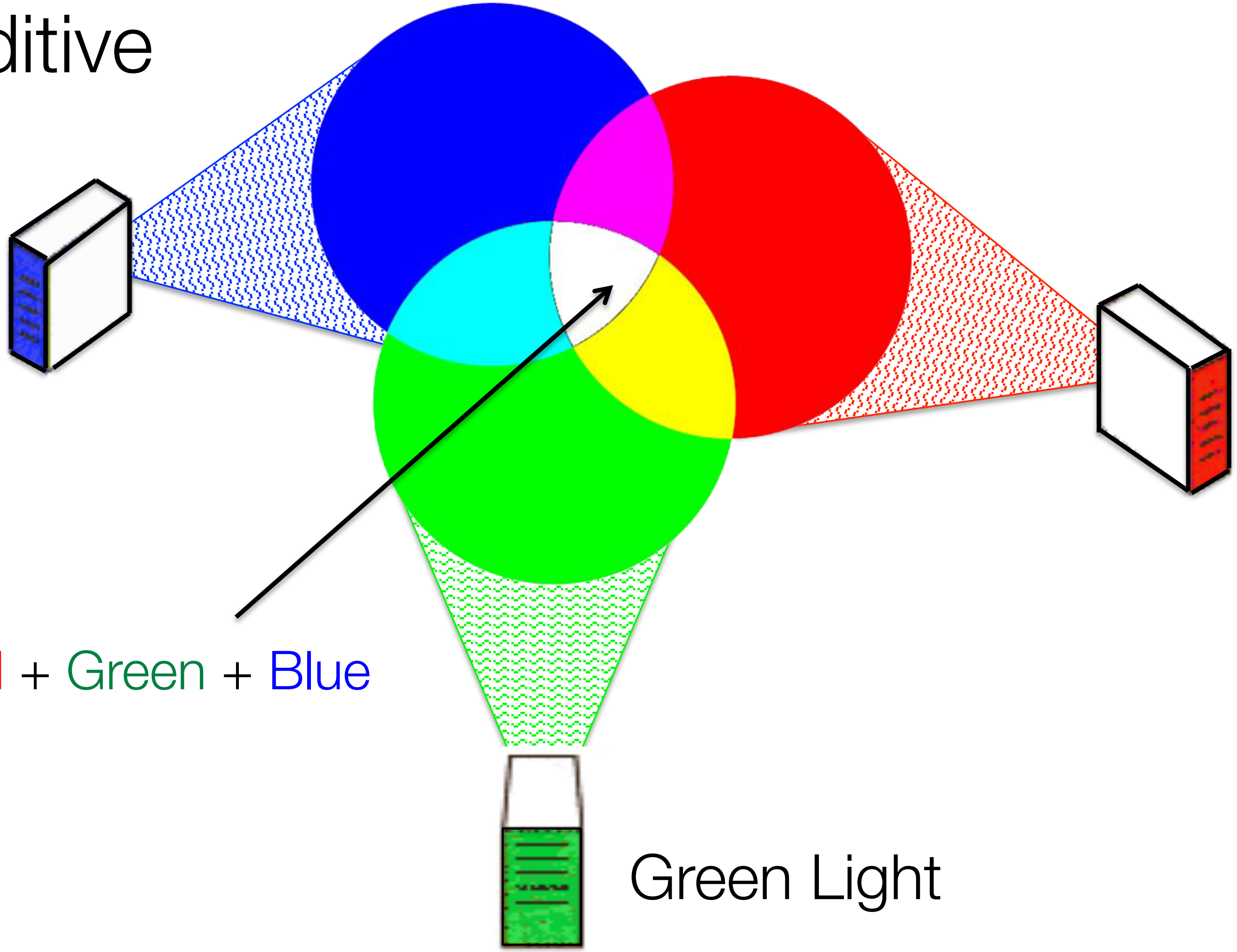
Color Management

Additive Color:

Red + Green + Blue Light Primaries

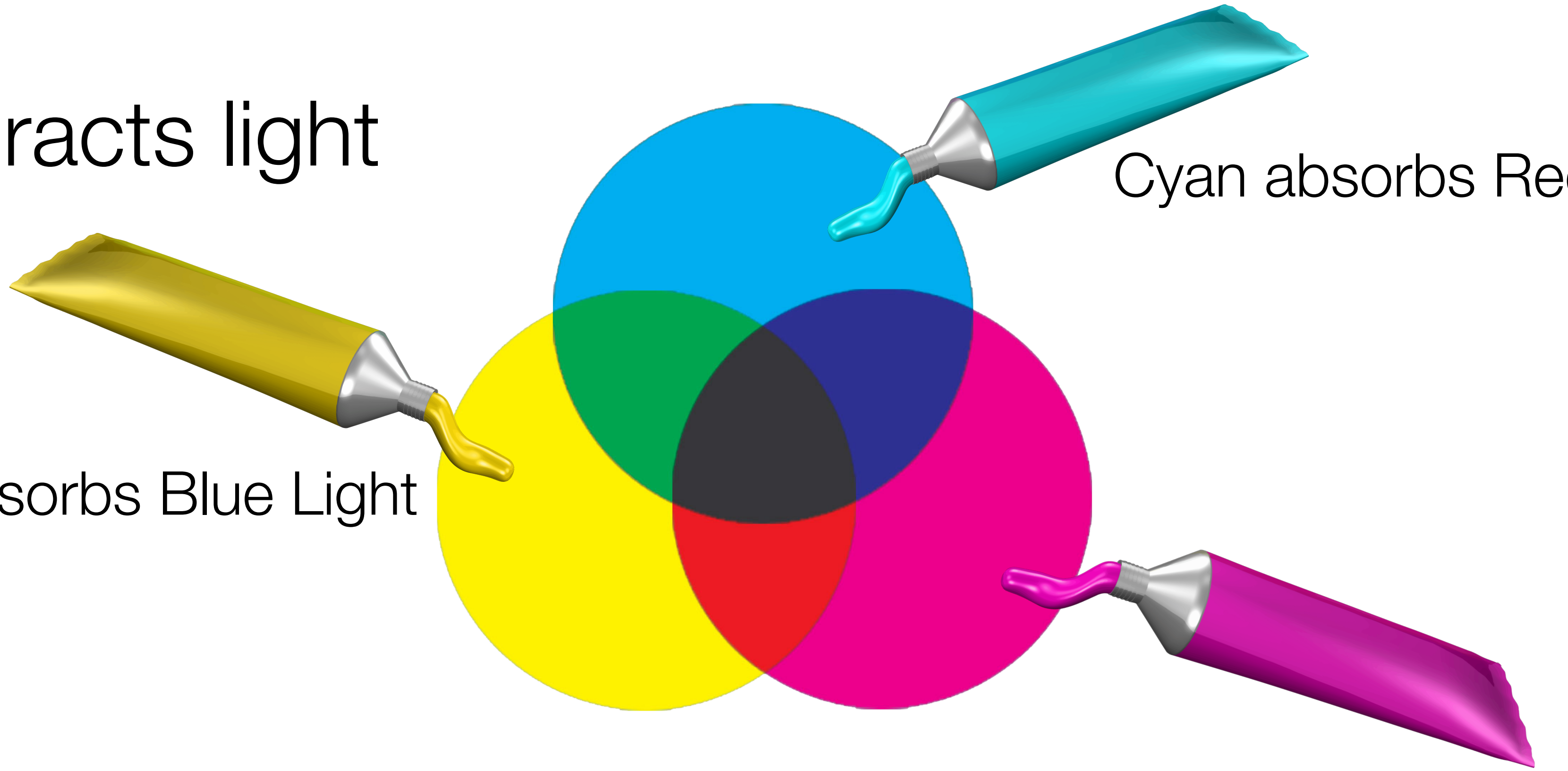
Light is additive



White Light = Red + Green + Blue

Subtractive Color

Ink subtracts light



Cyan absorbs Red Light

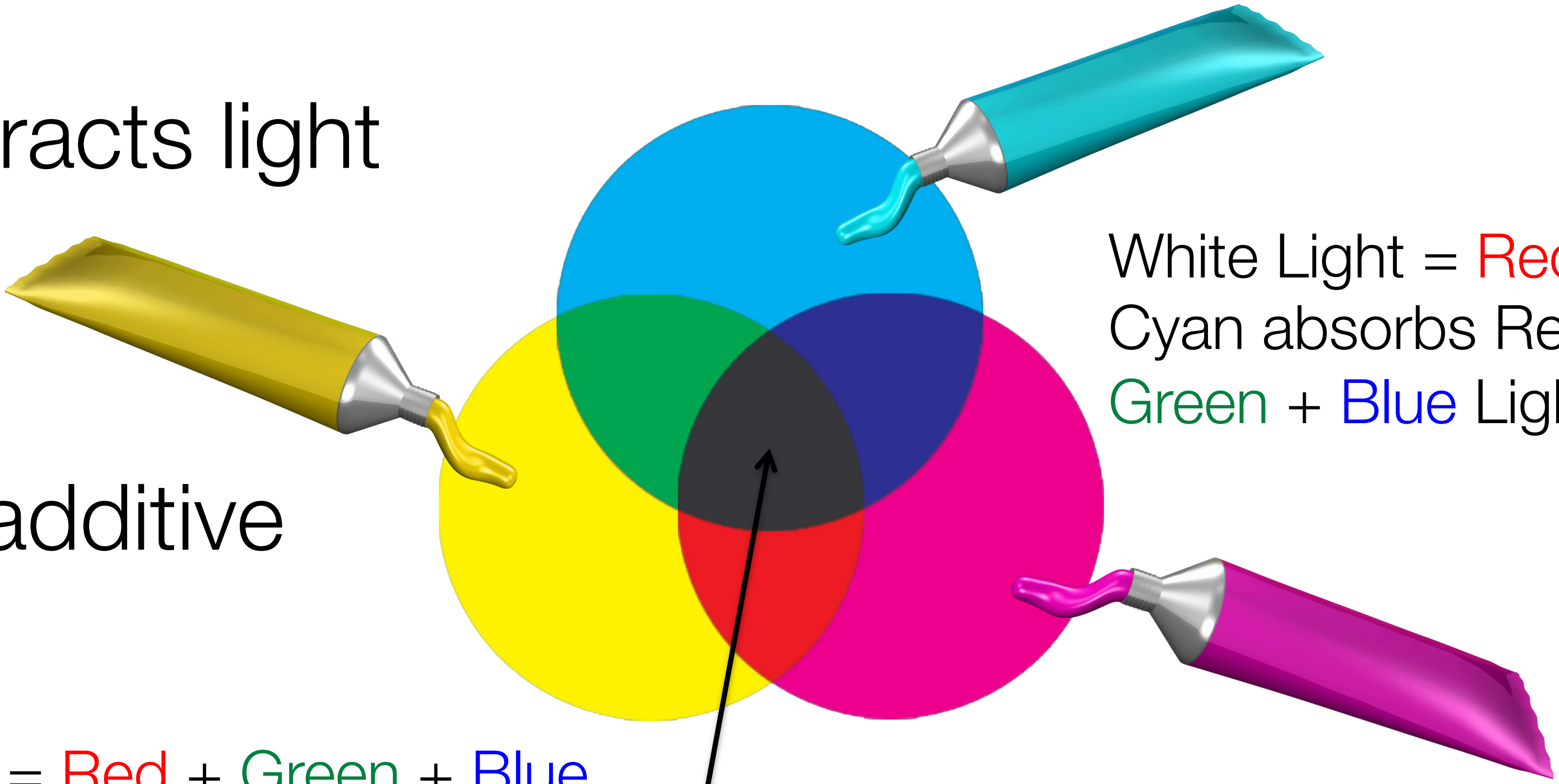
Yellow absorbs Blue Light

Magenta absorbs Green Light

Subtractive Color

Ink subtracts light

Light is additive



White Light = Red + Green + Blue
Cyan absorbs Red Light
Green + Blue Light appears Cyan

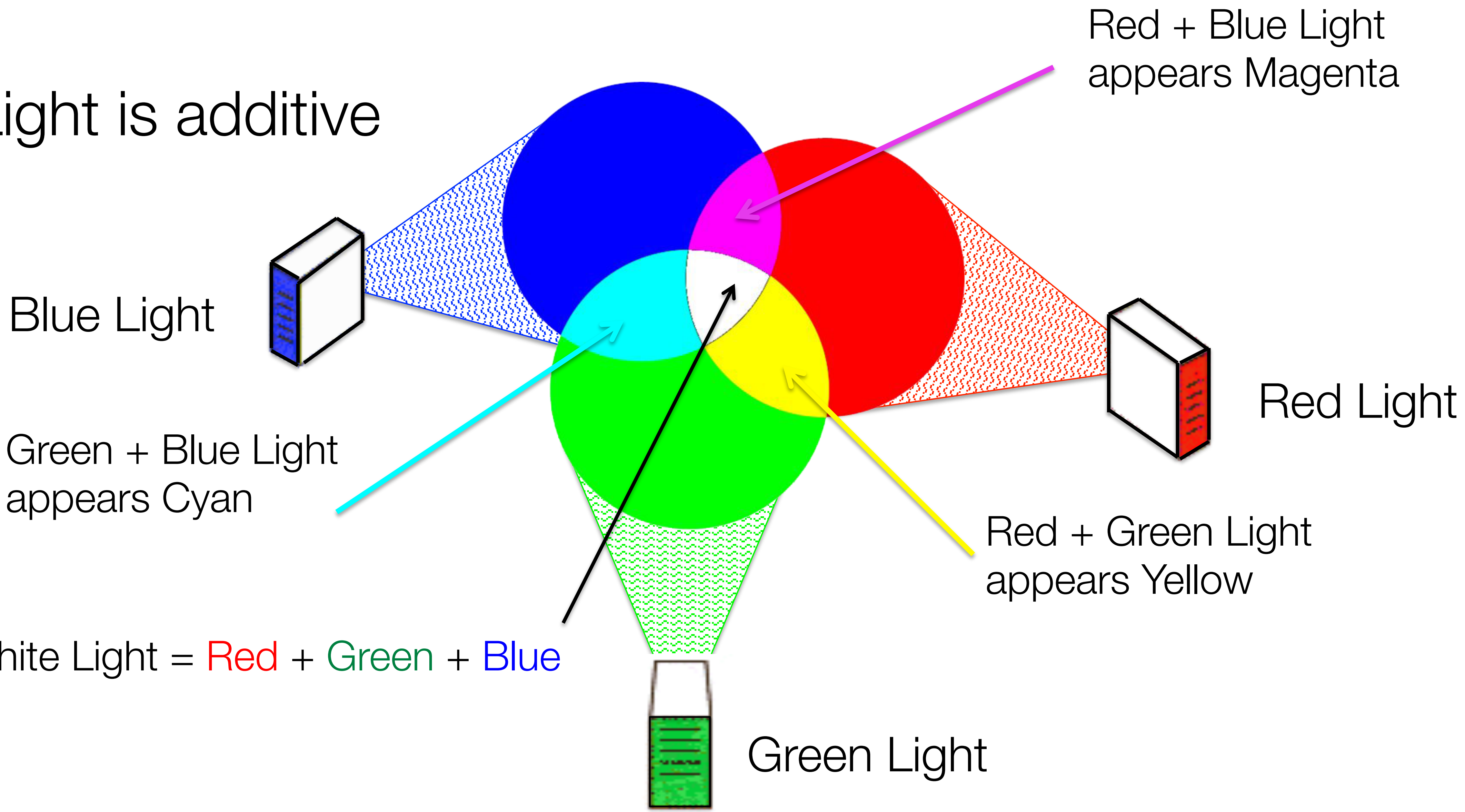
White Light = Red + Green + Blue
Yellow absorbs Blue Light
Red + Green Light appears Yellow

White Light = Red + Green + Blue
Magenta absorbs Green Light
Red + Blue Light appears Magenta

Black = subtract Red + Green + Blue light

Additive Color:

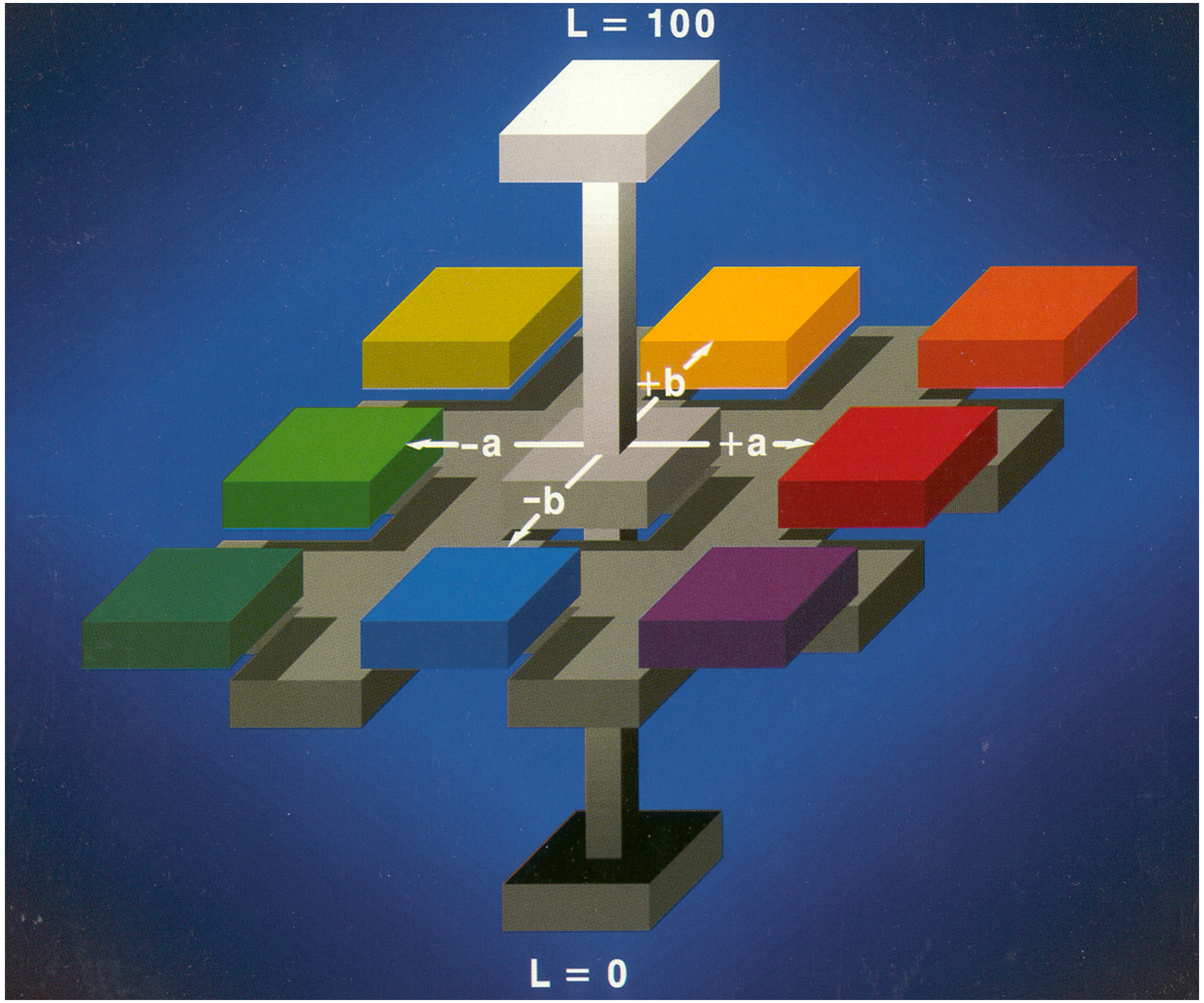
Light is additive



White Light = Red + Green + Blue

CIELAB and ΔE

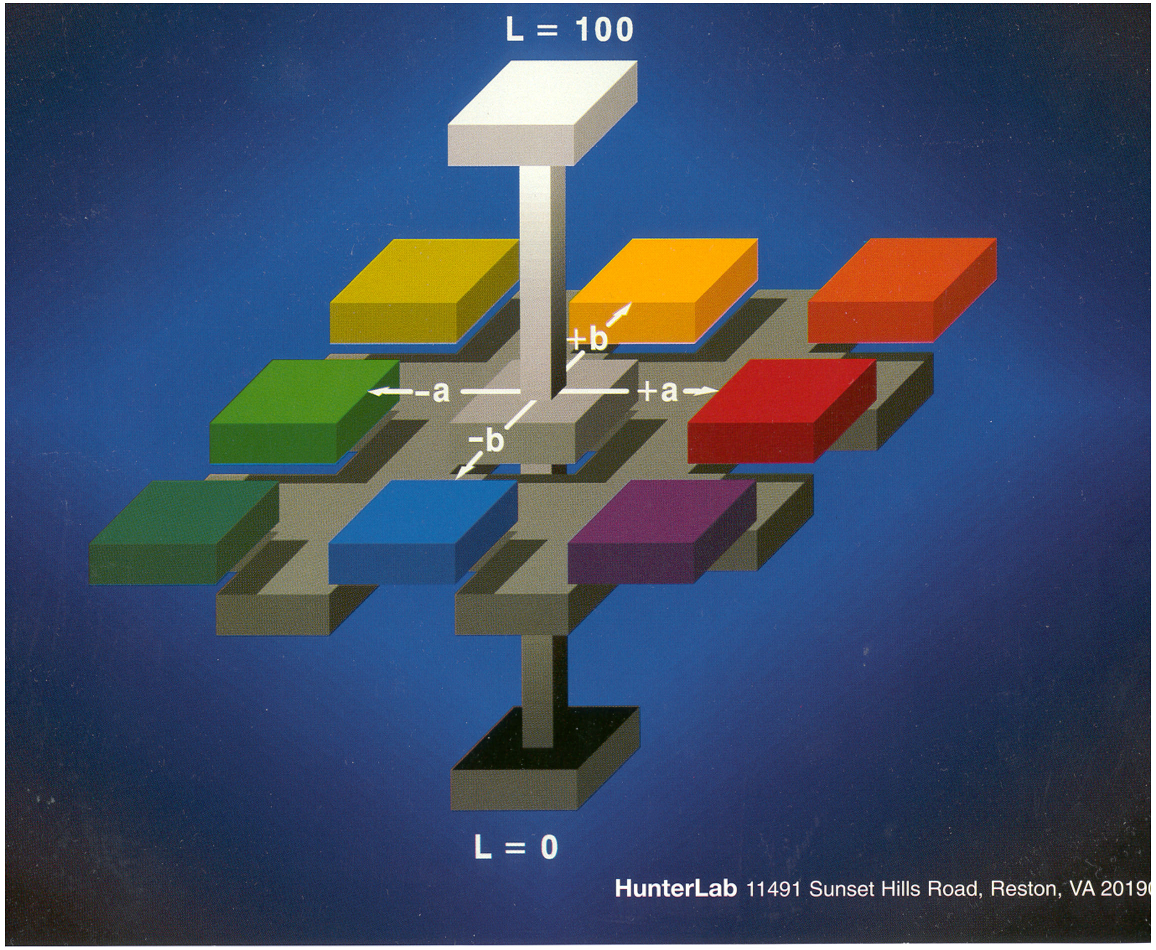
The CIELAB color difference metric is an international standard for a trichromatic observer that is based on the Euclidean distance between two colors in CIELAB space.



