ISO/TC 42/WG 23 **N 7088**

ISO/TC 42/WG 23 "Joint ISO/TC 42 - ISO/TC 130 - CIE WG: Extended colour encodings for digital image storage, manipulation and interchange"
Convenorship: **ANSI**
Convenor: **Foshee Scott Dr**

ICC and ISO 21496-1

Document type	Related content	Document date	Expected action
Project / Other	Project: ISO/WD 21496-1	2023-12-15	**INFO**

ICC and ISO 21496-1

• ICC and ISO

In 2003 the **ICC** entered into a “Co-operative Agreement between **ISO/TC130** and the International Color Consortium” which established the detailed procedures whereby **ISO/TC130** (Graphic technology) and the International Color Consortium (ICC) will cooperate to continue the development of a series of ISO standards based on the work of the ICC, including the ICC Profile Specification.

ICC.1:2004-08	<->	ISO 15076-1:2005 Image technology colour management - Architecture, profile format ,and data structure
ICC.1:2010	<->	ISO 15076-1:2010
ICC.2:2023	<->	ISO 20677:2019

Compared with ICC.1, ICC.2 has many extensions enabled for **HDR/WCG workflows.**

Now **ISO/TC42** is developing a standard ISO 21496-1 Digital photography - Gain map metadata for image conversion –Part 1: Dynamic range conversion, which defines a gain map used in **HDR digital photography applications**, for dynamic range conversion between image representations. This includes the definition of the gain map metadata and its attributes, how to apply a gain map, and how to specify the gain map and other associated metadata.

ISO/TC 42/WG 23 Joint **ISO/TC 42 - ISO/TC 130** - CIE WG: Extended colour encodings for digital image storage, manipulation and interchange

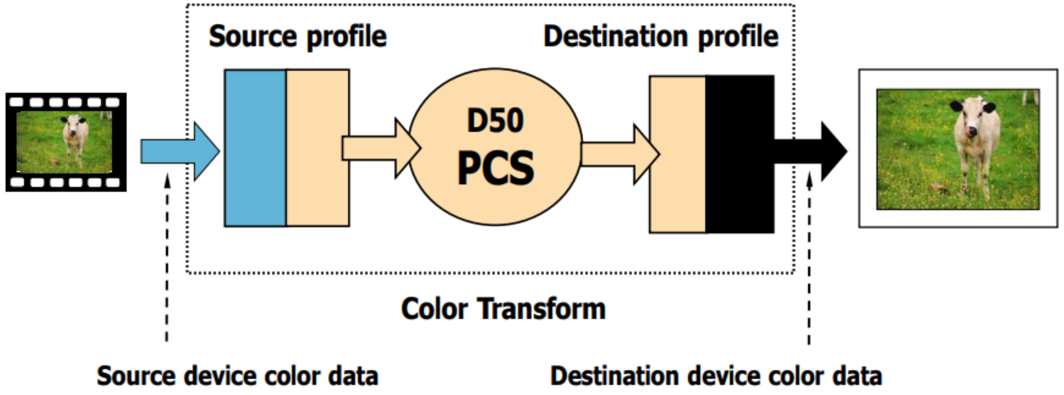
- ICC and ISO

ICC
Image technology colour management — Architecture, profile format, and data structure
support CMM workflows

ICC is more close to the definition of workflow or Architecture

ISO/TC 42/WG 23
Extended colour encodings for digital image storage, manipulation and interchange

