), respectively as exemplified in Fig. 1. Plugging (3) into (2) using the different mapping for frame \(k\) and \(l\), respectively:

\[
L_{n,l} = (L_{n,k} + 0.5) \frac{\log_2(Y_{\text{max},k}/Y_{\text{min},k})}{\log_2(Y_{\text{max},l}/Y_{\text{min},l})} + \frac{2^n - 1}{2^n - 1} \frac{\log_2(Y_{\text{max},k}/Y_{\text{min},k})}{\log_2(Y_{\text{max},l}/Y_{\text{min},l})}
\]

\[
= (L_{n,k} + 0.5) \cdot w + o.
\]

Apparently, the relation of two luma values \(L_{n,k}\) and \(L_{n,l}\) stemming from the same luminance value \(\hat{Y}\) is entirely defined by a scale \(w\) and an offset \(o\). \(w\) and \(o\) can be easily derived from the ranges \([Y_{\text{min},k}, Y_{\text{max},k}]\) and \([Y_{\text{min},l}, Y_{\text{max},l}]\).

H.264/AVC is the first international video coding standard defining the syntax for a weighted prediction (WP) tool [7]. The original intention of WP is to enhance the coding efficiency for fade-in and fade-out sequences where motion compensated prediction usually fails. It allows to explicitly signal a weight parameter \(\hat{w}\) and an offset parameter \(\hat{o}\) per slice. The parameters can be used to weight and shift the reference frame for enhancing the temporal prediction. Equation (4) shows that a change of the dynamic range of successive frames merely results in a weighting \(w\) and shifting \(o\) of identical luminance values in the luma space. Therefore, the WP syntax of H.264/AVC is perfectly suited to allow for an efficient temporal prediction despite any changes in the luminance range. Consider, e.g., the case that a nearly static scene is recorded by an HDR capable camera facing the bright sun. When the sun is now abruptly covered by a cloud, the dynamic range will change by several orders of magnitude whereas the luminance values of all the foreground objects will approximately remain constant. If we can use the WP tools to adapt the luma values of the reference frame, it allows for a perfect temporal prediction of the foreground pixels that stem from the same luminance values. Furthermore, the WP parameter information is sufficient to convey any necessary side information for a frame-wise adaptation of the luminance-to-luma mapping as it will shown in the following.

In H.264/AVC the precision and dynamic range of \(\hat{w}\) and \(\hat{o}\) is limited. Both parameters can take on integer values between \(-128\) and 127. The precision of \(\hat{w}\) is confined by a quantization interval of \(1/2^{\log_{\mathbb{P}}(w)}\). The parameter \(\log_{\mathbb{P}}(w)\) is signaled explicitly and can take on integer values from 0 to 7. Consequently, a higher \(\log_{\mathbb{P}}(w)\) value leads to a more fine-grained representation of the parameter \(w\). It also means that more bits are required for coding the weighting factors and a narrowing of the range of the effective scaling [7]. The step size of the offset parameter \(\hat{o}\) is defined by \(2^{n-s}\) in order to take into account the bit depth \(n\) of the luma representation in the H.264/AVC coder. Consequently, in order to allow for a perfect temporal prediction of unchanged luminance values from one frame to the next, it is necessary to quantize the change of the adaptive mapping function in such a way that it can be represented by the H.264/AVC WP parameters \(\hat{w}\) and \(\hat{o}\).

That is, given the dynamic luminance range covered by the mapping function of frame \(k\), \([Y_{\text{min},k}, Y_{\text{max},k}]\), we have to find the minimum \(\hat{Y}_{\text{min},l}\) and the maximum \(\hat{Y}_{\text{max},l}\) that fulfill

\[
\frac{\log_2(Y_{\text{max},k}/Y_{\text{min},k})}{\log_2(Y_{\text{max},l}/Y_{\text{min},l})} \cdot 2^{\log_{\mathbb{P}}(w)} = \hat{w}; \ {\hat{w} \in \mathbb{Z}} | -128 \leq \hat{w} \leq 127
\]

and

\[
\frac{\log_2(Y_{\text{max},k}/Y_{\text{min},k})}{\log_2(Y_{\text{max},l}/Y_{\text{min},l})} \cdot 2^{n-s} = \hat{o}; \ {\hat{o} \in \mathbb{Z}} | -128 \leq \hat{o} \leq 127
\]

under the constraints

\[
\hat{Y}_{\text{max},l} \geq Y_{\text{max},l} \quad \text{and} \quad \hat{Y}_{\text{min},l} \leq Y_{\text{min},l}.
\]

The latter two inequalities assure that the luminance range covered by the adaptive mapping covers at least the range of luminance range present in the current frame, \([Y_{\text{min},l}, Y_{\text{max},l}]\).

