SMPTE ST 2094-40:2020

Revision of ST 2094-40:2016

SMPTE STANDARD

Dynamic Metadata for Color Volume Transform — Application #4

Table of Contents Page

Approved April 9, 2020

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. SMPTE ST 2094-40 was prepared by Technology Committee 10E.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

Dynamic Metadata for Color Volume Transform Application #4 defines scene-based metadata designed to help reproduce the original intent of the creator of High Dynamic Range (HDR) and Wider Color Gamut (WCG) image essence on a display having a smaller color volume.

Scene-based Color Volume Mapping consists of scene adaptive tone mapping and optional scene-based color saturation mapping processing blocks. Descriptive scene characteristic metadata is provided for scene adaptive tone mapping.

These processing blocks can optionally make use of the upper limits of the peak luminance behaviors of the display used for mastering the input image essence (i.e. the “mastering display”) and of the targeted system display as display's ability to achieve peak luminance varies and large areas of bright pixels can change appearance with some displays due to power limitations and other constraints.

In addition to these adjustments that can be algorithmically determined from known display and content characteristics, Scene-based Color Volume Mapping also includes creatively approved adjustments with a creative's defined tone curve.

Application version 1 is introduced in the 2020 revision. Certain metadata elements have version dependencies, and these are identified for **ApplicationVersion** = 0 and **ApplicationVersion** = 1 in this document. The metadata items are **DistributionMaxRGB**, **BezierCurveAnchors** and **FractionBrightPixels**.

1 Scope

This standard specifies the metadata for Color Volume Transform Application #4, Scene-based Color Volume Mapping for ApplicationVersion = 0 and ApplicationVersion = 1. It is a specialization of the content-dependent transform metadata entries and processing blocks of the generalized color volume transform model defined in the SMPTE ST 2094-1 Core Components standard. It includes scene-adaptive tone mapping and scene-based color saturation mapping. Scene-based Color Volume Mapping also includes creatively approved adjustments with a creative's designed curve metadata.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; then formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this engineering document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this engineering document are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 2086:2018, Mastering Display Color Volume Metadata Supporting High Luminance and Wide Color Gamut Images

SMPTE ST 2094-1:2016, Dynamic Metadata for Color Volume Transform — Core Components

4 Terms and Definitions

SMPTE ST 2094-1, Section 4, Terms and Definitions applies to this document. This section defines additional terms and definitions used in this document.

4.1 actual display peak luminance

peak luminance (in units of cd/m^2) that a display is capable of delivering while rendering the scene

Note 1 to entry: dependent on the spatial distribution of the luminance levels of the pixels in the scene and the power consumption limits of the display.

4.2 brightest pixel

pixel with the maximum luminance (approximately) among the smoothed, selected pixels in the scene

4.3 input image essence

image essence to which the color volume transform is applied and having three color components labeled R, G and B where the R component corresponds to the Mastering Display Color Volume Display Primary with the largest x chromaticity coordinate, the G component corresponds to the Mastering Display Color Volume Display Primary with the largest y chromaticity coordinate, and the B component corresponds to the remaining Mastering Display Color Volume Display Primary

4.4 linearized

representing output linear light levels in the range [0,1]

Note 1 to entry: Signals encoded using SMPTE ST 2084 can be linearized by converting digital code values to output linear light levels using SMPTE ST 2084:2014 Equation 4.1 and are in the range [0,1]

4.5 linearized maxRGB

maxRGB value as defined in SMPTE ST 2094-1 applied to linearized RGB values

Note 1 to entry: linearized maxRGB is a single value per pixel

4.6 linearized RGB

linearized color component values in the range [0, 1]

Note 1 to entry: linearized RGB has three component values

EXAMPLE An RGB pixel represented by SMPTE ST 2084 10-bit full range digital code values of [0x0, 0x302, 0x3ff] has a linearized RGB representation of [0, 0.1008, 1] (rounded to 4 significant digits.)

4.7 luminance band

luminance of pixels between a minimum and maximum luminance

4.8 scene

selected pixels throughout all images in the time interval as defined in SMPTE ST 2094-1

4.9 selected pixels

pixels within a **ProcessingWindow** group as defined in SMPTE ST 2094-1 or within the intersection of a processing window and the external ellipse, if the processing window is extended using the elliptical pixel selector as defined in Section 6.

4.10 smoothed pixels

spatially-averaged pixels

Note 1 to entry: method defined in Section 10

5 Application Identification

The **ApplicationIdentifier** value shall be 4 and the **ApplicationVersion** value shall be either 0 or 1.

ApplicationIdentifier = 4 and **ApplicationVersion** being either 0 or 1 identifies this document as the defining document for the application-specific metadata specified in Section 9.1 which is identical for these versions.

6 Extension to the Window

6.1 Selection of Pixels for Processing

The **ProcessingWindow** group as defined in SMPTE ST 2094-1 may be extended with an elliptical pixel selector for **WindowNumber** > 0. The elliptical pixel selector shall consist of two aligned, concentric, similar ellipses (internal and external). The ellipse coordinate system shall be the same as the pixel coordinate system. An example of the elliptical pixel selector is shown in Figure 1, where $\theta,\ a_1,\ a_2,\ b_2$ indicate the rotation angle, the semi-major axis of the internal ellipse, the semi-major axis of the external ellipse and the semi-minor axis of the external ellipse respectively. (x_0, y_0) denotes the pixel coordinates of the center of the ellipse. The semi-major axes are along the direction of the rotation angle θ. Rotation angle $\theta = 0$ is parallel to the x-axis and it increases in a clockwise fashion.

Whether a pixel is within an elliptical selector shall be determined from the center position of each pixel in the ellipse coordinate system. If the center point of a pixel is within the ellipse equation, the pixel shall be included.

Pixels outside of the external ellipse shall not be affected by an elliptical pixel selector. Pixels inside the internal ellipse shall have full effect of the processing window.

The pixels between the internal and external ellipses form a transition region to aid in preventing visible boundary artifacts between the local processing region and the background. The transition weighting from the external ellipse to the internal ellipse should apply to the processed color pixel the least (0.0) to most (1.0) effect of the processing window based upon distance between the two ellipses.

The selected pixels shall be the pixels within the intersection of the external ellipse and the **ProcessingWindow** boundaries.

