CHROMA ADJUSTMENT FOR HDR VIDEO

Jacob Ström and Per Wennersten

Ericsson Research

ABSTRACT

This paper describes a pre-processing method for HDR video on linear RGB data before converting to a $Y'CbCr$ 4:2:0 representation. The method targets the luminance artifacts that can arise in saturated colors after compression when PQ $Y'CbCr$ 4:2:0 NCL is used. After processing, the resulting $Y'CbCr$ 4:2:0 representation is also more compressible, leading to objective BD rate results of -2.4%. Subjective improvements in compressed material are also clearly visible.

Index Terms— HDR video, YCbCr

1. INTRODUCTION AND BACKGROUND

Over the last decades, video quality has increased tremendously in terms of resolution. Another approach to improve quality is high-dynamic range (HDR) video, where the content has darker blacks (below 0.005 candela per square meter ($\mathrm{cd/m^2}$)) and brighter brights (up to 10000 $\mathrm{cd/m^2}$) than standard dynamic range (SDR) content, which was originally intended for the $[0.1, 100]$ $\mathrm{cd/m^2}$ range. This larger dynamic range has made it necessary to use a different electro-optical transfer function (EOTF) than for SDR data, which typically uses a power-law gamma function [1]. As an example, Larson uses a logarithmic transfer function operating in the $Yu'v'$ (CIE 1976) domain [2]. This paper will concentrate on the perceptual quantizer (PQ) EOTF proposed by Miller et al. [3], also known as SMPTE ST 2084 [4]. It converts a red signal component R'_{01} between zero and one to linear light R_{01} between zero and one,

$$R_{01} = tf(R'_{01}) = \left(\frac{(R'_{01})^{\frac{1}{m}} - c_1}{c_2 - c_3 (R'_{01})^{\frac{1}{m}}} \right)^{\frac{1}{n}}, \qquad (1)$$

where $tf(\cdot)$ denotes the EOTF function. The linear value R between 0 and 10000 can then be recovered by multiplying by the peak luminance $L_p = 10000$, $R = L_p R_{01}$. The green and blue channels are treated the same way.

As noted by François [5], using a highly nonlinear EOTF such as PQ can result in clearly visible luminance artifacts for some colors when subsamping to 4:2:0, even without compression. The reason is that the stored luma component, defined as $Y' = w_R tf^{-1}(R/L_p) + w_G tf^{-1}(G/L_p) + w_B tf^{-1}(B/L_p)$ is different from the luminance, which is defined as $Y = w_R R + w_G G + w_B B$ and which is what the human visual system is sensitive to. One way to avoid these artifacts is to use another color space, e.g., "constant luminance" $Y'_C CbCr$ as defined in BT.2020 [6] , or $IC_T C_P$ as defined in BT.2100 [7]. However, a problem with both of these approaches is that a lot of equipment in network infrastructure and consumer devices can only handle the traditional $Y'CbCr$ representation, referred to as "non-constant luminance" (NCL) in BT.2100 [7]. As an example, Blu-ray Disc Association has mandated PQ $Y'CbCr$ 4:2:0 NCL for playback of HDR data [8]. Therefore, a solution that can avoid luminance artifacts for such a decoder is of great value.

One solution called luma adjustment is presented by Ström et al. [9]: Since the luma Y' can be changed in every pixel, it can be set to a value that reproduces the correct luminance Y in the decoder, avoiding the artifacts. Norkin presents a fast version based on a linearization approximation [10]. While luma adjustment certainly helps, it may not be able to to fully remove the luminance artifacts for compression at low bit rates: The luma, which is calculated by the method with the help of the uncompressed Cb and Cr components, may not produce the correct luminance when used with the compressed Cb and Cr components. The proposed method, which we call chroma adjustment, targets this problem. It has the effect of producing smoother Cb and Cr channels, which means that the luminance artifacts due to subsampling will be greatly reduced even before luma adjustment takes place. Furthermore, the smoother chroma channels will survive compression better, greatly reducing luminance artifacts due to compression.