CIELAB Color Space

Distance between two colors designed to predict color difference visibility

$$\Delta E_{ab} = \left(\left(\Delta L^*\right)^2 + \left(\Delta a^*\right)^2 + \left(\Delta b^*\right)^2 \right)^{1/2}$$



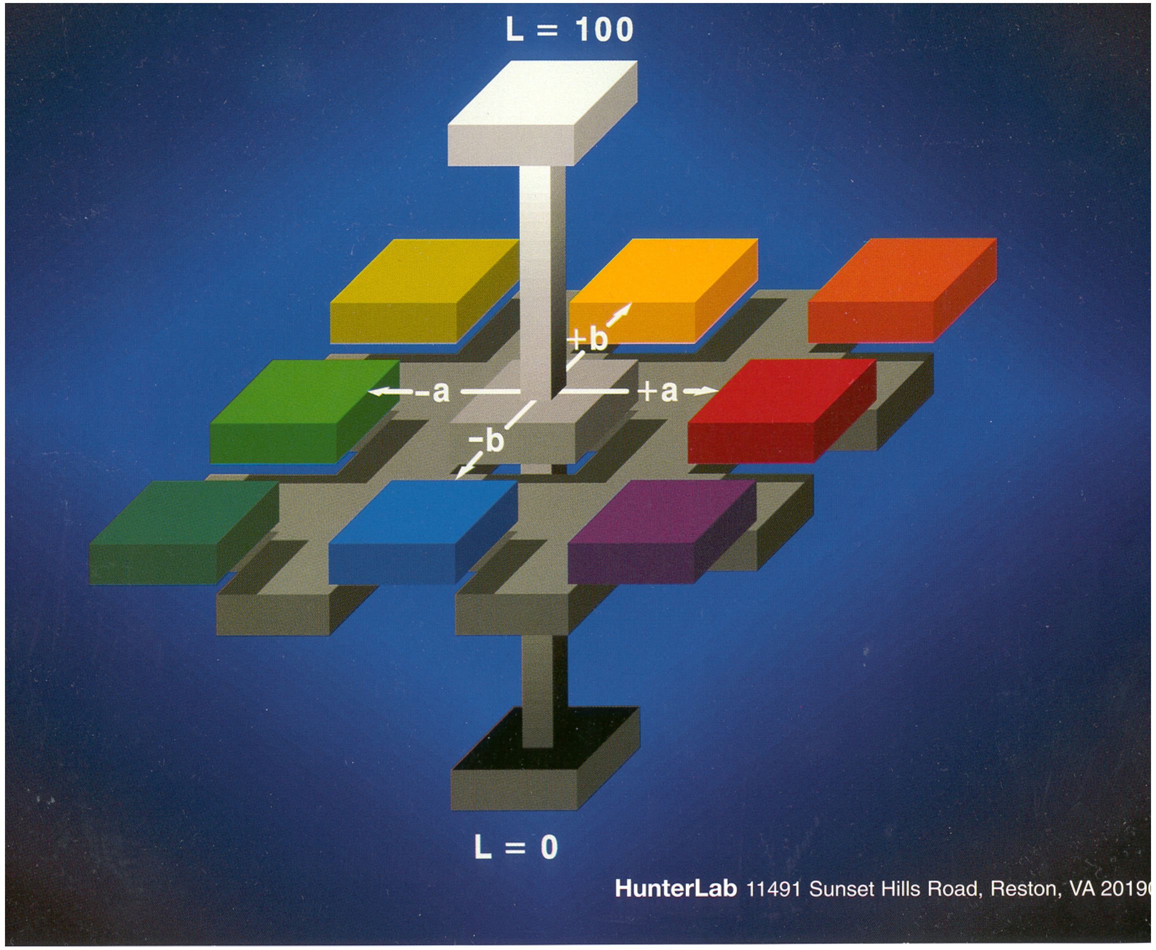
CIELAB Color Space

$$L^* = 116\left(\frac{Y}{Y_w}\right)^{1/3} - 16, \quad if\ \frac{Y}{Y_w} > .00856$$

$$L^* = 903.3\left(\frac{Y}{Y_w}\right), \quad otherwise$$

$$a^* = 500\left\{\left(\frac{X}{X_w}\right)^{1/3} - \left(\frac{Y}{Y_w}\right)^{1/3}\right\}$$ (red-green)

$$b^* = 200\left\{\left(\frac{Y}{Y_w}\right)^{1/3} - \left(\frac{Z}{Z_w}\right)^{1/3}\right\}$$ (blue-yellow)



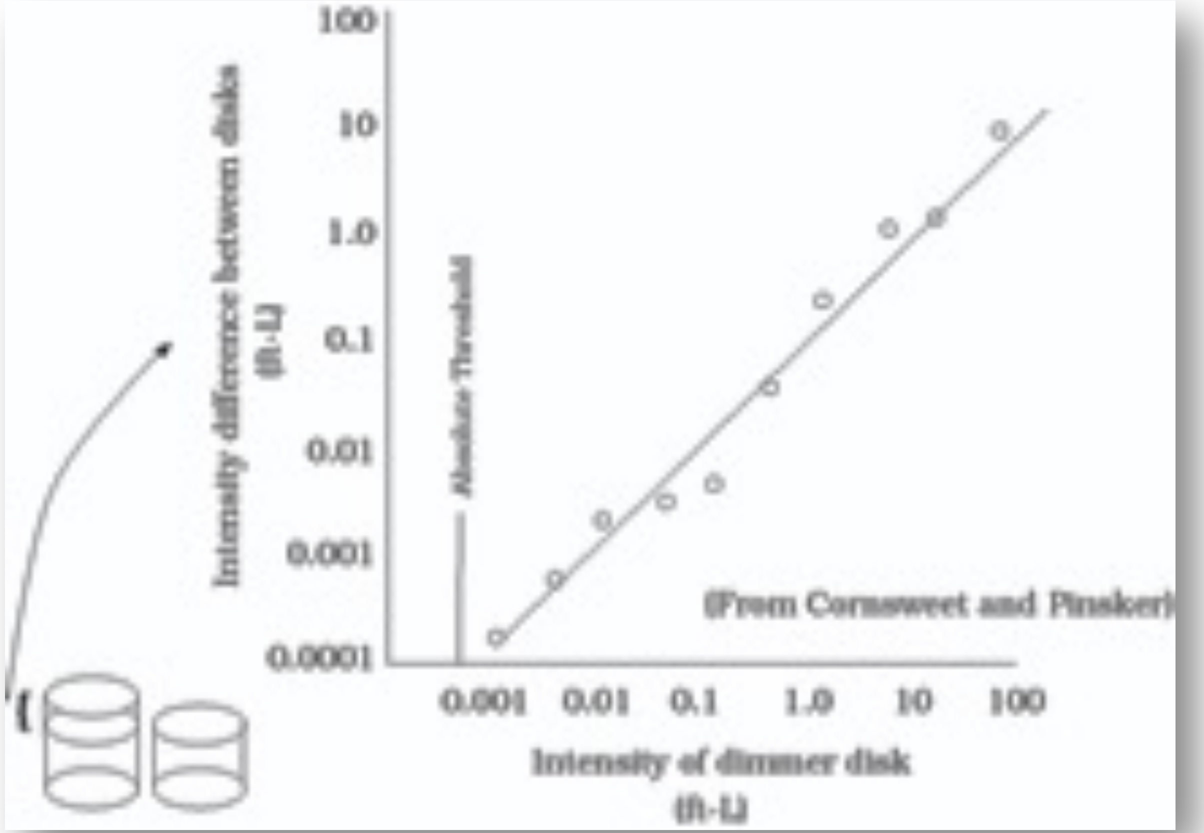
CIE Lightness

Perceived brightness (lightness) is a function of

- Background luminance
- Cube-root of the relative luminance

$$L^* = 116\left(\frac{Y}{Y_w}\right)^{1/3} - 16, \quad if\ \frac{Y}{Y_w} > .00856$$

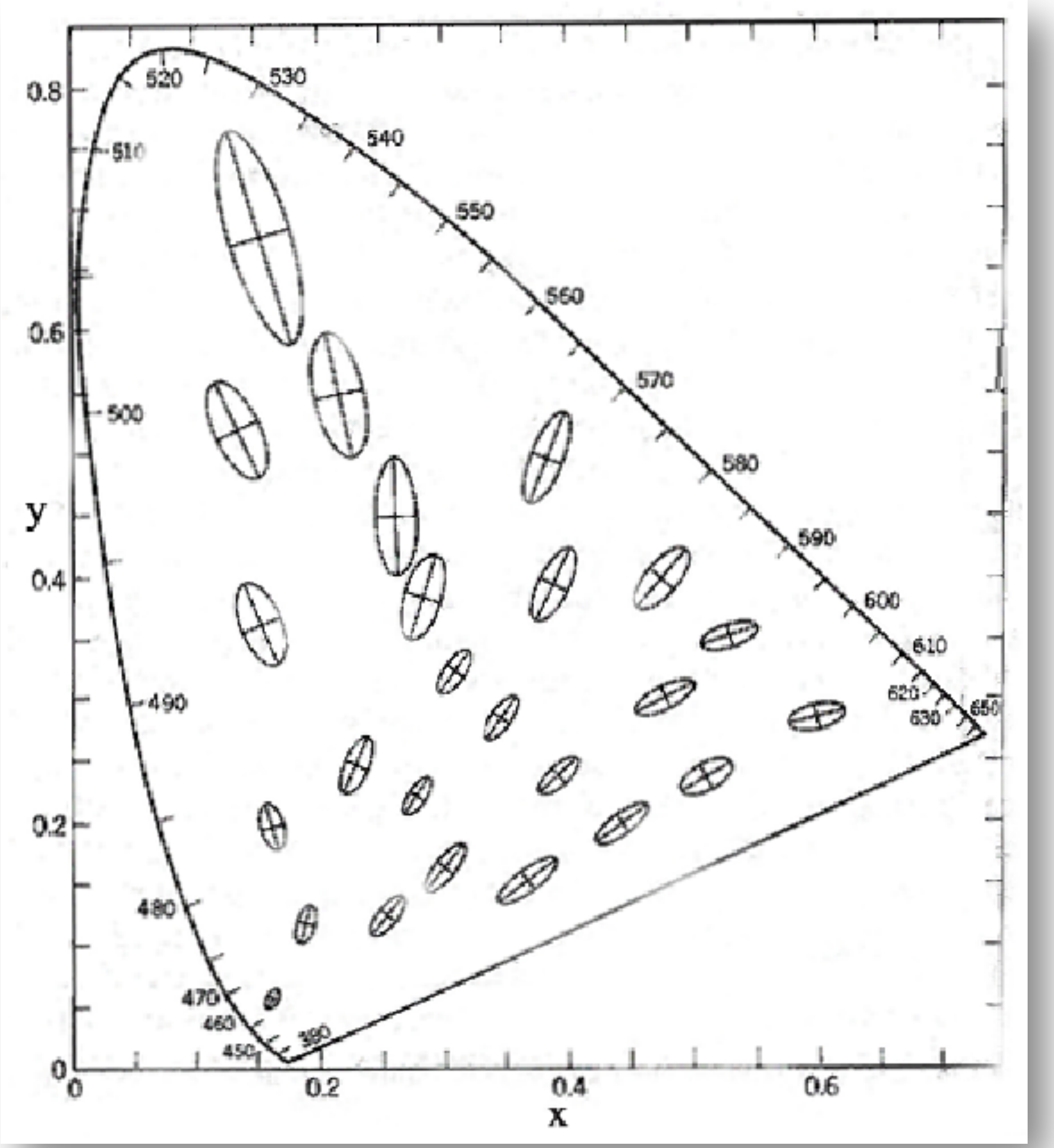
$$L^* = 903.3\left(\frac{Y}{Y_w}\right), \quad otherwise$$



Y = CIE luminance
Y_n = CIE luminance of a white surface
in the same viewing conditions

Macadam's Ellipses/Ellipsoids

- Ellipses shown at x10 threshold
- Kodak made 3D measurements
- Circularizing the ellipses is one of several metric constraints (Euclidean distance)



CIELAB Color Space

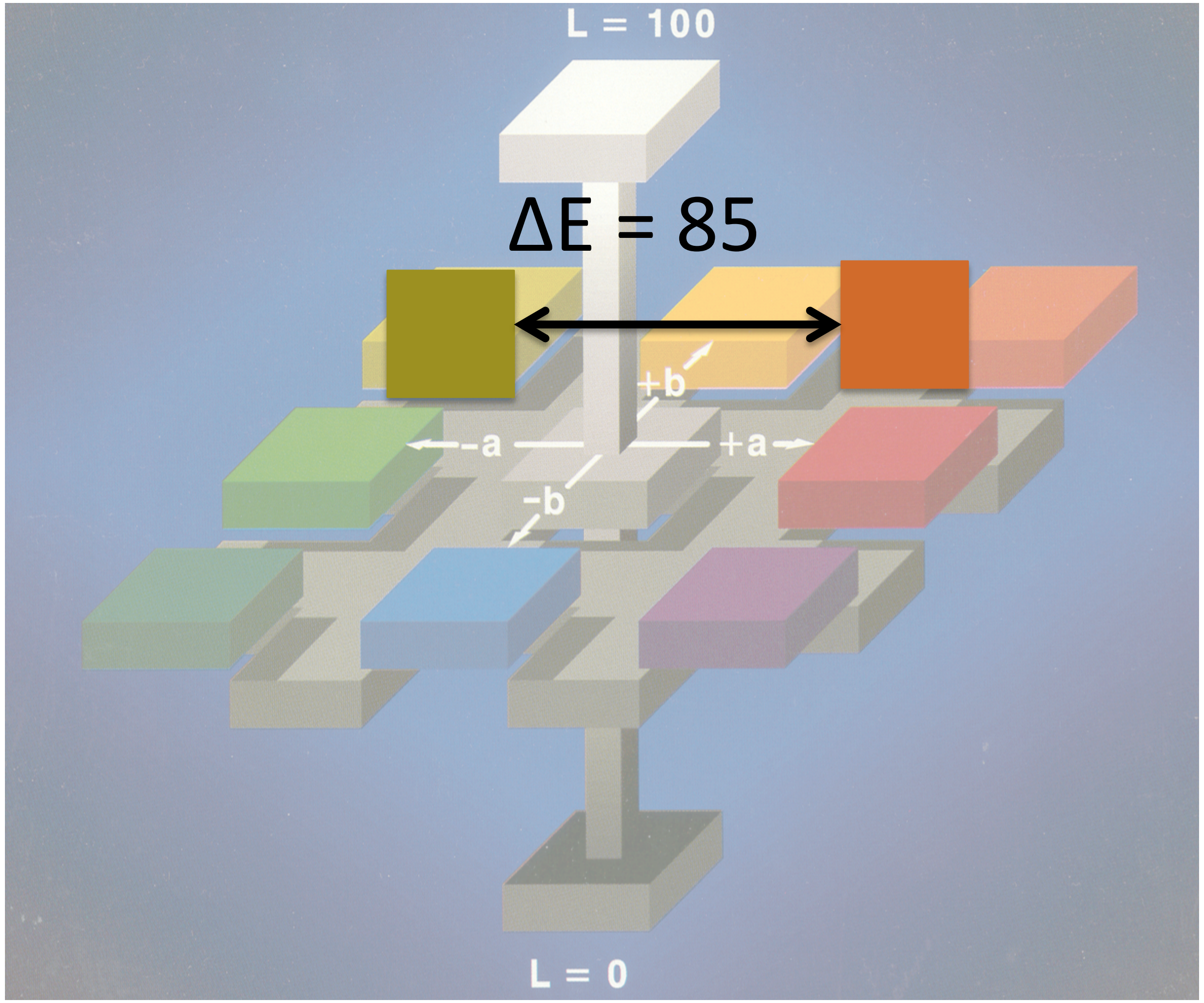
Distance between two colors designed to predict color difference visibility

$$\Delta E_{ab} = \left((\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2 \right)^{1/2}$$

CIELAB and ΔE

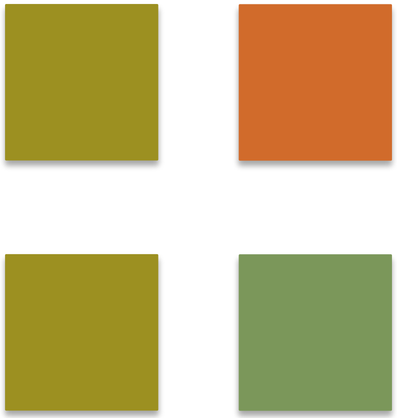
The metric is used to predict the visible difference (expressed in units of ΔE_{ab}) between two uniform color patches

ΔE = 85



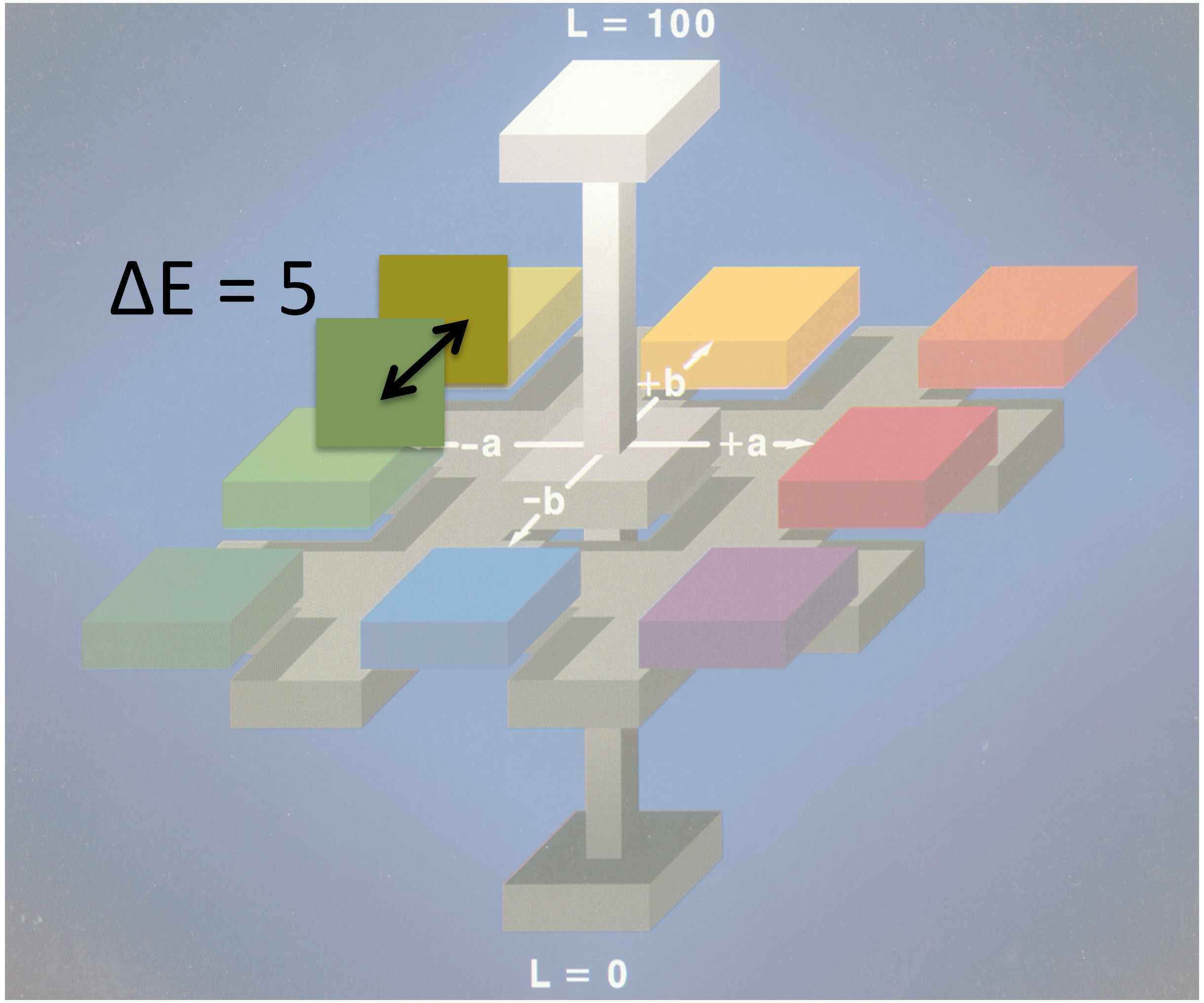
CIELAB and ΔE

The ability to detect the difference between two color patches decreases with ΔE



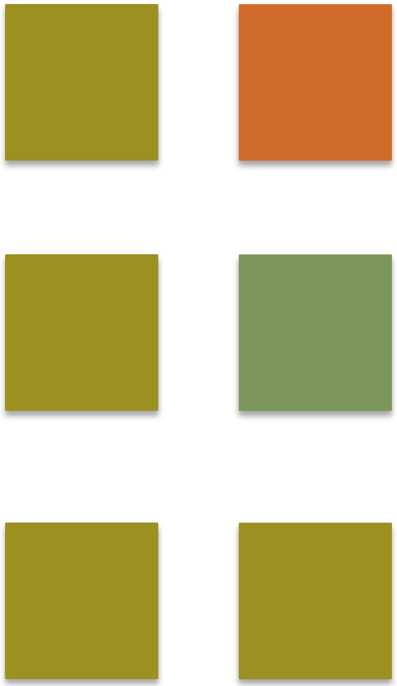
ΔE = 85

ΔE = 5



CIELAB and ΔE

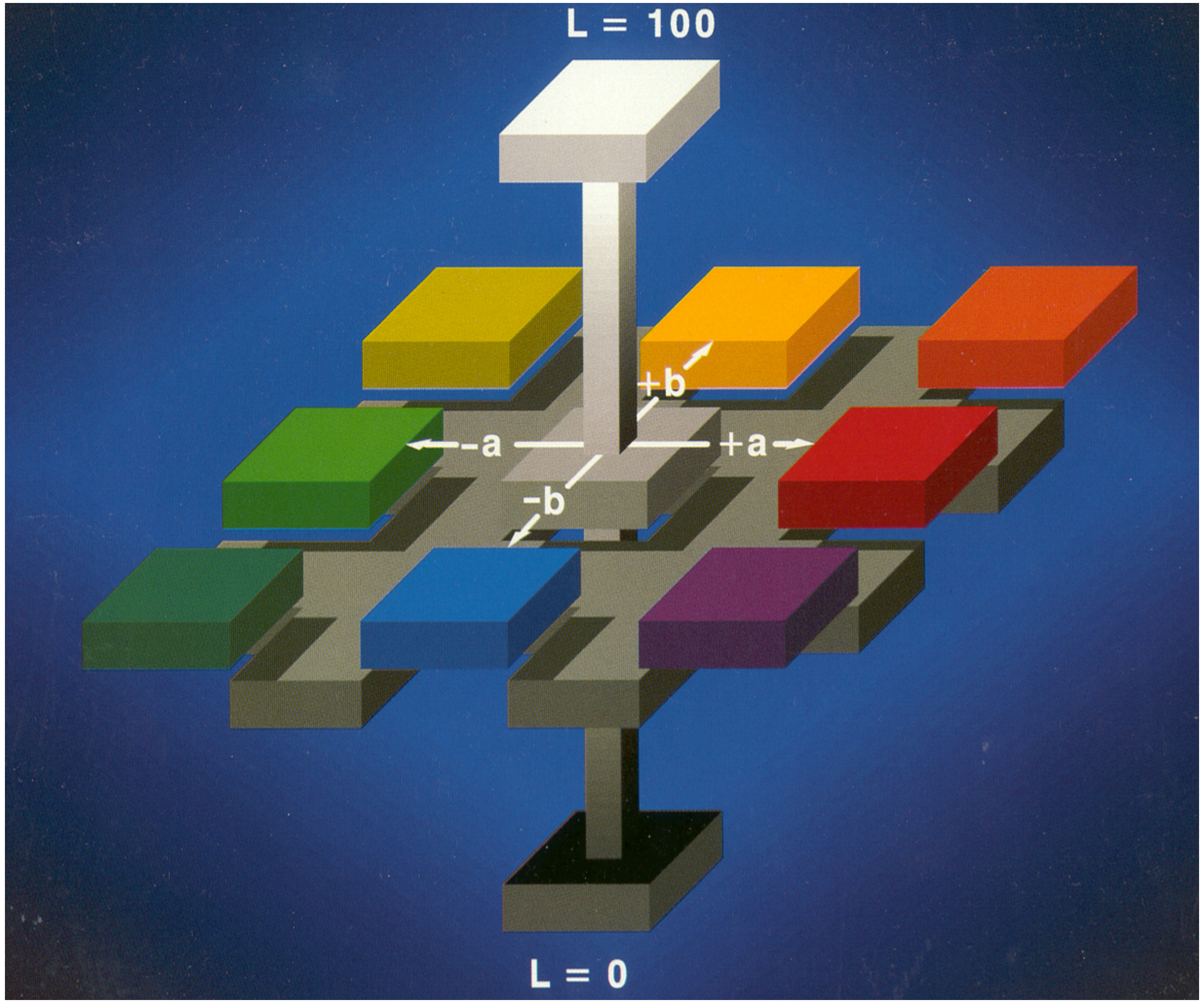
When ΔE is small enough, the difference will not be observed



ΔE = 85

ΔE = 5

ΔE = .05



Lab Color Space and ΔE

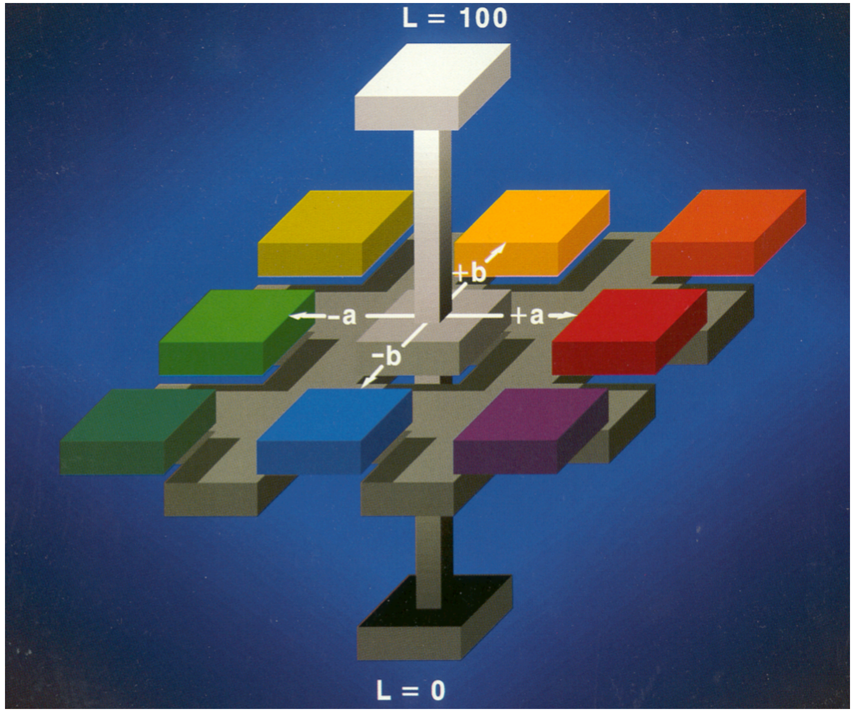
The value of Lab space is

- Intuitive (hue, chroma, lightness)
- Opponent color space (to be explained later)