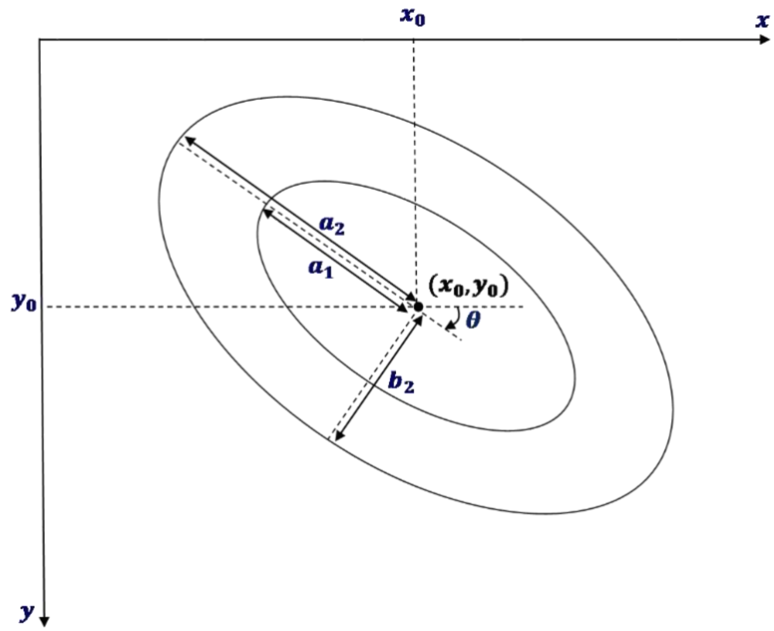

Figure 1 – An elliptical pixel selector consisting of two concentric ellipses: internal and external ellipses

6.2 Center of Ellipse

The **CenterOfEllipse** shall be a vector with two integers. This shall represent the center position (pixel coordinates) of the concentric internal and external ellipses of the elliptical pixel selector. The order of the vector elements shall be {x coordinate, y coordinate}. The x coordinate shall be in the range of [0, image width - 1] and the y coordinate shall be in the range of [0, image height - 1]. The x and y coordinates shall be in multiples of 1 pixel. The **CenterOfEllipse** is denoted by (x_0, y_0) in Figure 1.

6.3 Rotation Angle

The **RotationAngle** shall represent the clockwise rotation angle in degree of arc with respect to the positive direction of the x-axis of the concentric internal and external ellipses of the elliptical pixel selector centered at the **CenterOfEllipse**. **RotationAngle** shall be in the range [0,180] and in multiples of 1. The **RotationAngle** is denoted by θ in Figure 1.

6.4 Semi-Major Axis of the Internal Ellipse

The **SemiMajorAxisInternalEllipse** shall represent the semi-major axis value of the internal ellipse of the elliptical pixel selector in number of pixels. The **SemiMajorAxisInternalEllipse** shall be in the range of [1, 65535] and in multiples of 1 pixel. The **SemiMajorAxisInternalEllipse** is denoted by a_1 in Figure 1.

6.5 Semi-Major Axis of the External Ellipse

The **SemiMajorAxisExternalEllipse** shall represent the semi-major axis value of the external ellipse of the elliptical pixel selector in number of pixels. The **SemiMajorAxisExternalEllipse** shall not be less than the **SemiMajorAxisInternalEllipse**. The **SemiMajorAxisExternalEllipse** shall be in the range of [1, 65535] and in multiples of 1 pixel. The **SemiMajorAxisExternalEllipse** is denoted by a_2 in Figure 1.

6.6 Semi-Minor Axis of the External Ellipse

The **SemiMinorAxisExternalEllipse** shall represent the semi-minor axis value of the external ellipse of the elliptical pixel selector in number of pixels. The **SemiMinorAxisExternalEllipse** shall be in the range of [1, 65535] and in multiples of 1 pixel. The **SemiMinorAxisInternalEllipse** is denoted by b_2 in Figure 1.

6.7 Overlap Process Option

6.7.1 Definition

The **OverlapProcessOption** shall be an enumerator that indicates one of the two methods of combining rendered pixels in an image with processing window elliptical pixel selectors. The value of **OverlapProcessOption** shall be either 0 or 1 and overlapping elliptical pixel selectors shall have the same value.

Note: overlapping elliptical selectors can cause visible color artifacts or visual discontinuity in an image.

An example is provided in B.6.4.

6.7.2 Method 0 – Weighted Average Method

OverlapProcessOption = 0 shall specify a method that averages the rendered color of each processing window's elliptical selector with the rendered color in **WindowNumber** = 0.

6.7.3 Method 1 – Layering Method

OverlapProcessOption = 1 shall specify a method where a rendered color in the lowest **WindowNumber** > 0 with a non-zero weighting is averaged with the rendered color in **WindowNumber** = 0.

7 Targeted System Display

7.1 Introduction

The **TargetedSystemDisplay** metadata group associated with Application #4 Scene-based Color Volume Mapping adds the metadata item defined in Section 7.2.

7.2 Targeted System Display Actual Peak Luminance (2D LUT)

The **TargetedSystemDisplayActualPeakLuminance** shall be a two-input sampled function as defined in SMPTE ST 2094-1. The array of the **TargetedSystemDisplayActualPeakLuminance** two-input sampled function shall be limited to a maximum of 25 rows and a maximum of 25 columns. The array output values shall be in the range of [0,1] in multiples of 1/15, with 0 corresponding to 0 cd/m^2 and 1 corresponding to **TargetedSystemDisplayMaximumLuminance.**

The function takes as x (i.e. first input) **FractionBrightPixels** as defined in Section 8.6 and y (i.e. second input) **AverageMaxRGB** as defined in Section 8.4. It shall output a single value as the interpolated actual display peak luminance of the targeted system display.

8 Color Volume Mapping

8.1 Introduction

The color volume transform for Application #4 is based on Scene-adaptive Tone Mapping and Scene-based Color Saturation Mapping as shown in Annex A and further informatively described in Annex B. The **ColorVolumeTransform** metadata items associated with Application #4 are defined in Sections 8.2 to 8.8.

8.2 Mastering Display Actual Peak Luminance (2D LUT)

The **MasteringDisplayActualPeakLuminance** shall be a two-input sampled function as defined in SMPTE ST 2094-1. The array of the **MasteringDisplayActualPeakLuminance** two-input sampled function shall be limited to a maximum of 25 rows and a maximum of 25 columns. Array output values shall be in the range of [0,1], in multiples of 1/15, with 0 corresponding to 0 cd/m^2 and 1 corresponding to **MaximumDisplayMasteringLuminance**.

The function takes as x (i.e. first input) **FractionBrightPixels** as defined in Section 8.6 and y (i.e. second input) **AverageMaxRGB** as defined in Section 8.4. It shall output a single value of the interpolated actual display peak luminance of the mastering display used for mastering the image essence.

8.3 Maximum Scene Color Component Levels

The **MaxSCL** shall be a vector with three elements. They shall be the maximum of each component of linearized RGB values in the scene. The elements shall each be in the range [0,1] and in multiples of 0.00001. The order of the elements shall be {R,G,B}. A **MaxSCL** value of {0,0,0} should indicate that **MaxSCL** does not represent a calculated value.

8.4 Average MaxRGB

The **AverageMaxRGB** shall be the average of linearized maxRGB values in the scene. The value shall be a number in the range [0,1] and in multiples of 0.00001.

8.5 Distribution MaxRGB

8.5.1 Purpose

This metadata item provides descriptive metadata in compact form of the scene's cumulative frequency distribution (CFD). **DistributionMaxRGB** provides alternative metadata for displays to tone map when metadata associated with Section 8.7 is not present.

A CFD can be derived from a list of linearized maxRGB values of all pixels in the scene, sorted in ascending order by value, and retaining all duplicate values. There shall be n items in the CFD list, where n is the number of pixels in the scene.

8.5.2 Definition

The **DistributionMaxRGB** shall be a set of two equal-length vectors, **DistributionMaxRGBPercentages** and **DistributionMaxRGBPercentiles**, herein called J and V respectively. Subscript *i* denotes the position of an element in the vector. The first element is at position 0. The last element is designated as position Ω and shall be the position at the length of the vector minus 1.

