|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  23rd Meeting: San Diego, USA, 19–26 February 2016 | Document: JCTVC-W0046 / m37680 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Recommended conversion processes of YCbCr 4:2:0 and RGB SDR content in and between BT.709 and BT.2020 colour gamut** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Edouard Francois, Koohyar Minoo, René van der Vleuten, Alexis Tourapis | Tel: Email: | [Edouard.Francois@technicolor.com](mailto:Edouard.Francois@technicolor.com) [Koohyar.Minoo@arris.com](mailto:Koohyar.Minoo@arris.com) [rene.van.der.vleuten@philips.com](mailto:rene.van.der.vleuten@philips.com) [atourapis@apple.com](mailto:atourapis@apple.com) |
| *Source:* | Apple, Arris, Philips, Technicolor | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document describes the recommended processes for the following conversions, in the context of the JCTVC experiments on HDR with SDR backward compatibility:

1. Conversion of SDR BT.709 YCbCr 4:2:0 to SDR BT.709 R’G’B’
2. Conversion of SDR BT.709 R’G’B’ to SDR BT.709  YCbCr 4:2:0
3. Conversion of SDR BT.709 R’G’B’ to SDR BT.2020 R’G’B’
4. Conversion of SDR BT.2020 YCbCr 4:2:0 to SDR BT.2020 R’G’B’
5. Conversion of SDR BT.2020 R’G’B’ to SDR BT.2020  YCbCr 4:2:0
6. Conversion of SDR BT.2020 R’G’B’ to SDR BT.709 R’G’B’

This is an initial version to be completed based on the discussions and decisions made by the JCTVC.

# Introduction

In this document we will use the acronyms BT.709, BT.2020, and BT.2087 to refer to the following:

BT.709: ITU-R Recommendation BT.709-6: Parameter values for the HDTV standards for production and international programme exchange (06/2015). <https://www.itu.int/rec/R-REC-BT.709/en>

BT.1886: ITU-R Recommendation BT.1886-0: Reference electro-optical transfer function for flat panel displays used in HDTV studio production (03/2011). <https://www.itu.int/rec/R-REC-BT.1886/en>

BT.2020: ITU-R Recommendation BT.2020-2: Parameter values for ultra-high definition television systems for production and international programme exchange (10/2015). <https://www.itu.int/rec/R-REC-BT.2020/en>

BT.2087: ITU-R Recommendation BT.2087-0: Colour conversion from Recommendation ITU-R BT.709 to Recommendation ITU-R BT.2020 (10/2015). <https://www.itu.int/rec/R-REC-BT.2087/en>

The exploration work for an HDR and WCG extension of HEVC considers the support of SDR backward compatibility. With this feature, the content resulting from the HDR content reshaping (as described in document JCTVC-W0031) is SDR backward compatible and can be rendered on legacy SDR displays. According to the Common Test Conditions (MPEG document w15793), the HDR test sequences are processed in a BT.2020 container. The reshaped content is also supposed to be in a BT.2020 container. In case of SDR backward compatibility, this means that the SDR content resulting from the reshaping is in a BT.2020 container.

In order to properly render such content on a legacy BT.709 SDR display, the conversion process from SDR in BT.2020 container to SDR in BT.709 container needs to be described. Similarly, if reference SDR content is provided in a BT.709 container, for sake of comparisons, it is useful to provide a typical conversion process from BT.709 to BT.2020 container. Note that SDR displays, for both BT.709 and BT.2020, make use of the BT.1886 EOTF.

The conversion processes are described in the next sections. The up- and down sampling processes for BT.2020 use the co-sited chroma sample location (chroma location 2) that has been specified in BT.2020. This chroma sample location is different from the MPEG default chroma location (chroma location 0) that was used with BT.709. The conversion between BT.709 and BT.2020 R’G’B’ signals is performed according to Recommendation ITU-R BT.2087.

# Conversion process from SDR BT.709 YCbCr 4:2:0 to SDR BT.709 R’G’B’

The process consists of the following steps:

1. Upsampling from BT.709 4:2:0 DY’DCbDCr (10bit) to BT.709 4:4:4 DY’DCbDCr (10bit) by invoking section 7.1.2.
2. Inverse quantization from BT.709 4:4:4 DY’DCbDCr (10bit) into normalized BT.709 4:4:4 Y’CbCr (float) by invoking section 7.2.2 with BitDepthY and BitDepthC set to 10.
3. Conversion from BT.709 4:4:4 Y’CbCr (float) to BT.709 R’G’B’ (float) by invoking section 7.3.1.

# Conversion process from SDR BT.709 R’G’B’ to SDR BT.709 YCbCr 4:2:0

The process consists of the following steps:

1. Conversion from BT.709 R’G’B’ (float) to BT.709 4:4:4 Y’CbCr (float) by invoking section 7.3.3.
2. Quantization from BT.709 4:4:4 Y’CbCr (float) to BT.709 4:4:4 DY’DCbDCr (10bit) by invoking section 7.2.1 with BitDepthY and BitDepthC set to 10.
3. Downsampling from BT.709 4:4:4 DY’DCbDCr to BT.709 4:2:0 DY’DCbDCr by invoking section 7.1.1.