    L is a weighted combination of L and M

    a is red (L) / green (M) opponent representation

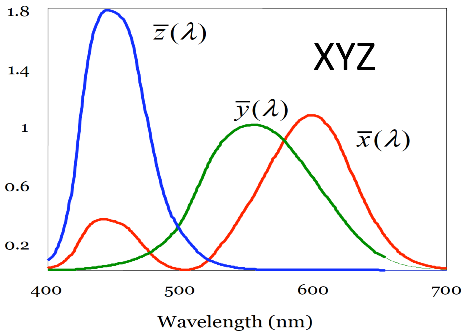
    b is blue (S) / yellow (R+G) opponent representation



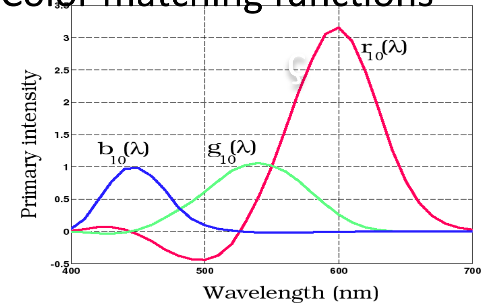
- Useful metric – ΔE based on Euclidean distance in Lab color space

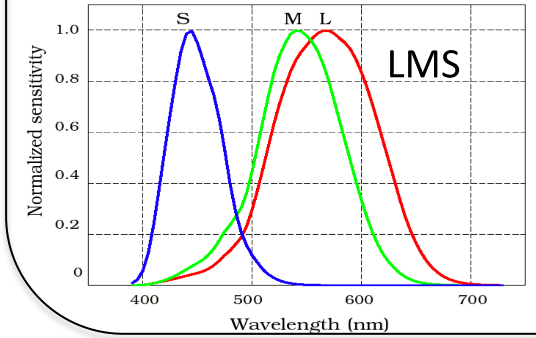
Review: Color Spaces

Sensor

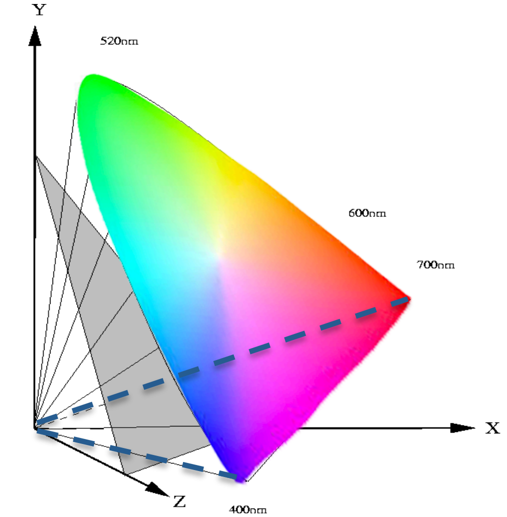


Color matching functions





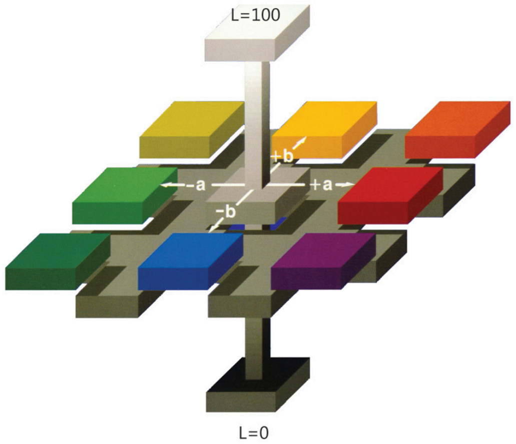
Chromaticity



$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

Color Difference



$$L^* = 116\left(\frac{Y}{Y_w}\right)^{1/3} - 16, \quad if\ \frac{Y}{Y_w} > .00856$$

$$L^* = 903.3\left(\frac{Y}{Y_w}\right), \quad otherwise$$

$$a^* = 500\left\{\left(\frac{X}{X_w}\right)^{1/3} - \left(\frac{Y}{Y_w}\right)^{1/3}\right\}$$

$$b^* = 200\left\{\left(\frac{Y}{Y_w}\right)^{1/3} - \left(\frac{Z}{Z_w}\right)^{1/3}\right\}$$

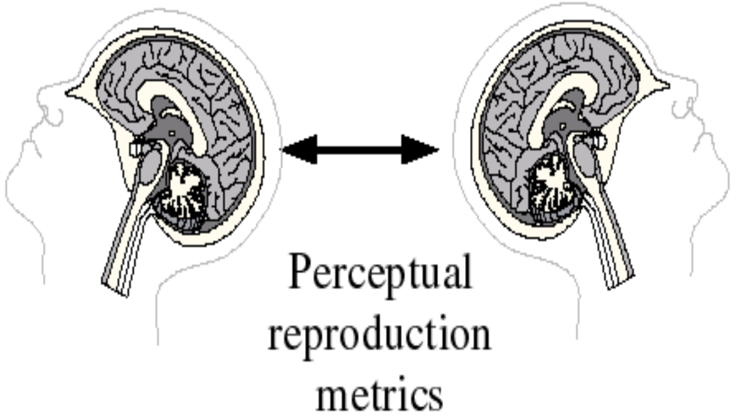
Color Management

What is Color Management

The goal of color management is to preserve the **appearance** of color images when they are rendered on different devices, such as

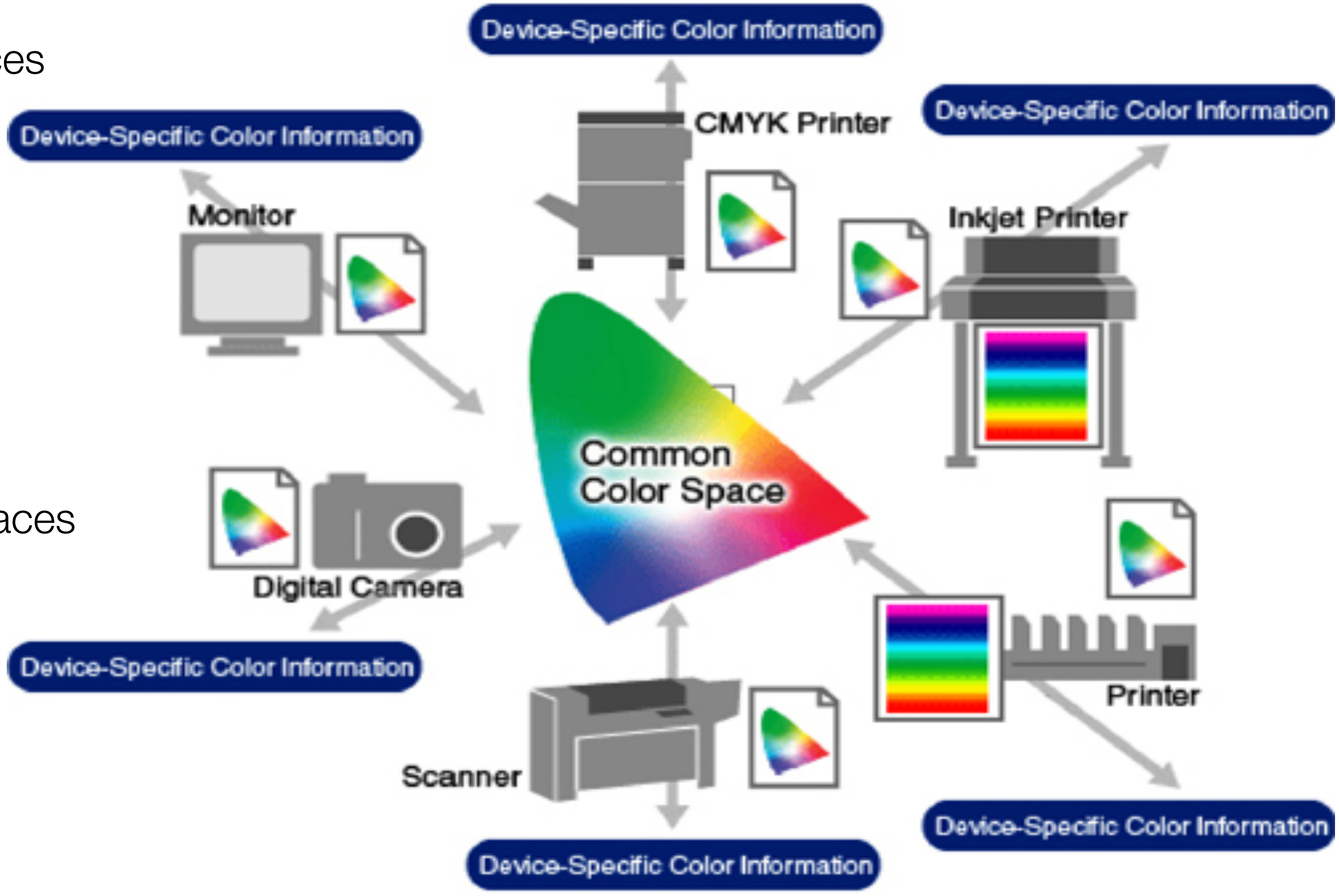
- Printers
- Displays

Color matching is not the same thing as color appearance



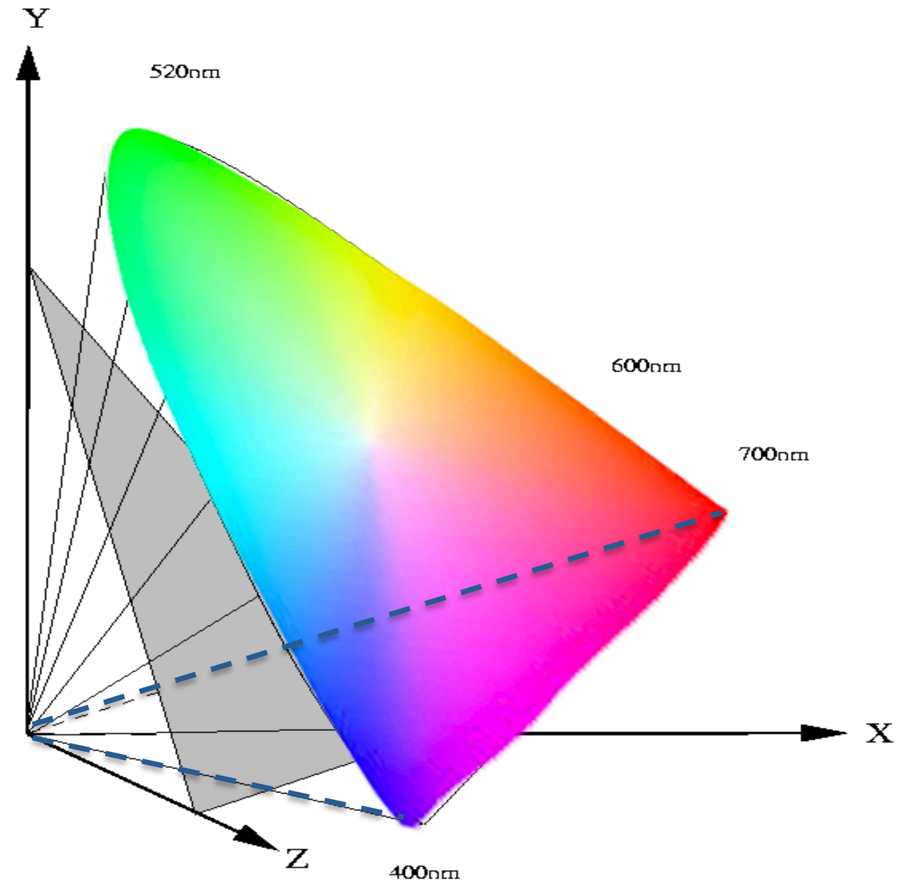
Color spaces

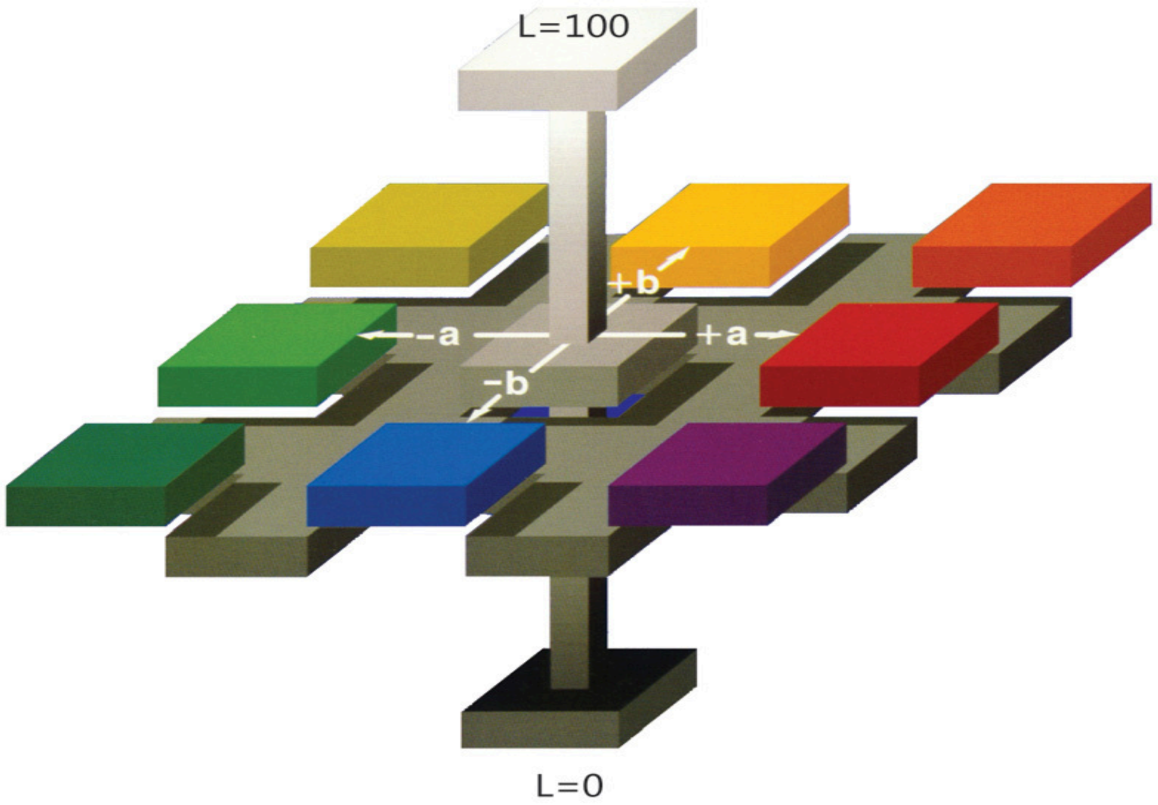
- ❑ Device-dependent color spaces
  - sRGB
  - Adobe RGB
  - ProPhoto RGB
- ❑ Device-independent color spaces
  - Chromaticity
  - CIE Lab



http://www.eizo.com/library/management/cms/

Color spaces





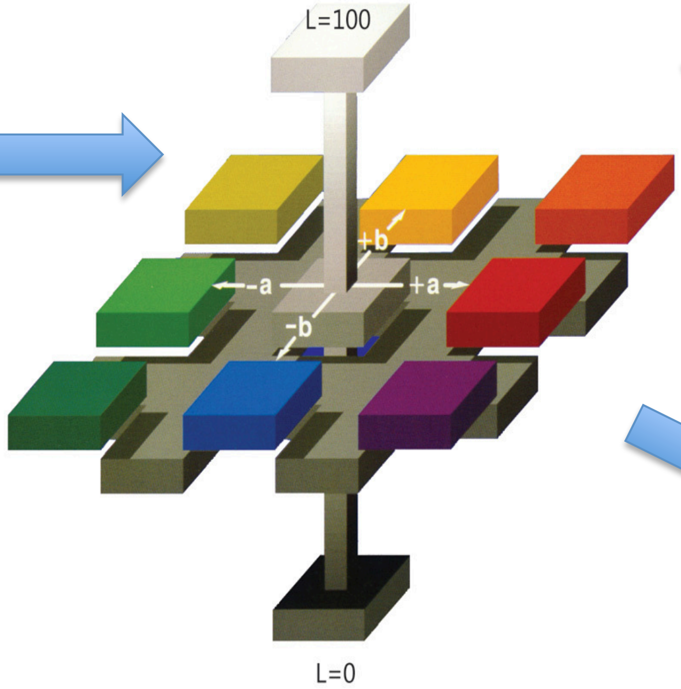
Pros: Visualization
Cons: Z projection can be confusing

Pros: Chroma, Hue, Brightness
Color Difference Metric
Cons: Dependence on white point

What is the White Point?

Sony Trinitron CRT

Lab Color Space

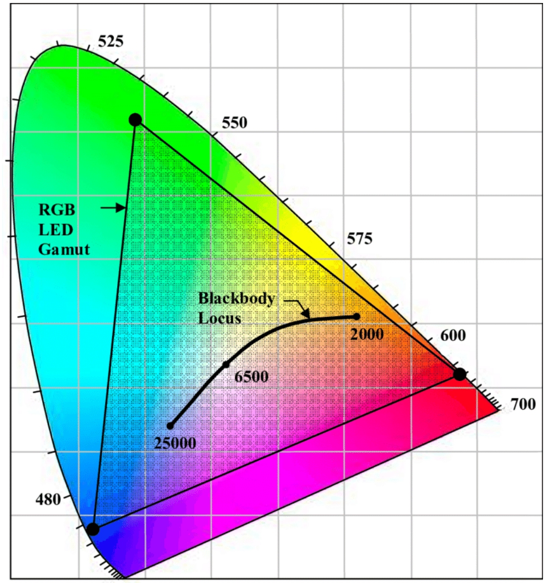


Printers

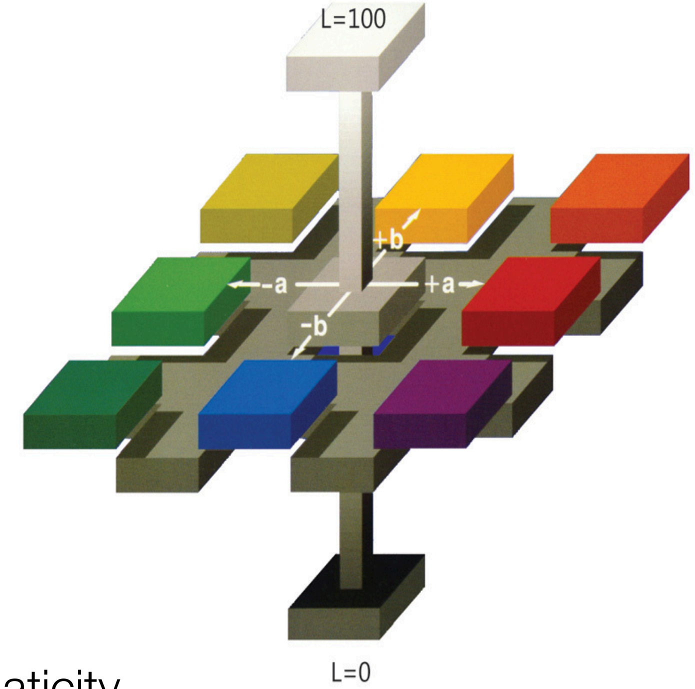
The “white point” is D65

De facto Industry standard

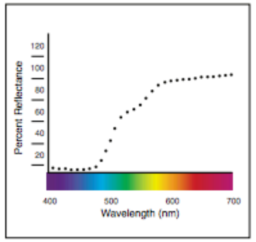
Set display “white point” to be D65



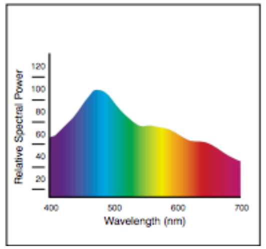
max R, G and B values produce chromaticity coordinates for D65 (x = 0.32371, y = 0.32902)
Normalizing for relative luminance (set Y = 100), the XYZ values that produce these coordinates are X = 95.047, Y =100, Z – 108.883



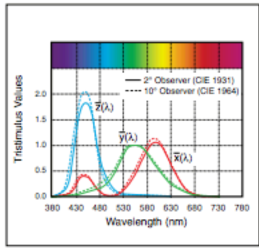
Measure XYZ of printed colors under “D65” illuminant