Each element in vector J shall be an integer value in the range [0,100] representing the percentage of the total length of the CFD list. The elements of vector J shall be in ascending order by value.

Each element in the second vector V shall be in the range [0,1] and in multiples of 0.00001. Except as specified in Section 8.5.4, V_i contains a linearized maxRGB value which is the smallest value in the CFD such that at least J_i percent of the CFD fall at or below V_i whereby V_i represents a percentile value of the CFD at J_i percent.

An example is provided in Annex C.

Note: The definitions of V_i can be understood in terms of a list of linearized MaxRGB values from the scene sorted in terms of linearized RGB from least to greatest, where the list includes duplicate values and there are n items in the list. The V_i value is the value of the $\lceil n \times J_i/100 \rceil$th element in the CFD list.

8.5.3 Constraints for ApplicationVersion = 0

When **ApplicationVersion** = 0, the constraints in this Section shall apply.

The maximum length of these vectors shall be 15.

Whenever J_i equal to 99 is present, the percentage value 99.98% should be used in the calculation of V_i.

8.5.4 Constraints for ApplicationVersion = 1

When **ApplicationVersion** = 1, the constraints in this Section shall apply.

The length of the vectors shall equal 9.

Whenever J_8 equal to 99 is present, the percentage value 99.98% shall be used in the calculation of V_8.

The J vector should contain the values in Table 1.

Table 1 Distribution Vectors

J =	**1**	**5**	**10**	**25**	**50**	**75**	**90**	**95**	**99**
V =	V_0	(V_1)	(V_2)	V_3	V_4	V_5	V_6	V_7	V_8

If and only if J_1=5 and J_2=10, the vector elements for V_1 and V_2 are not part of the CFD, V_1 shall be 0.00000, V_2 shall be 0.00255 and other values for V_1 and V_2 are reserved.

8.6 Fraction of Bright Pixels

FractionBrightPixels is a number representing the weighted fraction of pixels in the highest luminance band of the brightest frame in the scene. The brightest frame, the luminance band, and the weighting function are defined per **ApplicationVersion**. **FractionBrightPixels** is calculated on smoothed pixels which represent a spatial averaging method defined per **ApplicationVersion**.

FractionBrightPixels shall be in the range [0,1] and in multiples of 0.001.

FractionBrightPixels = 0 shall indicate that **FractionBrightPixels** does not represent a calculated value.

FractionBrightPixels > 0 shall indicate that **FractionBrightPixels** has been calculated as specified in Section 10, and then should be clamped to a minimum of 0.001 (one multiple).

8.7 Bezier Curve Tone Mapping

8.7.1 No metadata

If metadata items **KneePoint** and **BezierCurveAnchors** are not present, then no Bezier tone mapping curve is defined.

8.7.2 Knee Point

The **KneePoint** shall be a vector with two numbers. It represents the separation point (K_S, K_F) between the linear part and the curved part of the tone mapping function as illustrated in Figure 2. The value of the numbers shall be in the range [0,1] and in multiples of 1/4095.

8.7.3 Bezier Curve Anchors

BezierCurveAnchors shall be a vector of numbers whose elements represent the intermediate anchor parameters $(\boldsymbol{P_1}, \cdots, \boldsymbol{P_{N-1}})$ in Equation (1), with $P_0 = 0 \text{ and } P_N = 1$. The values of the vector elements shall be in the range [0,1] and in multiples of 1/1023, and as constrained in the Section 8.7.4 calculations.

8.7.3.1 Constraints for ApplicationVersion = 0

BezierCurveAnchors length shall be in the range [0,15]

8.7.3.2 Constraints for ApplicationVersion = 1

BezierCurveAnchors length shall be in the range [0,9]

8.7.4 Calculation of Bezier Curve Tone Mapping Function

s shall be a normalized input [0,1] of all linearized input pixel values. *s* = 1 can correspond to either **DistributionMaxRGBPercentiles[Ω]** or to max(**MaxSCL**) if non-zero.

A tone-mapping function, $F_N(s)$, shall be composed of a linear part and a curved part. The two parts shall be joined by the **KneePoint** (K_s, K_F) as depicted in Figure 2. Some image artifacts can be avoided by matching the slope of the linear section with the slope of the curved section at P_0, the first anchor by adjusting P_1.

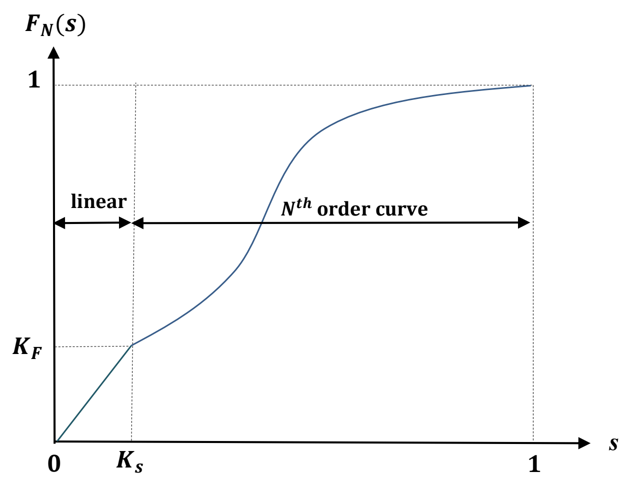

Figure 2 – A tone mapping function with a knee point

Parameters used in this section are:

K_s = horizontal knee point shown in Figure 2,

K_F = vertical knee point shown in Figure 2,

N = the order of the Bezier curve. N shall be equal to the length of **BezierCurveAnchors** + 1 and N shall be 1 if **BezierCurveAnchors** is not present. N is in the range [1,16],

The function $B_N(t)$ shall be defined by Equation (1):

$$B_N(t) = \sum_{\rho=0}^{N} \binom{N}{\rho} \times t^{\rho} \times (1-t)^{N-\rho} \times P_{\rho}, \qquad \text{for} \quad 0 \le t \le 1 \tag{1}$$

where:

t = function argument,

$\binom{N}{\rho} = \frac{N!}{\rho! \times (N-\rho)!}$,

$P_0, \cdots, P_N =$ anchor parameters that characterize the curved part of the tone mapping function with $P_0 = 0,\ P_N = 1,$ and where P_i for $i = 1, \ldots, N-1$ shall satisfy the condition for monotonicity defined in Equation (2) .

The result of Equation (1) satisfies $0 \leq B_N(t) \leq 1$.

Note: Equation (2) is the condition for monotonicity

$$\frac{d}{dt} B_N(t) = \sum_{\rho=0}^{N} \binom{N}{\rho} t^{\rho-1} \times (1-t)^{N-\rho-1} \times (\rho - N \times t) \times P_\rho \geq 0,\ \forall\, t \in [0,1]. \qquad (2)$$

The tone-mapping function, $F_N(s)$, depends upon the value of K_S.

