HDR color conversion with varying distortion metrics

Andrey Norkin
Netflix Inc., 100 Winchester Circle, Los Gatos, CA, USA 95032

ABSTRACT

The paper compares three algorithms, which attenuate artifacts that may appear in HDR video in Y'CbCr 4:2:0 format. The algorithms attenuate artifacts in colors at the color gamut boundaries while also improving the objective quality. Two closed form solutions demonstrate the same subjective quality as the iterative approach, while being computationally simpler. One of the closed form solutions also shows similar objective results to the iterative algorithm. The choice of the upsampling filter in the pre-processing stage is important and may negatively affect both the objective and subjective quality if there is a mismatch with the upsampling filter used in the reconstruction.

Keywords: High dynamic range video, HDR, video pre-processing, 4:2:0 chroma subsampling, HDR10, HEVC.

1. INTRODUCTION

High Dynamic Range (HDR) is considered one of the major improvements in quality of television pictures. To facilitate broadcasting of HDR video, a Digital Entertainment Content Ecosystems (DECE) has adopted an HDR10 specification [1] that prescribes HEVC Main10 encoding [2], ST.2084 transfer function [3], BT.2020 color space [4], Y'CbCr 4:2:0 non-constant luminance color format [4], and some optional supplemental enhancement information (SEI) messages. To support luminance levels in the region of 0 to 10,000 cd/m^2, ST.2084 defines a highly non-linear transfer function [3] to achieve perceptually transparent quantization.

Several MPEG documents identified a problem with subjective quality of HDR Y'CbCr non-constant luminance 4:2:0 color format [5], [6], [7]. As was suggested in [6], [7], the problem is likely to be caused by a steep slope of the opto-electrical transfer function (OETF) in the low-luminance range and the following color transform, which makes color components with low values have significant impact on Y', Cb, and Cr values. This may cause artifacts in saturated colors at the boundaries of the color gamut.

A solution to the problem described above was first proposed in [8], [9]. The solution was to downsample and upsample chroma components to simulate the process of chroma reconstruction and then iterate over different values of luma to find the value that results in the linear luminance closest to the one of the original pixel. A bisection method was applied, which enabled getting the result in ten iterations for each luma sample of a 10-bit signal. The iterations required computing a transfer function (or using a look-up table (LUT)) and applying a color transformation(s), and could therefore be rather slow even when the transfer function was approximated with a LUT.

Two closed form solutions to this problem have been proposed in [10], [11], and [12]. The solutions calculate the value of the luma sample in one step and attenuate artifacts in saturated colors, therefore improving subjective and objective quality of the video. Subjectively, the results look very similar to the iterative method results from [9] (and even visually identical for one of the closed form solutions). Regarding the objective metrics, the closed form solutions improve the results compared to the direct chroma subsampling, one of them yielding close results to the iterative method over a range of objective metrics.

The paper is organized as follows. Section 2 explains the motivation for this work. Section 3 describes the studied algorithms. Section 4 presents experimental results including both the subjective and objective quality while Section 5 addresses the computational complexity. Finally, Section 6 concludes the paper.

2. POTENTIAL ARTIFACTS IN HDR10

2.1 Y'CbCr 4:2:0 color conversion in HDR10 standard

The HDR10 standard [1] employs a so-called non-constant luminance approach, which is defined in the ITU-R Recommendation BT.2020 [4]. Strictly speaking, HDR10 standard specifies operations in the receiver/decoder, which, however, partially determine the encoding process. The processing chain for HDR10 is shown in Figure 1.

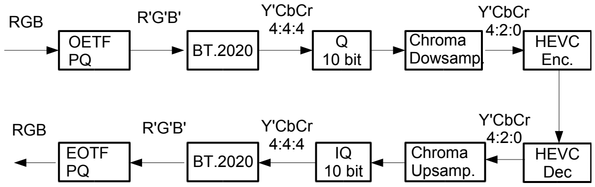


Figure 1. HDR10 processing chain.

The HDR10 processing can be summarized as follows. The OETF, which is the inverse of the electro-optical transfer function (EOTF) ST.2084 [3], is applied separately to the original linear light R, G, and B components yielding the R', G', and B' values. In the next step, Y'CbCr values are obtained by applying a color transform as in non-constant luminance BT.2020 [4]. Chroma components Cb and Cr are subsequently downsampled by a factor of two in vertical and horizontal dimensions. The decoding and display process is the inverse of that.

The inverse of the ST.2084 perceptual quantizer (PQ) transfer function, which is used to transform linear light to the transfer function domain, is shown below.

$$PQ\_TF(L) = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_3 L^{m_1}} \right)^{m_2}; \quad m_1 = 0.1593017578125,$$
$$m_2 = 78.84375, \quad c_1 = 0.8359375, \quad c_2 = 18.8515625, \quad c_3 = 18.6875. \tag{1}$$

The OETF function is chosen based on the higher human visual system sensitivity to changes in luminance when the luminance level is low. That is why the OETF allocates more code words to samples in the low luminance region.