In practice, we find the solution to this problem by solving (5) and (6), setting \(\hat{Y}_{\text{max},l} = Y_{\text{max},l}\) and \(\hat{Y}_{\text{min},l} = Y_{\text{min},l}\), and rounding
towards zero. This gives us the initial values for $\hat{w}$ and $\hat{o}$ and we can solve (5) and (6) w.r.t. $Y_{\text{min},l}$ and $Y_{\text{max},l}$, respectively:

$$Y_{\text{min},l} = 2^\frac{\log_2(Y_{\text{min},k}) - \hat{o} \cdot \log_2(\frac{2\log WD}{\hat{w}} \cdot (2^n - 1)) - \log_2(Y_{\text{max},k} / Y_{\text{min},k})}{\log_2(Y_{\text{max},k} / Y_{\text{min},k})}.$$  

$$Y_{\text{max},l} = 2^\frac{\log_2(Y_{\text{max},k} / Y_{\text{min},k}) + \log_2(Y_{\text{max},l})}{\log_2(Y_{\text{max},k} / Y_{\text{min},l})}.$$  

If the results violate one of the conditions in (7), we decrease $\hat{w}$ or increase $\hat{o}$ by 1, respectively and reevaluate (8) and (9).

After finding the best luminance range $[Y_{\text{min},l}, Y_{\text{max},l}]$ of frame $l$ w.r.t. frame $k$, we use these values for the mapping in (2). Furthermore, the weight and offset parameters $\hat{w}$ and $\hat{o}$ are readily available for usage in the weighted temporal prediction of the H.264/AVC video encoder. Finally, it can be seen from the relations in (5) and (6) that these parameters fully suffice to exactly recover the luminance range of the current frame given the range of the previous frame. No additional side information is necessary for the adaptive mapping when the mapping of the first frame (and possibly IDR frames) covers the maximal visible dynamic range. Otherwise, the range for the first frame must be signaled explicitly to the decoder.

### 2.3. Temporally Coherent Quantization

For each frame we map different luminance ranges to luma code values. Therefore, using the identical QP during the H.264/AVC encoding process, as it was, e.g., done in [6], will lead to a varying quantization of the luminance space, depending on the mapping. Therefore, we propose to take the luminance mapping range into account to find a suitable $\Delta\text{QP}$ for each frame, accordingly. Here, $\Delta\text{QP}$ denotes a QP offset for the current frame w.r.t. the reference QP that is used to encode the first frame. It can be easily seen in Fig. 1 that, in order to introduce the same effective quantization to the luminance values, the quantizer step sizes $Q_{\text{step},l}$ and $Q_{\text{step},k}$ of the current frame $l$ and an arbitrary reference frame $k$ have to be related according to

$$Q_{\text{ref},l,k} = \frac{Q_{\text{step},l}}{Q_{\text{step},k}} = \frac{\log_2(Y_{\text{max},k} / Y_{\text{min},k})}{\log_2(Y_{\text{max},l} / Y_{\text{min},l})}.  \tag{10}$$

Taking into account this fact, per definition $Q_{\text{step}}$ approximately doubles when the QP value is increased by 6 units we can state:

$$Q_{\text{ref},l,k} \approx 2^{\Delta\text{QP}_{l,k} / 6} \Rightarrow \Delta\text{QP}_{l,k} = \text{round}(6 \cdot \log_2(Q_{\text{ref},l,k})).  \tag{11}$$

In this work, we always use the first frame of a sequence as reference frame for calculating the QP offset values for each frame. That is, an arbitrary frame $l$ will be quantized with $\text{QP}_l = \text{QP}_1 + \Delta\text{QP}_{l,1}$.

### 3. EXPERIMENTAL RESULTS

For evaluating the proposed temporally coherent luminance-to-luma mapping, we performed coding experiments with three HDR test sequences: Panorama, Tunnel, and Sun. All sequences have a resolution of $640 \times 480$ pixel and a frame rate of 30 fps. The panorama test sequence was generated by panning a $8000 \times 4000$ pixel HDR panorama image [11]. It shows dark interior areas as well as very bright sun reflections from outside a window. Its overall dynamic range is of the order of $10^{10}$ : 1. Both, Tunnel and Sun were taken from inside a driving car with an HDR video camera and are freely available from Max-Planck Institute [8]. The former one shows a drive through a dark tunnel, the latter one shows a drive on a highway facing the bright sun. The overall dynamic range represented in these sequences is $10^5$ : 1 and $10^7$ : 1, respectively.