Case 1: if $K_S = 0$, then $F_N(s)$ shall be as defined in Equation (3).

$$F_N(s) = K_F + (1 - K_F) \times B_N(s) \qquad 0 \leq s \leq 1 \qquad (3)$$

Case 2: if $0 < K_S < 1$, then $F_N(s)$ shall be as defined in Equation (4).

$$F_N(s) = \begin{cases} \frac{K_F}{K_S} \times s, & 0 \leq s < K_S \\ K_F + (1 - K_F) \times B_N\left(\frac{s-K_S}{1-K_S}\right), & K_S \leq s \leq 1 \end{cases} \qquad (4)$$

Case 3: if $K_S = 1$, then $F_N(s)$ shall be as defined in Equation (5).

$$F_N(s) = K_F \times s \qquad 0 \leq s \leq 1 \qquad (5)$$

The range of $F_N(s)$ is [0,1] with 1 represented by a) if metadata item in Section 7.2 is present, otherwise by b):

a) the **TargetedSystemDisplayMaximumLuminance** multiplied by the output of the metadata item defined in Section 7.2, **TargetedSystemDisplayActualPeakLuminance**,

b) the **TargetedSystemDisplayMaximumLuminance.**

8.8 Color Saturation Weight

The **ColorSaturationWeight** shall be a number in the range [0,63/8] and in multiples of 1/8. This can be used to adjust the color saturation mapping gain as described in Annex B.5.3. The default value shall be 1.

9 Application Constraints

9.1 Metadata Set

If the underlying image set is modified, the metadata set might no longer apply.

A metadata set shall contain exactly one of each of the following:

- **ApplicationIdentifier** (= 4)
- **ApplicationVersion** (= 0 or 1)
- **TimeInterval**
  - which shall include one of each of the following metadata items defined in SMPTE ST 2094-1:

    TimeIntervalStart

    TimeIntervalDuration
- **ProcessingWindow**
  - which shall include one of each of the following metadata items defined in SMPTE ST 2094-1:

    UpperLeftCorner

    LowerRightCorner

    WindowNumber
  - and shall include zero or one of the **EllipsePixelSelector** metadata group which shall contain one of each of the following metadata items defined in this document:

    CenterOfEllipse

    RotationAngle

    SemiMajorAxisInternalEllipse

    SemiMinorAxisExternalEllipse

    SemiMajorAxisExternalEllipse

    OverlapProcessOption
- **TargetedSystemDisplay**
  - which shall include the following metadata item defined in SMPTE ST 2094-1:

    TargetedSystemDisplayMaximumLuminance
  - and shall include zero or one of the following metadata items defined in Section 7.2:

    TargetedSystemDisplayActualPeakLuminance
- **ColorVolumeTransform**
  - which shall include one of each of the following metadata items as defined in this document:

    MaxSCL

    AverageMaxRGB

    DistributionMaxRGB

    FractionBrightPixels
  - and shall include zero or one of the following metadata item as defined in Section 8.2:

    MasteringDisplayActualPeakLuminance
  - and shall include zero or one of the following metadata item as defined in this document:

KneePoint

if **KneePoint** is present, then shall include exactly one of the following metadata item as defined in Section 8.7.3, otherwise the item shall not be included:

BezierCurveAnchors

- and shall include zero or one of the following metadata item as defined in Section 8.8:

ColorSaturationWeight

9.2 Processing Window Constraints

For each targeted system there shall be a maximum of 3 processing windows within one image.

Processing Window 0 shall be always present and shall cover all pixels in an image and shall not be extended with the elliptical pixel selector.

Processing Window 1 and 2, if present, shall be extended with an elliptical pixel selector.

9.3 Metadata Set ApplicationVersion = 0 Constraints

The metadata set should not include the following optional items:

- **TargetedSystemDisplayActualPeakLuminance,**
- **MasteringDisplayActualPeakLuminance**,
- **ColorSaturationWeight**

Any **ProcessingWindow.WindowNumber** > 0 should not be present.

9.4 Metadata Set ApplicationVersion = 1 Constraints

The metadata set shall not include the following optional items:

- **TargetedSystemDisplayActualPeakLuminance,**
- **MasteringDisplayActualPeakLuminance**,
- **ColorSaturationWeight**

Any **ProcessingWindow.WindowNumber** > 0 shall not be present.

10 Computation of FractionBrightPixels

10.1 General

This section defines calculations for **FractionBrightPixels** for **ApplicationVersion = 0 and ApplicationVersion = 1**. A mathematically equivalent calculation may be performed while calculating the metadata set. **FractionBrightPixels** = 0 further indicates that this metadata item might not represent a calculation from the image statistics.

FractionBrightPixels is abbreviated in Section 10 as F_{BP}.

10.2 Common computations for Section 10

The following parameters are used:

k = frame index within a scene

k' = frame index of the brightest frame
K = the set of all frames in the scene
M = number of frames in the scene
z = selected pixel position in the frame

10.2.1 Calculating *Y*

Y_{linear}, the luminance of a pixel z in frame *k* shall be calculated using Equation (6).

$$Y_{linear}(z,k) = 0.2627 \times R_{linear}(z,k) + 0.678 \times G_{linear}(z,k) + 0.0593 \times B_{linear}(z,k) \qquad (6)$$

where

$R_{linear}, G_{linear}, B_{linear}$ = linearized RGB components of a pixel.

Note: This is the luminance equation for the ITU-R Rec BT.2020 color space with a D65 white point where the Y result is in the range [0,1] per Section 4.6.

10.2.2 Calculating count of all pixels in the frame

H = image height
W = image width
i = index over height
j = index over width

$Z = \{ (i,j) \mid i = 0,1,\ldots,H-1 \text{ and } j = 0,1,\ldots,W-1 \}$, the set of all pixels in the calculated frame

T = count (Z) = H x W

Note: H and W can be the size of the image; either original or proxy (as resized for calculation).

10.3 Computation of FractionBrightPixels for ApplicationVersion = 0

FractionBrightPixels should be 0 for **ApplicationVersion** = 0.

Note: Metadata sets generated according to ST 2094-40:2016 might have FractionBrightPixels > 0.

10.4 Computation of FractionBrightPixels for ApplicationVersion = 1

10.4.1 Description of Constraints

Section 10.4 shall apply when the ApplicationVersion = 1.

The brightest frame in the scene shall be the frame with the highest average luminance.

The luminance band shall range from Y_{max}, the luminance of the brightest of the smoothed pixels, as the maximum bound to a minimum luminance of $(Y_{max} - (\frac{5}{255}))$.

The smoothed pixels calculation method shall be creation of a ‘proxy frame’ that is a 5:1 image reduction resizing operation on the image.

The weighting function, f(ε), shall be

$$f(\varepsilon) = \begin{cases} 1, & \varepsilon < \mu 1 \\ \left(\frac{\varepsilon - \mu 2}{\mu 1 - \mu 2}\right), & \mu 1 \le \varepsilon < \mu 2 \\ 0.0, & \mu 2 \le \varepsilon \end{cases} \qquad (7)$$

$\mu 1 = \frac{1}{255}$,

$\mu 2 = \frac{5}{255}$

FractionBrightPixels shall be 0 for any **WindowNumber** > 0.

