Using the X-Rite CMM

x-rite

Using the X-Rite CMM

- Will show how to use the X-Rite CMM for a typical use case
- Will summarize the correct usage of rendering intents
- Will demonstrate how to use the X-Rite Printing Workflow target to verify the correct functionality of an application

The X-Rite CMM

- Cross-platform
- Intelligent CMYK to CMYK conversion that preserves the black channel
- Black-point compensation

The Use Case

- Given a RAW Camera image
  - Combine image with vector objects
  - View combined image on a LCD Display
  - Soft-proof combined image on a LCD Display
  - Convert combined image to CMYK
  - Proof the image on a desktop printer

The Demo App

- Will demonstrate a simple application written for Mac OSX that uses the X-Rite CMM for color management
  - Non-CMM related functionality will be handled by the OS or third party libraries

Quick Demo

First, Rendering Intents

- Need to understand what rendering intents do and how to use them
  - Please see ICC white papers for more info.

What is a rendering intent?

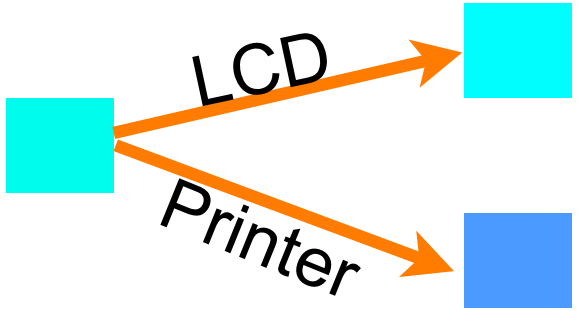
- The ICC profile is essentially a set of transforms that specify how to get from one color-space to another color-space
- These transforms can be performed in many different ways
- Each rendering intent specifies a goal to be achieved in the transform
- This discussion is based on the v4 definition of rendering intents. V2 profiles are problematic due to weaker definitions of rendering intents

The five rendering intents supported by the X-Rite CMM

- Perceptual
- Absolute Colorimetric
  - Transform minimizes colorimetric difference
- Relative Colorimetric
  - Similar to absolute, but maps
- Black-Point Compensation
- Saturation

Perceptual Rendering Intent

- Transforms is designed to provide a pleasing pictorial output
  - "Pleasing" is subjective and will depend on the use case and even individual observers
- Renders from the perceptual PCS to the color-space of the output device
  - The most pleasing transform can be very different for different device technologies

