Fast algorithm for HDR video pre-processing

Andrey Norkin
Netflix Inc.
Los Gatos, CA, USA
anorkin@netflix.com

***Abstract*—The paper addresses a problem of perceptual artifacts that appear in saturated colors of Y'CbCr non-constant luminance 4:2:0 HDR video. A computationally inexpensive method is proposed for converting 4:4:4 HDR video to Y'CbCr 4:2:0 non-constant luminance format. The method shows similar objective performance to an iterative algorithm and outperforms a previously published explicit solution. The method removes artifacts in areas with saturated colors, obtaining results in one step. Compared to the iterative algorithm, the proposed solution significantly decreases the worst-case and average complexity.**

I. INTRODUCTION

High Dynamic Range (HDR) is considered a major improvement in quality of television pictures. To facilitate transport and broadcasting of HDR video, a Digital Entertainment Content Ecosystems (DECE) has adopted a specification, called HDR10 [1] that demands HEVC Main10 encoding, ST.2084 transfer function [5], BT.2020 color space, Y'CbCr 4:2:0 non-constant luminance color format [4], and can use some optional supplemental enhancement information (SEI) messages. To support luminance levels between 0 and 10,000 cd/m^2, ST.2084 defines a highly non-linear transfer function [5] to achieve perceptually unnoticeable quantization.

A combination of non-constant luminance Y'CbCr 4:2:0 with highly non-linear transfer function can sometimes lead to subjective quality problems. Several MPEG contributions identified a problem with subjective artifacts in HDR Y'CbCr non-constant luminance 4:2:0 color format (used in HDR10) [6], [7], [8]. As was suggested in [7], [8] the problem is likely caused by the steep slope of the opto-electrical transfer function (OETF) in the low-luminance range and the color transform, which make 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 proposed in [10], [11]. The solution was to downsample and upsample chroma components to simulate the chroma upsampled in the receiver and iterate over different values of luma to choose the one that results in the linear luminance closest to the one of the original signal. A bisection method was applied, which enabled getting the result in ten iterations for each luma sample for a 10-bit signal. Each iteration required computing a transfer function for each color component (or using a look-up table (LUT)) and applying two color transforms and could therefore be rather slow.

An approximate explicit solution to this problem was proposed in [12], [13]. The solution calculated the value of a luma sample in one step and removed color artifacts in saturated colors, therefore improving subjective quality of the video with output visually similar to that of the iterative method from [11]. The method improved average PSNR and tPSNR [9] compared to straightforward downsampling of chroma planes. Compared to the iterative method of [10], [11], the closed-form solution produced somewhat lower objective metrics results, in particular tPSNR.

This paper presents another explicit solution for HDR Y'CbCr 4:2:0 conversion, which demonstrates objective performance close to the iterative approach [11]. The results of the algorithm are visually indistinguishable on the studied content from those of the iterative approach [11]. The solution has recently been included in a JCT-VC report on HDR conversion and coding practices [15].

The paper is organized as follows. Section II explains motivation for this work, Section III describes the proposed algorithm. Section IV summarizes experimental results including both the subjective and objective quality, while Section V addresses the computational complexity. Finally, Section VI concludes the paper.

II. COLOR ARTIFACTS IN HDR10 PROCESSING CHAIN

A. Y'CbCr 4:2:0 color conversion as in HDR10 standard

The HDR10 standard briefly described above employs a so-called non-constant luminance approach [1], which is defined in the ITU-R Recommendation BT.2020 [4]. As usual in video standards, HDR10 specifies operations in the receiver/decoder, which nevertheless partially define the encoding process. The processing chain for HDR10 is shown in Figure 1.

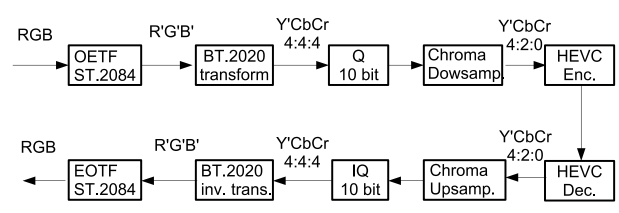


Figure 1. HDR10 processing chain.