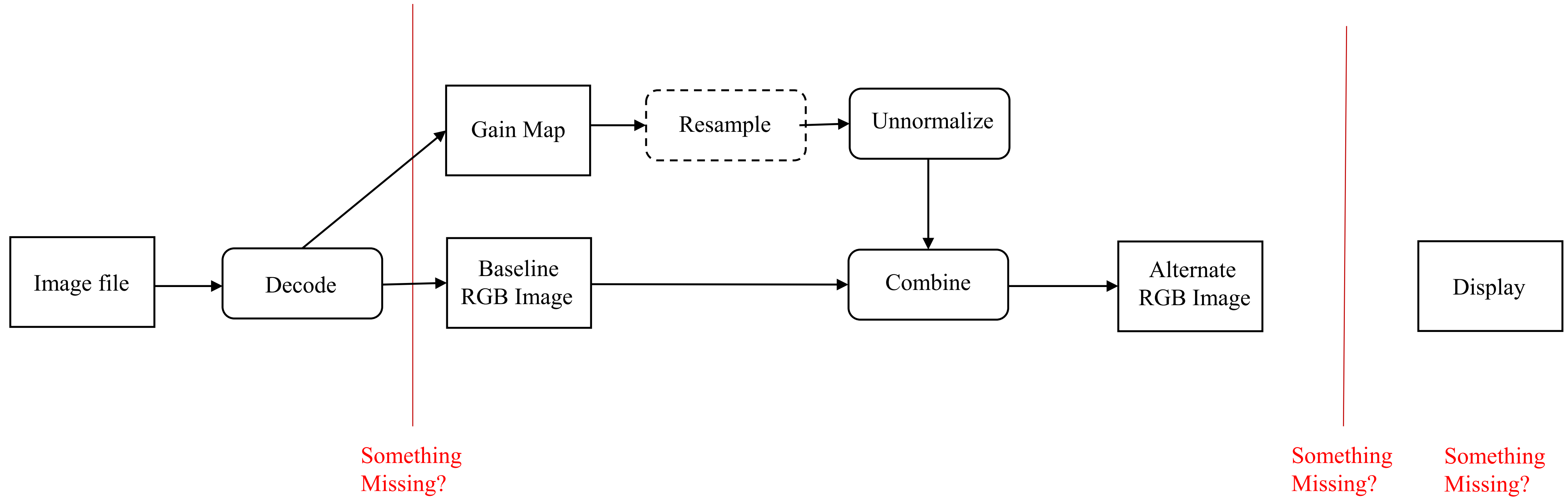
ISO 21496-1 Gain map metadata for image conversion



Cite:https://www.color.org/info_profiles2.xalter
Fundamentals_of_ICC_Color_Management

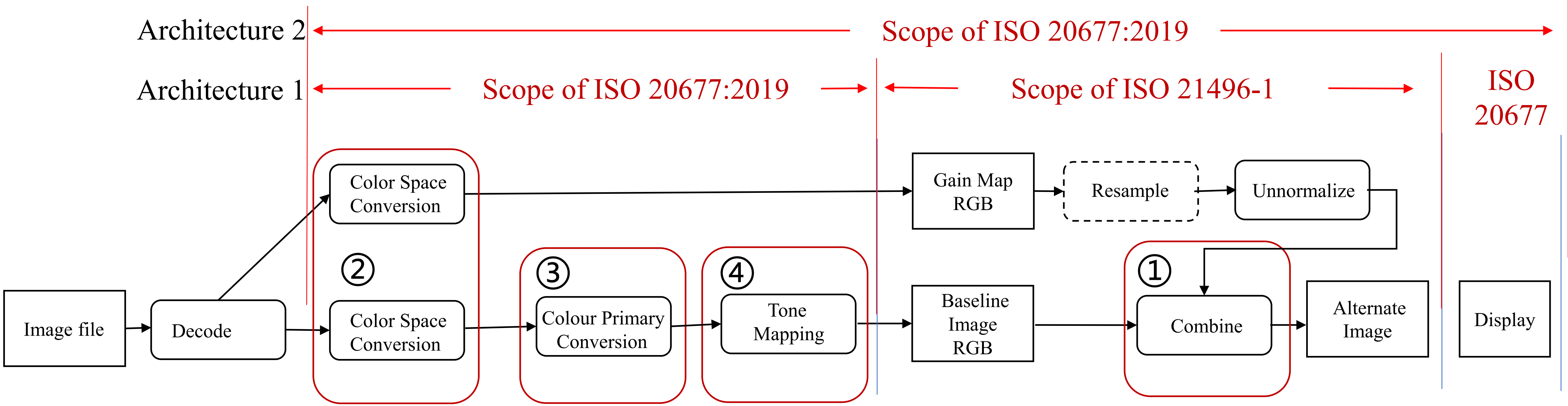
The problem is ISO 21496-1 also include how to do image conversion with Gain map Metadata and have its own workflow, which may have different Architecture with ICC.

- # Current workflow of ISO 21496-1



- The combination is performed in RGB
- Except for **color primary**, the color space of gain map and baseline image could be different from the application space, so **color space conversion** is necessary.
- **A (inverse) tone mapping function** to baseline image could efficiently reduce the energy of gain map.
- After combination, the alternate linear image shall be converted to or from the format such as PQ/HLG/Gamma.