Spectral reflectance



Spectral power of D65 illuminant

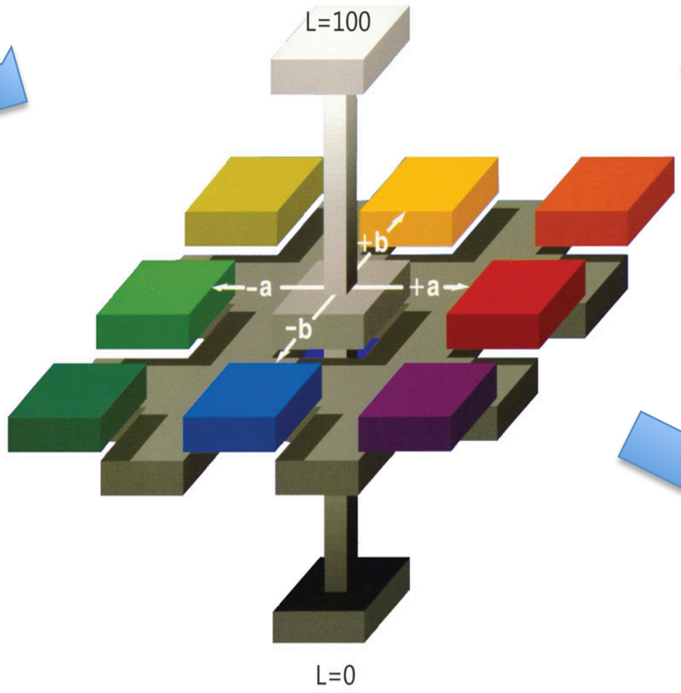


XYZ functions

Lab Color Space

Displays

Lab Color Space

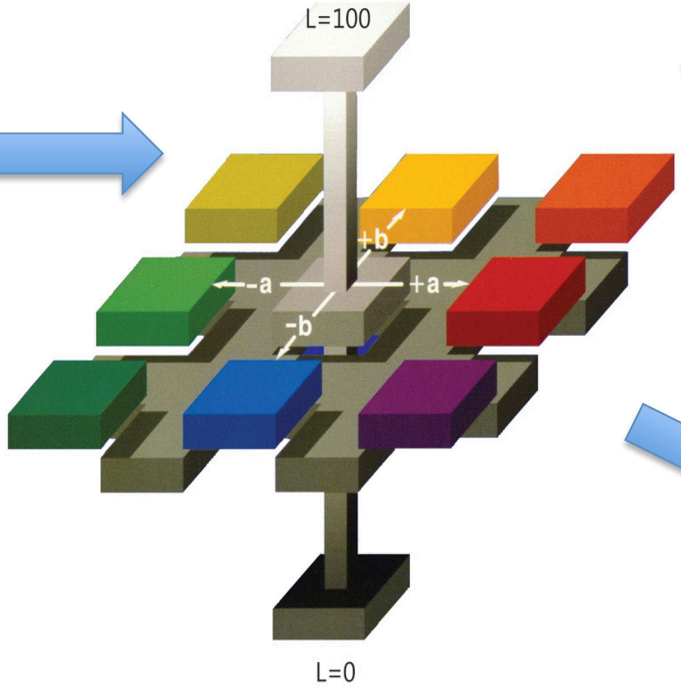


Printers

Let's make this even simpler : sRGB Standard

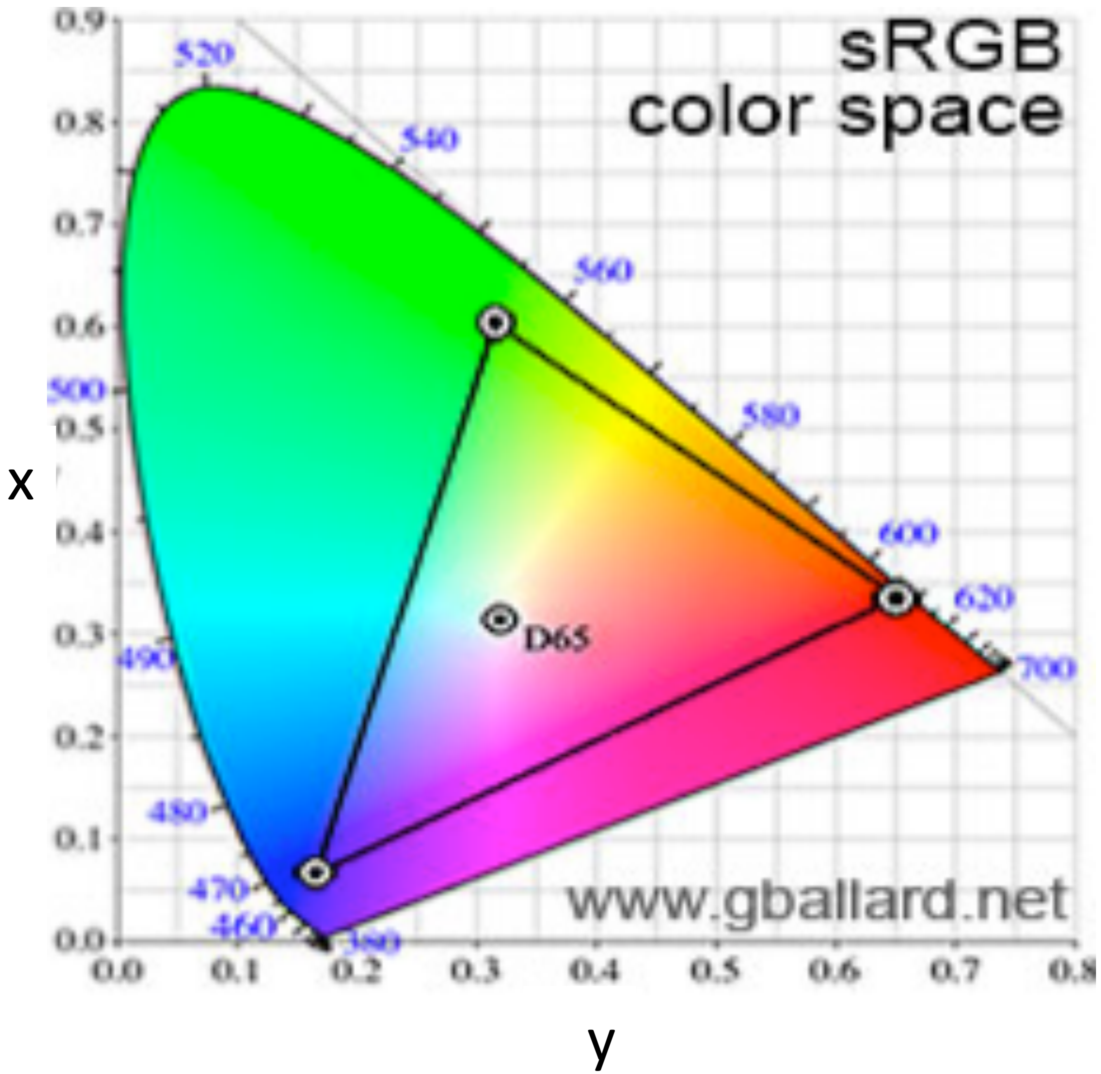
Sony Trinitron CRT

Lab Color Space

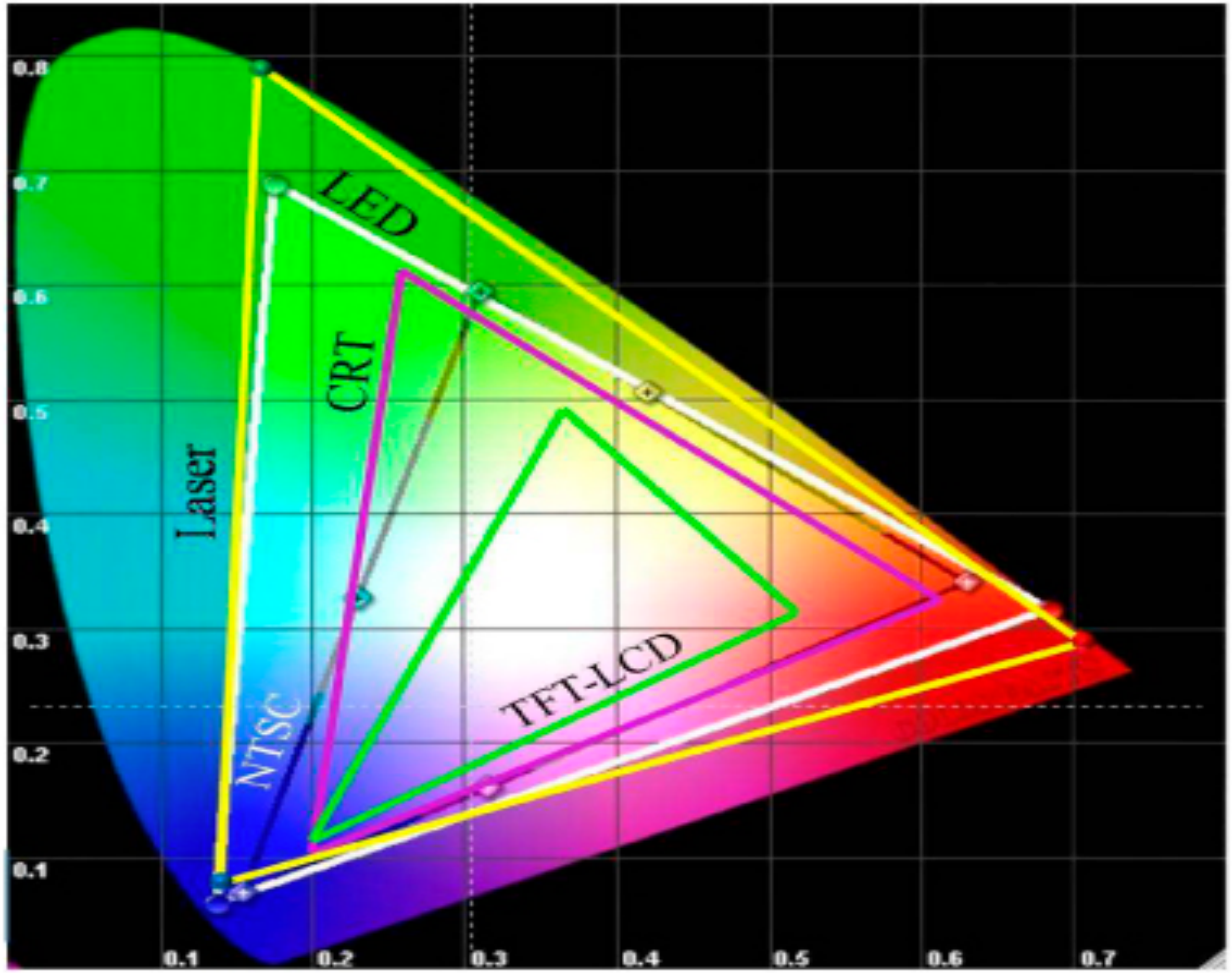


Printers

sRGB is the gamut for a Sony Trinitron CRT display

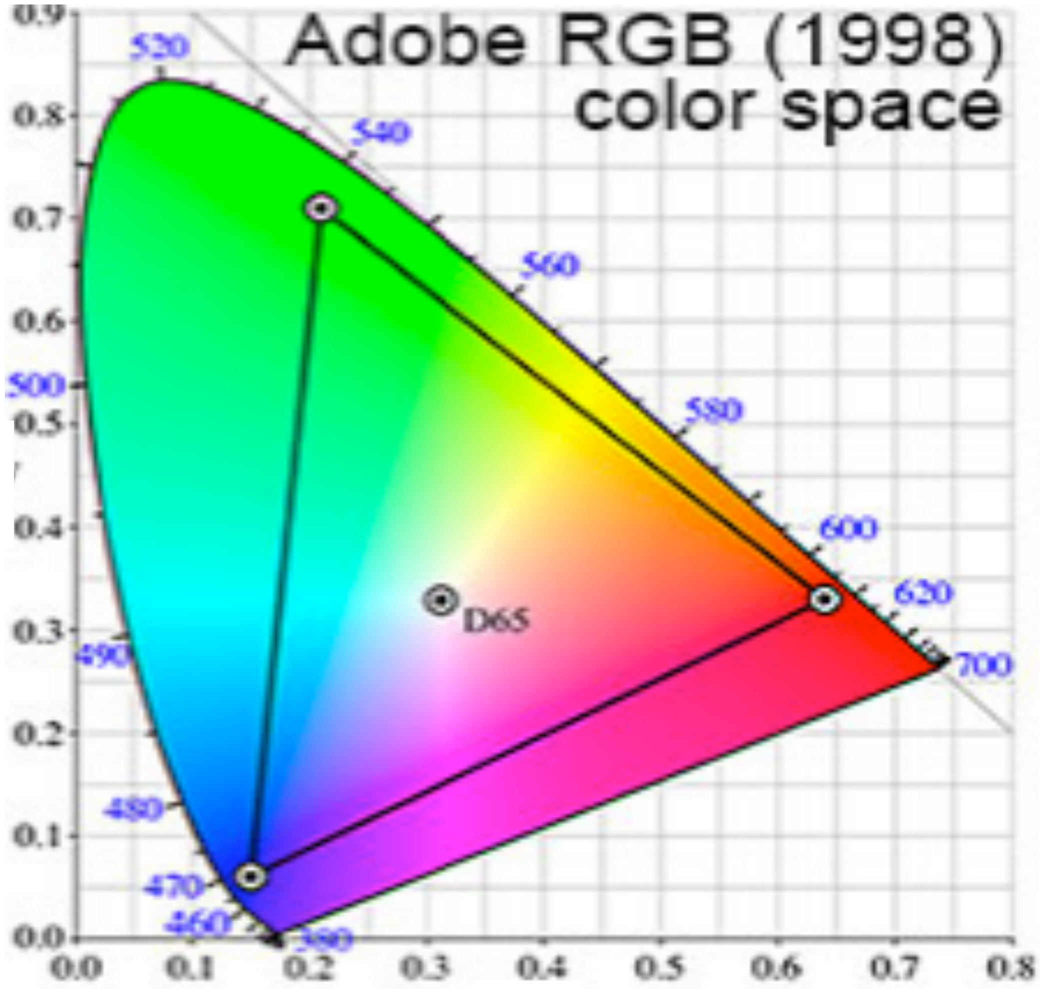


Display Gamuts



https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-18-5-4655&id=196021

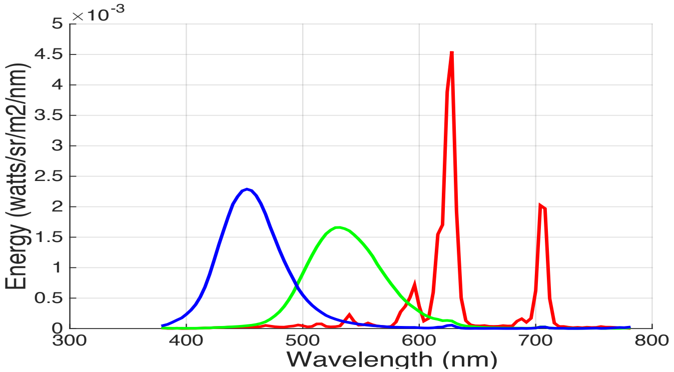
Adobe introduced a gamut that can include more displays

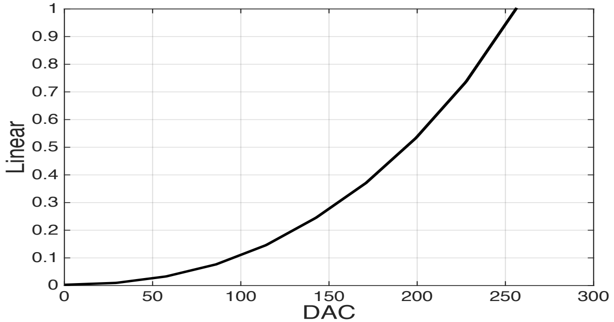


x
y

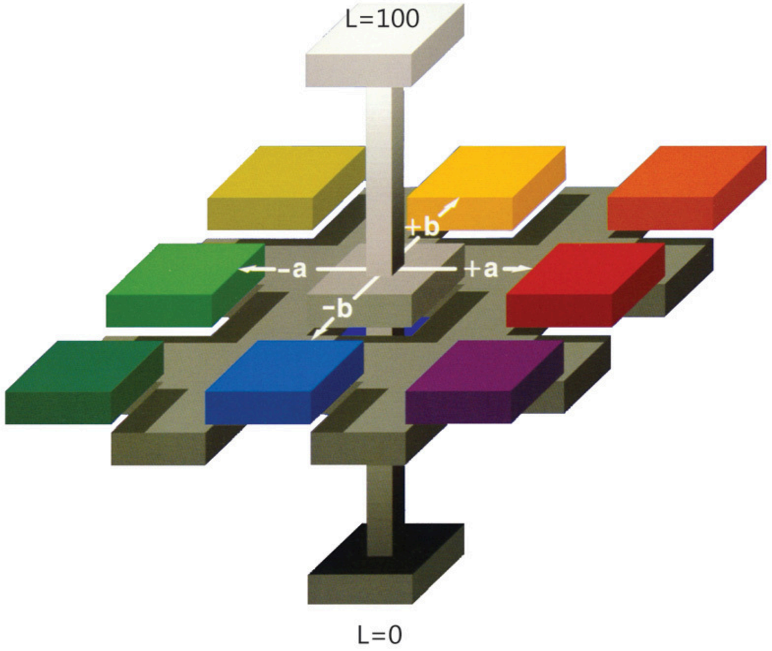
History of Color Management Systems

Calculate XYZ



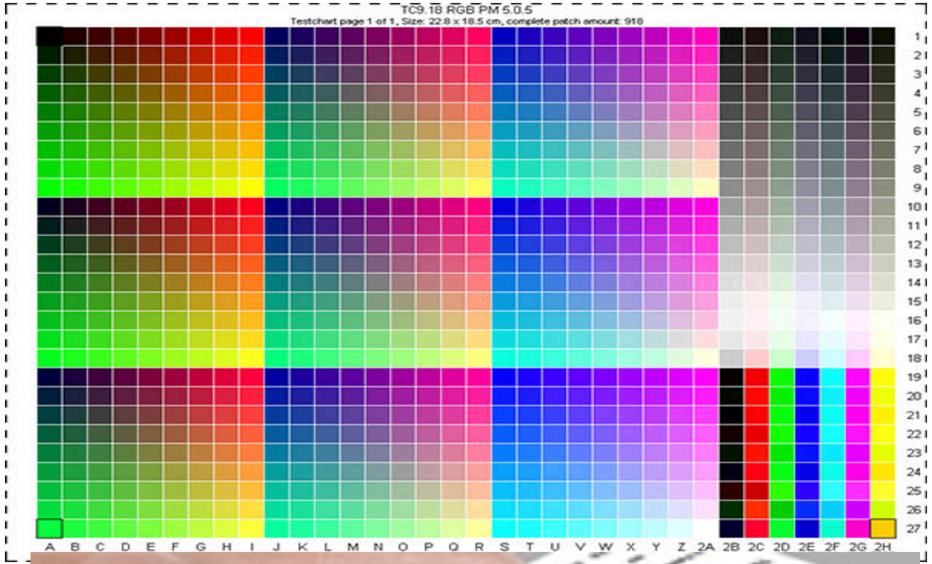


Lab LUT



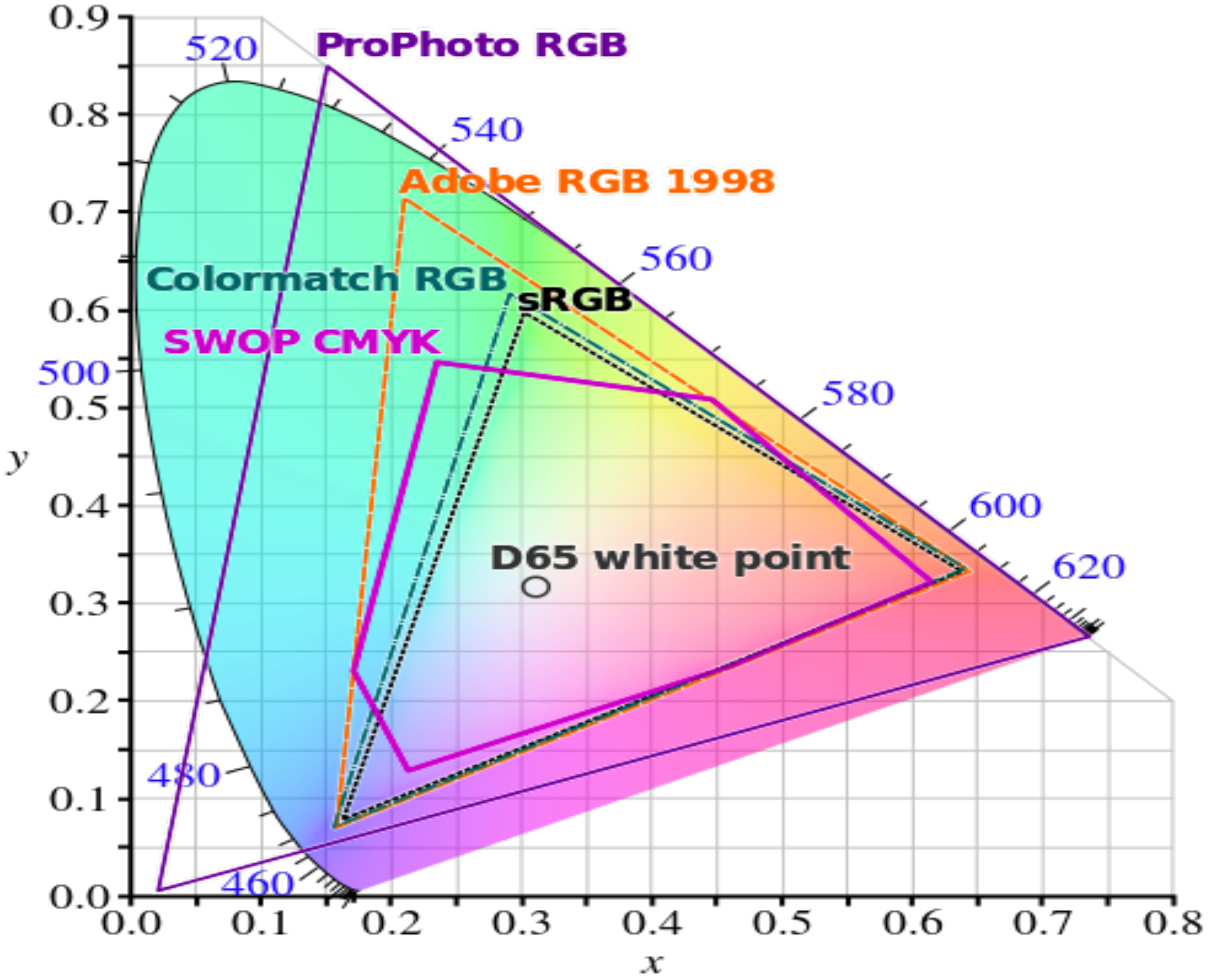
Early 1990s

Measure XYZ

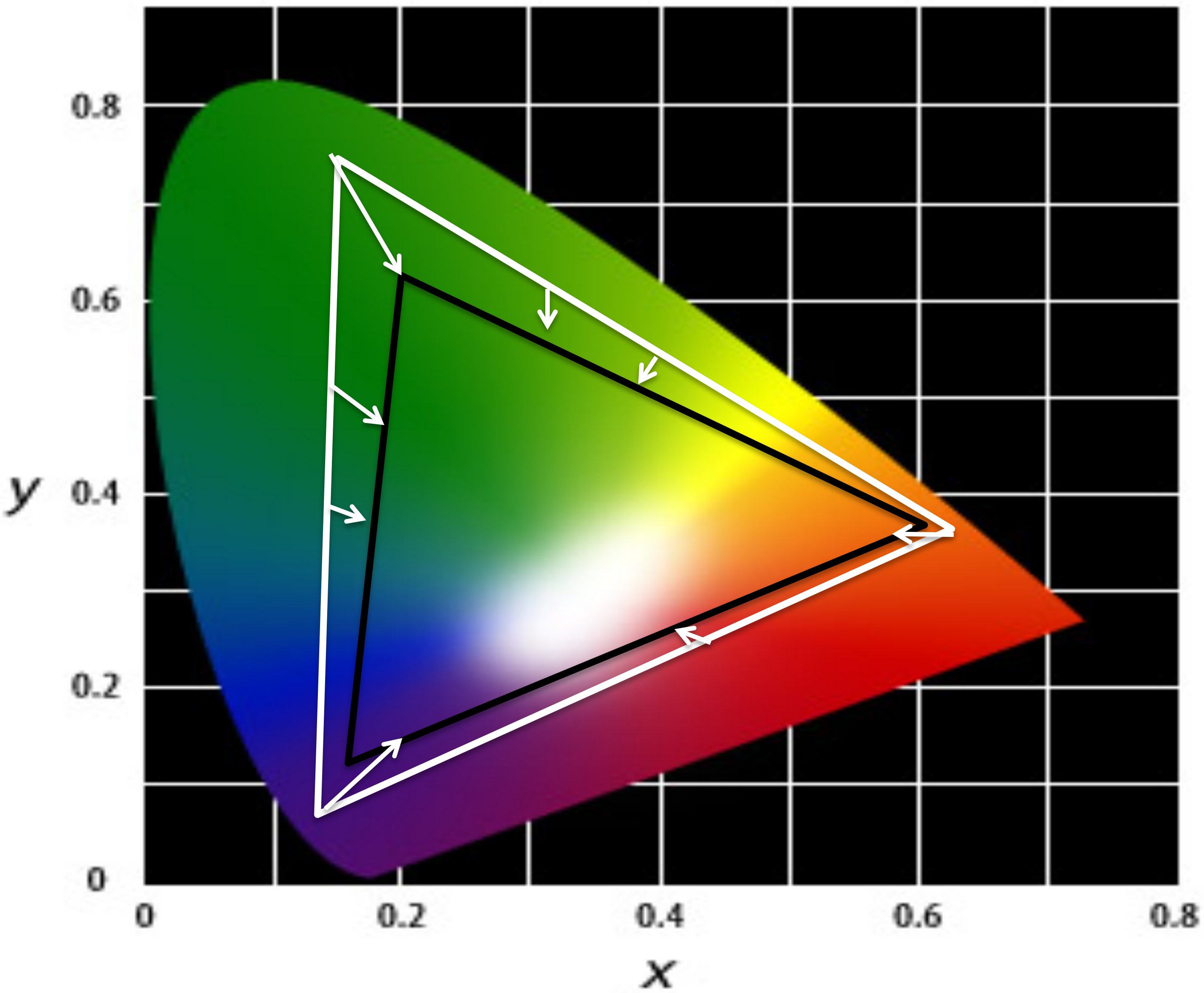


Gamut Mapping

- ❑ Display and Printer Gamut mismatch
- ❑ Mapping a large printer gamut to a smaller printer gamut
    - o Clipping (“Colorimetric”)
    - o Compression (“Perceptual”)
- ❑ Mapping a smaller display gamut to a larger display gamut
    - o Expansion