HDR10 processing can be summarized as follows. The OETF, which is the inverse of electro-optical transfer function (EOTF), here ST.2048 [5], is applied separately to each of R, G, and B components of the original linear light signal yielding the transfer function domain R', G', and B' values. Then, Y'CbCr signal is obtained by applying a color transformation (which is equivalent to a 3x3 matrix multiplication). Chroma components Cb and Cr are subsequently downsampled in both vertical and horizontal dimensions to exploit the human visual system's (HVS) higher sensitivity to high frequency details in luminance compared to the color. The decoding and display process is the inverse of that. The inverse of the ST.2048 perceptual quantizer (PQ) transfer function, which is used to transform linear light into the transfer function domain is shown below.

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

$m_2 = 78.84375,\ c_1 = 0.8359375,\ c_2 = 18.8515625,\ c_3 = 18.6875.$

The shape of the OETF is chosen based on the fact that HVS is more sensitive to changes in luminance when the luminance level is low. Therefore, ST.2084/PQ 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 applied as follows:

$$Y' = 0.2627\,R' + 0.6780\,G' + 0.0593\,B';$$

$$Cb = (B' - Y') / 1.8814;\ Cr = (R' - Y') / 1.4746. \tag{2}$$

B. Subjective quality problems caused by HDR non-constant luminance conversion

It has been reported in [6], [7], [8] that subsampling chroma components in non-constant luminance Y'CbCr can cause significant distortion in colors that are close to the color gamut boundaries. These artifacts look like additional details or noise that were not present in the original linear light signal. Figures 3(b) and 4(b) show examples of these artifacts, whereas Figures 3(a) and 4(a) show the original video. The HDR sequences used in the experiments have luminance level reaching 4000 nits. To produce Figures 3 and 4, a tone mapping was applied, but the artifacts are also well visible on HDR displays.

It was suggested by the authors of [6] that the reason for the artifacts is a steep slope of the OETF in the area close to zero (see Figure 2) and that the transfer function is applied to each color component separately in the non-constant luminance, which is the most widespread approach in the display and TV equipment. If a color component is close to zero, while other components have higher values (like in colors close to the gamut boundaries), the small intensity component, after highly non-linear mapping to the TF domain, has disproportionally high contribution to Y', Cb, and Cr values compared to as if (2) was applied to the original linear R, G, and B values. Therefore, small variations in the value of that component may cause significant variations of Y', Cb, and Cr values, even though linear RGB values are similar. Subsampling of chroma has a low-pass filtering effect on reconstructed chroma values, while luma values remain significantly different. When the inverse transform and EOTF are applied, these originally similar colors are reconstructed to significantly different values (as in Figures 3(b) and 4(b)).

It should be noted that the Market sequence (Figures 3 and 4) was shot in the BT.709 color gamut as well as many of other publicly available HDR sequences while the HDR10 uses wider BT.2020 primaries. To model the case of colors close to the color gamut boundary, as in wide color gamut (WCG) content, BT.709 color primaries and color transform coefficients have been used with BT.709 content. The same approach was used in the MPEG HDR and WCG Call for Evidence [9].

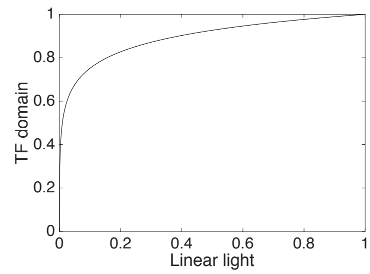


Figure 2. OETF PQ (ST.2084). OETF function has steep slope in the region close to 0 nits.

III. ALGORITHM DESCRIPTION

The proposed algorithm removes the artifacts described in Section II and improves objective metrics performance compared to the explicit solution from [13], while being computationally simpler than the iterative algorithm in [11]. The main difference of the proposed algorithm from the explicit solution presented previously in [13] is the use of a different distortion metric. This distortion metric takes into account human visual system sensitivity to different color components by adjusting the weights w_R, w_G, and w_B.

First, the HDR10 processing flow from Figure 1 is used until the downsampled chroma is obtained. This means that R_{org}, G_{org}, and B_{org} are transformed to R'_{org}, G'_{org}, and B'_{org} using (1). Given R'_{org}, G'_{org}, and B'_{org} components, Y'_{org}, Cb_{org}, and Cr_{org} are obtained by the color transformation as in (2). Then, chroma is subsampled to 4:2:0 representation and quantized. Following that, chroma is de-quantized and upsampled to its original resolution to simulate the reconstruction process in the decoder yielding the Cb_{new} and Cr_{new} values.