2. DETAILED DESCRIPTION

The reason why luminance artifacts occur is that some colors that are similar in RGB, such as $(4000, 0, 100)$ and $(4000, 4, 100)$, can get very different values once converted to 10 bit $Y'CbCr$ 4:4:4, in this example $(298, 627, 898)$ and $(436, 552, 802)$. Note that the luma component changes 138 code levels, i.e., 16% of the allowed code range $[64, 940]$ for a difference which is likely not visible. If we average the Cb and Cr components but keep the Y' value, we can approximate what happens in an encoder that subsamples without using luma adjustment. The result is $(298, 585.5, 850)$ and $(436, 585.5, 850)$, which after conversion back to RGB be-

Fig. 1. Top: Original, where each half is a single color. Middle: Converted to $Y'CbCr$ 4:2:0 and back without luma adjustment. Bottom: Same conversion, but of an image where both halves are the same color.

comes $(1927, 0.03, 45)$, which is far too dark, and $(8339, 2.2, 216)$, which is far too bright. This results in the dark and bright bands in the middle diagram of Figure 1. Note that if we would just have changed both of the original RGB colors to their average, $(4000, 2, 100)$, which would likely be an imperceptible difference, we would have been better off: Both pixels would then have the same $Y'CbCr$ representation, and would not have been influenced by the downsampling, as shown in the bottom diagram of Figure 1. The core idea of this paper is therefore to make imperceptible changes to the RGB input so that the resulting $Y'CbCr$ representation varies less. To do so we need to define what we believe is an imperceptible change, both in luminance and in chromaticity. In this paper we will say that two colors RGB_1 and RGB_2 are perceptually equivalent in terms of chromaticity if

$$sm_{u'}(RGB_1, RGB_2) = |u'_1 - u'_2| \leq \phi \text{ and} \quad (2)$$

$$sm_{v'}(RGB_1, RGB_2) = |v'_1 - v'_2| \leq \phi. \quad (3)$$

Here u' and v' are the CIE 1976 chromaticity coordinates

$$u' = \frac{4X}{X + 15Y + 3Z} \quad (4)$$

$$v' = \frac{9X}{X + 15Y + 3Z} \quad (5)$$

where XYZ are the CIE 1936 XYZ color space coordinates. If RGB is in BT.2020 color coordinates, these can be calculated using

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.636958 & 0.144617 & 0.168881 \\ 0.262700 & 0.677998 & 0.059302 \\ 0.000000 & 0.028073 & 1.060985 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = T \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (6)$$

According to Larson [2], it is not possible to see a difference if the change in chromaticity as measured in the $u'v'$ representation is smaller than $0.5/410$, which is therefore a reasonable value for ϕ. As for luminance, we say that RGB_1 and RGB_2 are perecptually equivalent if

$$sm_Y(RGB_1, RGB_2) = |(tf^{-1}(Y_1) - tf^{-1}(Y_2))| \leq \theta, \quad (7)$$

where Y_1 is the luminance (Y coordinate) from the XYZ representation of RGB_1. The function $tf^{-1}(\cdot)$ is the inverse of the PQ EOTF of Equation 1. A reasonable value for θ is $0.5/(940 - 64) = 0.5/876$, since this is equivalent to the quantization error (half a code level) for luminance for a gray color $RGB = (a, a, a)$ when quantized to 10-bit Y'CbCr.

2.1. Limits derivation

Assume we want to change the green component of a pixel (R, G, B) so that it becomes more like its neighbor. Assume furhter that we have already changed the red and blue components slightly, but that the new color (R_2, G, B_2) still satisfies the similarity measures for both luminance and chromaticity; $sm_Y((R, G, B), (R_2, G, B_2)) \leq \theta$ and $sm_{u'}, sm_{v'} \leq \phi$. We will now derive the allowed range $[Gmin, Gmax]$ for G_2 so that the new color (R_2, G_2, B_2) still satisfies the measures. Looking first at $sm_{u'}$, according to Equation 4 the u' component for the altered color u'_2 equals

$$u'_2 = \frac{4X_2}{X_2 + 15Y_2 + 3Z_2}. \quad (8)$$

Setting $(\delta R, \delta G, \delta B) = (R_2, G_2, B_2) - (R, G, B)$, we can calculate (X_2, Y_2, Z_2) as a function of the XYZ coordinate of the original color and of $(\delta R, \delta G, \delta B)$,

$$\begin{aligned} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} &= T \begin{bmatrix} R_2 \\ G_2 \\ B_2 \end{bmatrix} = T \left(\begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} \delta R \\ \delta G \\ \delta B \end{bmatrix} \right) \\ &= \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_{11}\delta R + t_{12}\delta G + t_{13}\delta B \\ t_{21}\delta R + t_{22}\delta G + t_{23}\delta B \\ t_{31}\delta R + t_{32}\delta G + t_{33}\delta B \end{bmatrix}. \end{aligned} \quad (9)$$