The R'G'B' to Y'CbCr non-constant luminance color transform in BT.2020 is performed as follows:

$$Y' = 0.2627\, R' + 0.6780\, G' + 0.0593\, B';$$
$$Cb = (B' - Y') / 1.8814; \; Cr = (R' - Y') / 1.4746. \tag{2}$$

2.2 Subjective quality problems in HDR non-constant luminance Y'CbCr 4:2:0 format

It has been reported in [5], [6], and [7] that subsampling chroma components in non-constant luminance Y'CbCr can cause significant distortions in colors which are close to the color gamut boundary. These artifacts look like details that are not present in the original video and can significantly decrease the picture quality. Figures 3 (b), 4 (b), and 5 (b) show examples of these subjective quality artifacts. Figures 3 (a), 4 (a), and 5 (a) show the original video. The HDR sequences used in the experiments have luminance level reaching 4000 nits. To produce Figures 3 – 8, a tone mapping process was applied. It should be noted that the test sequences were produced using either BT.709 [13] or DCI P3 color primaries. Therefore, BT.709 color primaries and color transform coefficients are used with BT.709 content to model the case when color values are close to the color gamut boundaries, using the same approach as in MPEG HDR and Wide Color Gamut (WCG) Call for Evidence [14].

It was suggested by the authors of [6] that the reasons for the artifacts is the steep slope of the OETF function in the area close to the zero (see Figure 2) and that the OETF is applied to each color component separately. If one component has a value close to zero while other components have higher values (like in colors close to the gamut boundaries), a small intensity component has disproportionally strong effect on Y', Cb, and Cr values compared to if (2) was applied to the linear R, G, and B values. Therefore, small variations in the value of that small component may cause significant variations in Y', Cb, and Cr values. When chroma subsampling is applied, it has an effect of smoothing the reconstructed chroma values, while luma values remain significantly different. When the inverse transform and EOTF are applied, these pixels are reconstructed to significantly different values (as in Figures 3 (b), 4 (b), and 5 (b)).

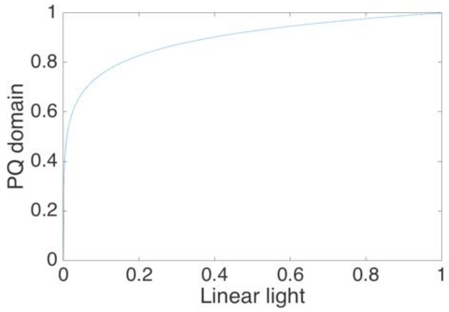


Figure 2. OETF (inverse of ST.2084 PQ). OETF function has steep slope in the region close to 0 nits.

3. DESCRIPTION OF ALGORITHMS

The algorithms shown below can mitigate the problem stated in Subsection 2.2. In general, these algorithms work as follows. First, having a linear RGB input, it is converted to the transfer function domain R'G'B' and then to Y'CbCr 4:4:4 (and quantized). Then, chroma components Cb and Cr are downsampled and upsampled back to 4:4:4 to simulate the reconstruction process after the decoding. This process is based on two assumptions, namely, that the upsampling filter after the decoding would produce similar results to the filter used in the pre-processing and that the chroma values are not significantly modified by compression. Then algorithms estimate a value of Y' that should result in a reconstruction to the linear light $\mathrm{RGB}_{\mathrm{new}}$ similar to this of the original image or video, i.e. $\mathrm{RGB}_{\mathrm{org}}$, according to a chosen distortion metric. Changing only the value of Y' while keeping Cb and Cr constant helps reducing the dimensionality of the problem. The distortion between two linear RGB pixels is denoted as

$$D = D\ (\ RGB_{new}(x,y) - RGB_{org}(x,y)\ ), \tag{3}$$

where *x* and *y* are horizontal and vertical positions of the sample respectively. Depending on the chosen metric *D* and a way to solve the optimization problem, different solutions can be obtained. In the following, three algorithms are described, the first based on the iterative approach, the second is a closed form solution minimizing the sum of the squared differences in R, G, and B components, and the third is the closed form solution based on the squared weighted sum of errors between the components.

3.1 Iterative approach

An iterative solution to the problem described in Subsection 2.2 was proposed by Ström et al. in [8], [9]. The solution used the distortion metric in the form of the difference between the linear luminance values of the original and the reconstructed pixels. At each step, the algorithm would pick a luma value Y', then convert the Y'CbCr to R'G'B', apply the EOTF to yield the linear RGB, and then calculate the linear luminance $\mathrm{Y}_{\mathrm{new}}$, which would be compared to the original linear luminance $\mathrm{Y}_{\mathrm{org}}$. A bisection search method could be used to find $\mathrm{Y}_{\mathrm{new}}$ instead of trying all possible luma values, which enabled getting the result in ten iterations for each luma sample of a 10-bit signal. The iterations required computing a transfer function (or using a look-up table) and applying color transformations at each step and could therefore be rather slow even when the transfer function was approximated with a LUT. Another speed optimization was to estimate the initial luma value boundaries before starting the bisection search, which allowed reducing the average number of iterations per pixel. Such optimization, however, did not decrease the worst-case complexity (and even increased it by adding the operations for the boundaries estimation), which may be a problem for real-time and hardware applications.