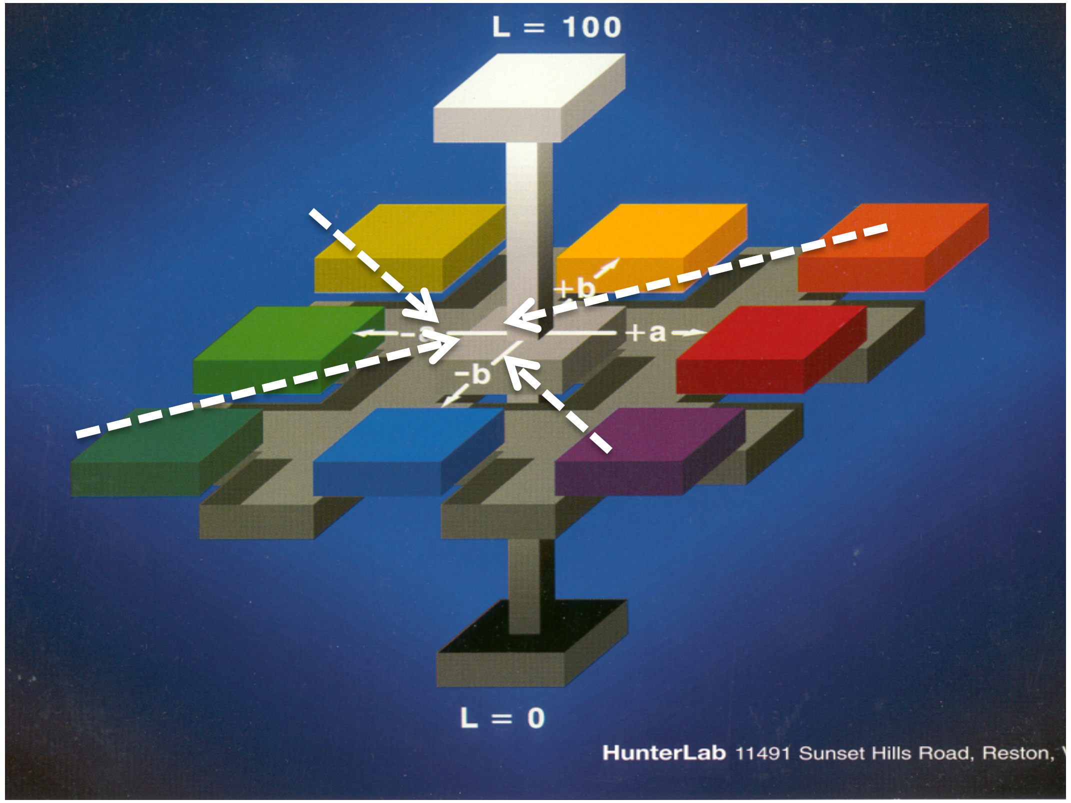


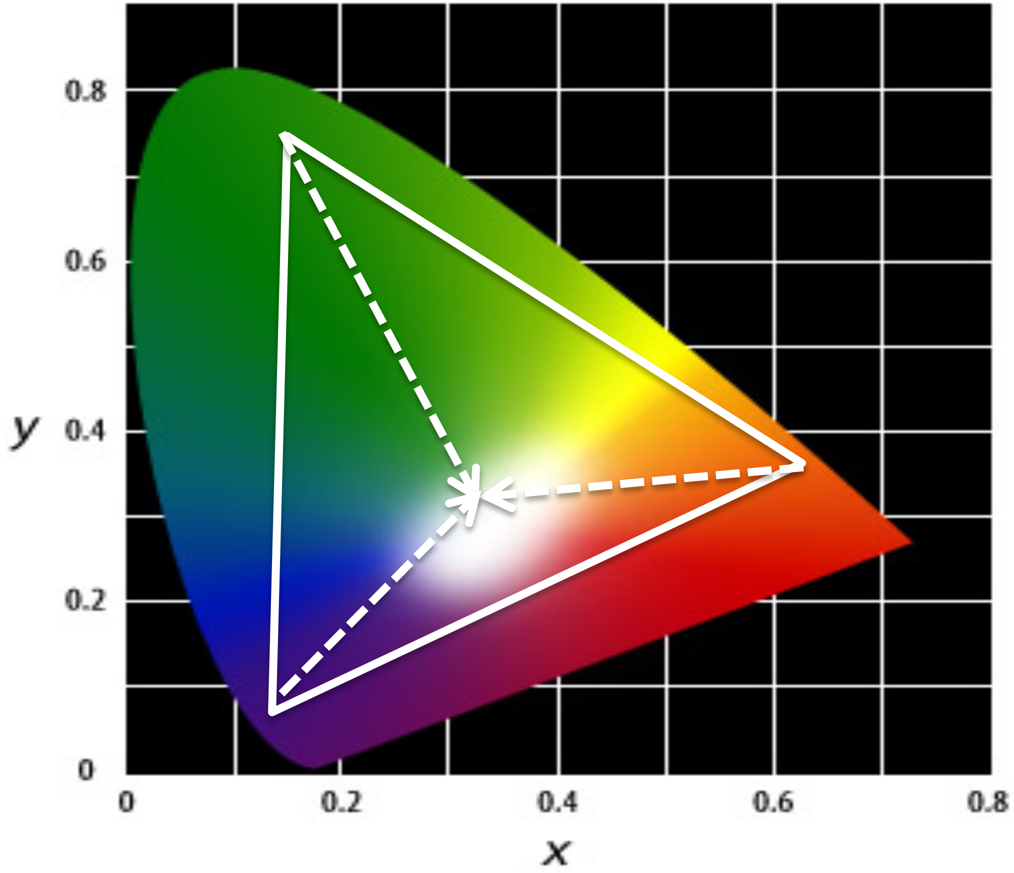
Gamut Clipping (“Colorimetric”)



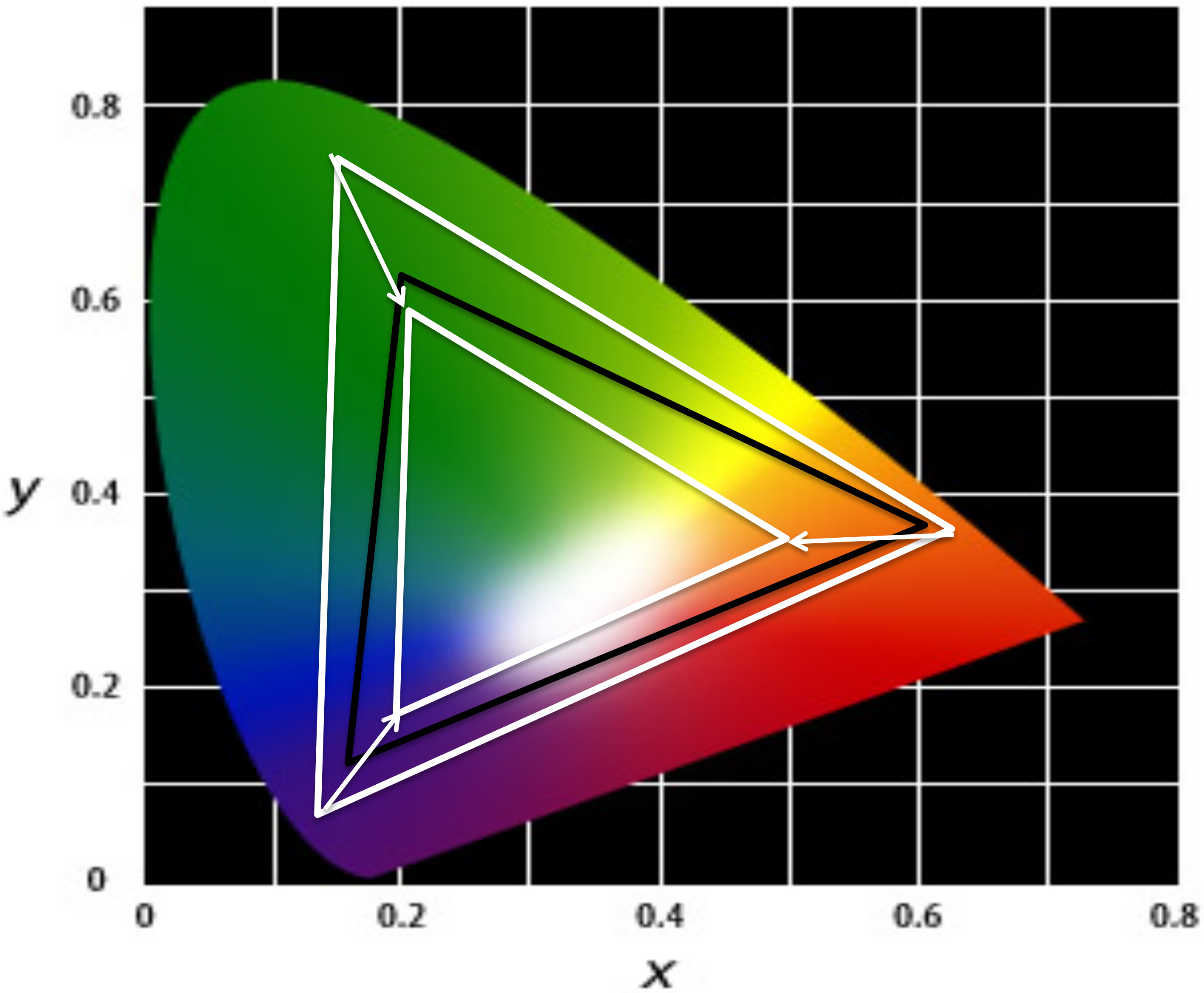
Lines of Constant Hue

Lines of constant hue, but decreasing chroma (saturation)

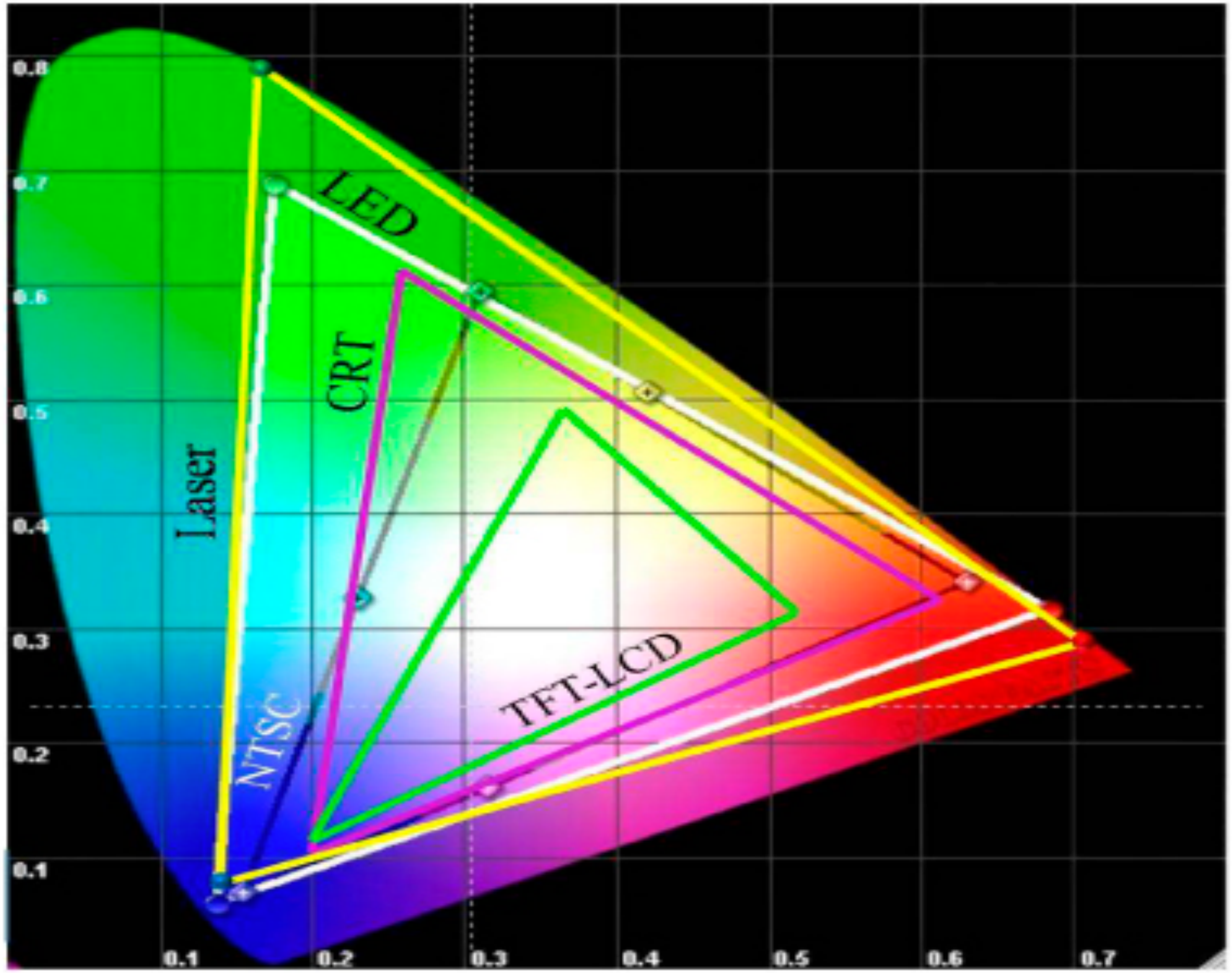




Gamut Compression ("Perceptual")



Displays (Gamut Expansion)



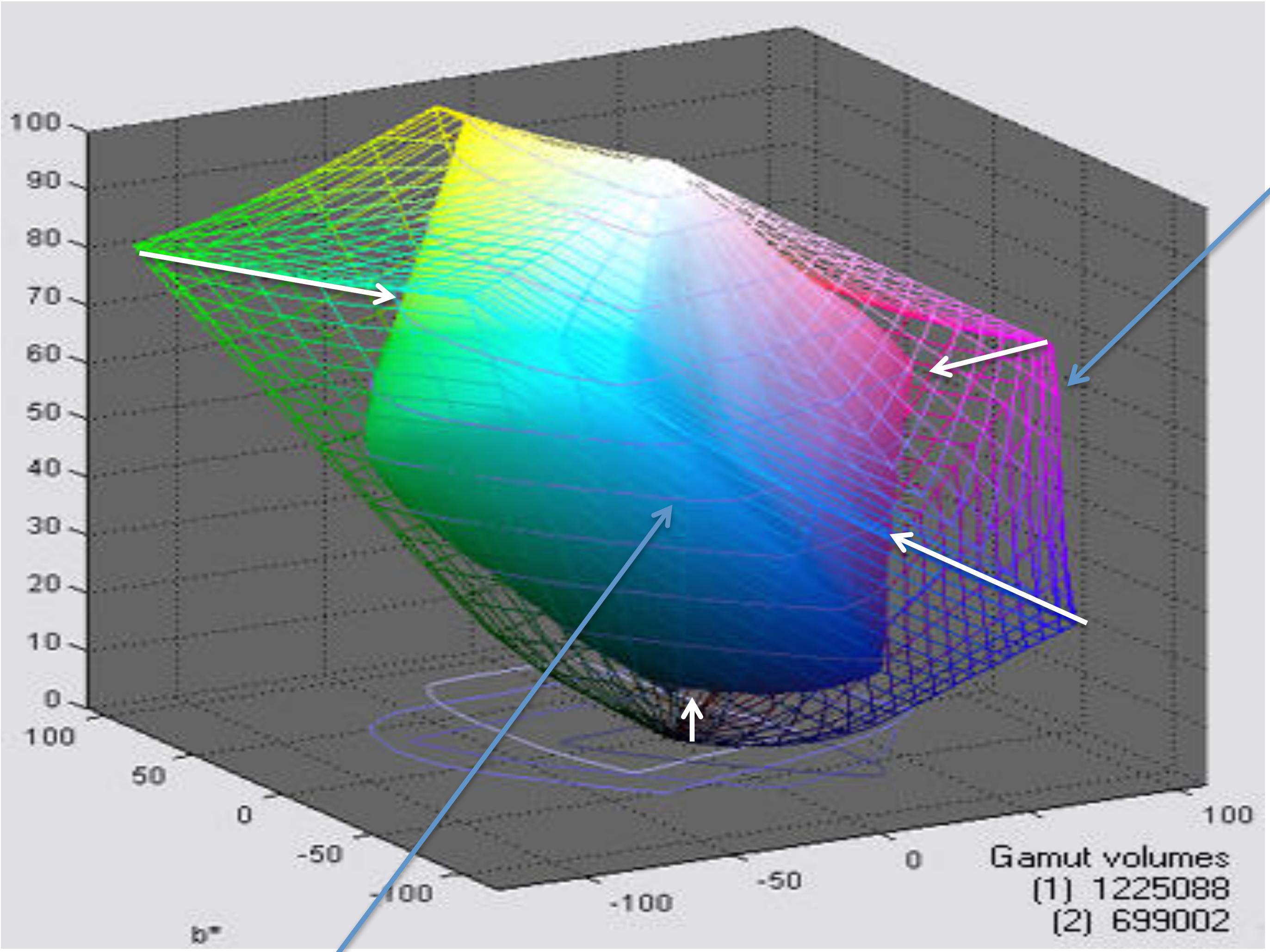
https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-18-5-4655&id=196021

Many different gamut mapping methods

Many methods – and inconsistent labeling

- Colorimetric
- “Relative Colorimetric”
- Perceptual
- Global versus local
- Clipping versus compression (linear, piecewise linear, sigmoidal)
- Gamut mapping and tone mapping

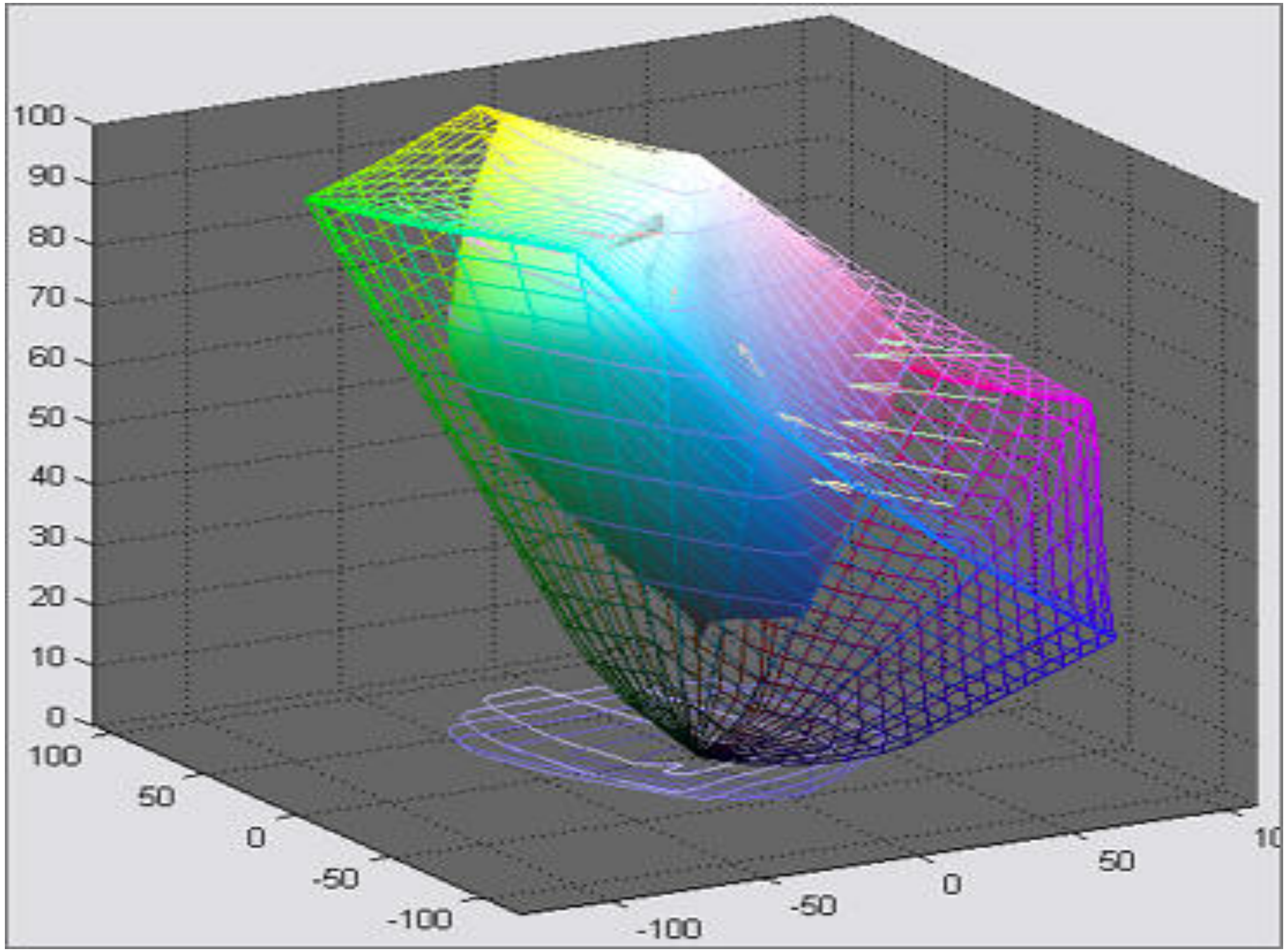
Gamuts in Lab Space



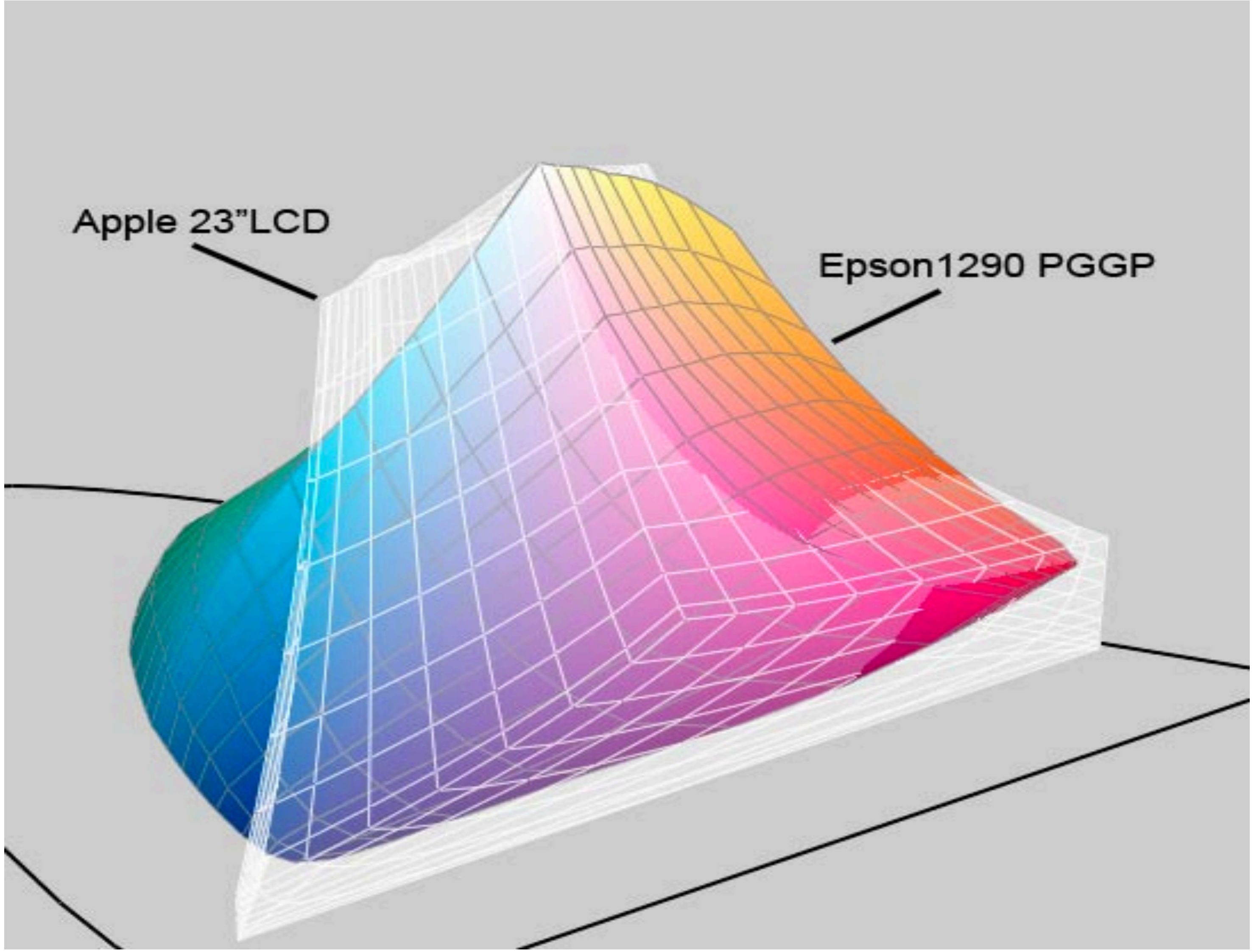
Adobe RGB

Inkjet printer on special paper

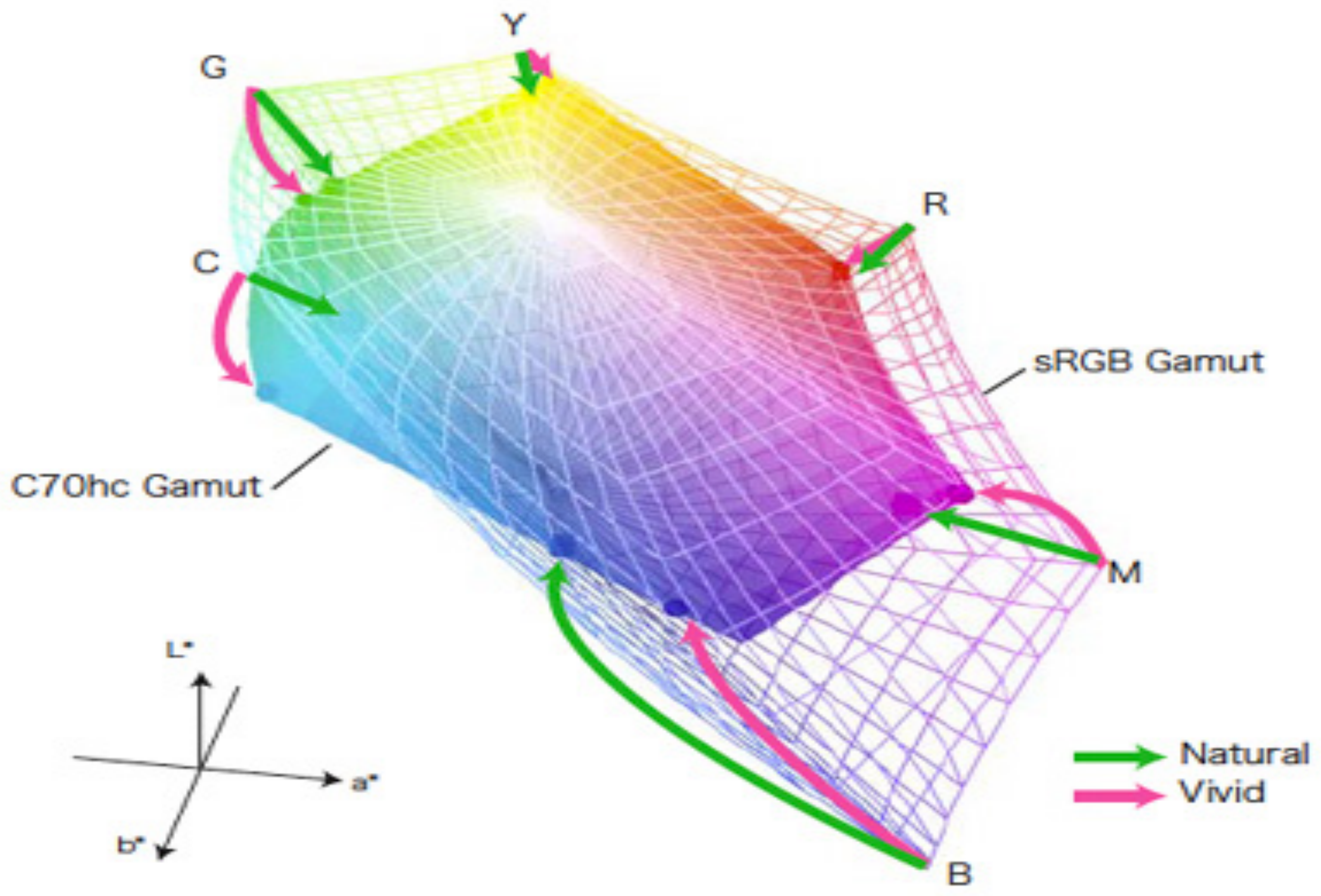
Gamuts in Lab Space



Gamuts in Lab Space



Different Gamut Mapping (Rendering Intents)