- Our proposed workflow of ISO 21496-1 and ISO 20677:2019



① Support combining baseline image and gain map in RGB, using linear, and may also PQ/HLG/Gamma for application space .

② Support color space conversion. Eg. Decoded Gain map & baseline image are YUV, gain map application space is RGB. YUV->RGB.

③ Support color primary conversion. Eg. Baseline image is BT709, alternate image and gain map application space is BT2020. BT709->BT2020.

④ Support (inverse) tone mapping for baseline image.

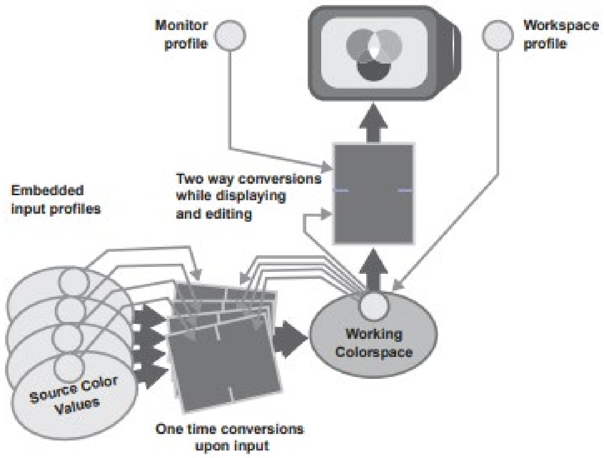
The relationship between ISO 20677:2019 and ISO 21496-1

Similar to: ⇨ Compound multi-pictures in workflow

Architecture 1: Refer ISO 20677:2019 in ISO 21496-1

- **Option 1: combination is done in working color space according to the ICC profile in baseline image (compulsory) and gain map (optional)**
  - ➢ All the required steps or functions in ISO 21496-1 could be achieved by creating and utilizing ICC profiles defined in ISO 20677:2019, no modification needed.
  - ➢ Adding unambiguous instructions to indicate how to refer ISO 20677:2019 in the processing pipeline of ISO 21496-1 is enough, which saves the time cost a lot.
- **Option 2:** only one ICC profile for alternate image
  - ➢ All color space/primary conversion of ISO 21496-1 should be defined in ISO 21496-1

- For faster implementation and higher efficiency, we request that the optimal method of combining this two is **option 1 in Architecture 1.**



Cite:https://www.color.org/info_profiles2.xalter
Why Color Management?

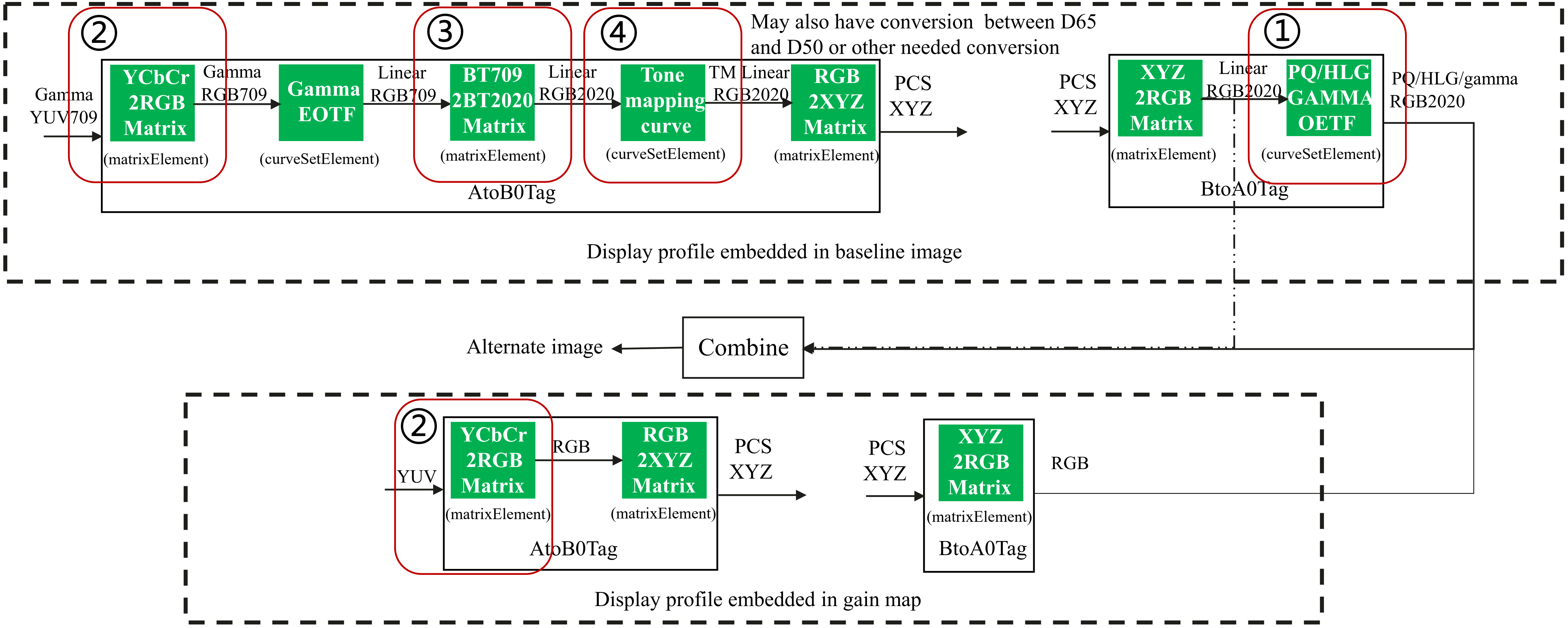
Architecture 2: Integrate ISO 21496-1 into ISO 20677:2019