The denominator in Equation 8 is thus equal to

$$\begin{aligned} &X + t_{11}\delta R + t_{12}\delta G + t_{13}\delta B + \\ &15(Y + t_{21}\delta R + t_{22}\delta G + t_{23}\delta B) + \\ &3(Z + t_{31}\delta R + t_{32}\delta G + t_{33}\delta B), \end{aligned} \quad (10)$$

which can be rewritten as

$$\begin{aligned} &(X + 15Y + 3Z) + (t_{11} + 15t_{21} + 3t_{31})\delta R + \\ &(t_{12} + 15t_{22} + 3t_{32})\delta G + (t_{13} + 15t_{23} + 3t_{33})\delta B \quad (11) \\ &= D + K_1\delta R + K_2\delta G + K_3\delta B, \quad (12) \end{aligned}$$

where D depends on the original color and K_1, K_2 and K_3 are constants. Rewriting also the nominator gives

$$u'_2 = \frac{4(X + t_{11}\delta R + t_{12}\delta G + t_{13}\delta B)}{D + K_1\delta R + K_2\delta G + K_3\delta B}, \quad (13)$$

and we can now solve for δG

$$\delta G = \frac{4(X + t_{11}\delta R + t_{13}\delta B) - u'_2(D + K_1\delta R + K_3\delta B)}{u'_2 K_2 - 4t_{12}}, \quad (14)$$

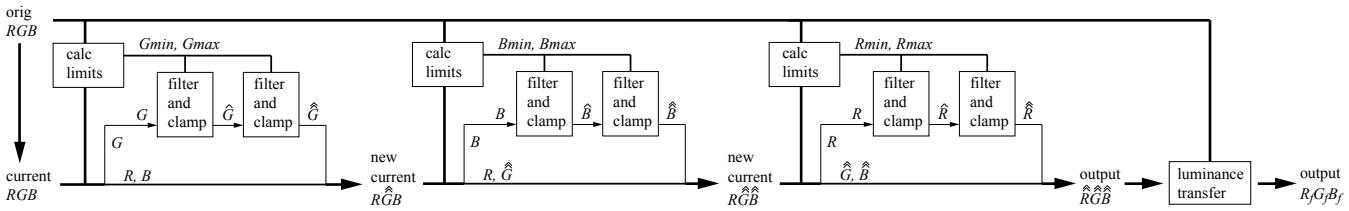

Fig. 2. Overview of the method.

and we call this function $\delta G = f_{u'}(u'_2)$. We can now find the δG on the border where $sm_{u'} = |u' - u'_2| = \phi$,

$$\delta G_a = f_{u'}(u' - \phi) \text{ and} \quad (15)$$

$$\delta G_b = f_{u'}(u' + \phi), \quad (16)$$

The smallest value G_2 that does not violate $sm_{u'} \leq \phi$ is thus $Gmin_{u'} = G + min(\delta G_a, \delta G_b)$, and the largest such value is $Gmax_{u'} = G + max(\delta G_a, \delta G_b)$. Analogously, we get

$$f_v(v'_2) = \frac{9(X + t_{11}\delta R + t_{13}\delta B) - v'_2(D + K_1\delta R + K_3\delta B)}{v'_2 K_2 - 9t_{12}}, \quad (17)$$

which in a similar manner can be used to calculate the allowed range $[Gmin_{v'}, Gmax_{v'}]$, that does not violate $sm_{v'} \leq \phi$.