10.4.2 Calculating the average luminance per frame

$S(k)$ is a function that returns the average luminance of the smoothed pixels for proxy frame, *k*, in the scene and shall be calculated using Equation (8). Equation (8) uses Equation (6) for calculation of luminance Y.

$$S(k) = \frac{1}{T} \sum_z Y_{linear}(z, k), \qquad \text{for } \forall\, k \in K,\ \forall\, z \in Z \qquad (8)$$

where:

$S(k)$ = average luminance value of proxy frame k

10.4.3 Selection of the brightest frame in the scene

The brightest proxy frame k' shall be the highest frame index, k, that satisfies condition (9).

$$S(k') \ge S(k), \quad \text{for } \forall\, k \in K \qquad (9)$$

10.4.4 Calculating the brightest pixel, Y, for the brightest frame

The largest value of Y in proxy frame k' is from Equation (10). Equation (10) uses Equation (6) for calculation of luminance Y.

$$Y_{max}(k') = \max\{\, Y(z, k') \mid \forall\, z \in Z \} \qquad (10)$$

10.4.5 Calculating FractionBrightPixels

FractionBrightPixels, F_{BP} , (shall be calculated using Equation (11). Equation (11) uses Equation (6) for calculation of luminance Y.

$$F_{BP} = \frac{1}{T} \sum_z f(\, Y_{max}(k') - Y(z, k')\,) \qquad (11)$$

where:

$f(\varepsilon) = \text{Equation } 7$ in Section 10.4.1

Annex A Mapping of Application #4 to the Generalized Color Transform Model (Informative)

The diagram in Figure A.1 describes Application #4 Scene-based Color Volume Mapping in the framework of the Generalized Color Transform Model described in SMPTE ST 2094-1. The processing blocks applied are the Scene-based Tone Mapping and the Scene-based Color Saturation Mapping.

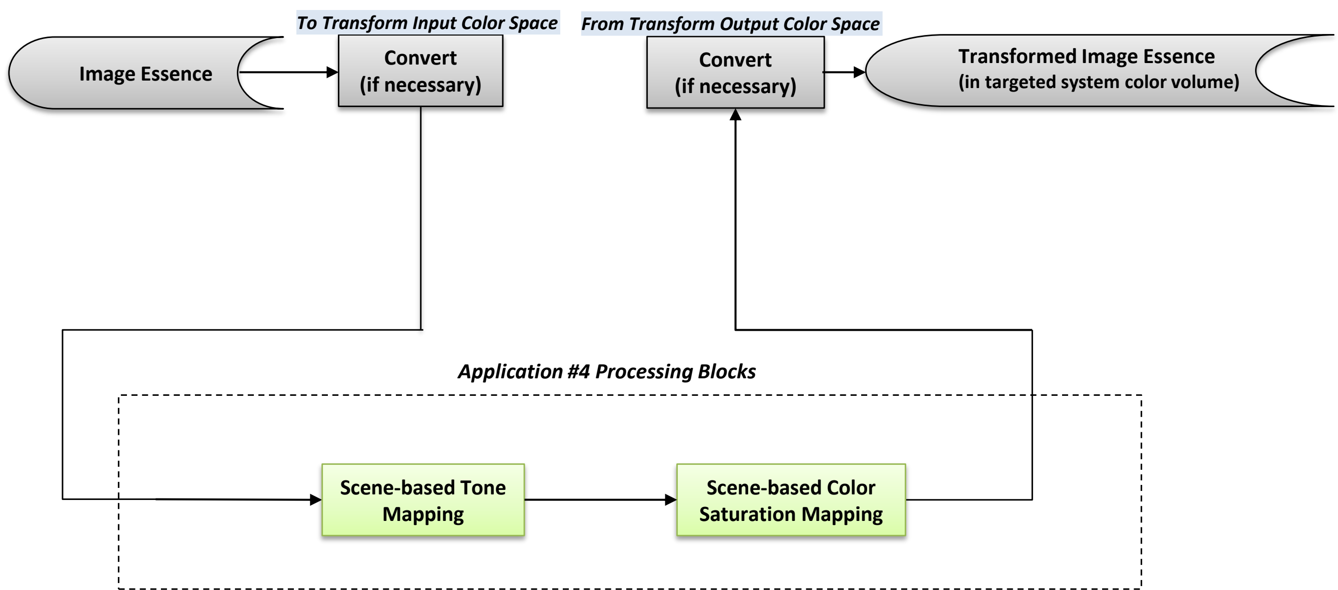

Figure A.1 – Processing blocks used by Application #4

Annex B Scene-based Color Volume Mapping Method Description (Informative)

B.1 Introduction

The scene-based color volume mapping for Application #4 is applied to the linearized color components of the source content. The proposed method uses a metadata set defined in Section 9.1 and processes the linearized RGB components, $R_{linear}, G_{linear}, B_{linear}$, that are defined in Section 4.5. The following are examples of the calculations involved, but a processing pipeline can use different mathematics with similar effect.

B.2 Actual Peak Luminance(s)

The actual display peak luminance of the mastering display and target display as they vary under real conditions are included as optional LUT metadata items. **AverageMaxRGB** is used as one index which can be proportional to the amount of power used in an LCD backlight. The other index, **FractionBrightPixels**, describes frames with different proportions of high luminance values. Together these can be used to roughly profile the upper limits of the performance of a display.

For populating the sampled function that returns actual display peak luminance, neutral-axis frames with the desired function characteristics of **FractionBrightPixels** and **AverageMaxRGB** can be used.

The output of the LUT can be used to support calculations in Annex B. An example of this use is in Equation B.13. Actual peak luminances can be computed as follows.

B.2.1 Source Normalized Actual Peak Luminance

When the metadata necessary for Equation (B.1) is present, M_P is the source normalized peak luminance of the input image essence as displayed on the mastering display used for mastering the image essence, otherwise M_P is not present. This value can be normalized by a non-zero **Maximum Display Mastering Luminance** (from SMPTE ST 2086) and given by Equation (B.1):

$$M_P = M_{RP} \times S_{MC} \times 10000/M_{ML} \tag{B.1}$$

where:

$M_{RP} =$ **MasteringDisplayActualPeakLuminance** (F_{BP}, S_{AMC}).

$M_{ML} =$ **MaximumDisplayMasteringLuminance** from SMPTE ST 2086,

$S_{MC} =$ max(**MaxSCL)** defined in Section 8.3, in the range [0,1],

$F_{BP} =$ **FractionBrightPixels**, in the range [0,1],

$S_{AMC} =$ **AverageMaxRGB**, in the range [0,1].

B.2.2 Target Normalized Actual Peak Luminance

When the metadata necessary for Equation (B.2) is present, T_P is the normalized and actual displayed peak luminance on the Targeted display, otherwise T_P is not present. This value can be normalized by a non-zero **TargetedSystemDisplayMaximumLuminance** and given by Equation (B.2):

$$T_P = T_{RP} \times \ S_{MC} \ \times 10000/T_{ML} \tag{B.2}$$

where

T_{RP} = **TargetedSystemDisplayActualPeakLuminance** (F_{BP}, S_{AMC}).

T_{ML} = **TargetedSystemDisplayMaximumLuminance**,

S_{MC} = max(**MaxSCL)** defined in Section 8.3 in the range [0,1],

F_{BP} = **FractionBrightPixels** in the range [0,1],

S_{AMC} = **AverageMaxRGB** in the range [0,1].

