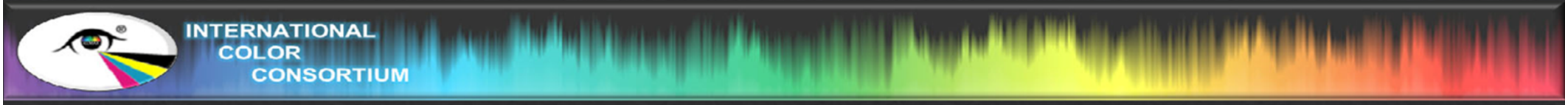


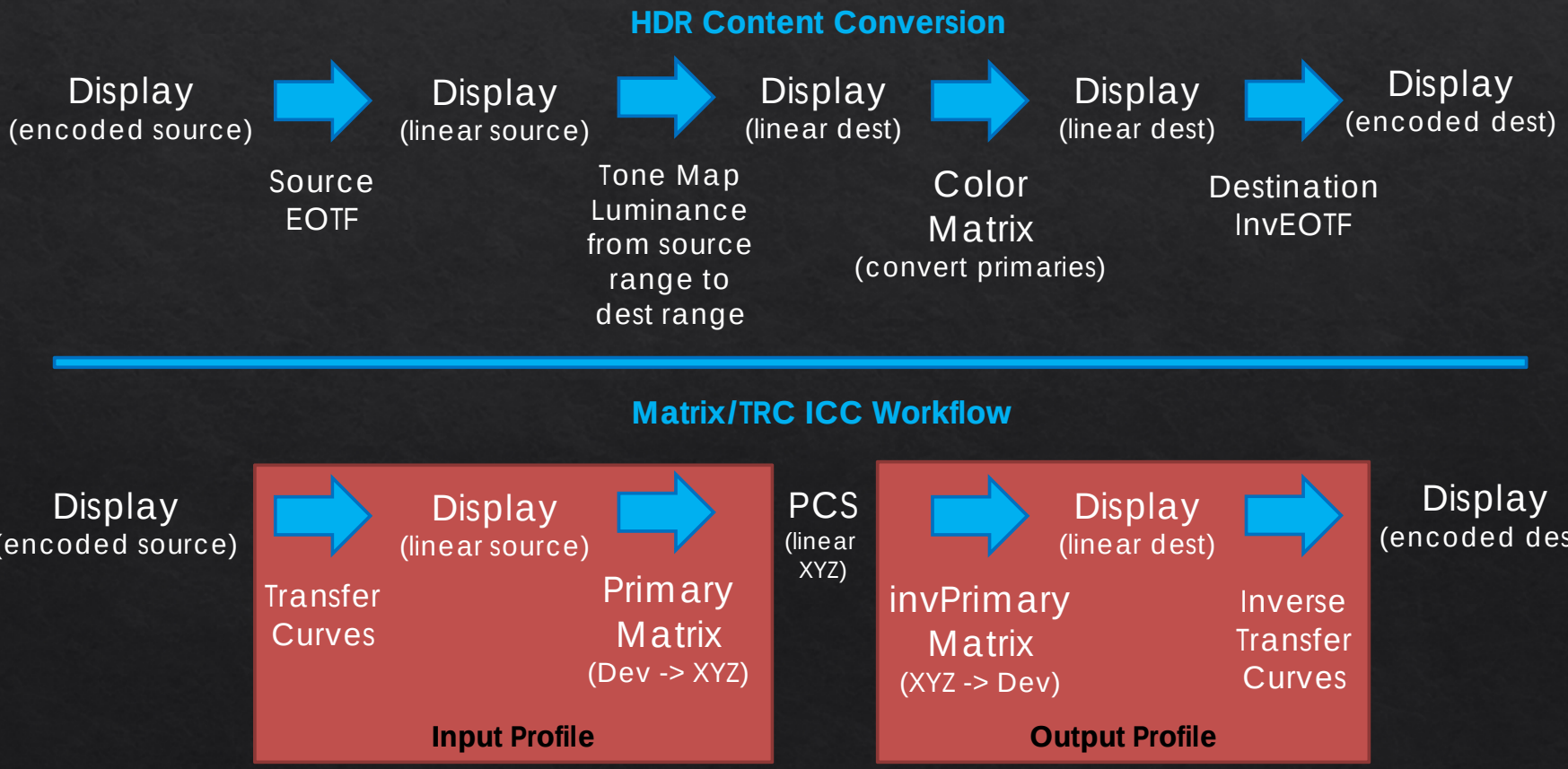
Thoughts on HDR Colour Management Using iccMAX (v5) Profiles