# Conversion process from SDR BT.709 R’G’B’ to SDR BT.2020 R’G’B’

The process consists of the following steps:

1. Conversion using BT.2087 EOTF from BT.709 R’G’B’ (float) to BT.709 RGB (float) by invoking section 7.4.2.
2. Conversion from BT.709 RGB (float) to BT.2020 RGB (float) by invoking section 7.5.2.
3. Conversion with BT.2087 inverse EOTF from BT.2020 RGB (float) to BT.2020 R’G’B’ (float) by invoking section 7.4.1.

# Conversion process from SDR BT.2020 YCbCr 4:2:0 to SDR BT.2020 R’G’B’

The process consists of the following steps:

1. Upsampling from BT.2020 4:2:0 DY’DCbDCr (10bit) to BT.2020 4:4:4 DY’DCbDCr (10bit) by invoking section 7.1.4.
2. Inverse quantization from BT.2020 4:4:4 DY’DCbDCr (10bit) into normalized BT.2020 4:4:4 Y’CbCr (float) by invoking section 7.2.2 with BitDepthY and BitDepthC set to 10.
3. Conversion from BT.2020 4:4:4 Y’CbCr (float) to BT.2020 R’G’B’ (float) by invoking section 7.3.2.

# Conversion process from SDR BT.2020 R’G’B’ to SDR BT.2020 YCbCr 4:2:0

The process consists of the following steps:

1. Conversion from BT.2020 R’G’B’ (float) to BT.2020 4:4:4 Y’CbCr (float) by invoking section 7.3.4.
2. Quantization from BT.2020 4:4:4 Y’CbCr (float) to BT.2020 4:4:4 DY’DCbDCr (10bit) by invoking section 7.2.1 with BitDepthY and BitDepthC set to 10.
3. Downsampling from BT.2020 4:4:4 DY’DCbDCr to BT.2020 4:2:0 DY’DCbDCr by invoking section 7.1.3.

# Conversion process from SDR BT.2020 R’G’B’ to SDR BT.709 R’G’B’

The process consists of the following steps:

1. Conversion using BT.2087 EOTF from BT.2020 R’G’B’ (float) to BT.2020 RGB (float) by invoking section 7.4.2.
2. Conversion from BT.2020 RGB (float) to BT.709 RGB (float) by invoking section 7.5.1.
3. Conversion with BT.2087 inverse EOTF from BT.709 RGB (float) to BT.709 R’G’B’ (float) by invoking section 7.4.1.

## Chroma resampling

### Chroma sample location 0 downsampling from 4:4:4 to 4:2:0

The chroma samples alignment is as follows:



[to be filled in by Alexis]

### Chroma sample location 0 upsampling from 4:2:0 to 4:4:4 (Y’CbCr domain)

The upsampling filter used is the same for both horizontal and vertical processes. First, vertical filtering is applied on the 4:2:0 picture, then horizontal filtering.

[to be filled in by Alexis]

### Chroma sample location 2 downsampling from 4:4:4 to 4:2:0

[to be filled in by Alexis]

### Chroma sample location 2 upsampling from 4:2:0 to 4:4:4 (Y’CbCr domain)

[to be filled in by Alexis]

## Quantization and inverse quantization

### Quantization from Y’CbCr into DY’DCbDCr

This process quantizes the input Y’CbCr signal into a signal of bit-depth BitDepthY for the Y component and BitDepthC for the chroma components (Cb, Cr).

with

Round( x ) = Sign( x ) \* Floor( Abs( x ) + 0.5 )

Sign ( x ) = -1 if x < 0, 0 if x=0, 1 if x > 0

Floor( x ) the largest integer less than or equal to x

Abs( x ) = x if x>=0, -x if x<0

Clip1Y( x ) = Clip3( 0, ( 1 << BitDepthY ) − 1, x )

Clip1C( x ) = Clip3( 0, ( 1 << BitDepthC ) − 1, x )

Clip3( x,y,z ) = x if z<x, y if z>y, z otherwise

### Inverse Quantization from DY’DCbDCr into Y’CbCr

This process dequantizes the input signal represented on BitDepthY bits for the Y component and BitDepthC bits for the chroma components (Cb, Cr) into a (float) signal Y’CbCr.

with

ClipY’ (x) = Clip3 ( 0, 1.0, x)

ClipC (x) = Clip3 ( -0.5, 0.5, x)

Clip3( x,y,z ) = x if z<x, y if z>y, z otherwise

## Colour transformation

### Y’CbCr to R’G’B’ with BT.709 primaries

* + R’ = clipRGB(Y’ + 1.5748 \* Cr)
  + G’ = clipRGB(Y’ – 0.18733 \* Cb – 0.46813 \* Cr)
  + B’ = clipRGB(Y’ + 1.8556 \* Cb)

### Y'CbCr to R’G’B' with BT.2020 primaries

* + R’ = clipRGB(Y’ + 1.47460 \* Cr)
  + G’ = clipRGB(Y’ – 0.16455 \* Cb – 0.57135 \* Cr)
  + B’ = clipRGB(Y’ + 1.88140 \* Cb)

with clipRGB( x ) = Clip3( 0, 1, x )