- Detail is after the description of Architecture 1

Architecture 1: Proposed Method for referencing ISO 20677:2019 in ISO 21496-1

Option 1 (Method 1): one ICC for each picture (baseline and gain map), combination is done in working color space

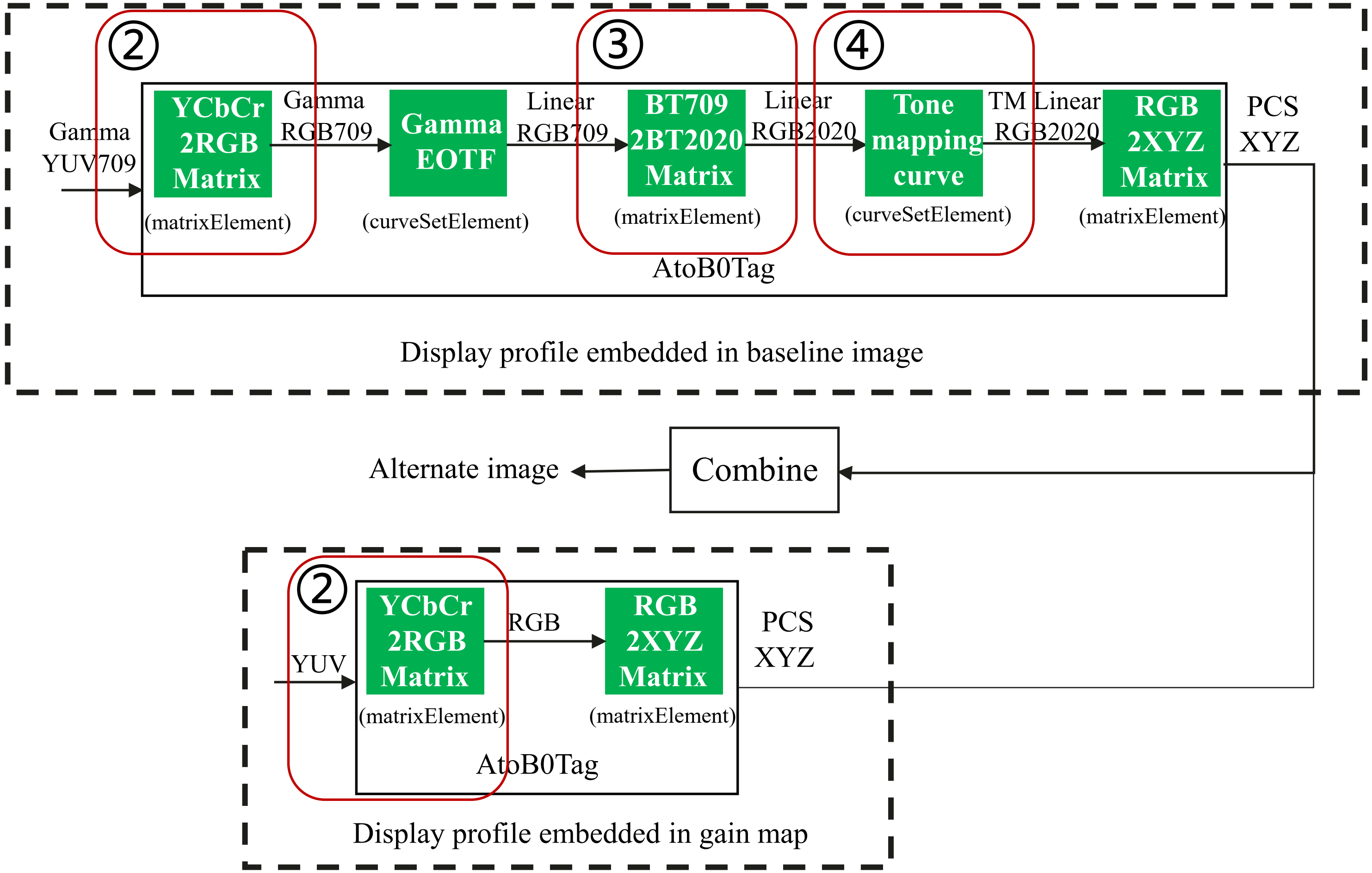
Add a cicpTag in the ICC-profile to indicate the format of alternate image.