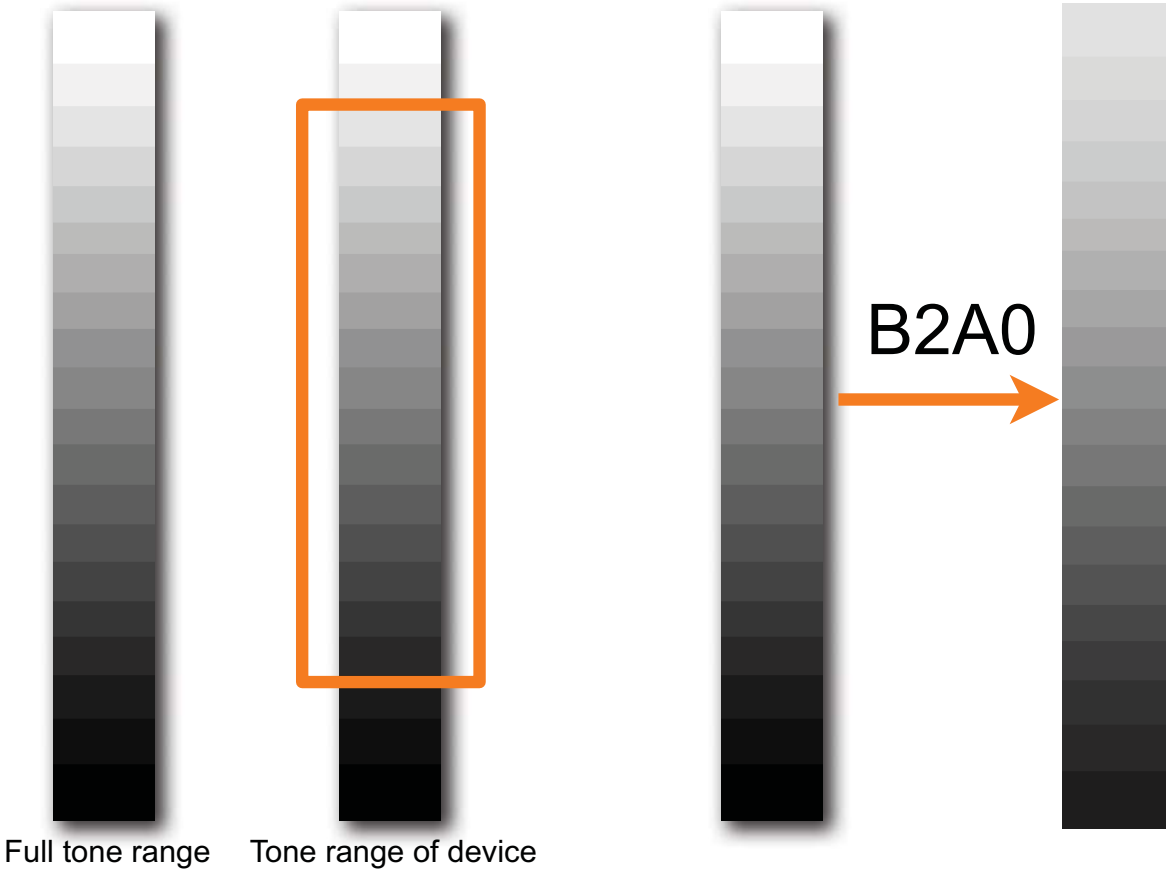


Perceptual (cont.)

- Note that the transform back to PCS is intended to be an inverse of the transform to PCS
  - –An image that is transformed to PCS and then back to its original color-space will not change significantly
  - –Not what you want if you are soft-proofing

Perceptual Tone mapping

- Should map tone range without clipping



Saturation intent

- A color re-rendering that is designed to preserve the vividness of colors
  - –Useful for charts
  - –May significantly alter hue

Screen

Printer

Absolute Rendering Intent

- The transform is designed to minimize colorimetric color differences for in gamut colors
  - The printed color measures the same as the input color
  - Provides the most accurate reproduction if one wants to proof an original
- Out of gamut colors are clipped to the surface in a vendor specific manner
- The transform is derived from the relative intent unless MPE absolute tags are used