3.2 Closed form solution based on sum of squared component errors (Closed Form 1)

In this approach, the distance between two RGB values is measured as the sum of the squared errors between each of the R, G, and B linear light components, i.e.

$$D = (\ R_{new}(x,y) - R_{org}(x,y)\ )^2 + (\ G_{new}(x,y) - G_{org}(x,y)\ )^2 + (\ B_{new}(x,y) - B_{org}(x,y)\ )^2. \tag{4}$$

In a more general case, the importance of each color component R, G, and B can be reflected in a weighting factor w_{X}, where X corresponds to a color component, i.e. w_{R}, w_{G} and w_{B}. After omitting pixel coordinates for simpler notation, the cost function would look as follows:

$$D = w_R\ (f(R'_{new}) - f(R'_{org})\ )^2 + w_G\ (f(G'_{new}) - f(G'_{org}))^2 + w_B\ (f(B'_{new}) - f(B'_{org}))^2, \tag{5}$$

where $f(\mathrm{X})$ denotes the EOTF of X. The transfer function domain values R', G' and B' can be obtained from Y'CbCr by applying the inverse color transform, which depends on the color space and in case of Y'CbCr in BT.709 and BT.2020 has the following form:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1 & a_{1,2} & a_{1,3} \\ 1 & a_{2,2} & a_{2,3} \\ 1 & a_{3,2} & a_{3,3} \end{pmatrix} \begin{pmatrix} Y' \\ Cb \\ Cr \end{pmatrix} \tag{6}$$

For an EOTF with a somewhat complex formula (such as ST.2084) finding a closed form solution for minimizing the cost function (5) may be difficult. In order to obtain a solution, the EOTF $f(\ \mathrm{X}\ )$ can be approximated with the first degree truncated Taylor series, i.e.

$$f(X_i + \Delta) = f(X_i) + f'(X_i)\ \Delta\ , \tag{7}$$

where $f'(\mathrm{X}_i)$ is the value of the derivative of the EOTF $f(\mathrm{X})$ with respect to X at point X_i. Substituting (7) into (5), the cost function is approximated as follows:

$$D = w_R\ (f'(R'_{org})\ \Delta_R\ )^2 + w_G\ (f'(G'_{org})\ \Delta_G)^2 + w_B\ (f'(B'_{org})\ \Delta_B)^2. \tag{8}$$

Then, we substitute Δ_{R} in (8) with ($\mathrm{Y}'_{\mathrm{new}} - e_{\mathrm{R}}$), do similar substitutions for Δ_{G} and Δ_{B}, differentiate the cost function D with respect to Y', set the derivative equal to zero, and solve the resulting equation with respect to Y'. The resulting solution for Y' is then obtained as follows. First, we calculate e_{R}, e_{G}, and e_{B}:

$$\begin{aligned} e_R &= Y'_{org} - (Cb_{\mathrm{new}} - Cb_{org})\ a_{1,2} - (Cr_{new} - Cr_{org})\ a_{1,3}\ , \\ e_G &= Y'_{org} - (Cb_{\mathrm{new}} - Cb_{org})\ a_{2,2} - (Cr_{new} - Cr_{org})\ a_{2,3}\ , \\ e_B &= Y'_{org} - (Cb_{new} - Cb_{org})\ a_{3,2} - (Cr_{new} - Cr_{org})\ a_{3,3}\ . \end{aligned} \tag{9}$$

Then, the value of Y' is equal to

$$Y'_{new} = \frac{w_R f'(R'_{org})^2 e_R + w_G f'(G'_{org})^2 e_G + w_B f'(B'_{org})^2 e_B}{w_R f'(R'_{org})^2 + w_G f'(G'_{org})^2 + w_B f'(B'_{org})^2} \tag{10}$$

or, if weights $\mathrm{w_R}$, $\mathrm{w_G}$, and $\mathrm{w_B}$ are all equal to 1, the Y' can be obtained as follows:

$$Y'_{new} = \frac{f'(R'_{org})^2 e_R + f'(G'_{org})^2 e_G + f'(B'_{org})^2 e_B}{f'(R'_{org})^2 + f'(G'_{org})^2 + f'(B'_{org})^2}\ . \tag{11}$$

The values of the EOTF derivative squared $f'(\mathrm{X})^2$ can be pre-computed and stored in a look-up table. The proposed method can work with various transfer functions, including ST.2084 [3] and BT.1886 [15]. The EOTF derivative can be obtained either by differentiating the EOTF or approximating the derivative numerically, for example by its definition (i.e. dividing the change in the function value by the change in the argument). The method can also be applied to different color spaces. Weights $\mathrm{w_R}$, $\mathrm{w_G}$, and $\mathrm{w_B}$ in expressions (5) – (10) can be set equal to one or chosen based on the desired precision or importance of each component.

3.3 Closed form solution based on squared weighted sum of component errors (Closed Form 2)

The third method uses the metric that is a squared sum of weighted errors between linear R, G, and B components of the new and the original RGB values. This cost function is defined as follows.

$$D = (\ w_R\ (\ R_{new} - R_{org}) + w_G\ (G_{new} - G_{org}) + w_B\ (B_{new} - B_{org})\ )^2, \tag{12}$$

which can be written as

$$D = (\ w_R\ (f(R'_{new}) - f(R'_{org})) + w_G\ (f(G'_{new}) - f(G'_{org})) + w_B\ (f(B'_{new}) - f(B'_{org}))\ )^2, \tag{13}$$

where $f(\mathrm{X})$ denotes the EOTF.

Minimizing this distortion function can be shown to be equivalent to minimizing the absolute value of the difference between the weighted sums of R, G, and B because the absolute value and the square functions reach their minima at the same point. In case the weights w_R, w_G, and w_B are set equal to the contributions of R, G, and B to linear luminance, optimizing for this cost function would be equivalent to optimizing for the error in linear luminance, which was used in the iterative algorithm in Section 3.1.