B.3 Color Components Normalization

Input linearized RGB components can be normalized using Equation (B.3):

$$\begin{bmatrix} R_{norm} \\ G_{norm} \\ B_{norm} \end{bmatrix} = \begin{cases} \mathrm{VecMin}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \dfrac{1}{M_{RP}} \times \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix}\right), & if\ M_{RP}\ \ available \\ \mathrm{VecMin}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \dfrac{1}{S_{norm}} \times \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix}\right), & otherwise \end{cases} \tag{B.3}$$

where:

$R_{norm}, \ G_{norm}, B_{norm}$ = normalized linearized color components,
S_{norm} = the maximum of the elements of **MaxSCL** defined in Section 8.3, if non-zero, or the value of **DistributionMaxRGBPercentiles**[Ω] as defined in Section 8.5.
$\mathrm{VecMin}()$ = a vector min( ) function working component-wise,
M_{RP} = **MasteringDisplayActualPeakLuminance** (F_{BP}, S_{AMC}) if present.

B.4 Scene Adaptive Tone Mapping

B.4.1 Descriptive Scene Characteristics

The following metadata items provide scene statistics of the content of a scene which can be used alone or in combination to create an optimized mapping to a particular target display range which can also be other than the **TargetedSystemDisplayMaximumLuminance** this metadata set was established for.

MaxSCL

AverageMaxRGB

DistributionMaxRGB

FractionBrightPixels

All of these items are required to be present in the metadata set and can be useful if **KneePoint** and **BezierCurveAnchors** are not present. **DistributionMaxRGB** can be used to identify where there are many image code values by comparing the percentages with the percentile linearized maxRGB values for all scene values that are above the target luminance.

There can be varying implementations of how the scene values are processed depending upon the vendor, the internal capabilities of their display device, and consumer settings. The range of the display device is possibly different than the target range of the metadata set.

Scene statistics can be compared to each other and to the luminance values available on a display device. In general, the range of the display device is compared to the target display of this metadata set and to the range of the content to determine where correction is needed and how best to preserve creative intent in the color reproduction. Tonal ranges in the scene are identified for either compression or expansion of the tone curve. Overall dimming or brightening to fit within the display device range and user display settings can be performed. **DistributionMaxRGB** can be used to create or modify the **BezierCurveAnchors** and **KneePoint** for the **TargetedSystemDisplayMaximumLuminance.**

B.4.2 Tone Mapping with the Bezier N-th Order Curve

If metadata items **KneePoint** and **BezierCurveAnchors** are present, this section can be used for calculation of $R_{stm}, G_{stm}, B_{stm}$ using $F_N(s)$ from Section 8.7

B.4.2.2 Tone Mapping Calculation

Normalized color components can be tone mapped using Equation (B.4):

$$\begin{bmatrix} R_{stm} \\ G_{stm} \\ B_{stm} \end{bmatrix} = \text{VecMin}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, r(s) \times \begin{bmatrix} R_{norm} \\ G_{norm} \\ B_{norm} \end{bmatrix}\right) \tag{B.4}$$

where:

$$r(s) = \begin{cases} \frac{F_N(s)}{s}, & s > 0 \\ 0, & s = 0 \end{cases}$$

$R_{norm}, G_{norm}, B_{norm}$ = normalized linearized color components (see B.3),

$R_{stm}, G_{stm}, B_{stm}$ = the tone-mapped normalized color components,

$s = \max(R_{norm}, G_{norm}, B_{norm})$,

$F_N(s)$ = the tone-mapping function as defined in Section 8.7.

B.4.3 Example: Tone Mapping with the Bezier 4th Order Curve

In this example, the tone-mapping function uses a 4th order Bezier curve defined in Equation (B.5)

$$F_4(s) = \begin{cases} \frac{K_F}{K_S} s, & 0 \le s < K_S, K_S \ne 0 \\ K_F + (1 - K_F) B_4\left(\frac{s - K_S}{1 - K_S}\right), & K_S \le s \le 1, K_S \ge 0 \end{cases} \tag{B.5}$$

where:

$0 < K_S$,

$B_4(t) = 4 \times t \times (1-t)^3 \times P_1 + 6 \times t^2 \times (1-t)^2 \times P_2 + 4 \times t^3 \times (1-t) \times P_3 + t^4$.

The three anchor parameters P_1, P_2, P_3 determine the behavior of the Bezier curve. For instance, Figure B.2 shows the curves for $P_1 = P_2 = P_3 = 1$ and $P_1 = 0,\ P_2 = P_3 = 1$.

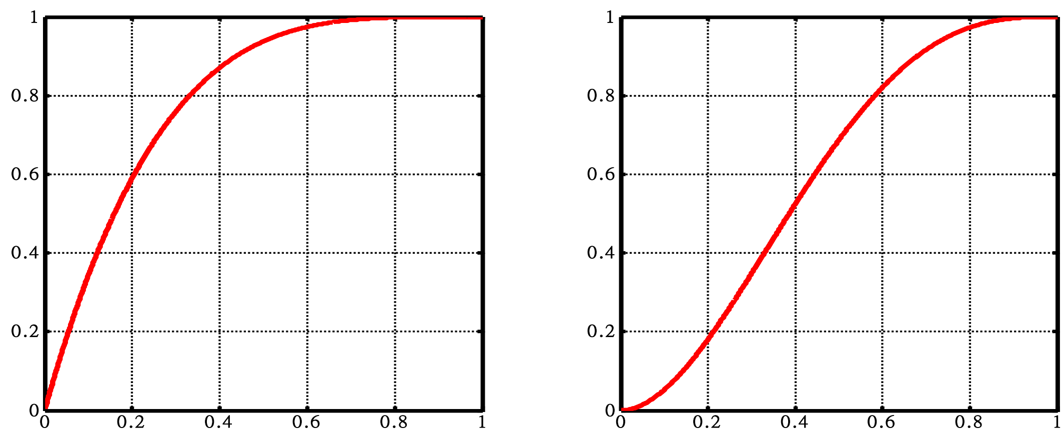

Figure B.2 – Examples of the 4th Order Bezier curves for $P_1 = P_2 = P_3 = 1$ and $P_1 = 0, P_2 = P_3 = 1$

B.5 Scene-based Color Saturation Mapping

B.5.1 Color Saturation Mapping

This section demonstrates use of optional **ColorSaturationWeight** if present. The saturation mapping can compensate the decrease in color saturation in the targeted system display with a smaller color volume. The color saturation is compensated according to the actual luminance difference between the source image and the tone-mapped image.

B.5.2 Color Saturation Mapping Process

Prior to color saturation mapping, color components are converted from RGB to $Y'C_B'C_R'$. Tone mapped RGB components (R_{stm}, etc.) exist either from the method of Annex B.4 or from other tone mapping methods that are out of scope of this standard. After color saturation mapping, they are converted back to RGB.