HDR Experts' Day

March 15, 2022

Comparing Workflows

- Previously, Luke Wallace made a presentation comparing differences and similarities of global HDR content conversions with ICC.1 workflows



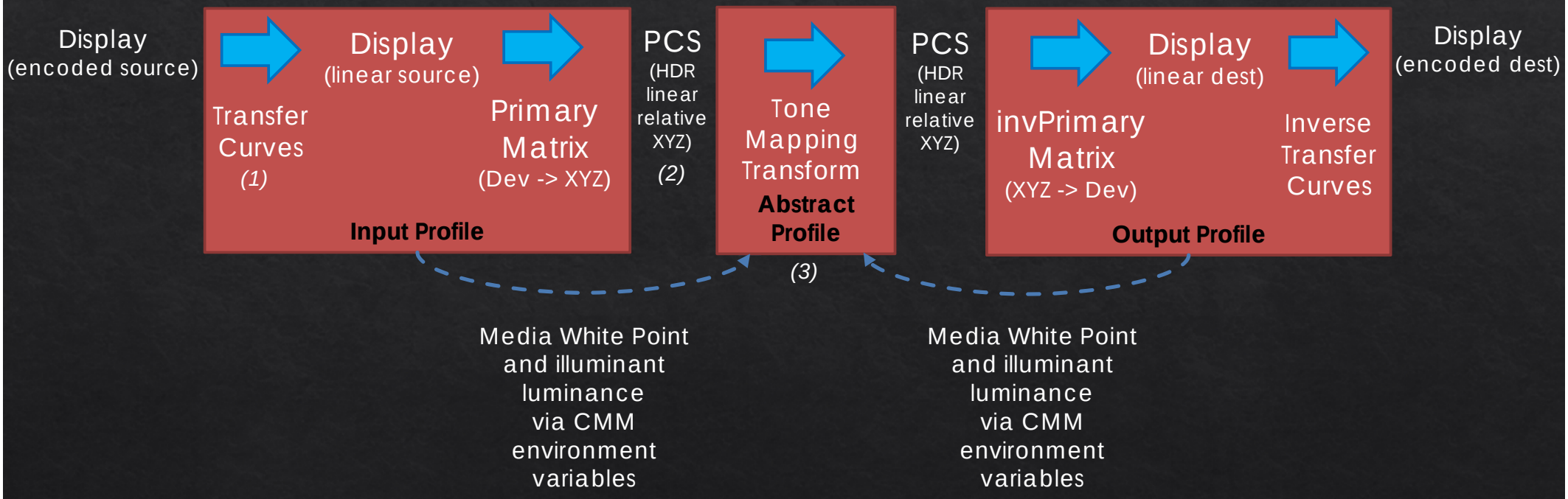
Luke's Conclusions

- Similarities:
  - Transfer functions = Parametric Curves
  - Matrices convert primaries (concatenation of Dev->XYZ & XYZ->Dev)
- Differences:
  - HDR parametric transfer functions (IE for HLG and PQ) do not exist in ICC.1 spec
  - Parametric 3D tone mapping functions do not exist in ICC.1 spec
  - Neither can these be approximated by 1-D or 3-D LUTs
  - Overall conversion depends on both source and destination
- One Additional Observation:
  - ICC.1 has a limited luminance range due to PCS encoding and this in effect restricts data to output-referred