In order to obtain a closed form solution for estimating Y', the EOTF $f(X)$ is approximated with a first degree Taylor polynomial as in (7), which results in the following expression for the cost function D:

$$D = \left(w_R f'(R'_{org})\, \Delta_R + w_G f'(G'_{org})\, \Delta_G + w_B f'(B'_{org})\, \Delta_B \right)^2. \tag{14}$$

Then, we substitute Δ_R in (14) with ($Y'_{new} - e_R$), do similar substitutions for Δ_G and Δ_B, differentiate cost function D with respect to Y', set the derivative equal to zero, and solve the resulting equation. The obtained solution is as follows. First, we calculate e_R, e_G, and e_B as in (9)

$$e_R = Y'_{org} - (Cb_{new} - Cb_{org})\, a_{1,2} - (Cr_{new} - Cr_{org})\, a_{1,3}\,,$$

$$e_G = Y'_{org} - (Cb_{new} - Cb_{org})\, a_{2,2} - (\mathrm{Cr}_{\mathrm{new}} - Cr_{org})\, a_{2,3}\,,$$

$$e_B = Y'_{org} - (Cb_{new} - Cb_{org})\, a_{3,2} - (\mathrm{Cr}_{\mathrm{new}} - Cr_{org})\, a_{3,3}\,. \tag{15}$$

The value of Y' is then equal to:

$$Y'_{new} = \frac{w_R f'(R'_{org}) e_R + w_G f'(G'_{org}) e_G + w_B f'(B'_{org}) e_B}{w_R f'(R'_{org}) + w_G f'(G'_{org}) + w_B f'(B'_{org})}\,. \tag{16}$$

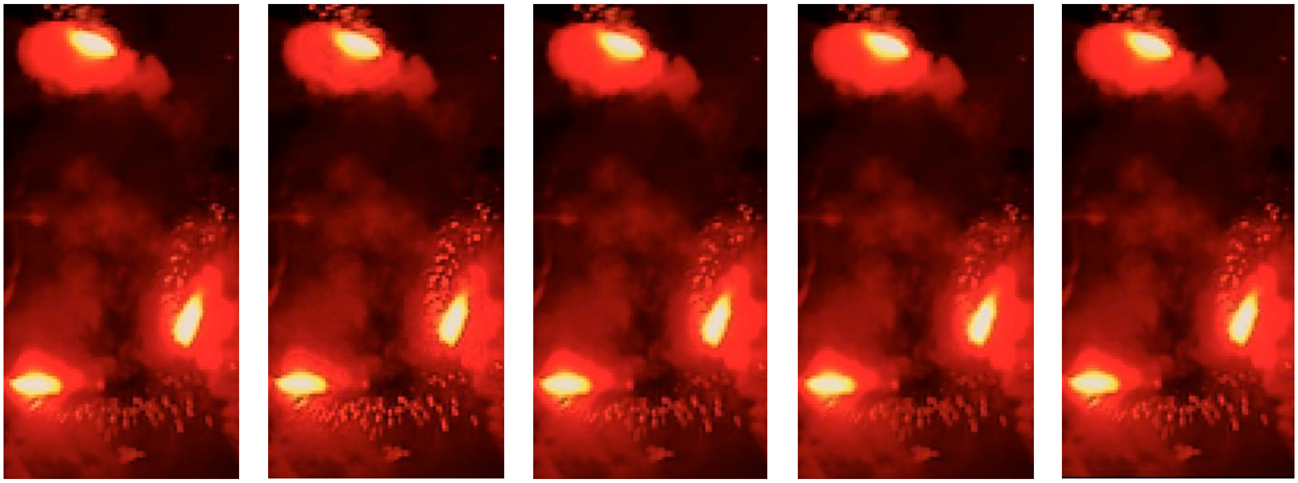
Values of the EOTF derivative $f'(X)$ can be pre-computed and stored in a look-up table. The weights w_R, w_G, and w_B in the expressions can be chosen based on the desired precision or importance of each component. For example, they can be set equal to one or based on the contribution of each color component to the luminance.

4. EXPERIMENTAL RESULTS

This section compares the objective and subjective performance of the studied algorithms. The algorithms from Subections 3.2 and 3.3 (Closed Form 1 and Closed Form 2, respectively) have been implemented in the HDRTools 0.11-dev software [16], which is used in HDR experiments in the Joint Collaborative Team on Video Coding (JCT-VC), a joint committee of ITU-T and MPEG. The Iterative algorithm [9] (briefly described in Section 3.1) has been implemented in the HDRTools software by algorithm's authors. The iterative algorithm can be turned on by setting a parameters closedLoopConversion=5 in the HDRTools configuration file, Closed Form 1 solution is turned on by setting closedLoopConversion=16, and Closed Form 2 by setting closedLoopConversion=17.