Architecture 1: Proposed Method for referencing ISO 20677:2019 in ISO 21496-1

How about applying gain map in PCSXYZ or CIEXYZ space?

Option 1 (Method 2): one ICC for each picture (baseline and gain map), combination is done in working color space

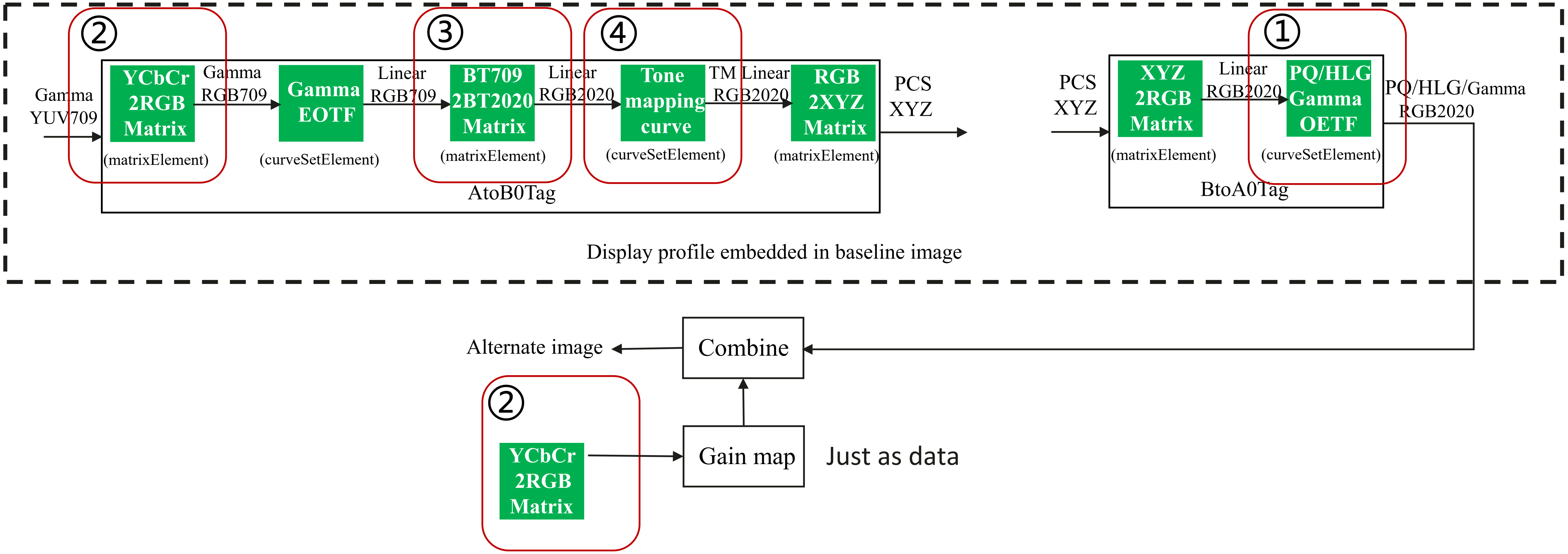


- XYZ color space is a more common color representation method, which is **independent of the characteristics of the device** and **closer to the perception of the human visual system.**