We next calculate in what range G_2 can vary while still statisfying the luminance constraint $|tf^{-1}(Y) - tf^{-1}(Y_2)| \leq \theta$ where Y is the luminance of the original pixel $Y = w_R R + w_G G + w_B B$ and $Y_2 = w_R R_2 + w_G G_2 + w_B B_2$ is the luminance of the color we vary. Here $w_R = t_{21}$, $w_G = t_{22}$ and $w_B = t_{23}$. The largest allowed luminance for Y_2 will happen when

$$tf^{-1}(Y_2) = tf^{-1}(Y) + \theta. \quad (18)$$

Taking $tf(\cdot)$ of both sides gives

$$Y_2 = tf(tf^{-1}(Y) + \theta) \quad (19)$$

which equals

$$w_R R_2 + w_G G_2 + w_B B_2 = tf(tf^{-1}(Y) + \theta). \quad (20)$$

Solving for G_2 gives the maximum G,

$$Gmax_Y = \frac{tf(tf^{-1}(Y) + \theta) - w_R R_2 - w_B B_2}{w_G}. \quad (21)$$

In a similar manner we get

$$Gmin_Y = \frac{tf(tf^{-1}(Y) - \theta) - w_R R_2 - w_B B_2}{w_G}. \quad (22)$$

The allowed range for G_2 that satisfies all three constraints is therefore $[Gmin, Gmax]$, where

$$Gmin = max(Gmin_{u'}, Gmin_{v'}, Gmin_Y) \quad (23)$$

$$Gmax = min(Gmax_{u'}, Gmax_{v'}, Gmax_Y). \quad (24)$$

Similar derivations can be used to find analytical expression for the allowed ranges $[Rmin, Rmax]$ and $[Bmin, Bmax]$.

2.2. Filtering, clamping and luminance transfer

An overview of the method is shown in Figure 2. Starting with the green component, for every pixel (R, G, B) in the linear input RGB image, the allowed range $[Gmin, Gmax]$ is found using the method above with $R_2 = R$ and $B_2 = B$. The green component is next FIR-filtered in both the horizontal and vertical direction. We use a 5-tap box filter $\{1, 1, 1, 1, 1\}/5$. The filtered green component is then clamped to $[Gmin, Gmax]$, producing $\hat{G}$. This is now filtered and clamped again to produce $\hat{\hat{G}}$. The current pixel is thus $(R, \hat{\hat{G}}, B)$ and the method continues processing the blue component. First the allowed range $[Bmin, Bmax]$ is calculated using the original pixel as the reference and $R_2 = R$ and $G_2 = \hat{\hat{G}}$. After two steps of filtering and clamping we get $(R, \hat{\hat{G}}, \hat{\hat{B}})$. The red component is finally treated in a similar way, giving the output $(\hat{\hat{R}}, \hat{\hat{G}}, \hat{\hat{B}})$. As a last step, the luminance of the original pixel is transferred to the processed pixel. The pixel $(\hat{\hat{R}}, \hat{\hat{G}}, \hat{\hat{B}})$ is converted to CIE1931 xy coordinates $x_p y_p$, and these are then combined with the original luminance Y to get the XYZ representation of the final pixel; $X_f = x_p Y / y_p$, $Y_f = Y$, $Z_f = Y(1 - x_p - y_p)/y_p$. The final pixel $R_f G_f B_f$is then converted from $X_f Y_f Z_f$ using T^{-1}. After this, conversion to $Y'CbCr$ 4:2:0 using luma adjustment according to [9] is performed.

3. RESULTS

The top row of Figure 3 shows Y' and Cb for two consecutive frames before compression. Since the luminance artifacts are strongest near the gamut edges, BT.709 content in a BT.709 container (i.e., using BT.709 color primaries) has been used. This is very similar to BT.2020 content in a BT.2020 container, but such material was not available at the time of writing. Figure 3a shows traditional processing where the linear input is first converted to $R'G'B'$ using $tf^{-1}(\cdot)$, then converted to $Y'CbCr$, quantized to 10-bits followed by subsampling of the Cb and Cr components using a $\{1, 6, 1\}/8$ filter, whereas the Y' component is kept as is. This processing is equivalent to the *simple reference model* processing in [11]. As can be seen in both Y' and Cb, the signal is very un-