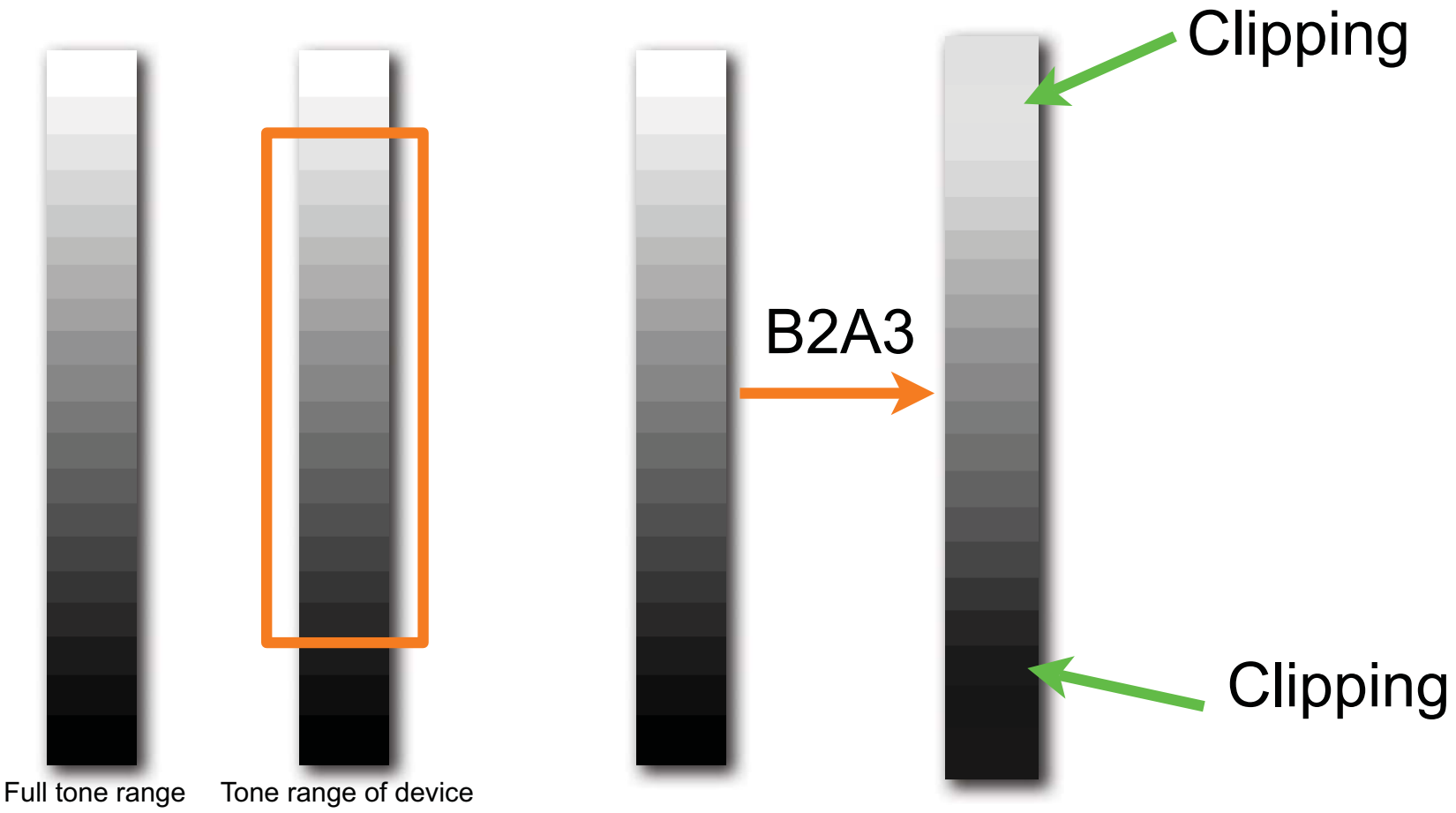
The Absolute Device to PCS transform

- This transform is also intended to be an inverse of the PCS to device transform. But only for in gamut colors.

(Assuming B2A3 & B2A3 were a real tags....)