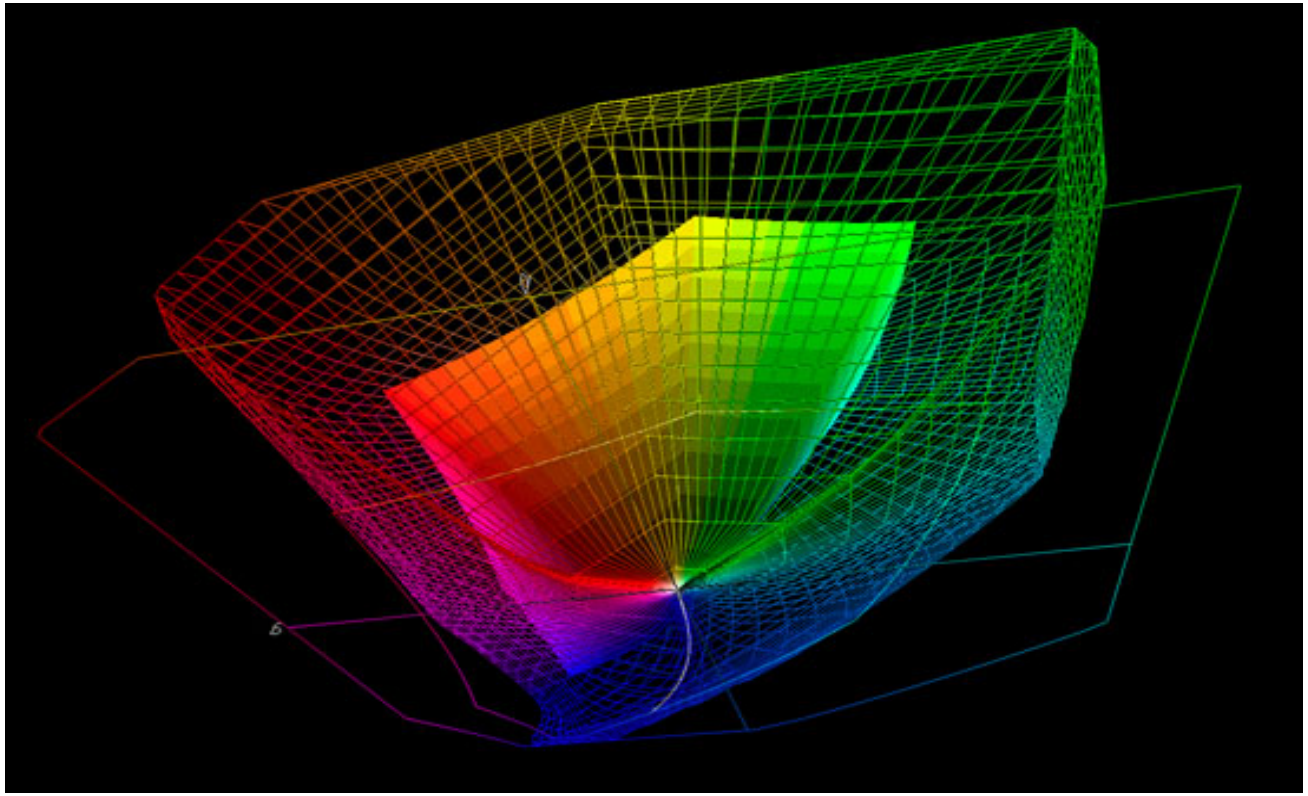
Optimizes mapping for use with the C70hc engine colour gamut.

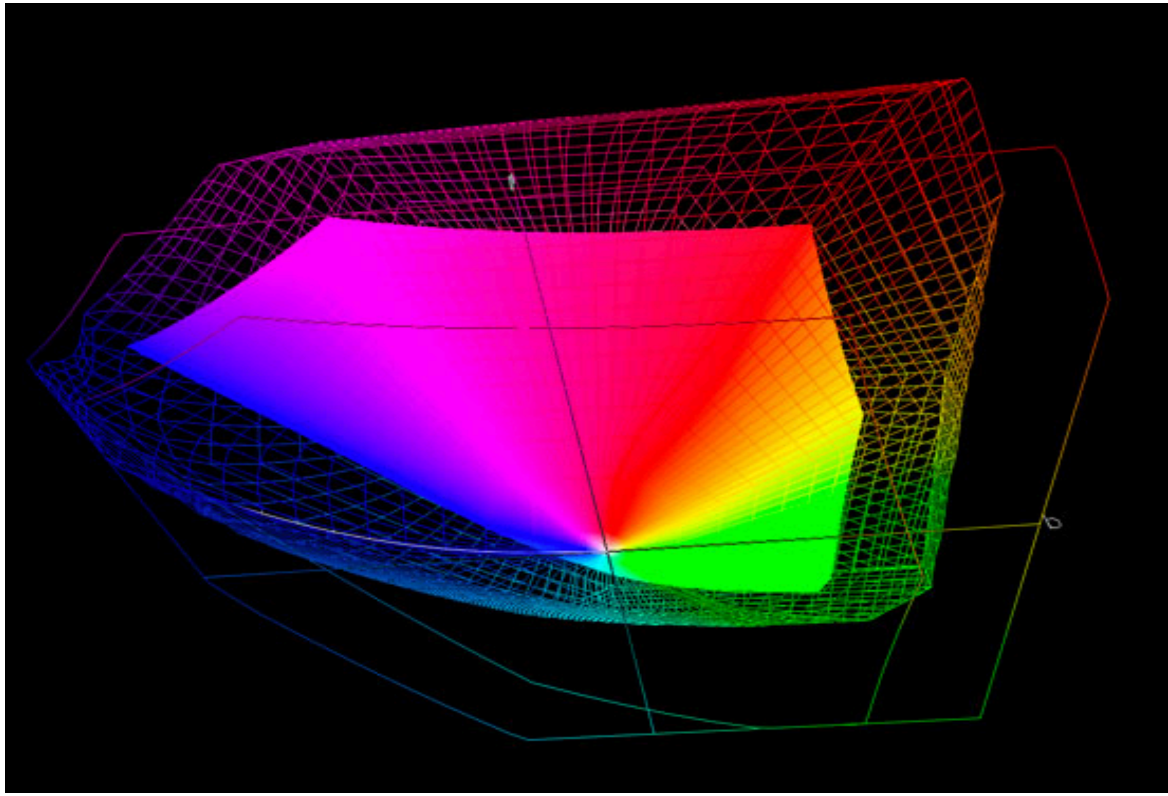
Adjusts the mapping pattern in a different mode for colours outside the gamut.

My gamut is bigger than your gamut

ProPhoto is a space developed by Kodak. It is bigger than Adobe RGB color space

completely contains the gamuts of all likely to be encountered **device dependent** colour spaces





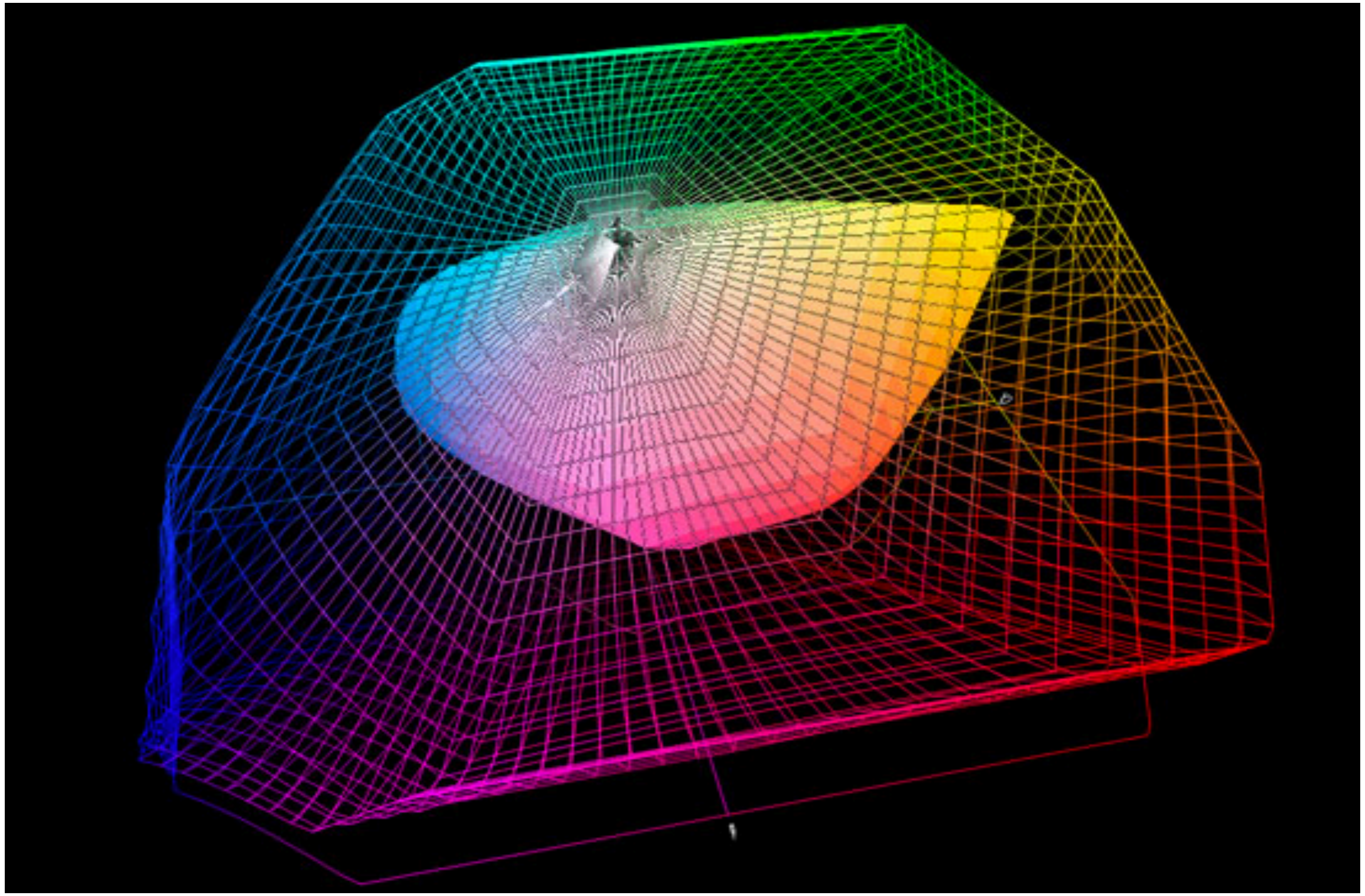
http://www.imagescience.com.au/kb/questions/85/January+2005+-+ProPhoto+or+ConPhoto

Yes, but …

Imagine that you defined your image in Prophoto color space. This is still a **device-dependent** space, however not many devices have this gamut

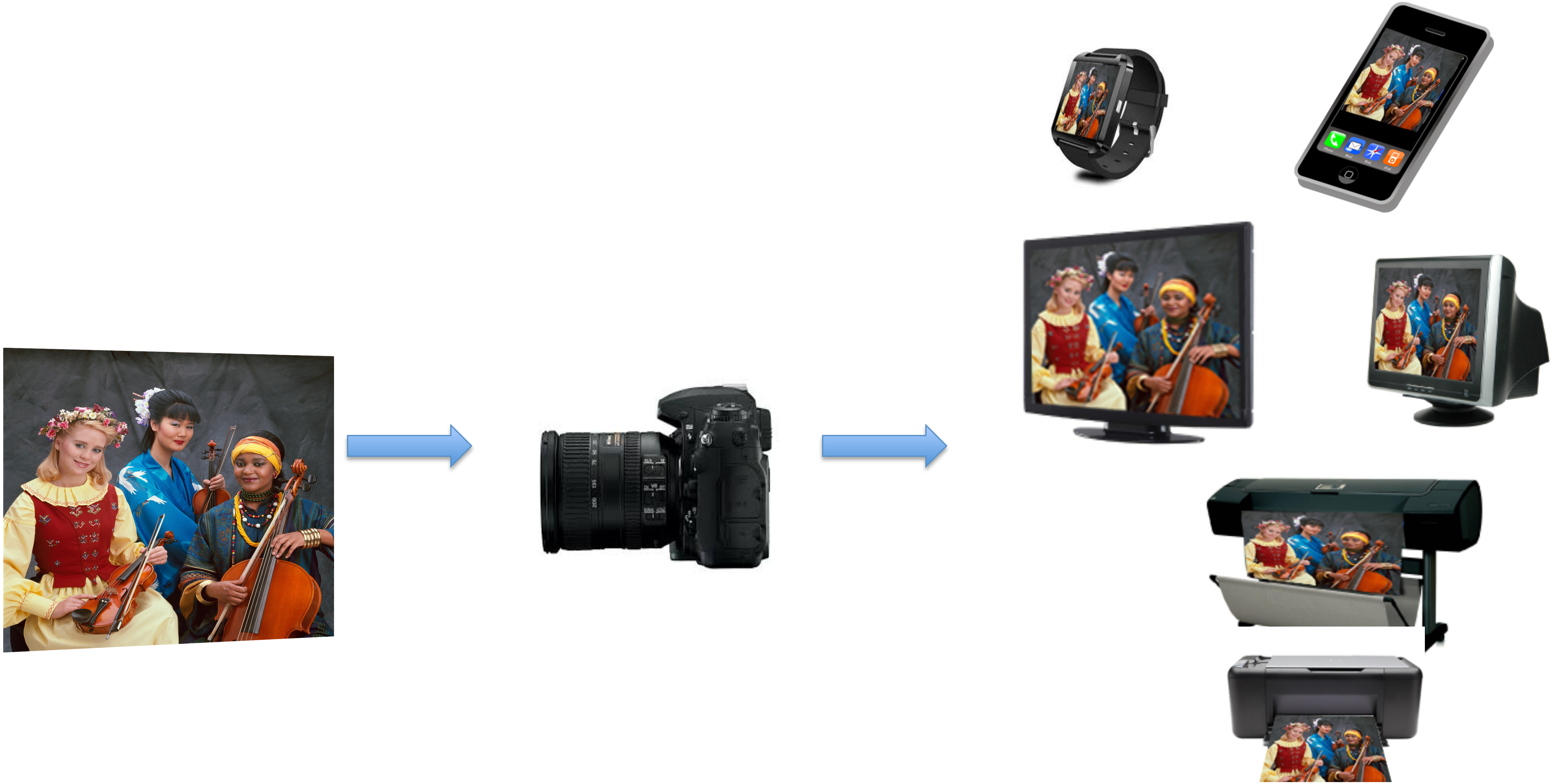
What if you have to print the image?

ProPhoto and an output profile for the Epson 2100



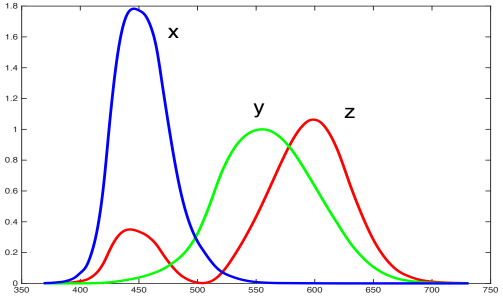
http://www.imagescience.com.au/kb/questions/85/January+2005+-+ProPhoto+or+ConPhoto

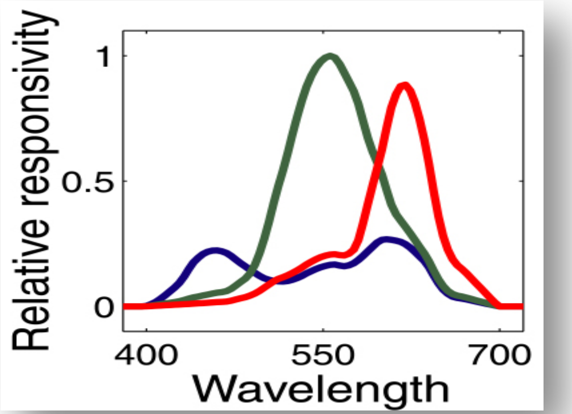
Color Management: Cameras to Displays



Color Management: Cameras to Displays

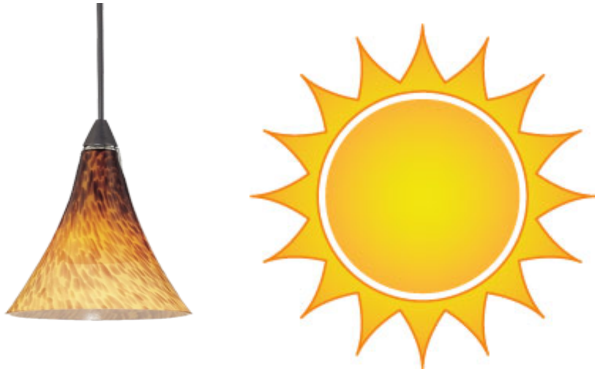
1. Sensor Correction





Color Management: Cameras to Displays

2. Illuminant Correction



Color management

ICC profiles

- Commonly used color processing terms
  - Used in different ways
  - I won't try to explain them

White balance

Color conversion

Color balance

Color correction

Color rendering

Illuminant transformation

Color constancy

Two critical color management steps

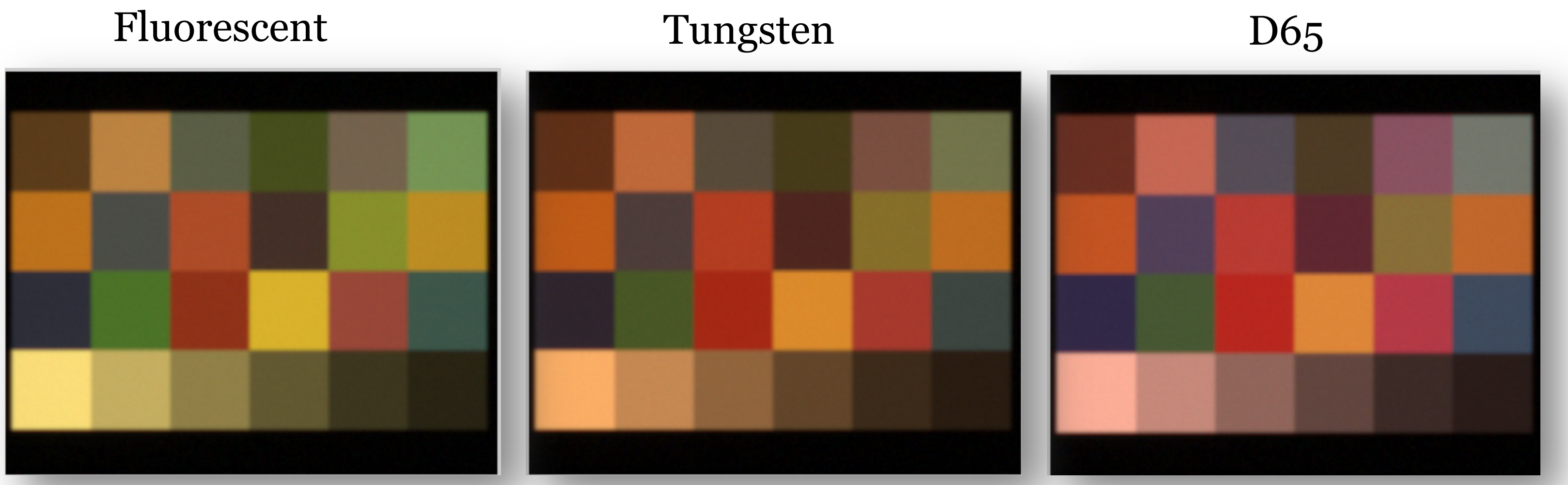
Sensor correction – adjust for the mismatch between the sensor filters and calibrated CIE (e.g., XYZ)

Illuminant correction – adjust for expected difference between imaging illumination and display illumination conditions

Color management: The problem

Copying sensor values directly to display graphics card or printer driver is a bad idea

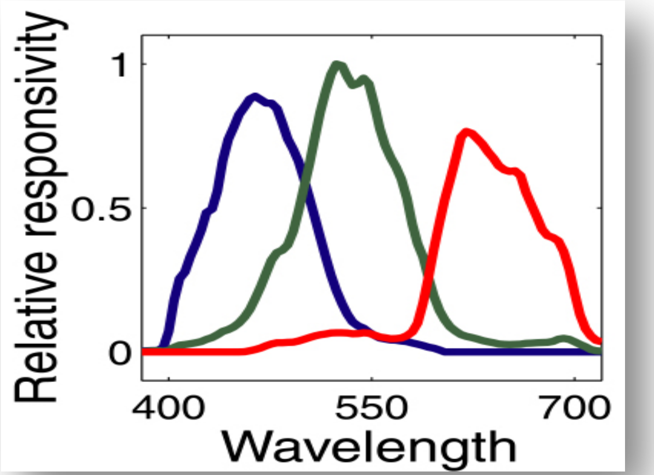
- Devices (cameras and displays) differ
- Illuminant changes influence camera data more than than these influence human perception



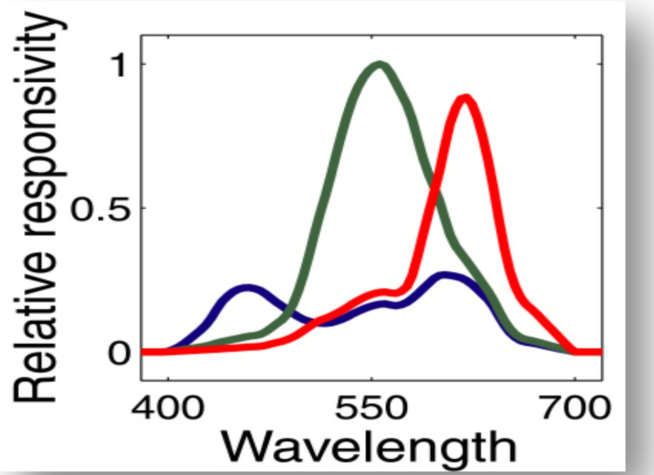
1. Sensor Correction

Same scene (same object, same light) captured by different cameras

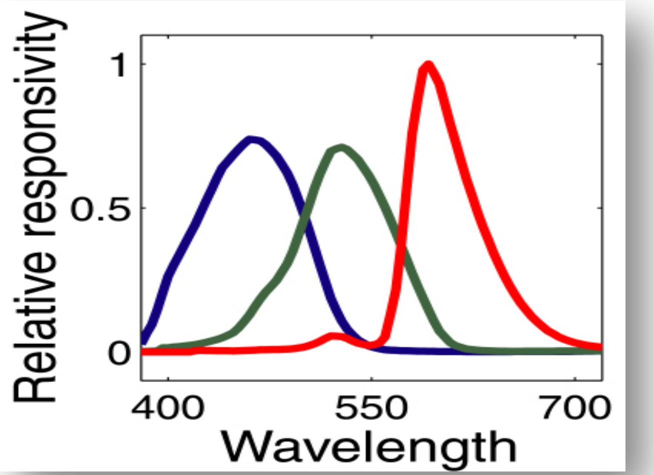
QImaging



Kodak



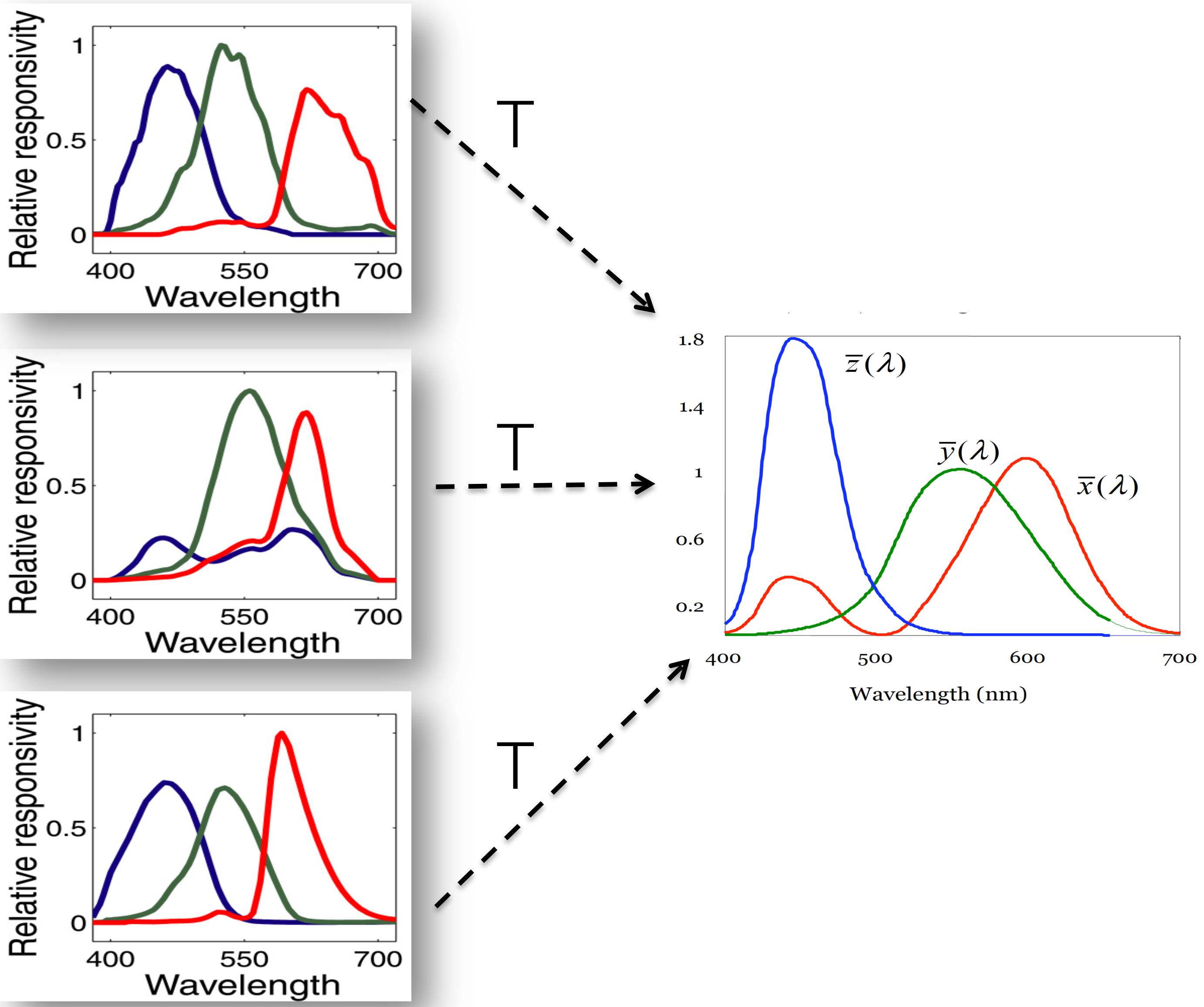
Nikon



1. Sensor Correction

Camera sensors are not XYZ-CMFs, and not even necessarily within a linear transform of the XYZ-CMFS

Sensor correction means deal with this problem.