Then, the algorithm estimates a luma value Y' that minimizes a chosen distortion metric D

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

between the reconstructed and the original linear light pixel values RGB_{new} and RGB_{org}. As the distortion measure, a squared sum of weighted differences between individual linear light R, G, and B components has been chosen:

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

If weights $w_{R,}$ w_R, and w_B are equal to the contribution of the linear light R, G, and B to luminance, the cost function will measure the squared difference between the new and the original luminance values. Expression (4) can also be written as

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

where $f(X)$ is a EOTF, for example PQ/ST.2084.

Finding a closed-form solution for Y' directly may be difficult because of a non-trivial form of the PQ transfer function. In order to obtain an explicit solution, EOTF $f(X)$ is approximated with a first degree polynomial using the truncated Taylor series expansion:

$$f(X_i + \Delta) = f(X_i) + f'(X_i)\,\Delta, \tag{6}$$

where $f'(X_i)$ is the value of the derivative of $f(X)$ with respect to X at point X_i. Substituting (6) into (5), the cost function is approximated as

$$D = (\, w_R f'(R'_{org})\,\Delta_R + w_G f'(G'_{org})\,\Delta_G + w_B f'(B'_{org})\,\Delta_B\,)^2. \tag{7}$$

Components R', G', and B' in the transfer function domain can be obtained from Y'CbCr (non-constant luminance) as:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} a_{RY} & a_{RCb} & a_{RCr} \\ a_{GY} & a_{GCb} & a_{GCr} \\ a_{BY} & a_{BCb} & a_{BCr} \end{bmatrix} \begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_{RCr} \\ 1 & a_{GCb} & a_{GCr} \\ 1 & a_{BCb} & 0 \end{bmatrix} \begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} \tag{8}$$

After substituting (8) into (7) we get

$$D = (\, w_R f'(R'_{org})\,(\,Y'_{new} - e_R\,)\,)$$
$$+\, w_G f'(G'_{org})\,(\,Y'_{new} - e_G) + w_B f'(B'_{org})\,(\,Y'_{new} - e_G\,)\,)^2, \tag{9}$$

where e_R, e_G, and e_B are defined as follows

$$e_R = Y'_{org} - (Cr_{new} - Cr_{org})\, a_{RCr},$$
$$e_G = Y'_{org} - (Cb_{new} - Cb_{org})\, a_{GCb} - (Cr_{new} - Cr_{org})\, a_{GCr},$$
$$e_B = Y'_{org} - (Cb_{new} - Cb_{org})\, a_{BCb}, \tag{10}$$

Then, to find the local minimum, we differentiate (9) with respect to Y' (note that e_R, e_G, and e_B do not depend on Y'), set the derivative equal to zero, and solve the resulting equation with respect to Y'. The value of Y' can be found as

$$Y'_{new} = \frac{w_R f'(R'_{org}) e_R + w_G f'(G'_{org}) e_G + w_B f'(B'_{org}) e_B}{w_R f'(R'_{org}) + w_G f'(G'_{org}) + w_B f'(B'_{org})} \tag{11}$$

Equations (10) and (11) are applied to each pixel location to get a luma value that minimizes the distortion metric (4) in linear RGB between the original and the reconstructed pixel.

The EOTF derivative $f'(X)$ can be computed explicitly or by the definition of a derivative, i.e. by dividing the change in the function value by the increment of the argument. Alternatively, values of $f'(X)$ can be pre-computed and stored in a look-up table (LUT). The algorithm can be used with any differentiable EOTF including ST.2084.

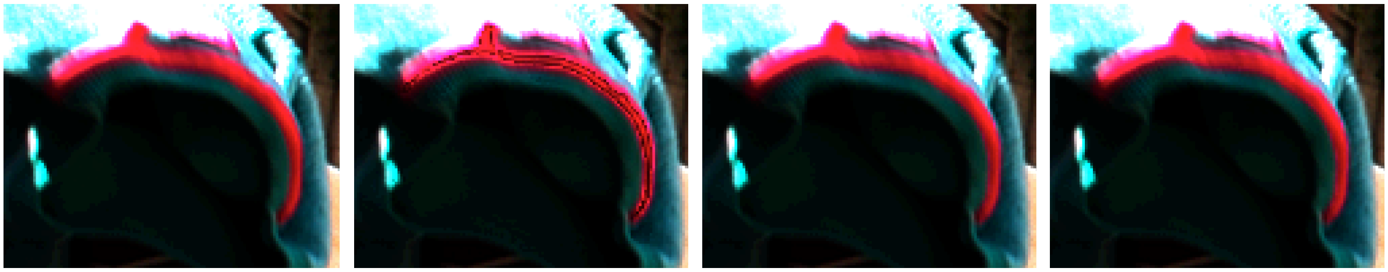
As mentioned earlier, weights w_R, w_G, and w_B can be chosen based on the desired precision or importance of each component. In the experiments described in Section IV the weights are set based on the contribution of each color component to luminance.