HDR Opportunities with iccMAX

- iccMAX transforms are defined as a sequence of 32-bit floating-point processing elements using:
  - Parametric/Segmented curves/1-D lookup curves
  - Matrices
  - N-Dimensional lookup tables
  - Calculator based elements
    - Use script to define transform functions
    - Viewing environment and white point/luminance information from neighboring profiles can be provided to script as CMM environment variables
    - Calculator Curves allow creation at startup of 1-D LUTs for fast processing
- iccMAX allows for full floating-point range PCS using D65 illuminant and standard 2° observer (commonly used by displays)

Proposed General iccMAX-based HDR Workflow



Note: Tone mapping transform may also be part of either the input or output profile

iccMAX HDR Transfer Curve Options

- Use Sampled Curve
- Use Parametric Curve
  - Possibly add HDR parametric curve types for HLG and PQ (possibly defined in v5.1)
- Use Calculator Element
  - Can be used to define entire transform
  - Possibly bad idea for fast speed transforms
- Use Calculator Curve (v5.1)
  - Calculator script used to establish 1-D LUT used for fast pixel processing (by GPU)
    - Allows for arbitrary 1-D functions to be defined using CMM environment variables to control behavior

HDR linear relative XYZ PCS

- XYZ PCS normalized relative to the illuminant in the profile header (Graphics White)
  - For HDR the illuminant doesn't necessarily represent the maximum device encoding
- The Media White Point (MWP) represents the ratio of the peak white (maximum device encoding) relative to the illuminant white
  - Note: For HDR Y_{MWP} is greater than 1.0
- The illuminant's absolute luminance ($\underline{Y_{illum}}$) is specified in a profile's Spectral Viewing Conditions Tag
- Radiance of profile PCS values is determined by the formula:

$$Y_{rad} = Y_{PCS} Y_{illum}$$

- The peak radiance of a profile is determined by the formula:

$$Y_{peak} = Y_{MWP} Y_{illum}$$

Luminance Tone Mapping using an Abstract Profile

- A basic tone mapping using a luminance-based scaling algorithm can be encoded as processing elements as follows:

$$\mathbf{c}_{tone} = f\left(\mathbf{XYZ}_{in}\right)$$

$$L_c = \text{luminance}\left(\mathbf{XYZ}_{in}\right)$$

$$\mathbf{c}_{map} = \text{tonemap}\left(L_c, Y_{mwp\_src}, Y_{illum\_src}, Y_{mwp\_dst}, Y_{illum\_dst}\right)\mathbf{c}_{tone}$$

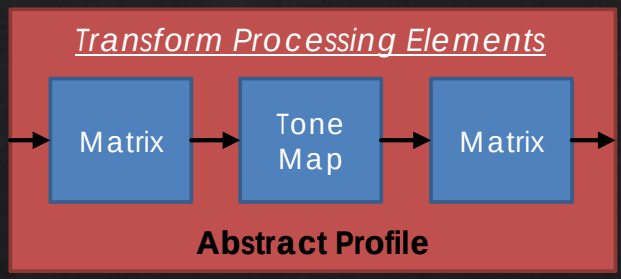
$$\mathbf{XYZ}_{out} = f^{-1}\left(\mathbf{c}_{map}\right)$$

Using a matrix element to convert XYZ PCS to relative Tone Space and Luminance value for tone mapping

Using a Tone Mapping element (v5.1) containing a calculator curve to define tone map 1-D LUT that provides scalar applied to color channels

Using a matrix element to convert to XYZ PCS

- The tonemap() function is defined as part of a calculator script allowing for pluggable tone mapping functions
- The Y_{mwp_src}, Y_{illum_src}, Y_{mwp_dst}, Y_{illum_dst} values represent Media White Points and illuminant luminance values of previous (src) and next (dst) profiles
  - These are accessed by the tonemap's calculator script as CMM environment variables



"Filmic" Tone Mapping using an Abstract Profile

- A basic filmic tone mapping algorithm can be encoded as processing elements as follows:

$$\mathbf{c}_{tone} = f\left(\mathbf{XYZ}_{in}\right)$$

$$\mathbf{c}_{map} = \begin{bmatrix} \text{calccurve}_1\left(c_{tone,1}, Y_{mwp\_src}, Y_{illum\_src}, Y_{mwp\_dst}, Y_{illum\_dst}\right) \\ \text{calccurve}_2\left(c_{tone,2}, Y_{mwp\_src}, Y_{illum\_src}, Y_{mwp\_dst}, Y_{illum\_dst}\right) \\ \text{calccurve}_3\left(c_{tone,3}, Y_{mwp\_src}, Y_{illum\_src}, Y_{mwp\_dst}, Y_{illum\_dst}\right) \end{bmatrix}$$