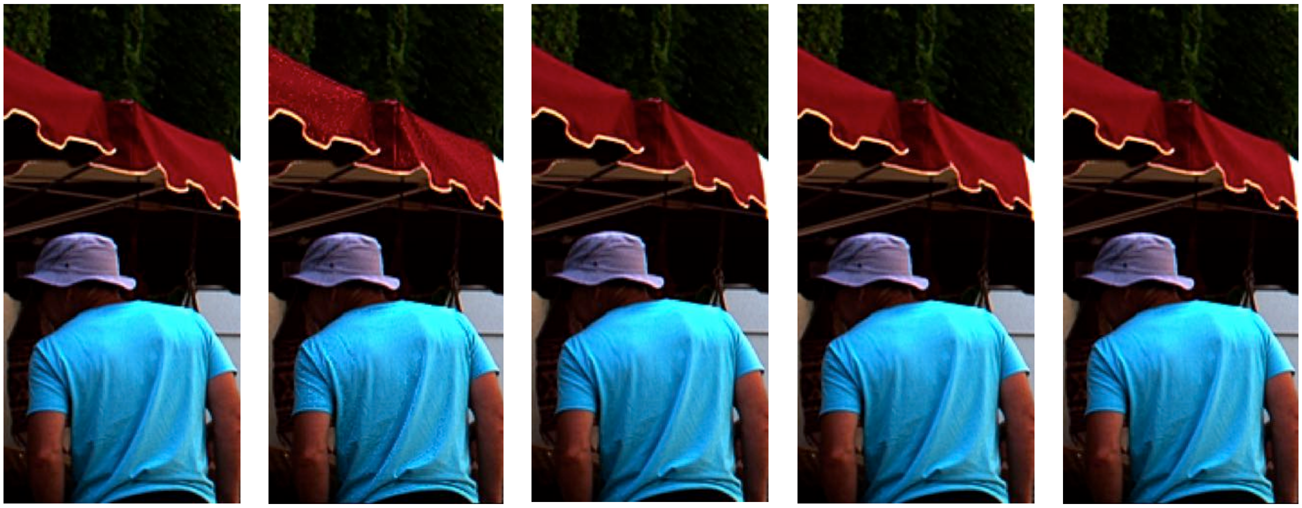
JCT-VC common test conditions for HDR [17] have been used in the experiments except the sequences StEM Magic Hour and StEM Warm Night, which were not available at the time of simulations. In addition to the JCT-VC common test conditions results, which use the BT.2020 container for coding HDR video, the objective results on BT.709 sequences using the BT.709 container have also been obtained to simulate the scenario when the video contains colors, which are close to the color gamut boundary. These BT.709 sequences used in the experiments (Tables 1 and 3) are: *Market, FireEater, EBUHurdles, EBUStarting, BaloonFestival*, and *Sunrize*. The simulations in BT.2020 container (Tables 2 and 4) additionally use the following P3 sequences: *GarageExit, ShowGirl, BikeSparklers1*, and *BikeSparklers2*. In all simulations, the weights w_R, w_G, and w_B in the Closed Form 1 solution have been set equal to 1 whereas in the Closed Form 2 solution, weights w_R, w_G, and w_B have been set equal to the contribution of R, G, and B components to the linear luminance.

An additional test has been performed on using a different chroma upsampling filter in the reconstruction than the filter that was used in the pre-processing. In this experiment, the pre-processing uses the chroma upsampling filter from the JCT-VC common test conditions [17], whereas the reconstruction uses the MPEG-2 upsampling filter (which can be turned on by setting ChromaUpsampleFilter=6 in the HDRTools configuration file). The pre-processing chroma upsampling filter (for a phase equal to 0.5) is (–16, 144, 144, –16)/256. The chroma upsampling filter used in the reconstruction is (21, –52, 159, 159, –52, 21) / 256.



(a) Original (b) Direct subsampling (c) Iterative (d) Closed Form 1 (e) Closed Form 2

Figure 3. Sequence FireEater. Comparison of four chroma subsampling schemes with the original. Direct subsampling (b) shows artifacts, solutions (c) – (e) closely resemble the original.



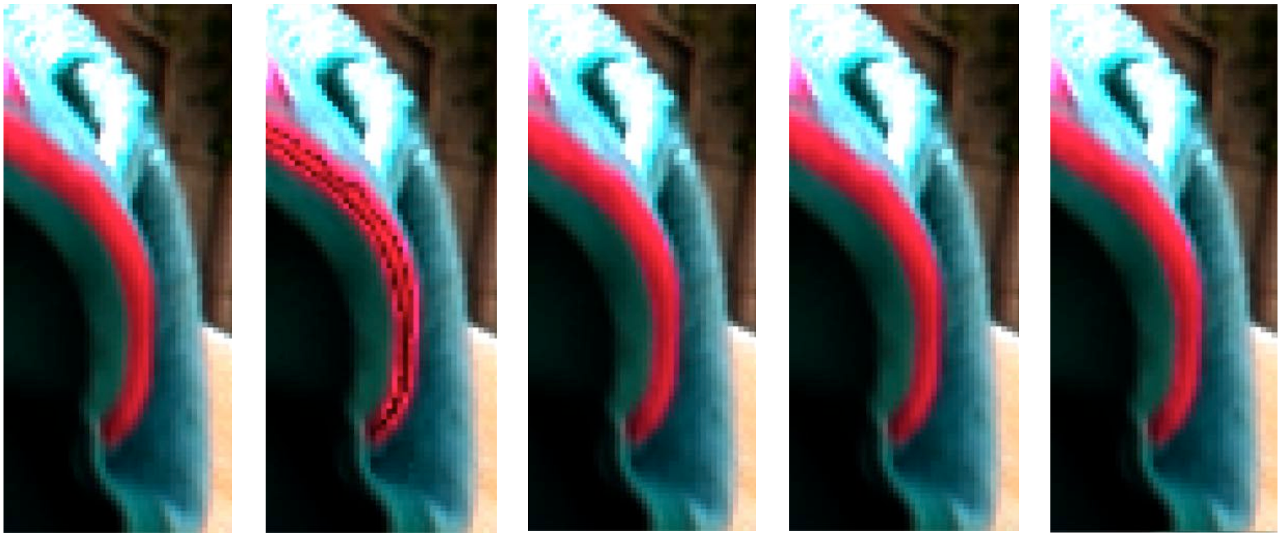
(a) Original (b) Direct subsampling (c) Iterative (d) Closed Form 1 (e) Closed Form 2

Figure 4. Sequence Market. Comparison of four chroma subsampling schemes with the original. Direct subsampling (b) shows artifacts, solutions (c) – (e) closely resemble the original.