Absolute Rendering Tone Mapping

- Clips both highlight and shadow



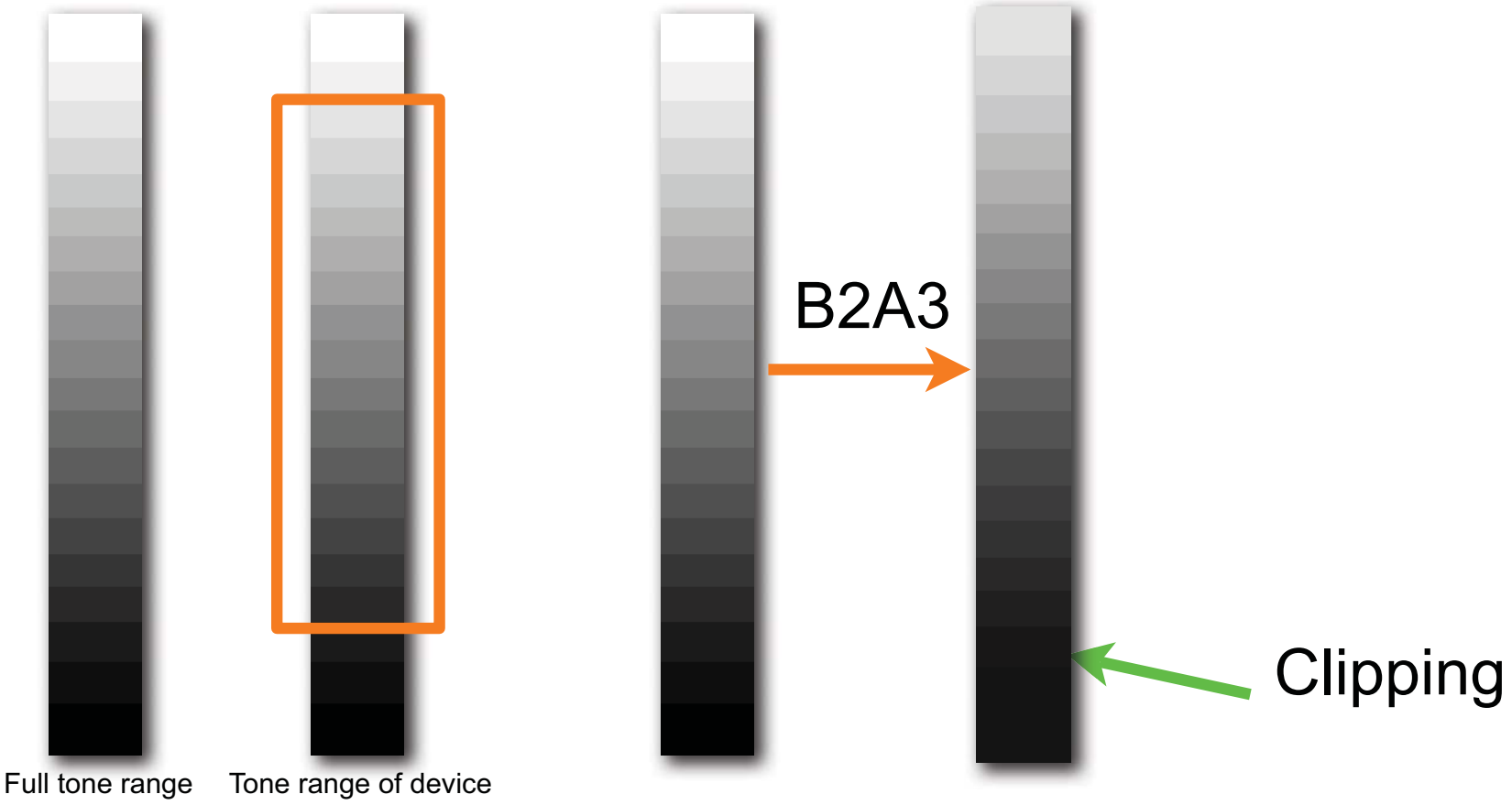
Media-Relative Colorimetric Rendering Intent

- Colorimetric reproduction that is relative to the white point
  - Colors are scaled so that 100% white maps to the paper white

Media-Relative Intent (cont)

- White point of paper is transformed back to 100% white. Colors that were clipped stay clipped