RGB Conversion to $Y'C_B'C_R'$ is performed using Equations (B.6) and (B.7)

$$\begin{bmatrix} R_{stm}' \\ G_{stm}' \\ B_{stm}' \end{bmatrix} = I_{1886}\left(\begin{bmatrix} R_{stm} \\ G_{stm} \\ B_{stm} \end{bmatrix}\right) \qquad \text{(B.6)}$$

$$\begin{bmatrix} Y_{stm}' \\ C_{B,stm}' \\ C_{R,stm}' \end{bmatrix} = Q_{2020} \times \begin{bmatrix} R_{stm}' \\ G_{stm}' \\ B_{stm}' \end{bmatrix} \qquad \text{(B.7)}$$

where:

$Y_{stm}',\ C_{B,stm}',\ C_{R,stm}'$ = Tone mapped color components,

I_{1886} = inverse of component-wise EOTF defined in ITU-R Rec. BT.1886-0,

Q_{2020} = RGB to $Y'C'_BC'_R$ transformation matrix from ITU-R Rec. BT.2020-2, given by

$$Q_{2020} = \begin{bmatrix} 0.2627 & 0.6780 & 0.0593 \\ -0.1396 & -0.3604 & 0.5000 \\ 0.5000 & -0.4598 & -0.0402 \end{bmatrix}.$$

Color saturation mapping can be performed using Equation (B.8):

$$\begin{bmatrix} Y'_{scsm} \\ C'_{B,scsm} \\ C'_{R,scsm} \end{bmatrix} = \begin{bmatrix} Y'_{stm} \\ S_{scsm} \times C'_{B,stm} \\ S_{scsm} \times C'_{R,stm} \end{bmatrix} \quad \text{(B.8)}$$

where:

$S_{scsm} = \min(f_{SCSM}(s), \eta)$,
$f_{SCSM}()$ = color saturation mapping function (as defined in Annex B.5.3),
$\eta =$ maximum color saturation mapping gain (as defined in Annex B.5.4).

Following color saturation mapping, $Y'_{scsm}, C'_{B,scsm}, C'_{R,scsm}$ can be converted back to RGB components using Equations (B.9) and (B.10).

$$\begin{bmatrix} R_{scsm}' \\ G_{scsm}' \\ B_{scsm}' \end{bmatrix} = M_{2020} \times \begin{bmatrix} Y'_{scsm} \\ C'_{B,scsm} \\ C'_{R,scsm} \end{bmatrix} \quad \text{(B.9)}$$

$$\begin{bmatrix} R_{scsm} \\ G_{scsm} \\ B_{scsm} \end{bmatrix} = E_{1886}\left(\begin{bmatrix} R_{scsm}' \\ G_{scsm}' \\ B_{scsm}' \end{bmatrix}\right) \quad \text{(B.10)}$$

where:

$R_{scsm}, G_{scsm}, B_{scsm}$ are color saturation mapping color components.

E_{1886} = component-wise EOTF defined in Rec. ITU-R BT.1886-0,

M_{2020} = inverse of Q_{2020} matrix given by

$$M_{2020} = \begin{bmatrix} 1.0000 & 0.0000 & 1.4746 \\ 1.0000 & -0.1646 & -0.5714 \\ 1.0000 & 1.8814 & 0.0000 \end{bmatrix}.$$

B.5.3 Color Saturation Mapping Weight

If **ColorSaturationWeight** and the other metadata terms in the equation are present, color saturation mapping is shown in Figure B.3 and can be calculated with Equation (B.11) and limited by Equation B.12:

(B.11)

$$f_{SCSM}(s) = 1 + \mathbf{ColorSaturationWeight} \times \max\left(0, \frac{\log(M_p \times M_{ML} \times s)}{\log\left(T_p \times T_{ML} \times F_N(s)\right)} - 1\right)$$

where:

$s = \max(R_{norm}, G_{norm}, B_{norm})$,

M_P = source normalized actual peak luminance as defined in Annex B.2.1,

T_P = target normalized actual peak luminance as defined in Annex B.2.2,

M_{ML} = **MaximumDisplayMasteringLuminance** from SMPTE ST 2086,

T_{ML} = **TargetedSystemDisplayMaximumLuminance**,

$F_N(s)$ = Tone-mapping function as defined in Section 8.7.

If either M_P or T_P are not present or undefined, both of them are omitted from the equation.

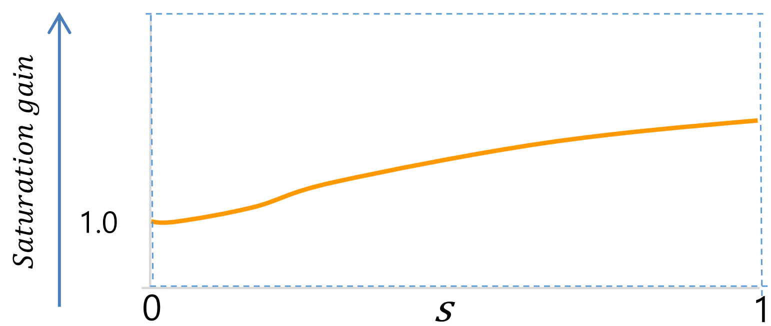

Figure B.3 – Color saturation mapping function

B.5.4 Color Saturation Mapping Maximum Gain

Color saturation gain can be limited to a maximum value to avoid clipping when compensated RGB values become negative or larger than unity. The maximum gain, η, satisfies the conditions of Equation (B.12):

$$\begin{gathered} 0 \le M_{11} \times Y'_{stm} + \eta(M_{12} \times C'_{B,stm} + M_{13} \times C'_{R,stm}) \le 1 \\ 0 \le M_{21} \times Y'_{stm} + \eta(M_{22} \times C'_{B,stm} + M_{23} \times C'_{R,stm}) \le 1 \\ 0 \le M_{31} \times Y'_{stm} + \eta(M_{32} \times C'_{B,stm} + M_{33} \times C'_{R,stm}) \le 1 \\ 0 \le \eta \le 4. \end{gathered} \tag{B.12}$$

where:

$[M_{ij}]$ = the elements of M_{2020}.

B.6 Processing Window With Elliptical Pixel Selector

B.6.1 Introduction

In a processing window extended with the elliptical pixel selector, the tone-mapping function, as described in Annex B.4.1, operates on the pixels in the intersection of the elliptical pixel selector and the corresponding **WindowNumber**. Local pixel processing primarily targets the pixels inside the internal ellipse and is not intended for the pixels outside of the external ellipse.

To prevent any visible boundary between the pixels selected by the elliptical pixel selector and the region outside of the external ellipse, we introduce a transition weighting function $w(x,y)$ at the position (x,y) by evaluating the normalized sum of distances to the foci of the external ellipse. The transition weighting function is 1 inside the internal ellipse, 0 outside of the external ellipse, and gradually change from 1 to 0 as the considered position moves from the internal ellipse to the external ellipse.

B.6.2 Definitions

From the metadata set for an extended window, there are definitions for:

(x0, y0) **= CenterOfEllipse**
θ **= RotationAngle**
a1**=SemiMajorAxisInternalEllipse**
b2**=SemiMinorAxisExternalEllipse**
a2**=SemiMajorAxisExternalEllipse**
OverlapProcessOption

An ellipse is the locus of points P(x,y), the sum of whose distances from two fixed points is constant.