IV. Experimental results

The proposed algorithm has been implemented in HDRTools v.0.12 reference software package [17]. The algorithm has been compared to the iterative approach [11], which was implemented by the authors of [11] in the same software and to the explicit solution from [13]. The test conditions are similar to the ones used for HDR proposals evaluation in JCT-VC [14]. The results on BT.709 sequences using BT.709 container have also been obtained to model the case with colors close to the color gamut boundary; the same procedure that was used in the MPEG Call for Evidence (CfE) for HDR and WCG Video Coding [9]. In the simulations, weights w_R, w_G, and w_B were set equal to the contribution of R, G, and B component to the linear luminance Y. The proposed algorithm can be simulated in the HDRTools software package by setting the configuration file parameter closedLoopConversion=17. The explicit solution from [13] can be evaluated by setting this parameter to 16, and the iterative solution by setting it to 5 [11].

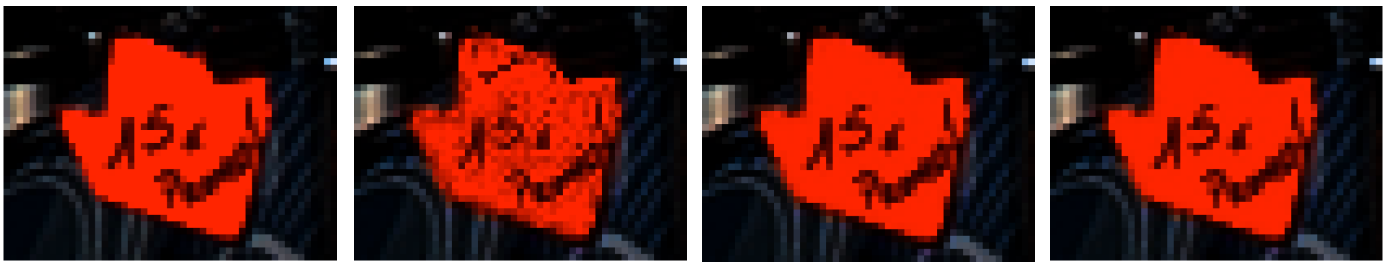
Reported objective metrics are the ones used by JCT-VC for evaluating HDR experiments. The description of the metrics can be found in [9], for implementation details the readers are referred to [17]. In general, tPSNR and tOSNR measures calculate the distortion in XYZ color space and take into account the effect of different HVS sensitivity to luminance variations at different luminance levels by incorporating transfer functions in the error calculation. DE0100 is similar to CIEDE2000 metric [16] and takes into accounts HVS sensitivity to changes in different colors.

Figures 3 and 4 demonstrate the effect of applying the algorithms to sequence *Market*. One can see that the result of the proposed algorithm (d) is much closer visually to the original (a) than the result of the direct chroma subsampling (b). The output of the proposed algorithm is visually identical to that of the iterative algorithm [11] (c) on Figures 3 and 4. The objective results are provided in Tables I - V below. Table I presents the results for BT.709 content in the BT.709 container, whereas Table II shows the results for BT.709 and P3 content in the BT.2020 container according to JCT-VC HDR common test conditions [14]. Tables III - V present sequence-wise results for three algorithms. It can be observed that the proposed algorithm demonstrates close performance to the iterative algorithm [11] and outperforms the previously proposed explicit solution from [13]. The proposed solution outperforms the direct chroma subsampling to the extent of almost 1.5dB in tPSNR on the content filling the color gamut. Results from Table II (where BT.709 and P3 content is not close to BT.2020 boundaries) indicate that the algorithm does not decrease the quality (and slightly improves it) for the content not filling the gamut. It has been observed that the results of the algorithm look visually identical to the iterative algorithm on the studied content. The small gap in objective performance with the iterative algorithm probably happens because of the transfer function approximation with its tangent. One can also notice that the proposed algorithm (in the same way as the iterative algorithm [11] and the explicit solution [13]) produces smoother luma in the areas that are smooth in the original video (compare luma component in Figures 5(a) and 5(b)). The luma component processed with the proposed algorithm has less variation and is likely to be better for compression with a video codec. It can also be seen that the "noise" on the label in Figure 5(a) is not present in the original video frame (Figure 5(a)) and is the result of the non-constant luminance representation.