- Clips shadow

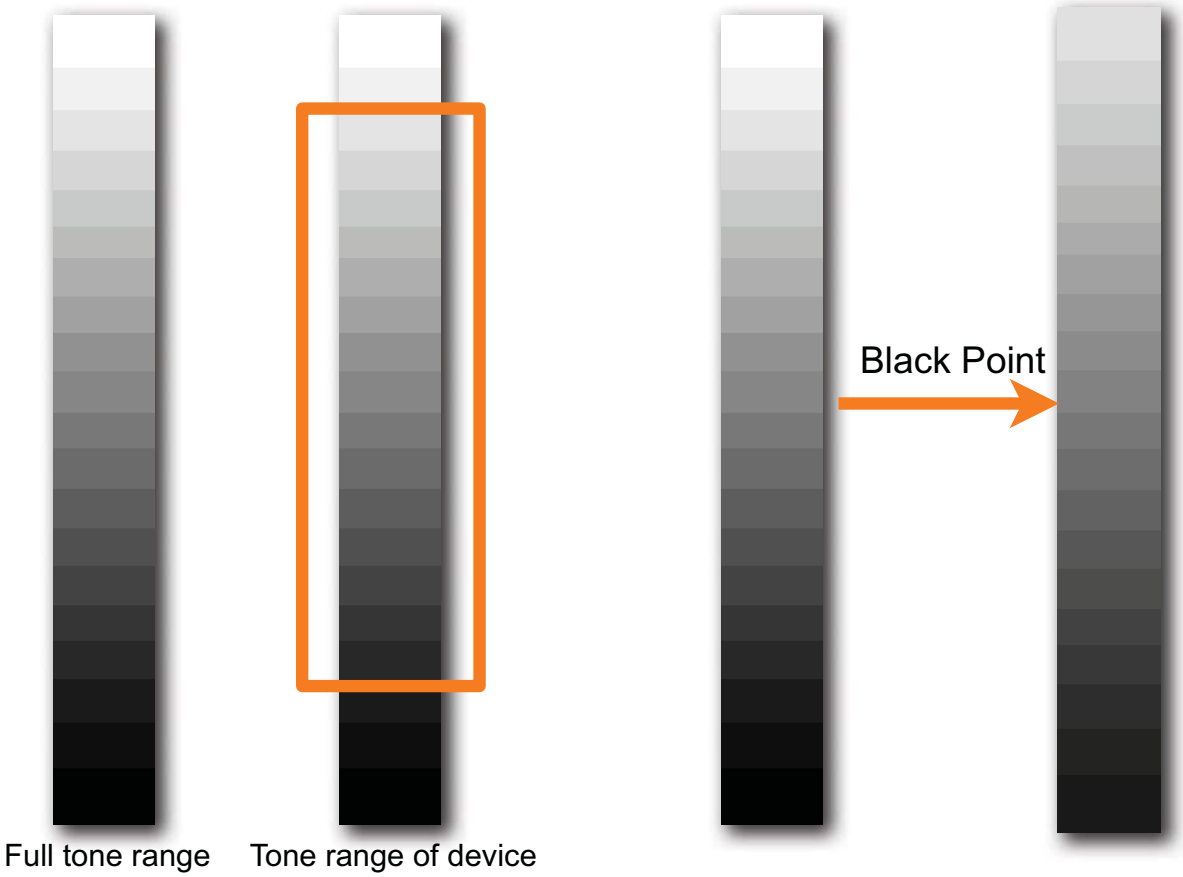


Black-Point compensation

- Similar to Media-Relative, but both white points and black points are mapped.
- Can result in similar results to Perceptual.
- Uses the media-relative colorimetric rendering combined with black-point scaling in the CMS

Black-Point Rendering Intent Tone Mapping

- Does not clip highlight or shadow



Rendering Intent Usage for Our Problem

- How should rendering intents be used for our problem?
- First, what are the transforms used?
  1. RAW image to screen
  2. Vector image to screen
  3. RAW image proof to screen
  4. Vector image proof to screen
  5. RAW image to press
  6. Vector image to press
  7. RAW image proof to printer
  8. Vector image proof to printer

RAW Image to Screen

- Want the image to be pleasing
  - Use perceptual intent to transform from image to PCS
  - For display on screen, want to avoid clipping of the tonal range, but don't want a significant re-rendering: use black-point compensation to go from PCS to RGB

Vector Image to Screen

- Want the colors to maintain vibrancy: use saturation intent to get to PCS
- Could use Perceptual, Black-Point, or Saturation to get to from PCS to RGB

RAW Image to Press & Soft-Proof

- Will use Perceptual to go from camera RGB to PCS
- Want a pleasing image on press, so also use perceptual to go from PCS to CMYK
- Want colorimetric match on screen so use absolute to go from CMYK to PCS as well as PCS to RGB

Vector Image to Press & Soft-Proof

- Will use Saturation to go from RGB to PCS
- Want a vivid image on press, so also use saturation to go from PCS to CMYK
- Want colorimetric match on screen so use absolute to go from CMYK to PCS as well as PCS to RGB