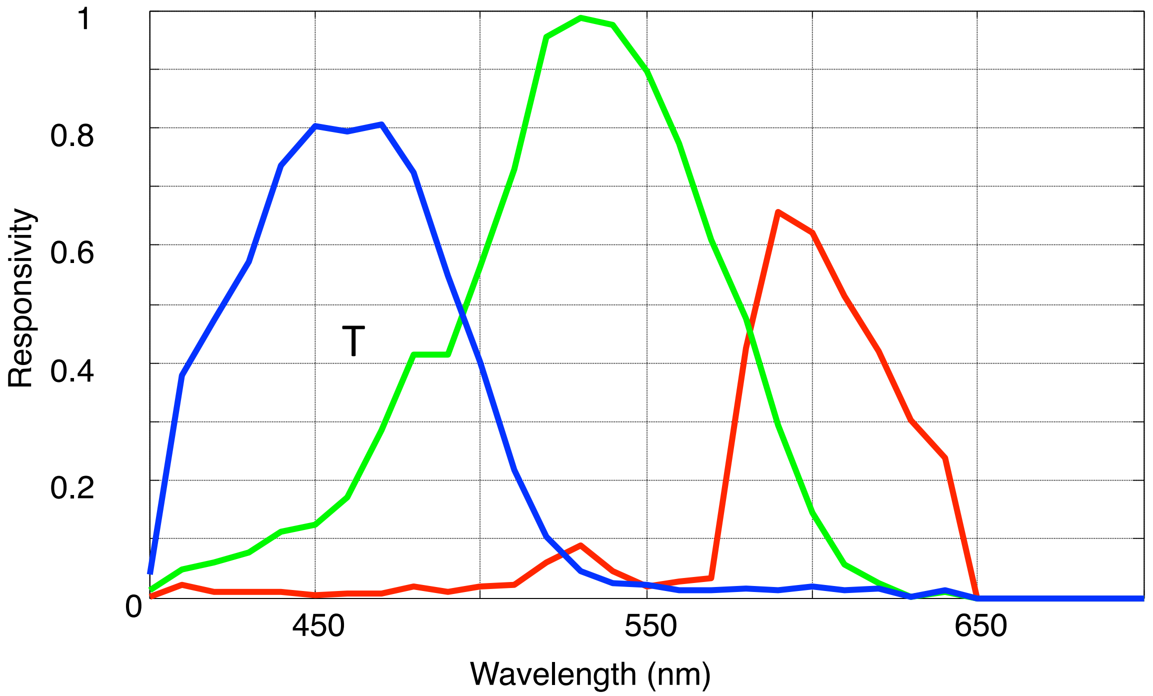


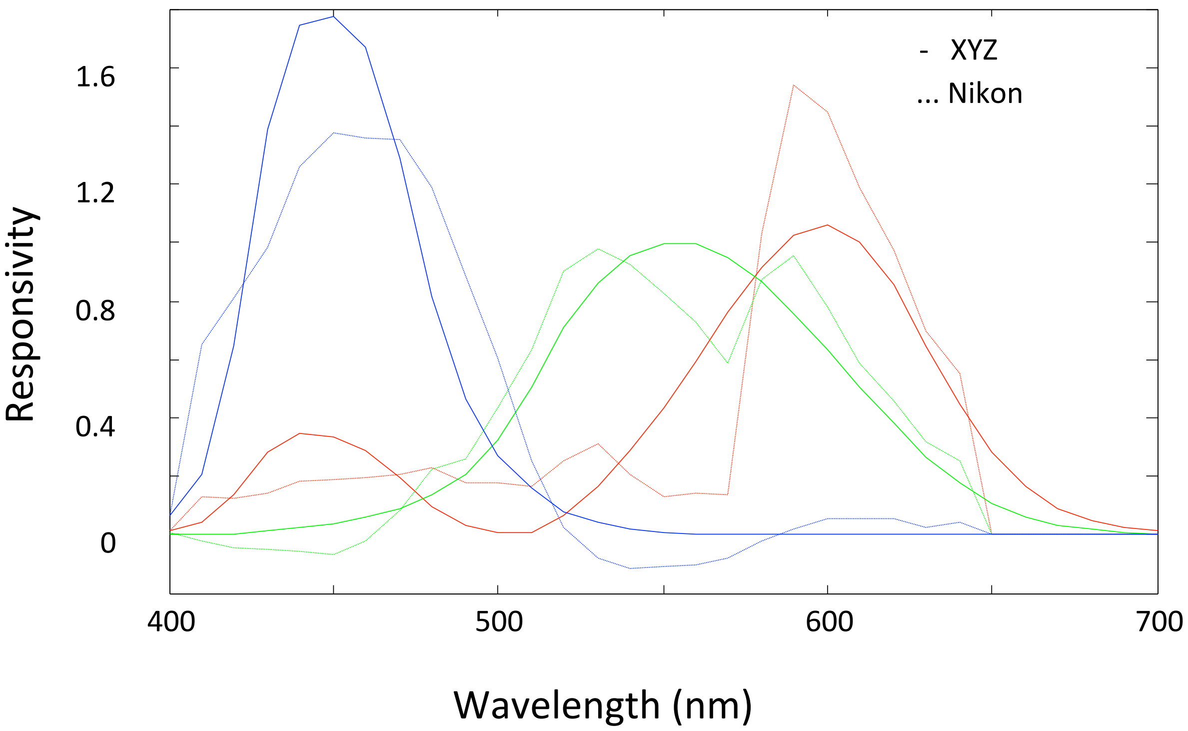
Sensor correction: RGB example

ISET: s_ipSensorConversion

$$xyz = T * sensor$$

2.2591	0.1328	0.1867
1.0159	0.9371	-0.2491
0.0156	-0.1201	1.7118



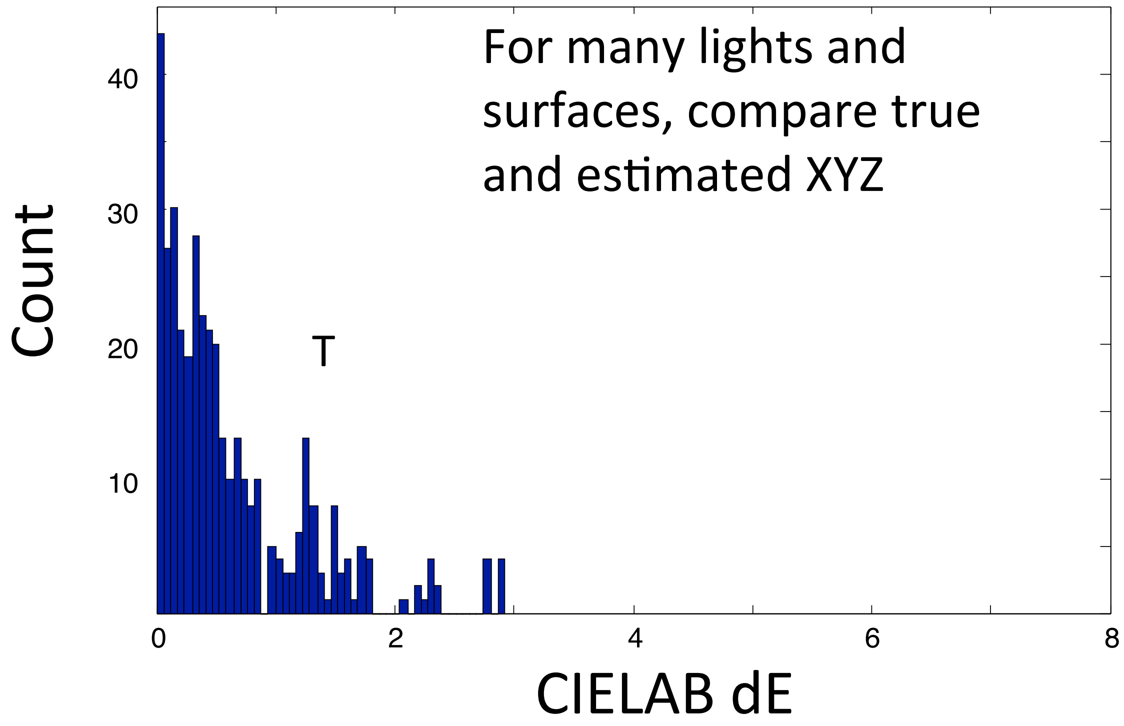


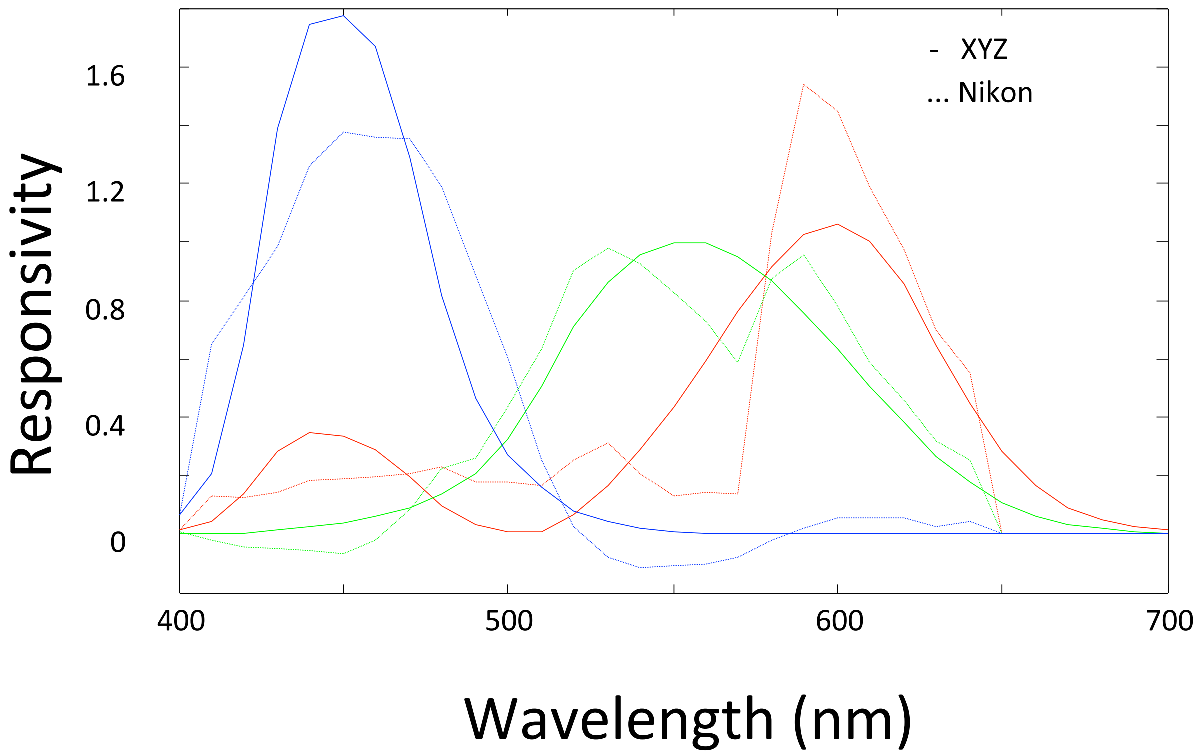
Sensor correction: RGB example

ISET: s_ipSensorConversion

$$xyz = T * sensor$$

2.2591	0.1328	0.1867
1.0159	0.9371	-0.2491
0.0156	-0.1201	1.7118



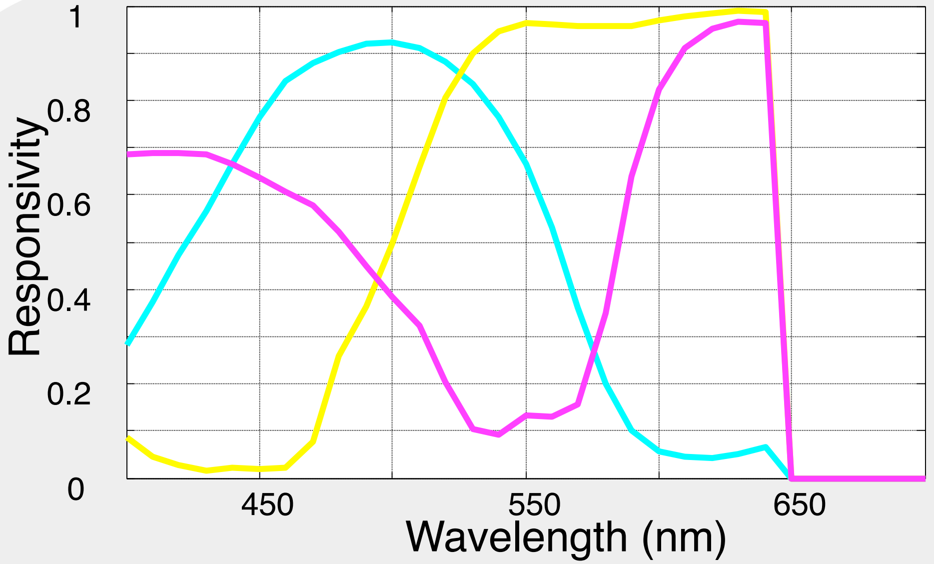


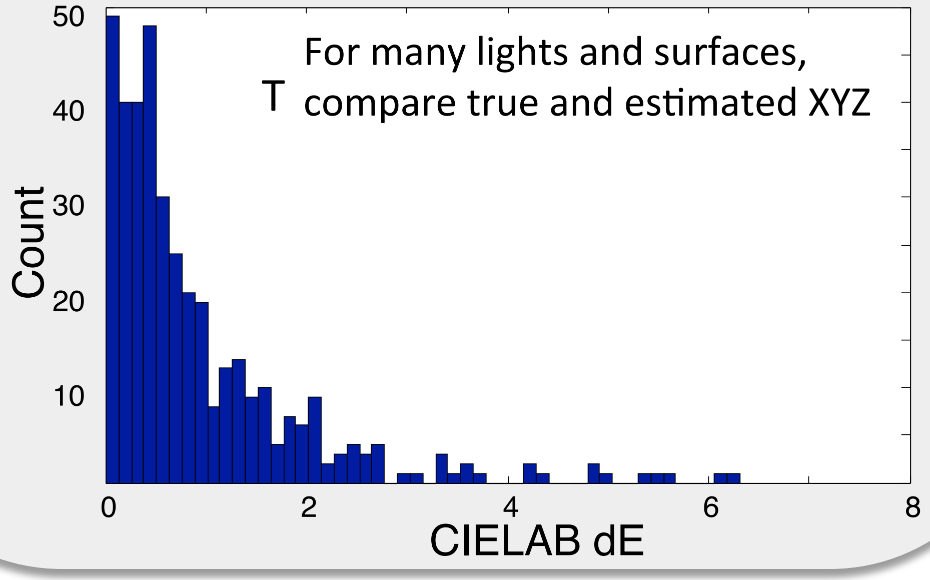
Sensor correction: CMY example

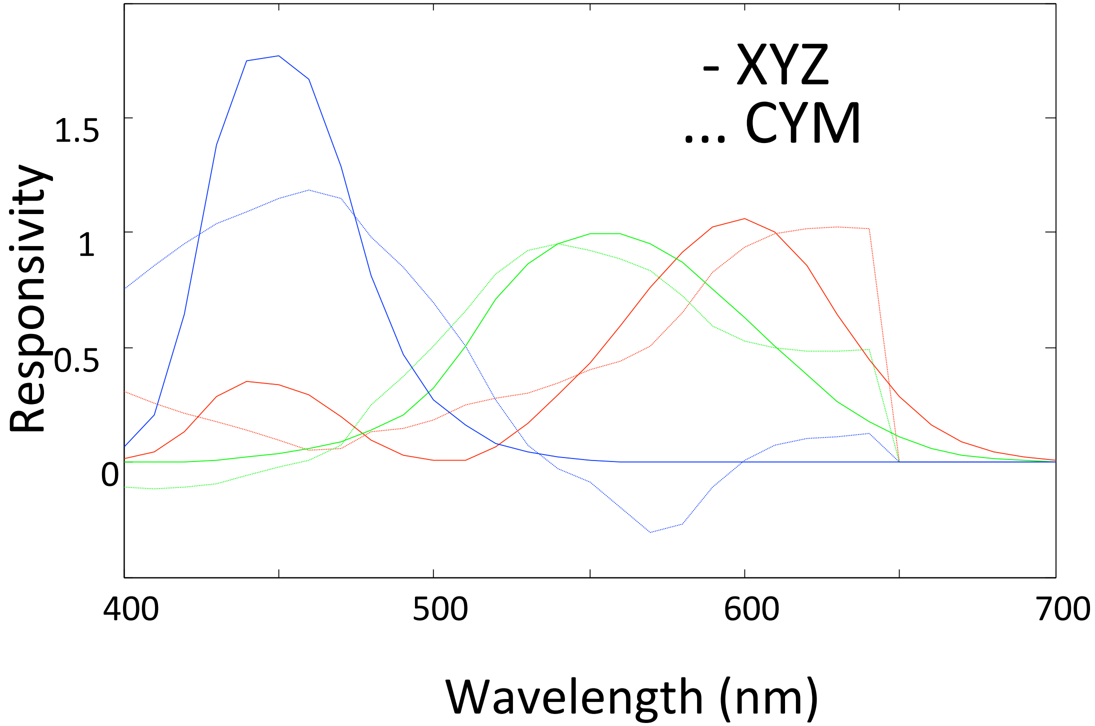
ISET: s_sensorCorrection

$$xyz = T * sensor$$

$$T = \begin{matrix} -0.2970 & 0.5812 & 0.4605 \\ 0.2581 & 0.8520 & -0.3963 \\ 0.8259 & -0.7436 & 0.8038 \end{matrix}$$

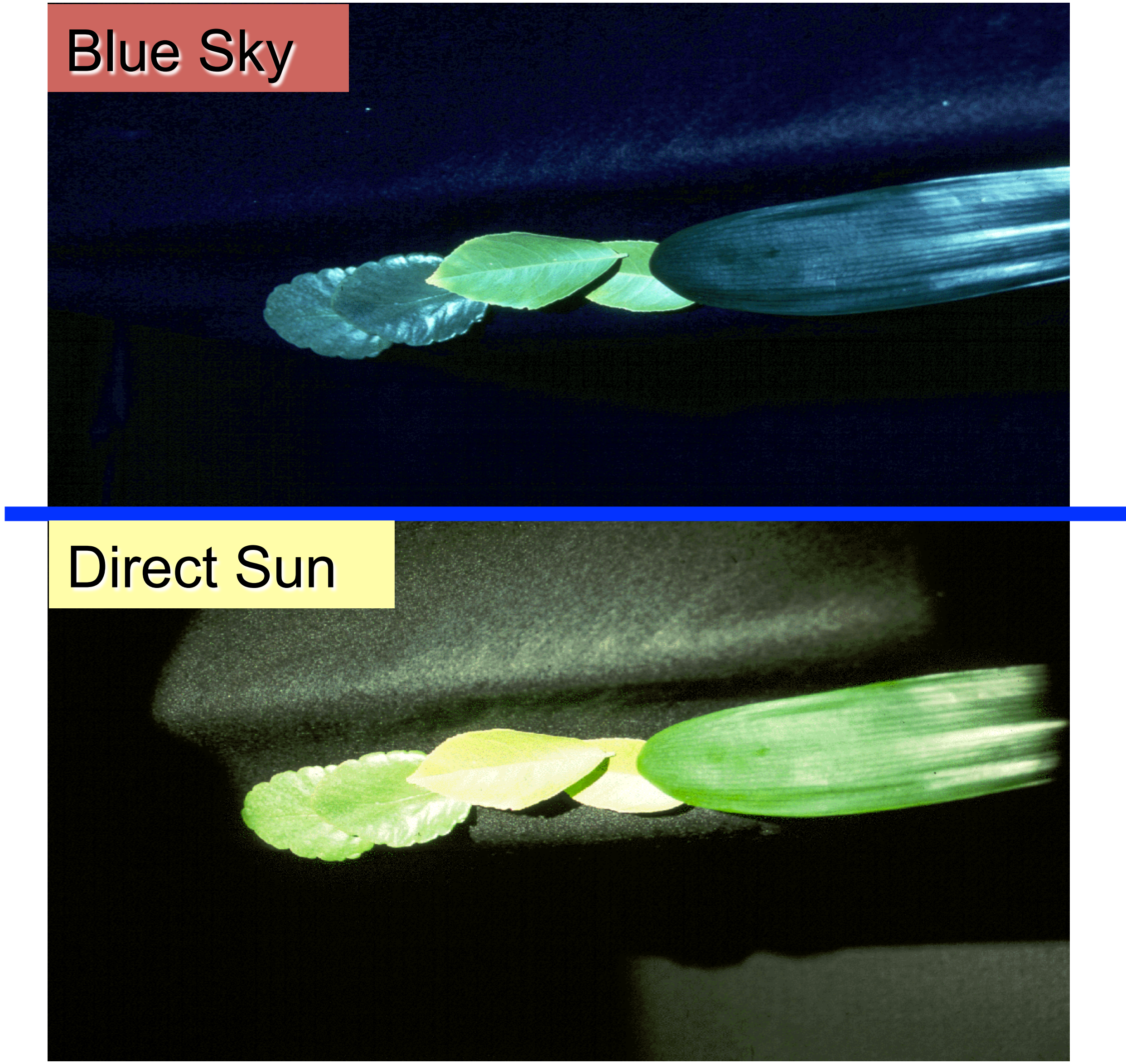






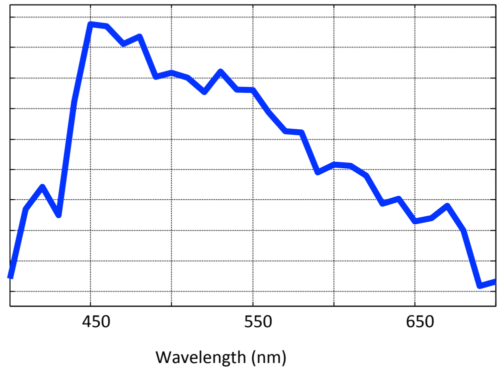
2. Illuminant correction

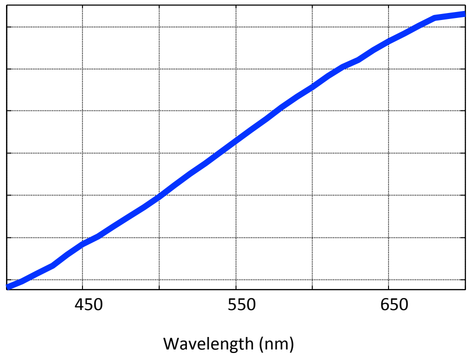
- Illuminant changes cause the scattered light to change
- The visual system adjusts to the illuminant (color constancy)
- If the pipeline doesn't adjust, the rendering has the wrong color balance
- Illuminant correction is a very important aspect of color balancing