Clip3( x,y,z ) = x if z<x, y if z>y, z otherwise

### R’G’B’ with BT.709 primaries to Y’CbCr

The ITU-R BT.709 standard specifies the following conversion process from R’G’B’ to Y’CbCr (non-constant luminance representation):

* + Y’ = 0.2126 \* R’ + 0.7152 \* G’ + 0.0722 \* B’

### R’G’B' with BT.2020 primaries to Y’CbCr

The ITU-R BT.2020 standard specifies the following conversion process from R’G’B’ to Y’CbCr (non-constant luminance representation):

* + Y’ = 0.2627 \* R’ + 0.6780 \* G’ + 0.0593 \* B’

## BT.2087 EOTF and inverse EOTF

### Conversion from RGB to R’G’B’

* + R’ = inverseEOTF( max(0, min(R, 1)) )
  + G’ = inverseEOTF( max(0, min(G, 1)) )
  + B’ = inverseEOTF( max(0, min(B, 1)) )

With

inverseEOTF(L) = L(1/2.4)

### Conversion from R’G’B’ to RGB

* + R = EOTF( R’ )
  + G = EOTF( G’ )
  + B = EOTF( B’ )

With

EOTF(V) = V2.4

## Colour space container conversion

This section describes the process of converting RGB samples encoded in and limited by one particular colour space, e.g. ITU-R Recommendation BT.709, BT.2020, into another RGB colour space specified with different primaries. We are particularly interested in the conversion of RGB BT.709 samples into RGB BT.2020 samples and vice versa.

### RGB conversion: from BT.2020 to BT.709

It is essential when displaying data on a display that can only operate with BT.709 primaries to appropriately convert them in that space before display. To do so, this can be done using either a two-step conversion process, from the current colour space of the data (i.e. ITU-R BT.2020) to XYZ, followed by a subsequent conversion to BT.709, or using a single step process. In particular, the two-step conversion process can be applied as follows:

* Conversion from R2020G2020B2020 to XYZ:
  + X = 0.636958 \* R2020 + 0.144617 \* G2020 + 0.168881 \* B2020
  + Y = 0.262700 \* R2020 + 0.677998 \* G2020 + 0.059302 \* B2020
  + Z = 0.000000 \* R2020 + 0.028073 \* G2020 + 1.060985 \* B2020
* Conversion from XYZ to R709G709B709 (BT.709):
  + R709 = clipRGB( 3.240970 \* X - 1.537383 \* Y - 0.498611 \* Z )
  + G709 = clipRGB( -0.969244 \* X + 1.875968 \* Y + 0.041555 \* Z )
  + B709 = clipRGB( 0.055630 \* X - 0.203977 \* Y + 1.056972 \* Z )

The above could be converted into a single step by combining the two matrix conversions above into a single matrix. This would result in the following, high precision matrix conversion:

* + R709 = clipRGB( 1.660490254890140 \* R2020

- 0.587638564717282 \* G2020 - 0.072851975229213 \* B2020 )

* + G709 = clipRGB( -0.124550248621850 \* R2020

+ 1.132898753013895 \* G2020 - 0.008347895599309 \* B2020 )

* + B709 = clipRGB( -0.018151059958635 \* R2020

- 0.100578696221493 \* G2020 + 1.118729865913540 \* B2020 )

We currently would recommend the single step approach, as above, for the conversion of RGB BT.2020 material to RGB BT.709.

### RGB conversion: from BT.709 to BT.2020

The process to convert RGB BT.709 samples to RGB BT.2020 samples is very similar to the process performed earlier for the inverse conversion. A two-step conversion involves first applying the RGB BT.709 to XYZ conversion process, followed by a conversion from XYZ to RGB BT.2020 using the appropriate conversion matrices. In particular, this is done as follows:

* Conversion from R709G709B709 (BT.709) to XYZ
  + X = 0.412391 \* R709 + 0.357584 \* G709 + 0.180481 \* B709
  + Y = 0.212639 \* R709 + 0.715169 \* G709 + 0.072192 \* B709
  + Z = 0.019331 \* R709 + 0.119195 \* G709 + 0.950532 \* B709
* Conversion from XYZ to R2020G2020B2020 (BT.2020)
  + R2020 = clipRGB( 1.716651 \* X – 0.355671 \* Y - 0.253366 \* Z )
  + G2020 = clipRGB( -0.666684 \* X + 1.616481 \* Y + 0.015768 \* Z )
  + B2020 = clipRGB( 0.017640 \* X - 0.042771 \* Y + 0.942103 \* Z )

Similarly, the single step and recommended method is as follows:

* + R2020 = clipRGB( 0.627404078626 \* R709 + 0.329282097415 \* G709 + 0.043313797587 \* B709 )
  + G2020 = clipRGB( 0.069097233123 \* R709 + 0.919541035593 \* G709 + 0.011361189924 \* B709 )
  + B2020 = clipRGB( 0.016391587664 \* R709 + 0.088013255546 \* G709 + 0.895595009604 \* B709 )