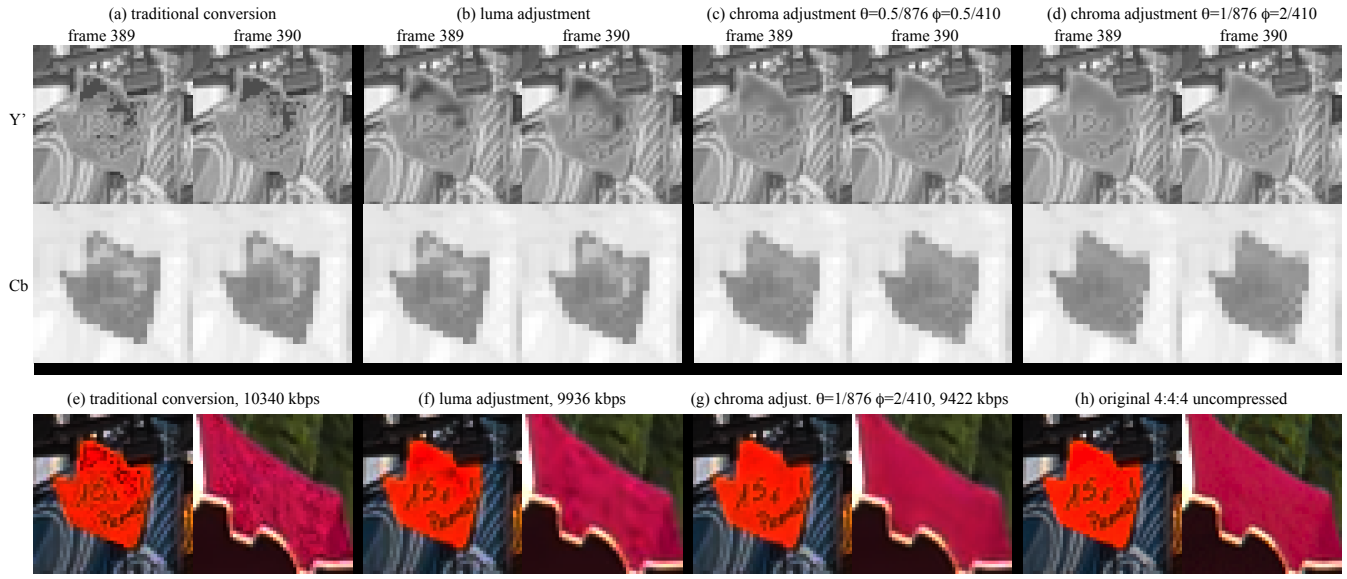

Fig. 3. Top: Y' and Cb cutouts before compression for two consecutive frames. Bottom: Compressed results QP 21, tonemapped to SDR for printed reproduction. Images courtesy of Technicolor and the NevEx project.

smooth and the consecutive frames are quite different from each other, even though the original RGB frames look identical (not shown). This makes it difficult for the encoder to predict across frames. In Figure 3b, the same processing has been carried out, except that luma adjustment has been used to obtain Y'. This is equivalent to the processing in [9] and to the *enhanced reference model* processing in [11] (iterative approach). Cb is the same as for the traditional processing, but the Y' is much smoother. The proposed scheme is shown in Figure 3c and 3d. The processing is the same as for 3b, except that the input RGB has undergone chroma adjustment, and that the downsampling filter of Cb and Cr is $\{1, 2, 1\}/4$. In Figure 3c, $\theta = 0.5/876$ and $\phi = 0.5/410$ have been used. The resulting change in RGB should therefore not be possible to see. Even so, the Y' and Cb components are much smoother, and more similar across frames. In practice it is possible to allow slightly higher values of θ and ϕ. In Figure 3d, $\theta = 1/876$ and $\phi = 2/410$ have been used. The resulting Y' and Cb components are now very smooth, which helps prediction across frames.

For the compressed results, HEVC Main 10 encoding using the HM reference software v16.7 has been used. The bottom row of Figure 3 shows results after compression with QP 21. The traditional conversion (Figure 3e) leads to large luminance artifacts. The results improve significantly with luma adjustment (Figure 3f), but the error is not completely eliminated. With the proposed scheme (Figure 3g) the remaining artifacts are substantially reduced. We have included a supplementary tonemapped SDR video which gives a better idea of the artifact reduction for running video. This will be available at http://ieeexplore.ieee.org. Tonemapping is done using $R_{SDR} = \text{clamp}(255(2^c R)^{\frac{1}{2.22}}, 0, 255)$, $c = -12$ and -8.