B.6.3 Transition Weighting Function

For pixels selected by the elliptical pixel selector, we rotate the pixel position, (x,y), in a clockwise direction, over the ellipse center by θ. That is, we get in Equation (B.13)

$$\begin{bmatrix} X \\ \mathrm{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{B.13}$$

Then, with simple mathematical manipulation, it can be shown by removing the offset (x0,y0), the ellipse is centered at the origin of the coordinate system and is aligned with the axes. D uses the canonical ellipse equation to sum the distances from the position (x,y) to the two external ellipse foci as shown in Equation (B.14).

$$D(x,y) = \sqrt{\frac{(X - x_0)^2}{a_2^2} + \frac{(Y - y_0)^2}{b_2^2}} \tag{B.14}$$

Because the internal and external ellipse trace a symmetrical locus, D represents the distance between the two ellipses. With this, the transition weighting function $w(x,y)$ can be designed as shown in Figure B.4. The transition weighting function is composed of a unity part and a curve part, which are separated by the point ($\frac{a_1}{a_2}$,1) as depicted in Figure B.4. The curve part of the transition weighting function can be any monotonically descending curve designed with continuous derivatives to the adjoining segments. $D(x,y) > 1$ when (x,y) is outside the external ellipse, $D(x,y) = 1$ when (x,y) is on the external ellipse, $\frac{a_1}{a_2} < D(x,y) < 1$ when (x,y) is between the internal and the external ellipses, $D(x,y) = \frac{a_1}{a_2}$ when (x,y) is on the internal ellipse, and $D(x,y) < \frac{a_1}{a_2}$ when (x,y) is inside the internal ellipse. An example Equation (B.15) for $w(x,y)$ is given by

$$w(x,y) = \begin{cases} 1, & D(x,y) < \frac{a_1}{a_2} \\ 0.5 + 0.5 \times \cos\left(\frac{(a_2 \times D(x,y) - a_1) \times \pi}{a_2 - a_1}\right), & \frac{a_1}{a_2} \le D(x,y) < 1 \\ 0, & 1 \le D(x,y) \end{cases} \quad \text{(B.15)}$$

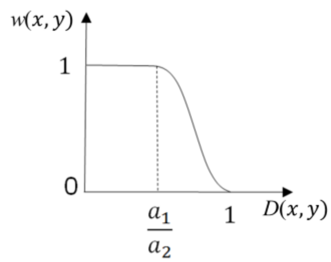

Figure B.4 – Transition Weighting Function

B.6.4 Output Pixel Values for up to Two Elliptical Pixel Selectors

B.6.4.1 Example Methods

Two example methods of pixel processing for up to two elliptical pixel selectors in one image are described in this section.

B.6.4.2 Weighted Average Method

As a first method of rendering, we describe a way of weighted averaging for two pixel selectors in one image. The output of the weighted average method, $\psi_{WA}(x,y)$, is given as Equation (B.16).

$$\psi_{WA}(x,y) = \begin{cases} \psi_{scsm,0}\,(x,y)\,, & if\ \sum_{l=1}^{Q} w_l(x,y) = 0 \\ \sum_{j=1}^{Q}\left[w_j(x,y) \times \psi_{scsm,j}\,(x,y) + \left(1 - w_j(x,y)\right) \times \psi_{scsm,0}\,(x,y)\right] \times \gamma_j(x,y)\,, & otherwise. \end{cases} \quad \text{(B.16)}$$

where:

Q = number of the elliptical pixel selectors in the image ∈ {1,2},

$w_j(x,y)$= transition weighting function for Processing Window j ∈ {1,2},

$\psi_{scsm,0}\,(x,y)$ = $(R_{scsm}, G_{scsm}, B_{scsm})$ output of Equation (12) for Processing Window 0 at pixel position (x,y),

$\psi_{scsm,j}\,(x,y)$ = $(R_{scsm}, G_{scsm}, B_{scsm})$ output of Equation (12) for Processing Window j ∈ {1,2} at pixel position (x,y).

$$\gamma_j(x, y) = \frac{w_j(x,y)}{\sum_{l=1}^{Q} w_l(x,y)}.$$

B.6.4.3 Layering Method

As a second method of rendering, we describe a way of layering for up to two pixel selectors in one image. Layering gives priority to the outputs of the processing windows extended with the elliptical pixel selector. Suppose the priority of the elliptical pixel selectors is given in accordance with the window number, that is, processing **WindowNumber**=1 has the higher priority (top layer) and processing **WindowNumber**=2 has the lower priority (bottom layer).

The output of the layering method, $\psi_L(x, y)$, is given as Equation (B.17)

$$\psi_L(x, y) = \begin{cases} w_1(x, y) \times \psi_{scsm,1}(x, y) + \left(1 - w_1(x, y)\right) \times \psi_{scsm,0}(x, y) & if\ w_1(x, y) \neq 0,\ else \\ w_2(x, y) \times \psi_{scsm,2}(x, y) + \left(1 - w_2(x, y)\right) \times \psi_{scsm,0}(x, y) & if\ w_2(x, y) \neq 0,\ else \\ \psi_{scsm,0}(x, y) & otherwise. \end{cases}$$

(B.17)

Annex C DistributionMaxRGB Example (informative)

In this example, linearized **MaxRGB** values are between the range 0.2 to 0.65 with each number having 10% of the total values and the number represented as the smallest in the range.

The histogram is

1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	frequency
0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	value

The CFD then is

1/10	2/10	3/10	4/10	5/10	6/10	7/10	8/10	9/10	10/10	cumulative frequency
0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	value

The **DistributionMaxRGB** would be

J = { 1, 25, 50, 75, 90, 95, 99 (i.e. 99.98) }

V = { 0.2, 0.3, 0.4, 0.55, 0.6, 0.65, 0.65 }

The values under 75% and 95% appear to be from the next highest bin, however, this satisfies the condition "smallest value in the CFD such that at least J_i percent of the CFD fall at or below V_i." The value in the CFD can be thought of as the highest value in that percentage bin, so the next highest value is needed for values that go above the stated percentage of the CFD.

Annex D Revision History (informative)

D.1 Initial Version

SMPTE ST 2094-40:2016

The example in SMPTE ST 2094-2:2016 for Application 4 matches SMPTE ST 2094-40:2016.

D.2 Revision 1

SMPTE ST 2094-40:201x

This edition supersedes the SMPTE ST 2094-40:2016 edition and edits include:

- Removed 2016 Annex C
- Added Annex C and D
- Editorial changes and clarifications.

ApplicationVersion = 0 aims to maintain compatibility with the 2016 edition and includes:

- Normative description of the **OverlapProcessOption** methods. (Section 6.7)
- **MaxSCL** = (0,0,0) does not represent a calculated value. (Section 8.3)
- In **DistributionMaxRGB.percentages,** use percentage value of 99.98 whenever ‘99’ appears. (Section 8.5.3)
- **FractionBrightPixels** is recommended to be 0 and 0 does not represent a calculated value. (Section 8.6)
- Calculation of Bezier curves adds cases for Ks = 0 and 1. (Section 8.7.4)
- Some optional metadata items and WindowNumber > 0 are not recommended. (Section 9.3)

ApplicationVersion = 1 addresses new constraints or requirements:

- **DistributionMaxRGB** has a 9-position Cumulative Frequency Distribution with two exception positions. Use percentage value 99.98 whenever ‘99’ appears. (Section 8.5.4)
- **BezierCurveAnchors** has length of [0..9]. (Section 8.7.3)
- Some optional metadata items and WindowNumber > 0 are not permitted. (Section 9.4)
- **FractionBrightPixels** has a new calculation method (Section 10)

Bibliography (Informative)

Recommendation ITU-R BT.1886-0 (3/2011), Reference Electro-optical Transfer Function for Flat Panel Displays used in HDTV Studio Production

Recommendation ITU-R BT.2020-2 (10/2015), Parameter Values for Ultra-high Definition Television Systems for Production and International Programme Exchange

SMPTE ST 2084:2014 High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays