# High Quality HDR Video Compression using HEVC Main 10 Profile

Jacob Ström*, Kenneth Andersson*, Martin Pettersson*, Per Hermansson*, Jonatan Samuelsson*, Andrew Segall†, Jie Zhao†, Seung-Hwan Kim†, Kiran Misra†, Alexis Michael Tourapis‡, Yeping Su‡ and David Singer‡
*Ericsson Research, †Sharp Laboratories of America, ‡Apple Inc.

***Abstract*—This paper describes high-quality compression of high dynamic range (HDR) video using existing tools such as the HEVC Main 10 profile, the SMPTE ST 2084 (PQ) transfer function, and the BT.2020 non-constant luminance Y′CbCr color representation. First, we present novel mathematical bounds that reduce complexity of luminance-preserving subsampling (luma adjustment). A nested look-up table allows for further speedup. Second, an adaptive QP scheme is presented that obtains a better bit allocation balance between dark and bright areas of the picture. Third, a method to control the bit allocation balance between chroma and luma by adjusting the chroma QP offset is presented. The result is a considerable increase in perceptual quality compared to the anchors used in the 2015 MPEG High Dynamic Range/Wide Color Gamut Call for Evidence. All techniques are encoder-side-only, making them compatible with a regular decoder capable of supporting HEVC Main10/PQ/BT.2020, which is already available in some TV sets on the market.**

## I. Introduction

With the digitalization of video, there has a been a tremendous increase in video quality. Resolution has been increased first to high definition (HD) and now a transition to 4K is under way.

Another way to increase video quality is to improve the contrast by increasing the dynamic range. The TV system that is used today was originally intended for luminance values between $0.1\ \mathrm{cd/m^2}$ (candela per square meter) and $100\ \mathrm{cd/m^2}$, typically referred to as standard dynamic range (SDR). However, the real world exhibits a much greater dynamic range; starlight is at about $10^{-8}\ \mathrm{cd/m^2}$ and an unclouded sun is at about $10^9\ \mathrm{cd/m^2}$. High-dynamic range (HDR) video refers to the capturing and reproducing video at a higher dynamic range. Playback of HDR data therefore requires new displays, but in return delivers enhanced realism. Furthermore, HDR is typically combined with wide color gamut (WCG) spaces, such as the ITU-R BT.2020 color space.

Unlike the increase in resolution and frame-rate, HDR presents several challenges especially for content delivery to consumer devices, which has typically been limited to only 8 bit data representations. The adoption of the HEVC Main 10 profile in several consumer decoder devices, however, opened the door for the consideration of a home delivery HDR format. More specifically, the Blu-ray Disk Association (BDA) included in their next generation Ultra HD Blu-ray disk format not only support for 10 bit and 4K resolutions but also for HDR and WCG video data. This HDR/WCG format included the use of the SMPTE ST 2084 (PQ) transfer function, BT.2020 color primaries, as well as the traditional 4:2:0 $Y'CbCr$ non-constant luminance representation. Delivery of the HDR data uses the HEVC Main 10 profile. There have been concerns on how suitable the format was for supporting very low bitrate applications, such as streaming.

To address these concerns, MPEG issued a ”Call for Evidence” (CfE) on HDR/WCG technologies [5]. The BDA format was used as an anchor, and two activities were conducted - a first to study the performance of new normative methods in improving HDR/WCG delivery performance, and a second to identify non-normative improvements to HEVC Main 10. As a result, MPEG discontinued the normative activity, concluding that the non-normative technologies were sufficient to enable HDR/WCG applications. The goal of this paper is to present an overview of some of these non-normative technologies.

## II. Previous Work

When the BDA was determining the HDR format to be used in their Ultra HD specification, it was felt that it would be difficult to require devices to be capable of receiving anything more than 10 bits per pixels. It was also preferred to use 4:2:0 given the memory and processing bandwidth benefits such a format can provide, especially when coupled with the desire to support 4K and 60fps content. It was also decided to utilize the non-constant luminance $Y'CbCr$ color representation, which was well understood and already heavily utilized for SDR applications. WCG was supported by allowing signaling using BT.2020 color primaries. However, a solution was needed that would enable signaling of sufficient quality HDR pixel data using a 10 bit representation, i.e. without the presence of banding (posterization) artifacts. Unfortunately, it was already known [4] that the power-law based transfer functions utilized by SDR applications were incapable of doing so, and hence a new transfer function was needed.

In the work by Larson [4] a logarithmic based transfer function operating in the $Yu'v'$ (CIE 1976) domain was utilized. Although that solution was quite effective, it used a different color representation and required 16 bits of precision. Instead, Miller et al. introduced the Perceptual Quantizer (PQ) [6], building on the contrast sensitivity model of Barten [2]. PQ aims at spreading out the code levels so that a change of one code level will be equally perceptually visible regardless of what luminance level this happens at. By using PQ, and by also limiting the peak brightness to 10000 $\mathrm{cd/m^2}$, a sufficient compromise in terms of peak brightness for consumer applications, it was possible to lower the number of bits down to 12 without any visible banding artifacts. In practice, lowering this further down to 10 bits typically produces banding-free results for most natural content.

A processing chain that can generate this format was defined by MPEG for their HDR/WCG CfE. This chain is shown in Figure 1. Starting with linear light $RGB$ in the BT.2020 color space, normalized coordinates $R_{01}, G_{01}, B_{01}$ are obtained by dividing by the peak luminance $L_p = 10000$. This is followed by the inverse of the PQ transfer function, yielding PQ-adjusted coordinates $R'_{01}G'_{01}B'_{01}$. A color transformation then gives normalized values $Y'_{01}Cb_{0.5}Cr_{0.5}$, which are quantized and chroma-subsampled to yield the $Y'_{420}, Cb_{420}, Cr_{420}$ samples that are fed into the encoder. The encoder is a standard HEVC Main-10 encoder with appropriate signaling of the PQ transfer function and BT.2020 color representations. The resulting bitstreams are fully compliant and decodable by an HEVC Main 10 decoder. After decoding, the values are upsampled, inverse quantized, inverse color transformed, linearized using the PQ transfer function and then scaled according to peak luminance.

The $Y'CbCr$ representation used in this system is typically referred to as the "non-constant luminance" (NCL) representation, since the luminance, defined as $Y_o = w_R \times R + w_G \times G + w_B \times B$, is carried not only in the non-subsampled $Y'_{420}$ component but to some extent also in $Cb_{420}$ and $Cr_{420}$. The non-subsampled component $Y'_{420}$ is therefore called luma in order to distinguish it from the true luminance $Y_o$, and the other two components are typically referred to as chroma. The term "chroma leakage" is used to describe the fact that changes in chroma can have an impact on the luminance in an NCL system. A high level of chroma leakage can be a problem since the human visual system is more sensitive to errors in luminance than to errors in chrominance. While this phenomenon occurs even for SDR processing, François [3] points out that this effect is even larger for the HDR processing chain in Figure 1, as just the chroma error introduced by the subsampling alone can cause visible luminance artifacts.

Ström et al. presented a method called Luma Adjustment to alleviate this problem [8]. It compensates for the chroma leakage by adjusting the luma value $Y'_{420}$ so as to have a signal with a reconstructed luminance value that is as close as possible to the original luminance value of $Y_o$. In detail, the decoded luminance equals to $Y_d = w_R \times \hat{R} + w_G \times \hat{G} + w_B \times \hat{B}$. Going backwards in the lower diagram of Figure 1 it is possible to arrive at

$$Y_d/L_p = w_R \times tf(\hat{R}'_{01}) + w_G \times tf(\hat{G}'_{01}) + w_B \times tf(\hat{B}'_{01}) \quad (1)$$

where $tf(\cdot)$ denotes the PQ transfer function. Going back one step further in the figure gives

$$\begin{aligned} Y_d/L_p =& w_R \times tf(Y'_{01} + a_{13} \times \hat{C}b_{0.5}) \\ &+ w_G \times tf(Y'_{01} + a_{22} \times \hat{C}b_{0.5} + a_{23} \times \hat{C}r_{0.5}) \\ &+ w_B \times tf(Y'_{01} + a_{32} \times \hat{C}b_{0.5}), \end{aligned} \quad (2)$$

where $a_{13}$, $a_{22}$, $a_{23}$ and $a_{32}$ are matrix coefficients in the matrix of the color transformation. Here no compression is assumed so $\hat{Y}'_{01} = Y'_{01}$. Since the luminance $Y_d$ is monotonously increasing with $Y'_{01}$, it is possible to do interval halving to arrive at the best $Y'_{01}$, i.e., the $Y'_{01}$ that will generate a luminance $Y_d$ closest to the original luminance $Y_o$. If a 10-bit representation is used, this process will take at most ten steps. Ström et al. also employ mathematical bounds that narrow the starting interval, lowering the average number of iterations. A speedup method based on a linearization approximation was also presented by Norkin [7]. However, that method can substantially differ from the exact solution in some cases.