Proof to Printer

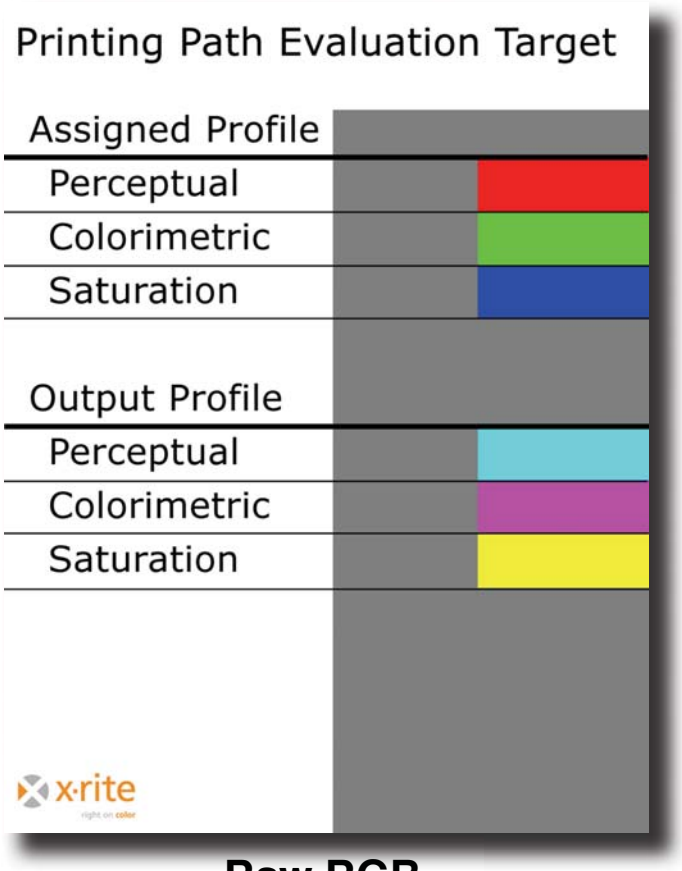
- For both RAW and Vector CMYK, transform using absolute.
  - Will use Smart CMM transform that preserves black

X-Rite Printing Path Evaluation Target

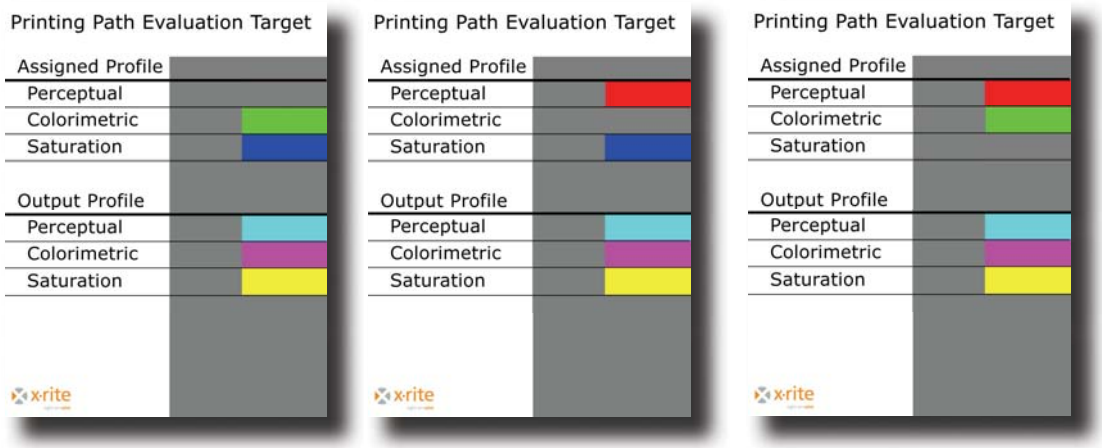
- How do you know that color management is being applied correctly?
- Need a test that will verify correct operation

Printing Path Evaluation Target

- **Image with probe profile imbedded**
- **Used to see if assigned profile is used and what rendering intent is used**