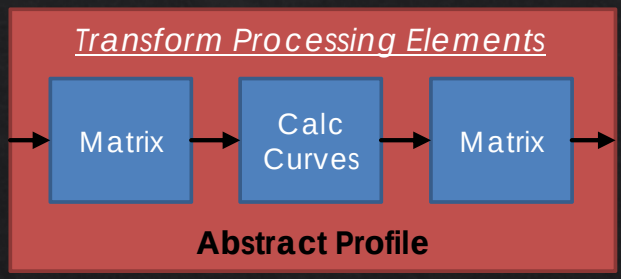
$$\mathbf{XYZ}_{out} = f^{-1}\left(\mathbf{c}_{map}\right)$$

Using a matrix element to convert XYZ PCS to space for tone mapping

Using calculator curves (v5.1) to define a tone map resulting in a 1-D LUT for each $\mathbf{c}_{tone}$ color channel

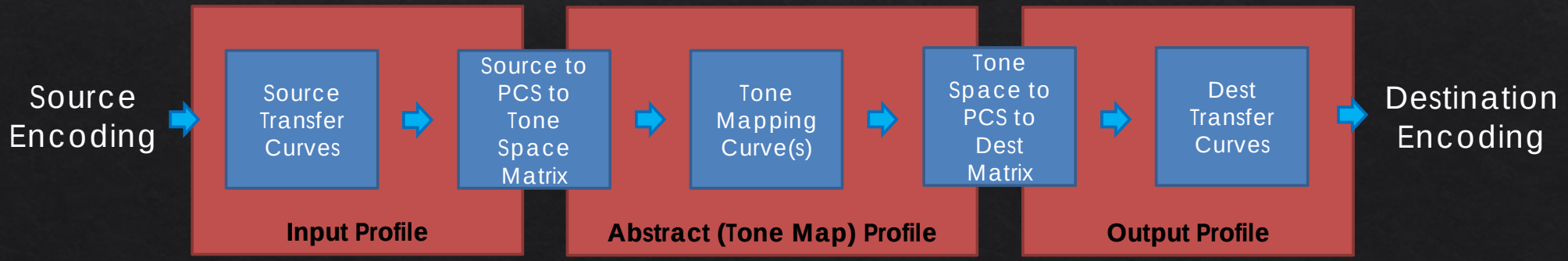
Using a matrix element to convert to XYZ PCS

- The calccurve$_x$() functions each contain a calculator script defining a pluggable tone mapping function
- The Y_{mwp_src}, Y_{illum_src}, Y_{mwp_dst}, Y_{illum_dst} values represent Media White Points and illuminant luminance values of previous (src) and next (dst) profiles
  - These are accessed by the calccurve calculator script as CMM environment variables



Resulting HDR Transform

- Combination of processing elements (IE matrix concatenation performed by CMM) reduces overhead of multiple processing elements provided in separate Input, Abstract (Tone Mapping) and Output profiles
- 1-D LUT generation at transformation startup allows complex calculations to be used to define transfer and tone map curves without extra processing overhead when applying to pixels



Still to be done (in V5.1)

- Define Parametric curve additions for HLG and PQ
- Define CMM environment variables for Media White Point and illuminant luminance of previous and next profiles that can be use in tone mapping calculator scripts
- Define Tone Map element using calculator script
- Implement various tone mapping approaches in actual profiles to ensure that capabilities / performance match expectations

Conclusions

- Challenges with implementing HDR content conversions using ICC.1 can be overcome using iccMAX (v5)
- Pluggable global HDR Tone Mapping can be performed utilizing an abstract profile to define tone mapping transforms
- New processing elements in iccMAX (v5.1) should allow for efficient processing of HDR image data (in a GPU) while retaining programmable flexibility

Thank You!