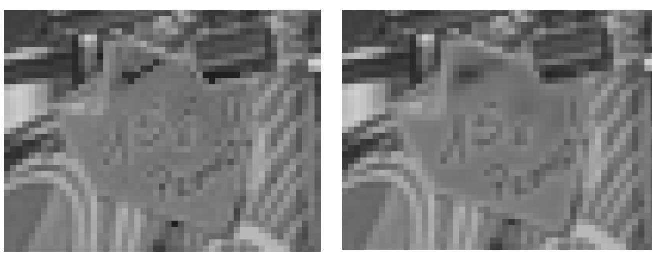
(a) Original (b) Direct subsampling (c) Iterative approach [11] (d) Proposed algorithm

Figure 3. Comparison of the proposed algorithm with iterative one and direct subsampling of chroma. Sequence Market.



(a) Original (b) Direct subsampling (c) Iterative approach [11] (d) Proposed algorithm

Figure 4. Comparison of the proposed algorithm with iterative one and direct subsampling of chroma. Sequence Market.



(a) Direct subsampling (b) Proposed algorithm

Figure 5. Luma component comparison, sequence Market.

TABLE I. BT.709 CONTENT IN BT.709 CONTAINER (AVERAGES)

Algorithm	tPSNR XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	50.35	50.70	39.47	22.41	45.62
Iterat.[11]	51.87	51.76	39.96	22.46	49.63
Explicit [13]	50.59	51.08	40.09	22.26	44.30
Proposed	51.82	51.74	39.96	22.40	49.27

TABLE II. BT.2020 CONTAINER (AVERAGES)

Algorithm	tPSNR XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Direct	48.42	47.34	37.96	22.78	48.72
Iterat.[11]	48.45	47.45	38.05	22.79	50.68
Explicit [13]	48.88	48.19	38.10	22.75	44.89
Proposed	48.45	47.45	38.05	22.77	50.43

TABLE III. BT.709 IN BT.709 CONTAINER (SEQUENCES). DIRECT SUBSAMPLING OF CHROMA

Sequence	tPSNR -XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Market	46.52	48.44	36.82	21.65	44.05
FireEater	55.40	54.29	47.80	24.52	51.29
Hurdles	49.06	47.88	36.87	22.22	42.62
Starting	47.12	47.05	36.27	21.26	44.11
BaloonFest.	48.98	51.67	40.68	20.97	44.53
Sunrize	54.99	54.87	38.38	23.81	47.13
Average	50.35	50.70	39.47	22.41	45.62

TABLE IV. BT.709 IN BT.709 CONTAINER (SEQUENCES). ITERATIVE ALGORITHM [11]

Sequence	tPSNR -XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Market	50.01	49.90	36.98	21.68	47.90
FireEater	57.75	56.19	50.04	24.64	58.67
Hurdles	50.80	49.31	36.99	22.24	47.34
Starting	48.11	47.87	36.33	21.28	47.71
BaloonFest.	49.46	52.34	41.02	20.98	48.25
Sunrize	55.11	54.91	38.40	23.98	47.94
Average	51.87	51.76	39.96	22.46	49.63

TABLE V. BT.709 IN BT.709 CONTAINER (SEQUENCES). PROPOSED ALGORITHM.

Sequence	tPSNR -XYZ	tOSNR-XYZ	DE0100	MD0100	L0100
Market	49.86	49.86	36.97	21.54	47.50
FireEater	57.73	56.17	50.02	24.57	58.32
Hurdles	50.78	49.30	36.99	22.22	47.06
Starting	48.05	47.85	36.33	21.24	46.98
BaloonFest.	49.45	52.33	41.02	20.96	47.86
Sunrize	55.04	54.91	38.40	23.90	47.88
Average	51.82	51.74	39.96	22.40	49.27

TABLE VI. OPERATIONS PER PIXEL IN ITERATIVE ALGORITHM [11] (10 ITERATIONS) AND PROPOSED ALGORITHM

Algorithm	Adds	Mults	Divs	TF (LUT).	Comparisons.	Divs by 2
Iterative [11]	55	69	0	30	10 (70)[1]	10
Explicit [13]	10	9	1	3	0 (2)[1]	0
Proposed	10	12	1	3	0 (2)[1]	0

[1] If clipping to (0, 1) range is implemented with comparison operations.

TABLE VII. RUNTIMES OF THE ALGORITHM IN HDRTOOLS IMPLEMENTATION (BT.709 CONTAINER).

Algorithm	Total running time, all sequences (s)	Ratio over direct subsampling
Direct subsampling	465	100.0%
Iterative [11]	2479	533.4%
Explicit [13]	959	206.3%
Proposed	990	212.9%

V. COMPLEXITY ESTIMATION

The proposed algorithm requires one iteration for each pixel. On the contrary, the iterative approach from [11] uses up to ten iterations for a 10-bit video, which include obtaining R'G'B' values, applying EOTF (can be approximated with a LUT), and calculating linear light luminance. Calculation of EOTF derivative $f'(X)$ can also be approximated with a LUT. In this case, the proposed approach and explicit approach from [13] have the same memory requirements as the iterative algorithm [11] since the iterative algorithm also needs an EOTF LUT, whereas the explicit approaches do not.

To compare complexity of the proposed algorithm with the worst-case complexity of the iterative approach, an approximate number of operations after obtaining the upsampled chroma has been estimated in Table VI. The number of operations required for color space conversion, down- and up-sampling of chroma is not included because it depends on the choice of up- and downsampling filters. Average number of operations per sample in iterative algorithm [11] can be decreased compared to the worst-case complexity in Table VI by computing tighter initial bounds to get the result in fewer iterations, which, however, adds additional operations to the worst-case complexity.

The complexity of the algorithms has also been compared based on the running time of HDRTools using LUT for TF and TF derivatives computation to convert linear light RGB input to Y'CbCr 4:2:0 (the runtimes include read/write operations). The results in Table VII indicate that the proposed algorithm takes about twice longer than the direct chroma subsampling, while the micro-grading algorithm takes five times longer. It is likely that the difference between the runtimes of the proposed approach and the iterative algorithm [11] is even more significant when other pre-processing steps are excluded from the time measurement. The runtime of the closed-form solution from [13] is similar to that of the proposed algorithm.

VI. CONCLUSIONS

The proposed algorithm removes artifacts in saturated colors of HDR video that may appear in non-constant luminance Y'CbCr 4:2:0 color subsampling. The results of the algorithm look visually identical to the iterative approach [11] on the studied content. Objectively, the proposed algorithm performs similar to the iterative approach [11] and outperforms a previously described explicit solution [13]. The proposed algorithm spends fixed number of operations per pixel and has lower complexity than the average and significantly lower than the worst-case complexities of the iterative approach [11]. This makes the proposed algorithm well-suited for format conversion in non-constant luminance 4:2:0 HDR systems, such as HDR10, especially in the real-time and hardware implementations.

ACKNOWLEDGEMENT

The author would like to thank A. Tourapis for providing HDRTools software package [17], authors of [11] for implementing their algorithm in the package, and Dolby, Technicolor, EBU, DCI, Hochschule der Medien, and Cable Labs for making their test sequences available.

REFERENCES

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

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

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

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

[5] SMPTE ST 2084: "High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays"

[6] E. François, P. Lopez, Y. Olivier, (Technicolor) "About using a BT.2020 container for BT.709 content", MPEG2013/m35255, Sapporo, July 2014

[7] 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.

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

[9] Call for Evidence (CfE) for HDR and WCG Video Coding, ISO/IEC (MPEG).

[10] 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.

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

[12] A. Norkin, "Closed form HDR 4:2:0 chroma subsampling (HDR CE1 and AHG5 related)", ITU-T and ISO/IEC document JCTVC-W0107, San Diego, CA, USA, Feb. 2016.

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

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

[15] J. Samuelsson, C. Fogg, A. Norkin, A. Segall, J. Ström, G. Sullivan, A. Tourapis, "Suggested new draft text of Conversion and Coding Practices for HDR/WCG Y'CbCr 4:2:0 Video with PQ Transfer Characteristics", ITU-T and ISO/IEC document JCTVC-Y0046, Chengdu, CN, Oct. 2016.

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

[17] HDRTools, software package (Apple). Alexis M. Tourapis. https://gitlab.com/standards/HDRTools