Three objective tests have been carried out. For the first test, the same sequences are used as in the Common Test Conditions used for HDR in JCT-VC [12], i.e., eleven sequences. Four are BT.709 and seven are P3DCI, but all are in a BT.2020 container. The sequences have been compressed at four QPs using the enhanced reference model, (i.e. using luma adjustment) both with and without chroma adjustment preprocessing ($\theta = 0.5/876, \phi = 0.5/410$). BD rate figures are calculated for both configurations and compared. In all cases the metrics are from [12] and are evaluated against the (non-chroma adjusted) original. The differences are small: tPSNRY (-0.1%), tPSNRXYZ (-0.0%), DE100(0.7%) and L100 (-0.1%). This is reasonable; since no colors are close to the gamut edge, the chroma adjustment step will not make much of a difference. Therefore, for the second test a BT.709 container has been used with BT.709 content (class A' from the MPEG Call for Evidence [13]). Again, $\theta = 0.5/876, \phi = 0.5/410$ are used for chroma adjustment. This time the differences are bigger: tPSNRY (-0.7%), tPSNRXYZ (-0.8%), DE100(-0.5%) and L100 (-0.5%). The last test is identical to the second, but uses $\theta = 1/876, \phi = 2/410$. This boosts the results even more: tPSNRY (-2.4%), tPSNRXYZ (-2.0%), DE100(-0.3%) and L100 (-1.7%).

In summary, this paper presents a pre-processing method that makes the $Y'CbCr$ 4:2:0 NCL representation more compressible, as well as reducing luminance artifacts during compression. The system parameters θ and ϕ can be set so that the processed image is indistinguishable from the original.

4. REFERENCES

[1] ITU-R, "Reference electro-optical transfer function for flat panel displays used in hdtv studio production," *Recommendation ITU-R BT.1886-3*, April 2011.

[2] G. W. Larson, "The logluv encoding for full gamut, high dynamic range images," *Journal of Graphics Tools*, vol. 3(1), pp. 15–31, March 1998.

[3] S. Miller, M. Nezamabadi, and S. Daly, "Perceptual signal coding for more efficient usage of bit codes," *Motion Imaging Journal*, vol. 122, pp. 52–59, May 2013.

[4] SMPTE ST 2084:2014, "Dynamic range electro-optical transfer function of mastering reference displays," *Society of Motion Picture and Television Engineers*, August 2014.

[5] E. François, "Mpeg hdr ahg: about using a bt.2020 container for bt.709 content (not public)," *110th MPEG meeting in Strasbourg*, October 2014.

[6] ITU-R, "Parameter values for ultra-high definition television systems for production and international programme exchange," *Recommendation ITU-R BT.2020-2*, October 2015.

[7] ITU-R, "Image parameter values for high dynamic range television for use in production and international programme exchange," *Recommendation ITU-R BT.2100-0*, July 2016.

[8] Blu ray Disc Association, "Bd-rom: Audio visual application format specifications version 3," *http://www.bluraydisc.com/assets/Downloadablefile/BD-ROM_Part3_V3.0_WhitePaper_150724.pdf*, July 2015.

[9] J. Ström, J. Samuelsson, and K. Dovstam, "Luma adjustment for high dynamic range video," in *Proceedings of the IEEE Data Compression Conference (DCC), Snowbird*. IEEE, 2016.

[10] A. Norkin, "Fast algorithm for hdr color conversion," in *Proceedings of the IEEE Data Compression Conference (DCC), Snowbird*. IEEE, 2016.

[11] ISO/IEC, "Conversion and coding practices for hdr/wcg y'cbcr 4:2:0 video with pq transfer charactersitics," *ISO/IEC TR 23008-14*, February 2017.

[12] E. François, J. Sole, J. Ström, and P. Yin, "Common test conditions for hdr/wcg video coding experiments," *JCTVC-X1020, 24th JCT-VC meeting, Geneva*, July 2016.

[13] A. Luthra, E. François, and W. Husak, "Call for evidence (cfe) for hdr and wcg video coding," *N15083, 111th MPEG Meeting, Geneva, Switzerland*, February 2015.