Architecture 1: Proposed Method for referencing ISO 20677:2019 in ISO 21496-1

How about taking gain map format as the default sRGB? Nearly all systems can do the YUV2RGB conversion correctly

Option 1 (Method 3): only one ICC for baseline image, combination is done in working color space



- **Gain map does not include and utilize ICC profile, it is just regarded as some kind of image data with default sRGB format.**

Solution for issue ①（Transfer Curves)

- CurveSetElement(ISO 20677:2019 11.2.2)

The Curve Set element encodes multiple one-dimensional curves. The encoding is shown in Table 107. The sampledClaculatorCurve curve type allows for the encoding of 1-dimensional curves as an arbitrary function using an embedded element of type calculatorElement. This allows for more optimal implementation as the embedded calculator element can be applied at transform setup to populate a sampled 1-dimensional lookup table that is then used for performing curve transforms. The encoding of a sampledClaculatorCurve is shown in Table 109.

Table 107 — curveSetElement encoding

Byte position	Field length (bytes)	Content	Encoded as...
0...3	4	'cvst' (63767374h) type signature	
4...7	4	Reserved, shall be 0	
8...9	2	Number of input channels (P)	uInt16Number
10...11	2	Number of output channels (Q)	uInt16Number
12..11+8P	8P	Curve positions (offset and size)	positionNumber[...]
12+8P to end		Data	

Table 109 – sampledCalculatorCurve encoding

Byte Position	Field Length (bytes)	Content	Encoded as...
0..3	4	'clcf' (636c6366h) type signature	
4..7	4	Reserved, must be set to 0	
8..12	4	Recommended number of interpolation samples for the curve (R)	uInt32Number
12..15	4	Starting input range value of curve (S)	float32Number
16..19	4	Ending input range value of curve (E)	float32Number
20.21	2	Lookup Extension type (L)	uInt16Number
22.23	2	Reserved, must be set to 0	
16+20..end		Curve Definition	calculatorElement

In our case, take **PQ OETF** as an example, the input P and output channels are all set to **3,** which means that the PQ transfer curves shall be applied to R,G,B channels separately. PQ curve is defined according to ITU-R BT.2100 by embedded **calculator element** as:

$$L_{PQ} = \left(\frac{c1+c2L^{m1}}{1+c3Y^{m1}}\right)^{m2}$$

*where m1= 2610/16384,m2=2523/4096*128,c1=3424/4096,c2=2413/4096*32,c3=2392/4096*32*

The curves shall be defined between **0(S) and 1(E)** , with Lookup Extension type(L) set to 0 for clipping abnormal value.

Solution for issue ②（Color Space Conversion）

- matrixElement(ISO 20677:2019 11.2.10)

The matrix is organized as an array of P x Q elements, where P is the number of input channels to the matrix, and Q is the number of output channels. The matrix element encoding is shown in Table 120.

array = $[e_{11}, e_{12}, \ldots, e_{1P}, e_{21}, e_{22}, \ldots, e_{2P}, \ldots, e_{Q1}, e_{Q2}, \ldots, e_{QP}, e_1, e_2, \ldots, e_Q]$

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_Q \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1P} \\ e_{21} & e_{22} & \cdots & e_{2P} \\ \cdots & \cdots & \cdots & \cdots \\ e_{Q1} & e_{Q2} & & e_{QP} \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_P \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ e_Q \end{bmatrix}$$

Table 120 — Matrix Element encoding

Byte position	Field length (bytes)	Content	Encoded as...
0..3	4	'matf' (6D617466h) type signature	
4..7	4	Reserved, shall be 0	
8..9	2	Number of input channels (P)	uInt16Number
10..11	2	Number of output channels (Q)	uInt16Number
12..end	4x(P+1)xQ	Matrix elements	float32Number[...]