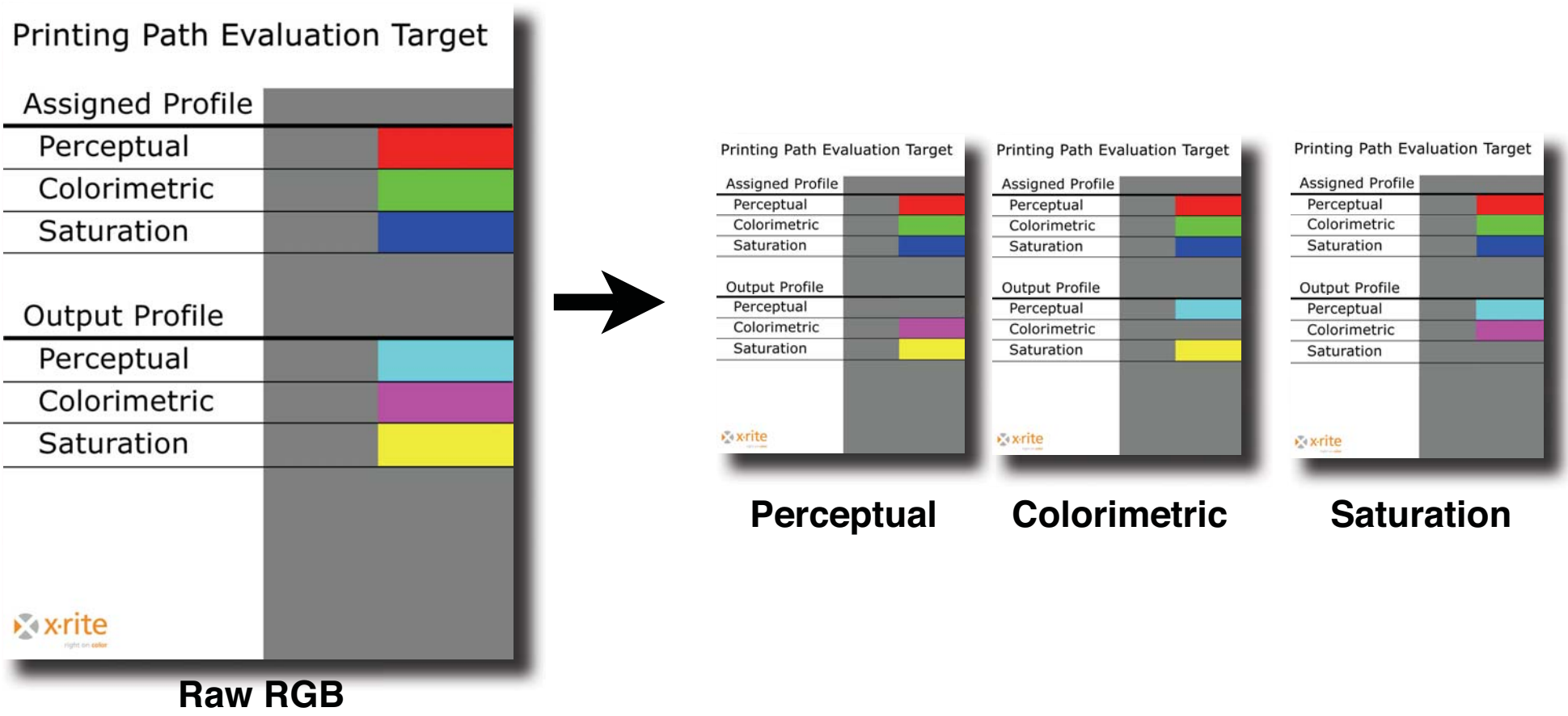
Raw RGB



Perceptual **Colorimetric** **Saturation**

Printing Path Evaluation Target

- **Output profile maps C,M,Y to gray for perceptual, colorimetric, saturation respectively**



Here is the code:

Any questions?

Will look at how CMM is called

- Apply profile to RAW image for display
- Apply profile to vector for display
- Apply profiles to RAW image for soft-proof
- Apply profiles to vector for soft-proof
- Apply profiles for saving image
- Apply profiles for Proof on printer

- (Note for draft version: API not yet finalized)

Apply Profile to Image

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10

Apply Profile to Vector Elements

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10

Apply Profile to Image Elements

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10

Apply Profile to Vector Elements

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10

Save

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10

Print

- 10 i = 0
- 20 print i
- 30 i = i+1
- 40 goto 10