Figures 3 – 5 demonstrate the effect of the algorithms on artifacts in the FireEater and Market sequences. One can see that the pictures (c), (d), and (e) resulting from applying the algorithms from Subsections 3.1, 3.2, and 3.3, respectively, are much closer visually to the original (a) than the results of the direct chroma subsampling (b). Another observation (not shown in the figures) is that the algorithms from Section 3 produce smoother luma component compared to the direct chroma subsampling. Therefore, gains in the subsequent compression of the pre-processed video can be expected.

The reported objective metrics are the ones used in the JCT-VC HDR experiments. The description of the metrics can be found in [14], for more details on the metrics' implementation the readers are referred to [16]. In particular, tPSNR calculates the distortion in the XYZ color space in the transfer function domain. The DE0100 metric is a CIEDE2000 based metric [18] and takes into accounts different eye sensitivity for changes in different colors.

The objective results (averages over the sequences) are provided in Tables 1 – 4. Table 1 shows the results for BT.709 content in BT.709 container, whereas Table 2 shows the results for the BT.709 and P3 content in BT.2020 container. Table 3 and 4 make the same comparison except that the chroma upsampling filter used in reconstruction is different from the one used in pre-processing. It can be observed from the results that Closed Form 2 solution closely



(a) Original (b) Direct subsampling (c) Iterative (d) Closed Form 1 (e) Closed Form 2

Figure 5. Sequence Market. Comparison of four chroma subsampling schemes with the original. Direct subsampling (b) shows artifacts, solutions (c) – (e) closely resemble the original.

Table 1. BT.709 in BT.709 container (averages over BT.709 sequences).

Algorithm	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	50.96	55.04	47.97	50.30	50.29	39.28	22.59	45.48
Iterative	56.51	69.78	47.64	51.83	51.35	39.76	22.65	49.63
Closed Form 1	59.37	54.02	46.79	50.55	50.64	39.90	22.47	44.13
Closed Form 2	56.50	67.02	47.61	51.79	51.33	39.76	22.59	49.29

Table 2. BT.2020 container (averages over BT.709 and P3 sequences).

Algorithm	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	52.65	62.76	45.52	49.41	48.27	38.11	22.67	48.36
Iterative	54.33	69.66	45.22	49.45	48.39	38.20	22.68	50.27
Closed Form 1	56.59	54.72	46.05	49.77	49.07	38.27	22.64	44.95
Closed Form 2	54.36	68.46	45.22	49.45	48.39	38.20	22.66	50.03

Table 3. Mismatch in upsampling filters in pre-processing and reconstruction. BT.709 in BT.709 container (averages over BT.709 sequences).

Algorithm	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	51.17	55.24	48.09	50.45	50.34	39.27	22.38	45.49
Iterative	55.87	64.56	47.72	51.77	51.18	39.68	22.42	48.33
Closed Form 1	56.80	53.36	46.81	50.33	50.25	39.71	22.24	43.78
Closed Form 2	55.86	63.38	47.68	51.72	51.16	39.68	22.37	48.13

Table 4. Mismatch in upsampling filters in pre-processing and reconstruction. BT.2020 container (averages over BT.709 and P3 sequences).

Algorithm	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	52.82	62.95	45.66	49.55	48.34	38.12	22.69	48.39
Iterative	54.37	68.43	45.36	49.57	48.44	38.21	22.69	49.90
Closed Form 1	55.78	54.50	46.11	49.78	48.90	38.25	22.66	44.84
Closed Form 2	54.40	67.32	45.36	49.57	48.44	38.21	22.68	49.68

resembles the performance of the iterative algorithm [9]. One can also notice that the performance drop because of the mismatch between the chroma upsampling filters in pre-processing and reconstruction is larger than the difference between the Iterative algorithm and Closed Form 2 solution.

Finally, it has been observed that the upsampling filter mismatch in pre-processing and reconstruction sometimes causes subjective artifacts, often in the same areas, which have objectionable artifacts in case of the direct chroma subsampling. Figures 6 – 8 demonstrate this problem. It should be noted, however, that the artifacts due to upsampling filters mismatch are significantly smaller than the artifacts that appear when using the direct chroma subsampling. It is possible that the artifacts due to the filters mismatch can be attenuated by considering different filters at the pre-processing stage and choosing the one that results in smaller distortion over the range of filters used in the reconstruction. However, this is a subject of further investigation.

5. COMPLEXITY ESTIMATION

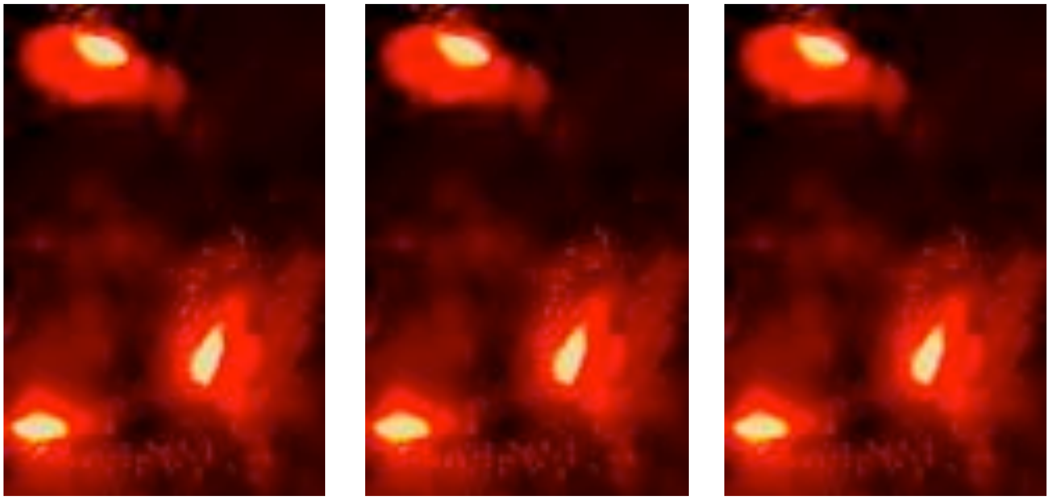
To compare the complexity of Closed Form 1 and 2 solutions with the worst-case complexity of the Iterative algorithm [9], an approximate number of operations for each approach has been reported in Table 5. Closed Form 2 solution uses different weights w_R, w_G, and w_B while Closed Form 1 sets them equal to 1. The number of operations required for color space conversion, downsampling and upsampling of chroma components is not included because it depends on the choice of up- and downsampling filters and on the implementation details. For example, when chroma is co-located with (0,0) luma position, as in HDR10 [1], chroma down- and up-sampling steps can be combined since 4:2:0 chroma samples are co-located with (and equal to) the 4:4:4 chroma samples at these positions. The numbers in Table 5 take into account some obvious optimizations, such as avoiding to perform the same operations more than once.

Table 5. Comparison of the number of operations per luma sample of Closed Form 1, Closed Form 2 solutions and the worst case (10 iterations) of the Iterative algorithm [9].

Algorithm	Adds	Mults	Divs	Table look-ups, (TF or TF deriv.).	Compari-sons	Shifts (div by 2)
Iterative algorithm	55	69	0	30	10 (70)*	10
Closed Form 1 (no weights)	10	9	1	3	0 (2)*	0
Closed Form 2 (with weights)	10	12	1	3	0 (2)*	0

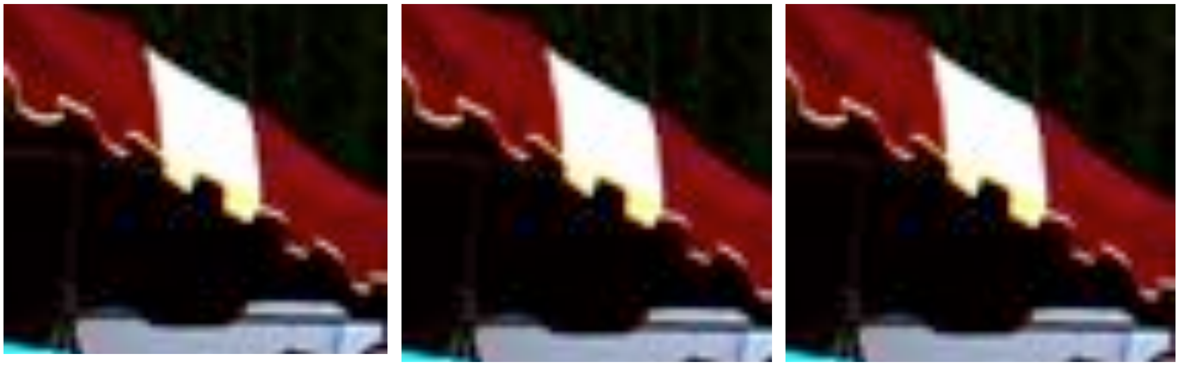
* When clipping of R', G', and B' in Iterative algorithm to the range (0, 1) or Y' in closed form solution is implemented with comparison operations and included in operations count.

One can see from Table 5 that the worst case complexity of the Iterative algorithm is significantly higher than the complexity of the closed form solutions, which find the value of Y' in one iteration and spend a fixed number of operations for each pixel irrespectively of the input. The iterative approach [9], if run with the highest accuracy, uses up to 10 iterations for a 10-bit video, which includes a color transform, applying EOTF (can be approximated with a LUT), and calculating the luminance at each iteration. A number of iterations in the Iterative algorithm [9] may be limited to a number smaller than 10 at the expense of decreasing the algorithm performance. The average number of operations per sample can also be decreased compared to the worst case complexity by computing tighter initial bounds in order to get the result in fewer iterations. This, however, would add additional operations to the worst-case complexity. It can be concluded that Closed Form 1 and 2 solutions have good complexity/quality trade-off and a constant number of operations per sample, which makes them well suited for use in real-time and hardware systems.



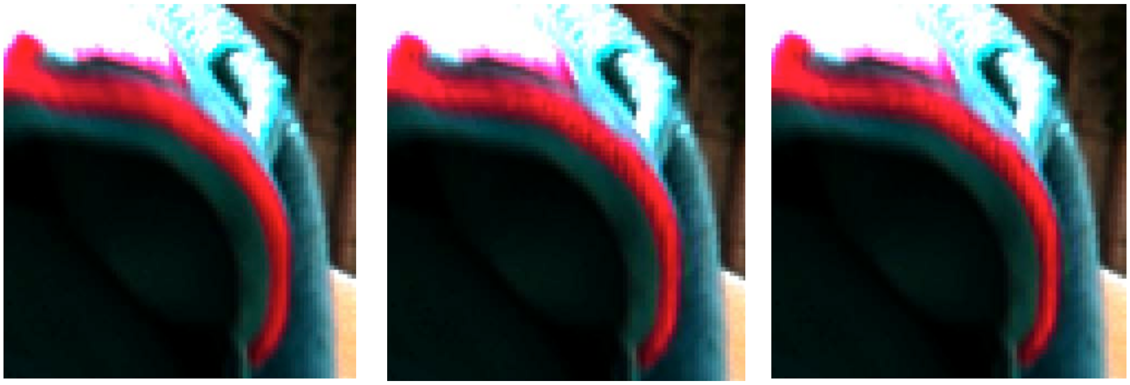
Original Iterative Closed Form 1

Figure 6. Sequence FireEater. Mismatch between chroma upsampling filters. Comparison of Iterative algorithm and Closed Form 1 solution with the original.



Original Iterative Closed Form 2

Figure 7. Sequence Market. Mismatch between chroma upsampling filters. Comparison of Iterative algorithm and Closed Form 2 solution with the original.



Original Iterative Closed Form 1

Figure 8. Sequence Market. Mismatch between chroma upsampling filters. Comparison of Iterative algorithm and Closed Form 1 solution with the original.

6. CONCLUSIONS

The paper studies performance of three algorithms, which attenuate subjective artifacts that may appear in Y'CbCr 4:2:0 HDR color format. It has been found that the subjective performance of two closed form solutions closely resembles the quality obtained with the iterative algorithm [9]. The objective performance of Closed Form 2 solution is also very close to the performance of the iterative algorithm.

It has been found that the choice of the upsampling filter in the decoder is important for the overall system performance. A mismatch between the upsampling filters used in the pre-processing and reconstruction stages may lead to objective and subjective quality degradation. In particular, the difference between the performance of Closed Form 2 solution and the Iterative algorithm [9] is smaller than the drop in performance of either of these two algorithms in some cases of the mismatch between the upsampling filters in reconstruction and pre-processing. The visual artifacts in case of the upsampling filters mismatch have also been observed in all three algorithms. The artifacts, however, are significantly smaller than those when the direct chroma subsampling is used.

The results therefore indicate that the closed form solutions, especially the Closed Form 2, can be used as a practical alternative to the iterative algorithm, especially in the case when a low-complexity algorithm with a fixed number of operations per sample is desirable.

ACKNOWLEDGEMENTS

The author would like to thank A. Tourapis (Apple) for providing HDRTools software package [16], authors of [9] for implementing their algorithm in the HDRTools software, which made comparison with their algorithm easily feasible, and Dolby, Technicolor, EBU, Stuttgart Media University, and CableLabs for making their HDR test sequences available.

REFERENCES

[1] DECE, "Common File Format & Media Formats Specification version 2.1", Digital Entertainment Content Ecosystem (DECE).

[2] ITU-T H.265 and ISO/IEC 23008-2, “High efficiency video coding” (Apr. 2015).

[3] SMPTE ST 2084: “High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays”.

[4] Recommendation ITU-R BT.2020: “Parameter values for ultra-high definition television systems for production and international programme exchange”.

[5] E. François, P. Lopez, Y. Olivier, (Technicolor) “About using a BT.2020 container for BT.709 content”, ISO/IEC (MPEG) document m35255, Sapporo (July 2014).

[6] J. Stessen, R. Nijland, R. Brondijk, R. Goris, “Chromaticity based color signals for wide color gamut and high dynamic range”, ISO/IEC (MPEG) document m35065, Strasbourg, France (Oct. 2014).

[7] J. Ström, “Investigation of HDR color subsampling”, ISO/IEC (MPEG) document m35841, Geneva, Switzerland (Feb. 2015).

[8] J. Ström, J. Samuelsson, M. Pettersson, K. Andersson, P. Wennersten, R. Sjöberg, “Ericsson's response to CfE for HDR and WCG”, ISO/IEC (MPEG) document m36184, Warsaw, Poland (June 2015).

[9] J. Ström, J. Samuelsson, and K. Dovstam, ”Luma Adjustment for High Dynamic Range Video”, in Proc. IEEE Data Compression Conf. (DCC), Snowbird, UT, USA (March 2016).

[10] A. Norkin, “Closed form HDR 4:2:0 chroma subsampling (HDR CE1 and AHG5 related)”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-W0107, San Diego, CA, USA (Feb. 2016).

[11] A. Norkin, "Fast algorithm for HDR color conversion", in Proc. IEEE Data Compression Conf. (DCC) 2016, Snowbird, UT, USA (March 2016).

[12] A. Norkin, “On closed form HDR 4:2:0 chroma subsampling (AHG13 related)”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-W0107, Geneva, Switzerland (May – June 2016).

[13] Recommendation ITU-R BT.709: “Parameter values for the HDTV standards for production and international programme exchange”.

[14] Call for Evidence (CfE) for HDR and WCG Video Coding, ISO/IEC (MPEG), http://mpeg.chiariglione.org/standards/exploration/high-dynamic-range-and-wide-colour-gamut-content-distribution/call-evidence

[15] Recommendation ITU-R BT.1886: "Reference electro-optical transfer function for flat panel displays used in HDTV studio production".

[16] HDRTools software package (Apple). Alexis M. Tourapis. https://gitlab.com/standards/HDRTools (May 2016).

[17] E. François, J. Sole, J. Ström, P. Yin, "Common Test Conditions for HDR/WCG Video Coding Experiments", ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-W1020, San Diego, CA, USA (Feb. 2016).

[18] Joint ISO/CIE Standard: Colorimetry — Part 6: CIEDE2000 Colour-Difference Formula, ISO/CIE 11664-6:2014(E).