In our experiments we use two metrics to evaluate the quality of the decoded HDR videos: the HDR visible difference predictor (VDP) [9] and the perceptually uniform peak signal-to-noise ratio (PU PSNR) [10]. The former one estimates the percentage of pixels in a pair of images that an observer will notice to be different with a probability of more than 75%. The latter metric is a straightforward extension of the common PSNR metric to HDR. For LDR images it is assumed that the gamma corrected pixel code values are perceptually uniform, that is, equal error amplitudes are equally visible in bright and dark regions of an image. However, this assumption does not hold for HDR images and therefore, the code values must be scaled to a perceptually uniform space before meaningful PSNR values can be calculated [10].

For encoding the sequences, they are first transformed from RGB floating-point values to the LogLuv space and then encoded with the H.264/AVC reference software JM 17.2. The luma component is encoded with a bit depth of 12 bit/sample, the $u'$ and $v'$ components are subsampled by a factor of two vertically and horizontally and encoded with 8 bit/sample. We use the same configuration of the H.264/AVC high profile with $8 \times 8$ transform, IPPP GOP structure, intra frame period of 15, and CABAC enabled for all experiments. A fixed reference QP is selected for each encoder run and no rate-control is enabled. However, the frame-wise QP may deviate from this reference QP as described in Sec. 2.3. After decoding the sequences, they are mapped back to RGB floating-point values and their quality is evaluated according to the metrics described before.

Fig. 2 shows the coding results for all test sequences in terms of the VDP averaged over all decoded frames (upper row) and in terms of mean PU PSNR of the luminance component (lower row). The proposed method ("proposed") is compared with two reference methods in Fig. 2: straightforward frame-wise adaptation of the luminance-to-luma mapping to the dynamic range of each frame without taking into account the temporal coherence ("frame-wise") [6], and constant mapping of the whole perceivable luminance range $[10^{-4}, 10^{8}]$ ("visual range"). In the latter case, the luminance range of the mapping function might exceed the range of occurring luminances in many HDR video sequences. However, in a real-time coding application it is not possible to narrow the mapping range to the absolute luminance range of a sequence, because this would require the processing of the whole sequence before encoding.

Fig. 2 clearly shows that the proposed mapping significantly outperforms the reference methods for all test sequences. It is worth noting here that the VDP metric is a threshold metric that only offers an estimate about if a pixel is perceived erroneously or not. It does not state how annoying this error is for an observer. Thus, e.g., the results in Fig. 2(a) can be interpreted as follows: if we allow about 1% of the pixels to be perceived erroneously, with the proposed mapping, we only need a bit rate of less than 2500 kbit/s. This is a reduction of about 50% (25%) compared to the 5000 kbit/s (3250 kbit/s) we have to spend to achieve the same VDP value in the "visual range" ("frame-wise") scenario. Likewise, huge rate savings can be observed for the Tunnel and Sun test sequences in Figs. 2(b) and (c).

As expected, the PU PSNR results in Figs. 2(d)–(f) depict similar performance characteristics as the VDP results for all sequences. Furthermore, they allow a quantitative conclusion of the gain in quality that can be achieved with the proposed method for a large range of bit rates. E.g., for the Panorama sequence the PU PSNR value of the proposed method exceeds the PU PSNR value of the "visual range" mapping by 3 dB at 3250 kbit/s (cf. Fig. 2(d)). This means that the mean squared error in the perceptually uniform luminance
space is halved at the same bit rate and the visual quality is increased significantly.

It is worth noting, that for the Panorama sequence the frame-wise adaptive mapping has a very detrimental effect on the coding efficiency compared to the non-adaptive “visual range” mapping. This sequence exhibits very large and fast variations of its dynamic range and therefore, in the case of the frame-wise adaptive mapping, the temporal prediction fails (cf. Figs. 2(a),(d)). On the other hand, it can be observed in Figs. 2(b) and (e) that the proposed method performs almost identical to the “frame-wise” mapping. In this sequence, the temporal changes of the dynamic range are very smooth.

In our experiments we further observed that for the “frame-wise” mapping there exist strong temporal variations of the bit rate and quality whenever the dynamic range changes significantly. This negative effect is circumvented by the temporally coherent quantization and mapping of the proposed method.

4. CONCLUSIONS

In this paper an adaptive luminance-to-luma mapping has been proposed that allows the compression of floating-point high dynamic range video data with the state-of-the-art H.264/AVC video coding standard. Unlike other methods the mapping is adapted to the dynamic range of each frame. Nevertheless, temporal coherence is sustained by exploiting the weighted prediction tools of H.264/AVC and by applying a frame-wise adaptation of the quantization parameter in accordance with the mapping function. No additional side information is needed and significant bit rate savings of up to 50% compared to non-adaptive methods can be observed at the same quality.

We would like to thank Stefan Steuerwald and Alex Waggershauser from LizardQ GbR for providing the source image [11] for the Panorama test sequence.

5. REFERENCES