In our case, take **YUVtoRGB2020** as an example, the input P and output channels Q are all set to **3**, the total transfer can be expressed as:

$$\begin{matrix} R \\ G \\ B \end{matrix} = \begin{bmatrix} 1.0000 & -0.0000 & 1.4746 \\ 1.0000 & -0.1645 & 0.5713 \\ 1.0000 & 1.8814 & -0.0001 \end{bmatrix} * \begin{matrix} Y \\ U \\ V \end{matrix} + \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}$$

So the array could be written as $array = [1.0000, -0.0000, 1.4746, 1.0000, -0.1645, 0.5713, 1.0000, 1.8814, -0.0001, 0, 0, 0]$

Solution for issue ③ （Color Primary Conversion）

- matrixElement(ISO 20677:2019 11.2.10)

The matrix is organized as an array of P x Q elements, where P is the number of input channels to the matrix, and Q is the number of output channels. The matrix element encoding is shown in Table 120.

$$\text{array} = [e_{11}, e_{12}, \ldots, e_{1P}, e_{21}, e_{22}, \ldots, e_{2P}, \ldots, e_{Q1}, e_{Q2}, \ldots, e_{QP}, e_1, e_2, \ldots, e_Q]$$

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \ldots \\ Y_Q \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \ldots & e_{1P} \\ e_{21} & e_{22} & \ldots & e_{2P} \\ \ldots & \ldots & \ldots & \ldots \\ e_{Q1} & e_{Q2} & & e_{QP} \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ \ldots \\ X_P \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \ldots \\ e_Q \end{bmatrix}$$

Table 120 — Matrix Element encoding

Byte position	Field length (bytes)	Content	Encoded as...
0..3	4	'matf' (6D617466h) type signature	
4..7	4	Reserved, shall be 0	
8..9	2	Number of input channels (P)	uInt16Number
10..11	2	Number of output channels (Q)	uInt16Number
12..end	4x(P+1)xQ	Matrix elements	float32Number[...]

In our case, take **RGB709toRGB2020** as an example, the input P and output channels Q are all set to **3**, the total transfer can be expressed as:

$$\begin{matrix} R2020 \\ G2020 \\ B2020 \end{matrix} = \begin{bmatrix} 0.6274 & 0.3293 & 0.0433 \\ 0.0691 & 0.9195 & 0.0114 \\ 0.0164 & 0.0880 & 0.8956 \end{bmatrix} * \begin{matrix} R709 \\ G709 \\ B709 \end{matrix} + \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}$$

So the array could be written as $array = [0.6274, 0.3293, 0.0433, 0.0691, 0.9195, 0.0114, 0.0164, 0.0880, 0.8956, 0, 0, 0]$

Solution for issue ④ （Tone mapping)

- CurveSetElement(ISO 20677:2019 11.2.2)

The Curve Set element encodes multiple one-dimensional curves. The encoding is shown in Table 107. The sampledClaculatorCurve curve type allows for the encoding of 1-dimensional curves as an arbitrary function using an embedded element of type calculatorElement. This allows for more optimal implementation as the embedded calculator element can be applied at transform setup to populate a sampled 1-dimensional lookup table that is then used for performing curve transforms. The encoding of a sampledClaculatorCurve is shown in Table 109.

Table 107 — curveSetElement encoding

Byte position	Field length (bytes)	Content	Encoded as...
0...3	4	'cvst' (63767374h) type signature	
4...7	4	Reserved, shall be 0	
8...9	2	Number of input channels (P)	uInt16Number
10...11	2	Number of output channels (Q)	uInt16Number
12..11+8P	8P	Curve positions (offset and size)	positionNumber[...]
12+8P to end		Data	

Table 109 – sampledCalculatorCurve encoding

Byte Position	Field Length (bytes)	Content	Encoded as...
0..3	4	'clcf' (636c6366h) type signature	
4..7	4	Reserved, must be set to 0	
8..12	4	Recommended number of interpolation samples for the curve (R)	uInt32Number
12..15	4	Starting input range value of curve (S)	float32Number
16..19	4	Ending input range value of curve (E)	float32Number
20.21	2	Lookup Extension type (L)	uInt16Number
22.23	2	Reserved, must be set to 0	
16+20..end		Curve Definition	calculatorElement

In our case, take **Reinhard tone mapping** as an example, the input P and output channels are all set to **3,** which means that we have three one-dimensional curves(R,G,B separately). All the three curves apply Reinhard tone mapping function defined by embedded **calculator element** as:

$$L_{TM} = \frac{L * (1 + \frac{L}{L_{white}^2})}{1+L}$$

The curves could be defined between **0(S) and 1(E)** , with Lookup Extension type(L) set to 1 for avoiding clipping.

Draft Modifications for Architecture 1 Option 1 in ISO 21496-1 (Preferred)

1. 3.2 baseline image
data structure that contains pixels and image-related data, and is stored in the file as the main image with ICC profile
The baseline image data in RGB or XYZ is then derived using the ICC profile.
Note: ICC profile contains the color conversion and tone conversion information.

2. 3.3 gain map
data structure that contains local quotients used for conversion between two image representations with or without ICC profile
Note: ICC profile contains the color conversion information, The gain map data in RGB or XYZ is derived using the ICC profile. If no ICC profile exist, the gain map use default sRGB for the potential color conversion.

The relationship between ISO 20677:2019 and ISO 21496-1

Architecture 1: Refer ISO 20677:2019 in ISO 21496-1

Architecture 2: Integrate ISO 21496-1 into ISO 20677:2019

- add new mechanism into AtoBx or some other tag with coordinate, store gain map as a lut or somewhere, using coordinate to seek for the value.
- add new mechanism into AtoBx with an opening design with only label and offset to gain map data to allow citing other image conversion standard.
- Extend the abstract profile or adding a special profile with an opening design.

- Need massive amendment on ISO 20677:2019 , how to and when can support current structure of ISO 21496-1(two images, baseline image and gain map) in ISO 20677:2019 is not clear.

Architecture 2: Integrate ISO 21496-1 into ISO 20677:2019

The following methods can be used separately or together

- add new coordinate mechanism into AtoBx or some other tag or tag type, store gain map as a lut or somewhere, using coordinate to seek for the value.
  - **Examples of things that cannot be done by a profile:**
    - — Conditional rendering intent selection
    - — Spatial operations
    - — Variable transforms
    - — Channel preservation

  Cite:https://www.color.org/info_profiles2.xalter
  Fundamentals_of_ICC_Color_Management

- add new mechanism into AtoBx or some other tag with an opening design with only label and offset to gain map data to allow citing other image conversion standard.

➢ Extend the Tag type, with one additional tag Type with label 'gain' that can be registered by other standard.
➢ The input and output colour space should refer to ISO 21496-1.

- Extend the abstract profile or adding a special profile with an opening design.

➢ Extend abstract profile with one additional tag with label 'extn' that can be registered by other standard. With tag Type of 'gain' as using ISO 21496-1.
➢ Extend the Tag type, with one additional tag Type with label 'gain' that can be registered by ISO 21496-1.
➢ Some constraints should be added in ISO 21496-1 on the usage of PCS to PCS.
➢ PCS range conversions tag should be set accordingly
➢ Some flag in profile header should be added to indicate the usage of abstract profile.

Use of alternate HDR to SDR conversions can be complished using an abstract profile containing a passthrough (identity) colorimetric transform, an HDR to SDR tag transform, and the use of extended-range PCS is flagged in the abstract profile header. The abstract profile's HDR to SDR transform will be applied when it is placed between an extended-range PCS source profile and a non-extended range PCS profile – thus allowing overrides of HDR to SDR transforms to be performed.

Cite: ICCMax Spec

8.9 Abstract profile

In addition to the tags listed in 8.2 an abstract profile shall contain one or more of the following tags: AToB0Tag (see 9.2.1), DToB0Tag (see 9.2.76).

This profile represents an abstract transform and does not represent any device model. Colour transformations using abstract profiles are performed from PCS to PCS. Abstract profiles cannot be embedded in images.

Draft Modifications for Architecture 2

- In ISO 21496-1
  - Adding Annex E
  - The tag type of gain map application is ‘gain’.
  - (optional) Add all the gainmap metadata into the structure of this type.
  - Set the Added flag in the profile header to notify that the gain map application should be used, for example flag is 1 means using Abstract Profile with ‘extn’ tag who have the type of ‘gain’.

- In amendment of ISO 20677:2019
  - Adding The tag type of ‘gain’, the structure of this type is the same as in ISO 21496-1 Annex E, the procedure is as in ISO 21496-1.
  - Adding The additional tag with label ‘extn’, with one of the tag type as ‘gain’.
  - Adding The additional tag to Abstract Profile.
  - Modified the constrain on Abstract Profile to allow embedded in image.
  - Adding some flag in the profile header to notify that the Abstract Profile should be used, for example flag is 1 means using Abstract Profile with ‘extn’ tag who have the type of ‘gain’.

Need massive amendment on ICC, how and when to support current structure of ISO 21496-1(two images, baseline image and gain map) in ISO 20677:2019 is not clear.

Summary

Architecture 1: Refer ISO 20677:2019 in ISO 21496-1

- **Option 1: combination is done in working color space according to the ICC profile in baseline image (compulsory) and gain map (optional)**

Thank You!