## III. Speedup of Luma Adjustment

The Luma Adjustment method can help in considerably alleviating the chroma leakage effect. However, even with the mathematical bounds the average number of iterations per pixel is still high, typically between 4 and 5.

### *A. New Mathematical Bounds*

The original luminance $Y_o$ can be written as $Y_o = w_R \times R + w_G \times G + w_B \times B$, and thus

$$Y_o/L_p = w_R \times tf(R'_{01}) + w_G \times tf(G'_{01}) + w_B \times tf(B'_{01}). \quad (3)$$

Comparing this with Equation 1 it is easy to see that $Y_d$ can never be equal to $Y_o$ if $\hat{R}'_{01} < R'_{01}$, when also $\hat{G}'_{01} < G'_{01}$ and $\hat{B}'_{01} < B'_{01}$. Let $Y'_R$ be the value of $Y'_{01}$ that makes $\hat{R}'_{01} = R'_{01}$. Since $\hat{R}'_{01} = Y'_{01} + a_{13} \times \hat{C}b_{0.5}$, we can calculate $Y'_R$ as

$$Y'_R = R'_{01} - a_{13} \times \hat{C}r_{0.5} \quad (4)$$

Likewise, let $Y'_G$ be the luma value for which $\hat{G}'_{01} = G'_{01}$ and $Y'_B$ be the luma value for which $\hat{B}'_{01} = B'_{01}$, then

$$\begin{aligned} Y'_G =& G'_{01} - a_{22} \times \hat{C}b_{0.5} - a_{23} \times \hat{C}b_{0.5} \\ Y'_B =& B'_{01} - a_{32} \times \hat{C}b_{0.5} \end{aligned} \quad (5)$$

Thus if we choose $Y'_{01}$ so that it is strictly larger than the smallest of $Y'_R$, $Y'_G$, and $Y'_B$, we are guaranteed that $\hat{R}'_{01} < R'_{01}$, $\hat{G}'_{01} < G'_{01}$, and $\hat{B}'_{01} < B'_{01}$ will never occur simultaneously. We thus have found a lower bound $Y'_{min} = \min\{Y'_R, Y'_G, Y'_B\}$ for $Y'_{01}$. In a similar manner, we have an upper bound $Y'_{max} = \max\{Y'_R, Y'_G, Y'_B\}$.

These new bounds can be combined with the previous bounds from Ström et al., which we can call $Y'_{prev-min}$ and $Y'_{prev-max}$. The combined bounds can be calculated as

$$\begin{aligned} Y'_{low} &= \max\{Y'_{prev-min}, Y'_{min}\} \\ Y'_{high} &= \min\{Y'_{prev-max}, Y'_{max}\} \end{aligned} \quad (6)$$

Note also that if $Y'_R = Y'_G = Y'_B$, then using this value will bring $\hat{R}'_{01} = R'_{01}$, $\hat{G}'_{01} = G'_{01}$ and $\hat{B}'_{01} = B'_{01}$. Therefore we add a test to see if $Y'_R$, $Y'_G$ and $Y'_B$ round to the same 10-bit integer. In that case we use that value and avoid iterations altogether. While this may differ by one code level from the result by Ström et al. [8], the lowered complexity makes it a reasonable trade-off.

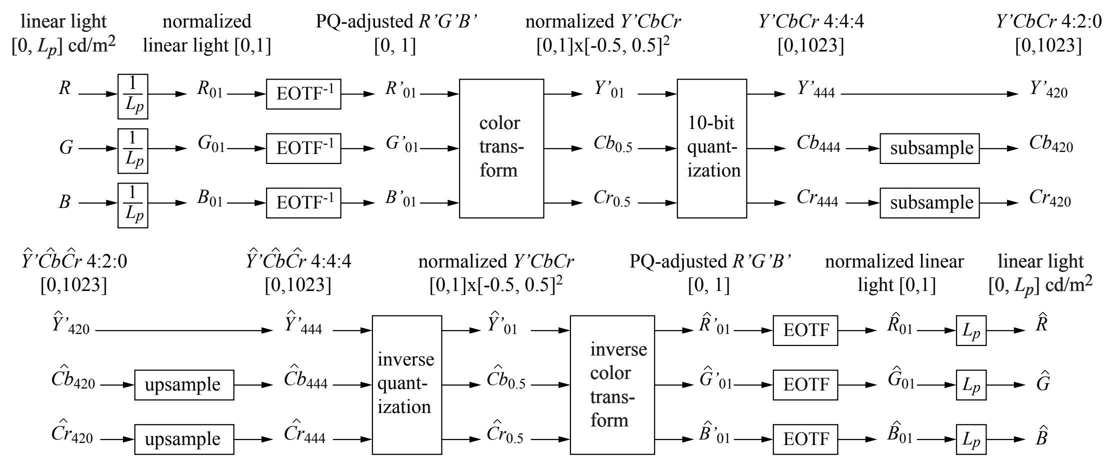


Fig. 1. Top: Going from linear light to $Y'CbCr$. Bottom: Going from $\hat{Y}'\hat{C}b\hat{C}r$ back to linear light.

### *B. Nested Look-up Table*

A lot of the computation when executing Luma Adjustment is spent in calculating the forward and inverse PQ transfer function, since it contains divisions and power functions. Therefore it is of interest to investigate look-up table (LUT) approaches. Unfortunately, the steep slope of the PQ makes the use of a single LUT prohibitive since it would need to be too large ($\approx 10^8$ bytes) to give reasonable precision.

Therefore, this paper proposes to use a two level look-up table. In the first level 10 segments are used, with the segments determined using a log10 distance relationship. In particular, starting from $i = 0$, we can define the first segment as covering the range from [0.0, $10^{-9}$], while for all other $i$ values the range would be equal to [$10^{-10+i}$, $10^{-9+i}$]. Then, for each segment, $10000$ uniformly spaced lookup table entries could be specified. This limits the memory requirements for such a look-up table to $10 \times 10000 \times 8 = 800,000$ bytes. Fewer entries could possibly be used but have not been tested. As a final stage, and to further improve precision, bilinear interpolation is used.

## IV. Adaptive Luma QP

Many video compression standards and codecs have the ability to control bit allocation between regions within an image. This is used by many existing encoders in order to steer away bits from areas where it is hard to spot errors towards areas where errors are more easily detected. As an example, in an image containing both a face (low variance) and a tree against a bright sky (high variance), using the same quantizer in both areas would likely result in more bits spent on the tree. However, at the same time, errors on the tree would likely be less detectable. An encoder that could steer bits from high variance areas (the tree) to low variance areas (the face) could potentially increase the perceived quality.

In HEVC, the primary method for controlling bit allocation is to adaptively change the quantization parameter (QP) of a block based on content characteristics. We will refer to this method as the "adaptive QP" method. An encoder may change QP depending upon, among other things, the variance, brightness, and motion of the block and its neighbors. Commonly, during the development of new video coding standards within MPEG, fixed QP parameters for an entire image are utilized. This is done in an effort to better isolate the performance of a proposed video coding tool from non-normative modifications, even though it is well appreciated that much better perceptual results could be achieved through the use of an adaptive QP method. This has also been the case during the development of the HEVC specification. In its HM reference software the default configuration used for all experiments utilizes fixed QP values for an entire image. This configuration is typically referred to as "fixed QP".

### *A. Background*

The development of HEVC and the HM has been done on SDR data (peak brightness of $100~\mathrm{cd/m^2}$) using SDR processing (BT.709 transfer function and BT.709 color space). One key insight is that by just changing to HDR processing (PQ transfer function and BT.2020 color space) the same SDR data (peak brightness of $100~\mathrm{cd/m^2}$) will be treated differently by an encoder. In particular, going from SDR to HDR processing will steer a lot of bits away from bright areas (around $100~\mathrm{cd/m^2}$) to dark areas (around $0.01~\mathrm{cd/m^2}$).

As an example, if SDR processing is used, and for a 10 bit limited range $Y'CbCr$ representation, the $100~\mathrm{cd/m^2}$ material will occupy all code levels from $64$ to $940$, but only code levels $64$ to $509$ if PQ is used. A perturbation of $\pm 1$ code level around $509$ ($100~\mathrm{cd/m^2}$) in the HDR processing chain will be equivalent of roughly $\pm 4$ code levels around code level $940$ (also $100~\mathrm{cd/m^2}$) in the SDR processing chain. At the same time, a perturbation of one code level at around $0.01~\mathrm{cd/m^2}$ will be roughly the same for both the SDR and the HDR processing chain. Thus, if an encoder is wired to treat an error of one code level in the same manner regardless of its value, such an encoder will allow luminance errors that are four times larger in bright areas using the HDR as compared to the SDR processing chain. Hence, changing from SDR to HDR processing will redistribute bits from bright to dark areas even for the same material. An adaptive QP method is presented in this paper that results in a more balanced bit distribution between bright and dark areas for HDR content.

### B. Implementation

As an example of what such a mechanism can look like, we show how it is possible to change the QP value in the HM encoder that otherwise employs fixed QP compression.

The picture is first divided into $64 \times 64$ blocks (Coding Tree Units in HEVC). For every block, the average luma value is calculated. The QP for a particular block is then obtained by adding the picture QP with the dQP obtained by using the look-up table shown in Table I.

TABLE I
LOOK-UP TABLE OF THE dQP VALUE FROM THE AVERAGE LUMA VALUE.

| luma $Y'_{ave}$ range | dQP |
|---|---|
| $Y'_{ave} < 301$ | 3 |
| $301 \le Y'_{ave} < 367$ | 2 |
| $367 \le Y'_{ave} < 434$ | 1 |
| $434 \le Y'_{ave} < 501$ | 0 |
| $501 \le Y'_{ave} < 567$ | $-1$ |
| $567 \le Y'_{ave} < 634$ | $-2$ |
| $634 \le Y'_{ave} < 701$ | $-3$ |
| $701 \le Y'_{ave} < 767$ | $-4$ |
| $767 \le Y'_{ave} < 834$ | $-5$ |
| $Y'_{ave} \ge 834$ | $-6$ |

### C. Discussion

That a redistribution of bits from dark to bright can bring perceptual benefits may seem counter-intuitive: PQ is based on Barten's model of contrast sensitivity, and thus a change of one code level should be equally visible regardless of if it happens in dark or bright areas. The unchanged HM software, which treats all code levels the same, should therefore produce perceptually pleasing results. However, PQ assumes best-case adaptation. When the eye is adapted to $1.0\ \mathrm{cd/m^2}$, Barten predicts that the eye can differentiate between $1.0\ \mathrm{cd/m^2}$ and $1.02\ \mathrm{cd/m^2}$, which is equal to one code level step. However, when some pixels are at $1000\ \mathrm{cd/m^2}$, the eye will be differently adapted, and the difference between 1.0 and 1.02 can no longer be seen, which makes it wasteful to spend bits there. This masking effect might be one of the reasons why the redistribution presented here improves perceptual quality.

## V. CHROMA QP OFFSET

Another difference between SDR and HDR is how the chroma samples $Cb$ and $Cr$ are distributed within their ranges. For SDR data ($100\ \mathrm{cd/m^2}$ peak brightness) and processing (BT.709 transfer function and BT.709 color space), most of the range is typically used: The $Y'$ component will use up its full range of $[64, 940]$ and $Cb$ and $Cr$ will populate most of its allowed range of $[64, 960]$. For HDR data ($10000\ \mathrm{cd/m^2}$ peak brightness) and HDR processing (PQ transfer function and BT.2020 color space), most of the allowed range for $Y'$ will still be used, but the distributions of $Cb$ and $Cr$ will cluster closer to 0. This effect will be especially pronounced if the original content does not exercise the entire BT.2020 color space. The MPEG CfE used content that exercised the BT.709 and P3D65 color spaces only.

### A. TF impact on Chroma

Assume an encoder that can provide a balanced quality allocation between luma and chroma for SDR data with SDR processing. Now if the same encoder with the same settings is fed HDR data with HDR processing, the variance of the $Cb$ and $Cb$ distributions will go down substantially, while the luma variance will be similar to the SDR case. Thus the encoder will spend less bits on chroma and more on luma, relative to the SDR case at the same bitrate. At lower rates this behavior may result in visible color artifacts, especially for colors near white, where mis-colorations in the direction of cyan and magenta become visible. This can be seen in the left column of Figure 2. Furthermore, due to chroma leakage, a poor quality chroma signal can even affect the luminance negatively. To ameliorate this, this paper proposes applying a negative chroma QP offset value. This has the effect of moving bits back from luma to chroma, avoiding much of the artifacts. This is especially beneficial at low bit rates, where such artifacts otherwise become apparent. However, at a sufficiently high bit rate, the chroma may be considered to be good enough, and a negative chroma QP offset may instead hurt coding performance. Therefore it is proposed to let the chroma QP offset be a function of the QP, and set it to zero for sufficiently low QPs.

A special case occurs when it is known that the content is in a restricted subset of the BT.2020 color space. As an example, if the mastering display used to grade the content was using the P3D65 color space, the content will never venture outside this space. Hence $Cb$ and $Cr$ will never go outside a certain interval, which is smaller than the allowed $[64, 960]$. In such circumstances it may be advantageous to use a larger negative chroma QP offset compared to the case when using material that cover the entire BT.2020 color space.

### B. Implementation

In HEVC version 1 the $Cb$ and $Cr$ QP offsets can be controlled individually on a picture and slice level for each color component. Therefore, in this proposal the $Cb$ and $Cr$ QP offsets are signaled once per picture using the picture parameter set (PPS). In particular, we are interested in the following 3 capture and representation color space scenarios, a) when both color spaces are identical, b) when the capture color space is P3D65 and the representation space is BT.2020, and c) when the capture color space is BT.709 and the representation space is BT.2020. The model is expressed as

$$\mathrm{QPoffset}_{Cb} = \mathrm{clip}(\mathrm{round}(c_{cb}(k \times QP + l)), -12, 0) \quad (7)$$

$$\mathrm{QPoffset}_{Cr} = \mathrm{clip}(\mathrm{round}(c_{cr}(k \times QP + l)), -12, 0), \quad (8)$$

where $\mathrm{clip}(x, a, b)$ clips $x$ to the interval $[a, b]$. Furthermore, in scenario a) $c_{cb} = 1$ and $c_{cr} = 1$, in scenario b) $c_{cb} = 1.04$ and $c_{cr} = 1.39$, in scenario c) $c_{cb} = 1.14$ and $c_{cr} = 1.78$. The linear model described by $k$ and $l$ is the same for both $Cb$ and $Cr$ components, and all color spaces. It models that a larger negative chroma QP offset is needed for smaller rates (i.e. large QPs) and was determined empirically to be equal to $k = -0.46$ and $l = 9.26$.

TABLE II
ITERATIONS PER PIXEL (IPP) AND EXECUTION TIME FOR [8] AND THE PROPOSED METHOD.

| Seq. nbr | ipp [8] | ipp prop. | savings (%) | time [8] (ms) | time prop. (ms) | savings % |
|---|---|---|---|---|---|---|
| 1 | 4.96 | 1.16 | 76.6% | 4150 | 733 | 82.3% |
| 2 | 3.85 | 2.49 | 35.4% | 4369 | 920 | 78.9% |
| 3 | 5.29 | 1.75 | 66.8% | 5242 | 827 | 84.2% |
| 4 | 3.74 | 1.59 | 57.4% | 4447 | 826 | 81.6% |
| 5 | 4.68 | 1.15 | 75.4% | 4698 | 765 | 83.7% |
| 6 | 4.22 | 1.48 | 65.0% | 4603 | 827 | 82.0% |
| 7 | 4.24 | 1.52 | 64.2% | 4649 | 793 | 81.9% |
| 8 | 4.13 | 1.74 | 58.0% | 4618 | 905 | 80.4% |
| 9 | 3.65 | 1.89 | 48.2% | 4384 | 842 | 80.8% |
| 10 | 3.42 | 1.96 | 42.9% | 4182 | 746 | 81.7% |
| *Ave.* | *4.22* | *1.67* | *60.3%* | *4537* | *820* | *81.9%* |

## VI. RESULTS

Table II presents the results of the speedup techniques for a number of sequences; BalloonFestival (1), BikeSparkles (2), EBU Hurdles (3), EBU Starting (4), FireEater (5), GarageExit (6), Market (7), ShowGirl (8), MagicHour (9), and WarmNight (10). The second column shows the number of iterations per pixel (ipp) for [8] measured over the entire sequence. The results for the proposed method are presented in the third column. The decrease in percentage is shown in the fourth column and it can be seen that the number of iterations can be reduced by an average of 60%. The fifth column shows the execution time for luma adjustment for frame 50 of each sequence using the bounds from [8] and no look-up table (LUT). The sixth column shows the execution time with new bounds and LUT, and the last column shows the reduction as a percentage. Average execution time is reduced by 82%. To illustrate the benefit of the Adaptive Luma QP and the Chroma QP offset tools described above, we evaluated the visual performance of several sequences using the MPEG test conditions and HM software model. Two examples are shown in Figure 2, where the left image of each image pair corresponds to the HM result at the time of the CfE, and the right image using the modifications described above. (Note: the images are tone mapped for publication.) As can be seen, large chroma artifacts are present in the HM coded data. For example, artifacts are visible on the white shutters, inside the umbrella, and around the model's lips and ear. Additionally, luma detail loss is also observed on the textured wall. In all cases, enabling the described tools significantly reduces these artifacts.

Two formal subjective evaluations of video quality have been carried out between the techniques presented here (denoted Anchor 3.2) and the CfE anchor (denoted Anchor 1.0), one in Stockholm and one in Rome. The results from these tests are reported by Vittorio et al. [1]: A perceptual benefit for Anchor 3.2 is demonstrated for some sequences, and averaged over all sequences the bit rate savings is 27% for equal mean opinion score (BD-MOS).

Fig. 2. Left column: Traditional processing (MPEG CfE Anchor). Right column: Proposed method, Anchor 3.2 in MPEG/JCT-VC at a similar but lower bit rate. Note how the color artifacts that are present on the left are ameliorated on the right. Texture is also more detailed. Top sequence courtesy of Technicolor and the NevEx project.

## VII. CONCLUSION

We have presented three techniques for efficient compression of HDR images. The first employs new mathematical bounds to speed up the implementation of the luma adjustment method. The use of look-up tables is also considered. Here the average number of iterations and the total execution time go down by 60% and 82% respectively. The second and third techniques rebalance bits between bright and dark areas of the image and chroma and luma respectively, which can result in perceptual quality improvements. These techniques have been been incorporated in the latest anchor (version 3.2) used for HDR evaluation in JCT-VC, and give a bit rate savings of 27% for equal MOS score [1].

## REFERENCES


[1] V. Baroncini, K. Andersson, A. K. Ramasubramonian, G. Sullivan, "Verification Test Report for HDR/WCG Video Coding Using HEVC Main 10 Profile", JCTVC-X1018, *24th JCT-VC meeting*, Geneva, Switzerland, Jun. 2016.

[2] P. G. J. Barten, *Contrast Sensitivity of the human eye and its effects on image quality*, SPIE Optical Engineering Press, 1999.

[3] E. Francois, "MPEG HDR AhG: about using a BT.2020 container for BT.709 content,", m35255, *110th MPEG meeting*, Strasbourg, France, Oct. 2014.

[4] G. W. Larson, "The LogLuv Encoding for Full Gamut, High Dynamic Range Images", *Journal of Graphics Tools*, 3(1):15-31, 1998.

[5] A. Luthra, E. François, W. Husak, "Call for Evidence (CfE) for HDR and WCG Video Coding", N15083, *111th MPEG Meeting*, Geneva, Switzerland, Feb. 2015.

[6] S. Miller, M. Nezamabadi, and S. Daly, "Perceptual Signal Coding for More Efficient Usage of Bit Codes," *Motion Imaging Journal*, 122:52–59, 2013.

[7] A. Norkin, "Fast Algorithm for HDR Color Conversion", *Proceedings of the IEEE Data Compression Conference (DCC)*, Snowbird, March 2016.

[8] J. Ström, J. Samuelsson, and K. Dovstam, "Luma Adjustment for High Dynamic Range Video", *Proceedings of the IEEE Data Compression Conference (DCC)*, Snowbird, March 2016.