Illuminant correction

- Illuminant changes cause the scattered light to change
- The visual system adjusts to the illuminant (color constancy)
- If the pipeline doesn't adjust, the color appears wrong
- Illuminant correction is important



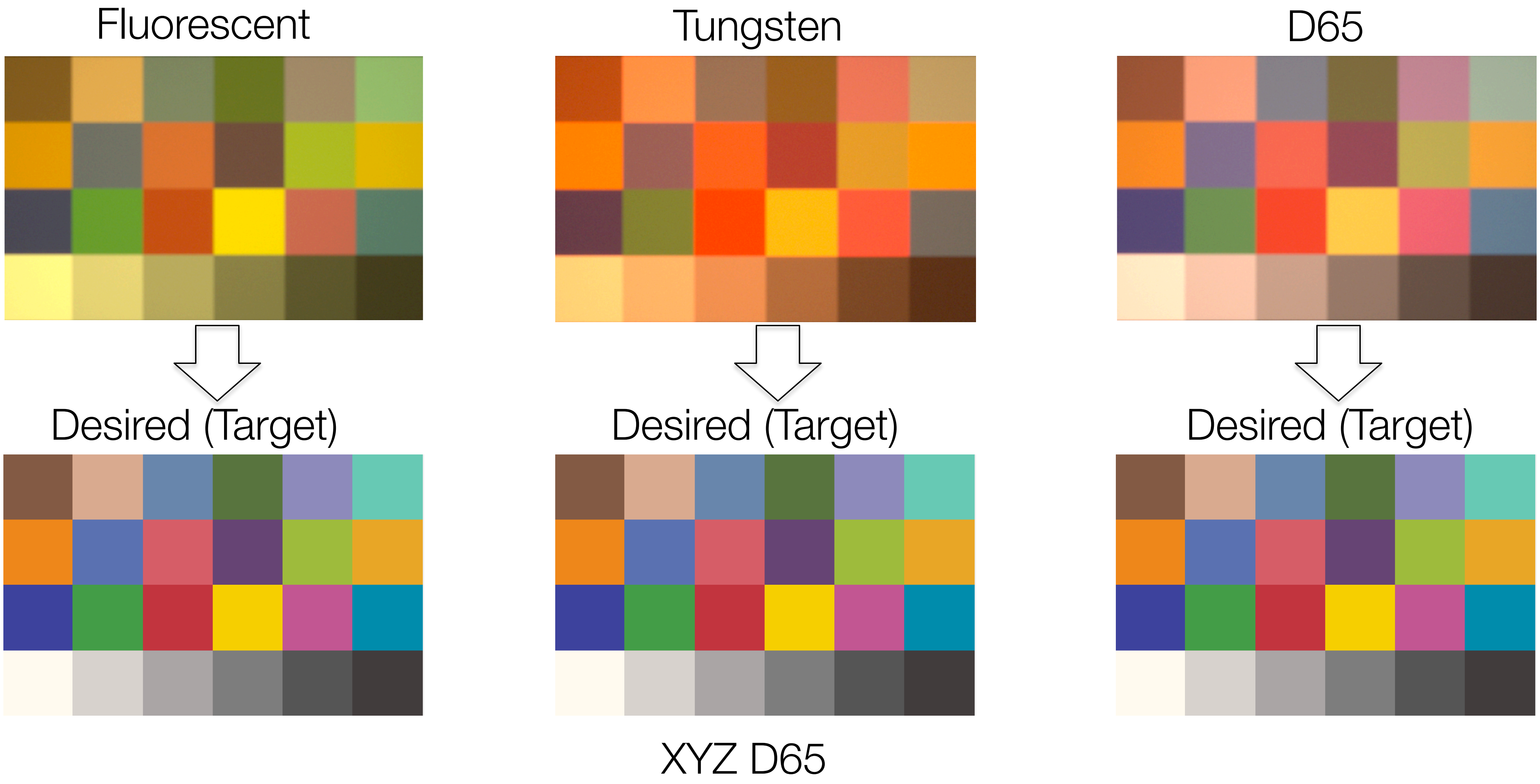


Illuminant Correction Strategy

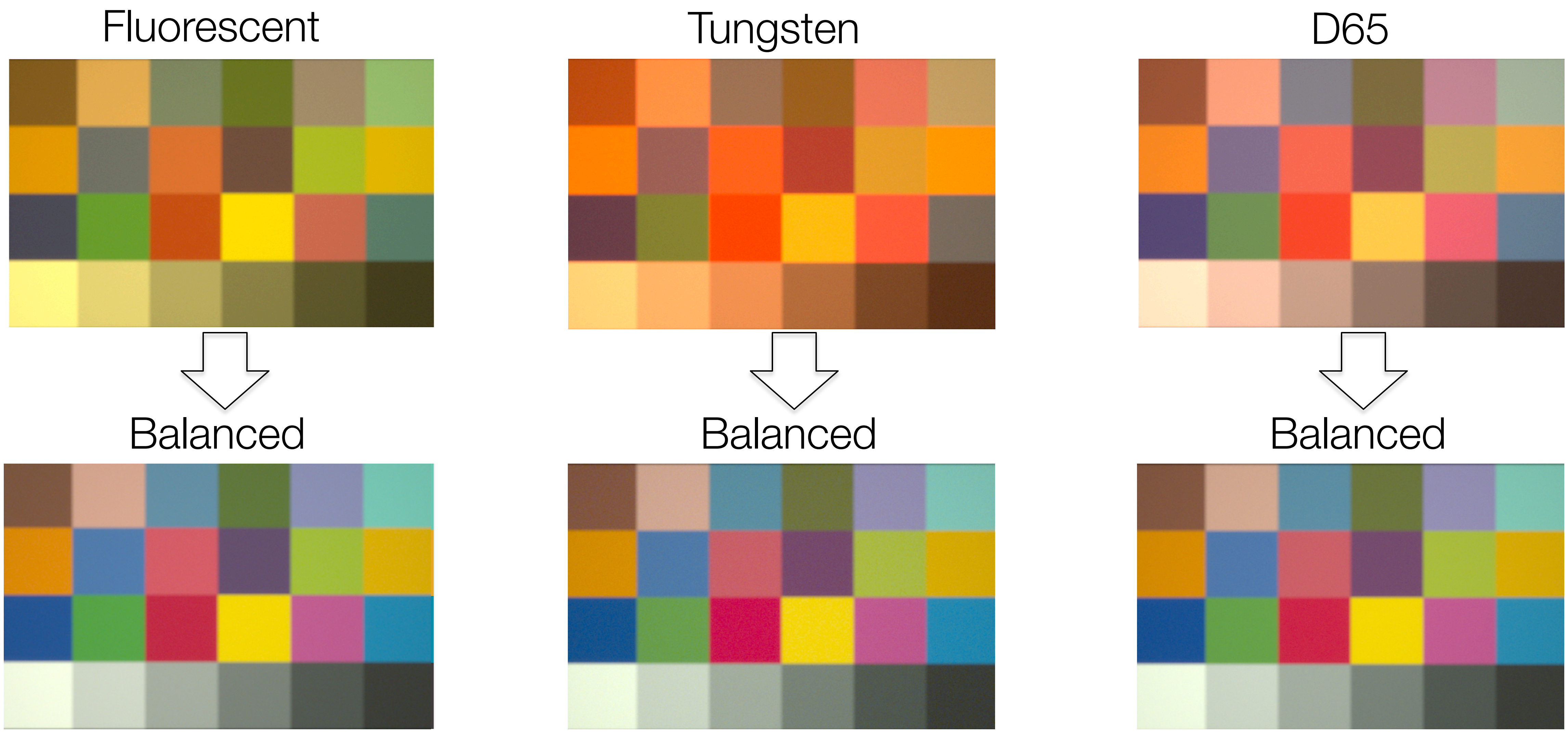
- Color constancy is not fully modeled or explained …
  - Von Kries : gain change in LMS sensor space
  - this is a question for vision science.

- Engineering solution
  - People typically like to see how objects look like under daylight
  - Estimate the illuminant
  - Find a 3x3 mapping from XYZ values under the estimated illuminant into XYZ values under daylight

Illuminant Correction



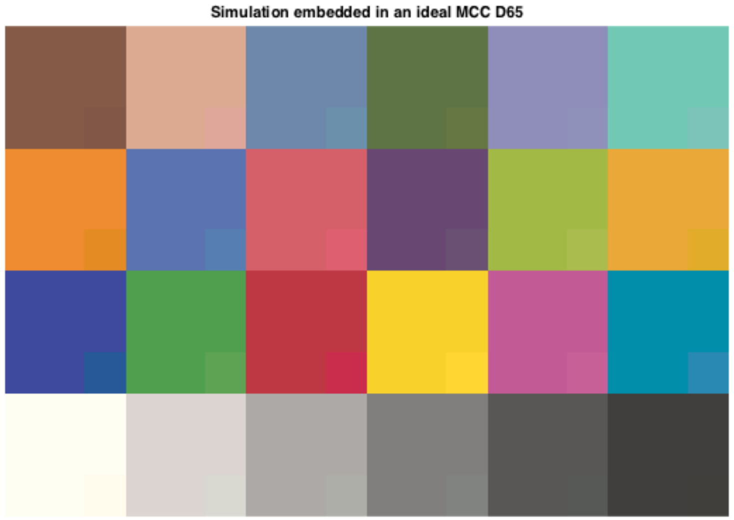
Illuminant Correction



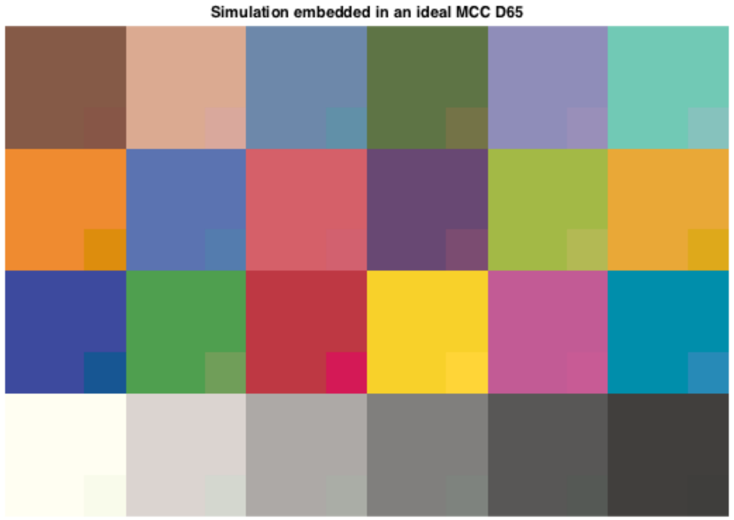
XYZ D65

Comparison of Desired and Rendered Colors

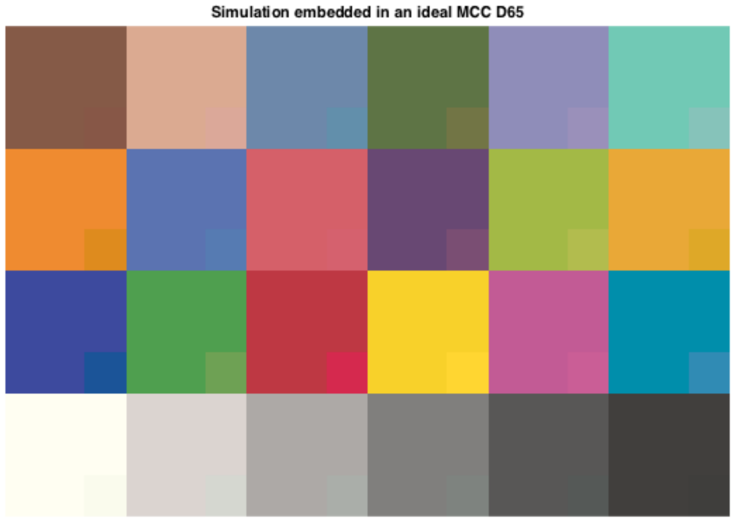
Fluorescent



Tungsten



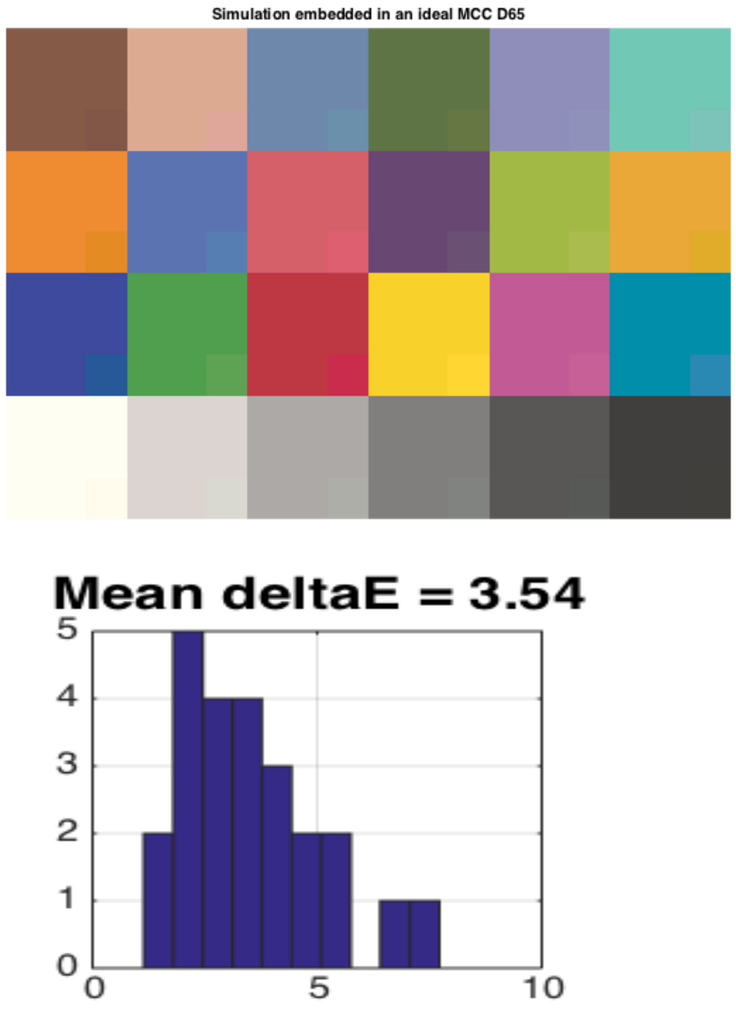
D65



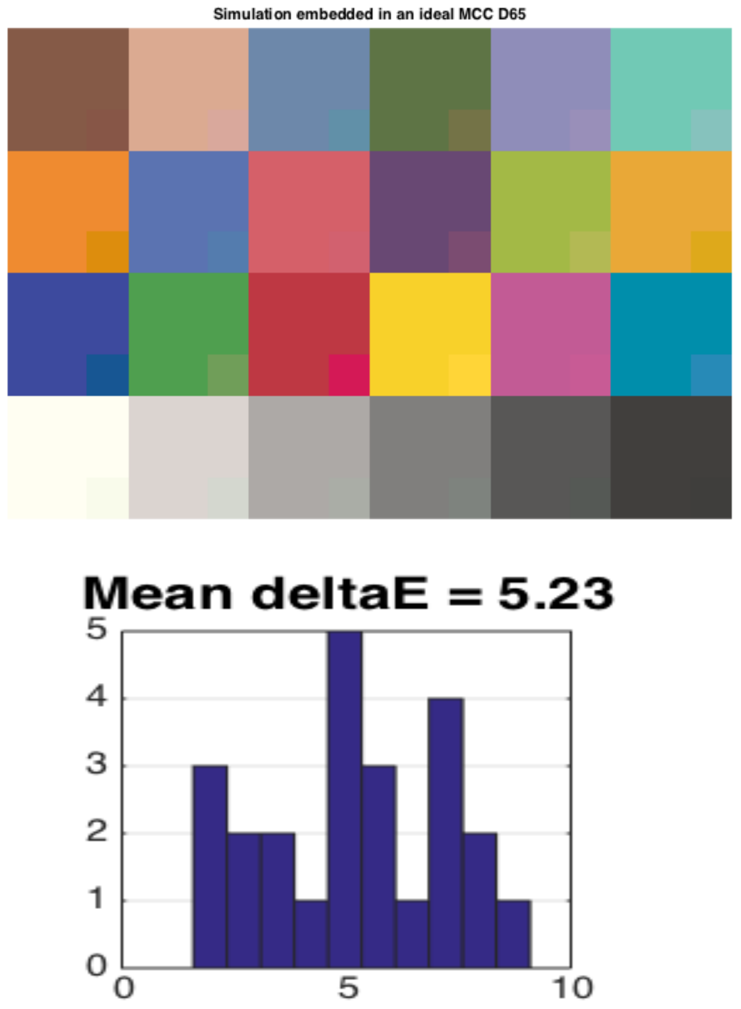
How would you measure the visible difference between the desired and rendered colors?

Comparison of Desired and Rendered Colors

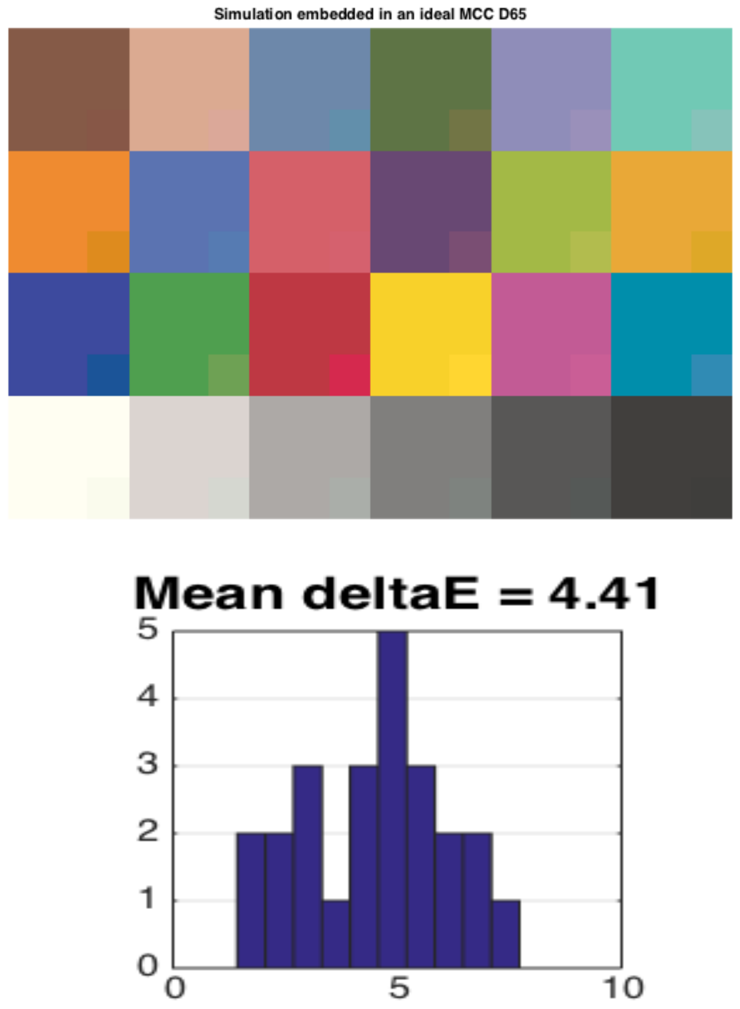
Fluorescent



Tungsten



D65



Color Management Strategy

1. Illuminant Estimation

2. Color Conversion
   - 3x3 matrix to convert sensor RGB values to a calibrated color space (e.g. XYZ)
   - Illuminant-sensor dependent transformation
   - Feng et al study that shows that conversion of RGB to XYZ produces better results

3. Illuminant transformation
   - 3x1 or 3x3 matrix to transform XYZ to XYZ_{D65}
   - Illuminant-sensor dependent transformation

4. Display Rendering
   - 3x3 matrix to convert XYZD65 to linear display RGB
   - Display dependent transformation
   - Apply gamma to convert linear RGB to DAC values

Step 1: Illuminant Estimation

- Most important aspect of color balancing
- Largest source of error



Step 1: Illuminant Estimation

- Most important aspect of color balancing
- Largest source of error

We will return to discuss methods for illuminant estimation

Step 2: Color Conversion

3x3 matrix to convert camera sensor RGB values to calibrated color space (e.g. XYZ)

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

↑

Matrix coefficients depend on sensor and illuminant

Step 3: Illuminant Transformation

3x3 matrix to transform XYZ_t to XYZ_{D65}

$$\begin{pmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix}$$

↑

Matrix coefficients depend on sensor and illuminant

Step 4: Display Rendering

– 3x3 matrix to convert XYZ_{D65} to linear display RGB

– Display dependent transformation

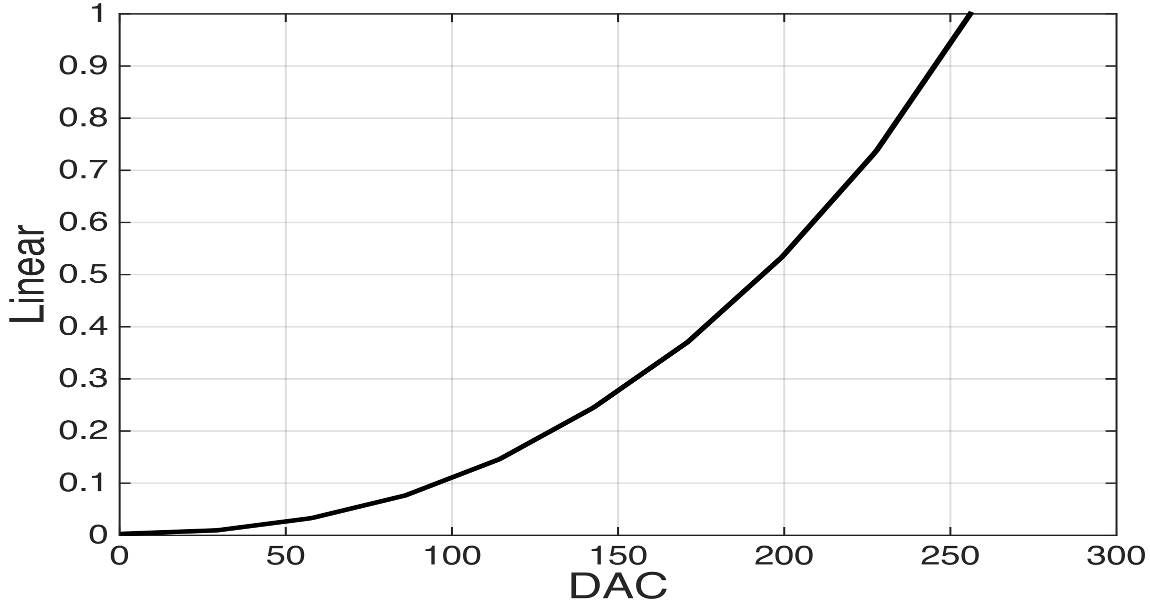
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{pmatrix}$$

↑

Matrix coefficients depend on display

Step 4: Display Rendering

- 3x3 matrix to convert XYZ_{D65} to linear display RGB
- Display dependent transformation
- Apply gamma to convert linear RGB to DAC values



Display Rendering

Common practice: Create a sRGB image and assume that each display has an sRGB display profile

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.1406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X_{\mathrm{D65}} \\ Y_{\mathrm{D65}} \\ Z_{\mathrm{D65}} \end{pmatrix}$$

Single 3x3 color transform

Combine

- sensor correction (fixed 3x3),
- illuminant transformation (customized for illuminant)
- display rendering (fixed 3x3)

into one 3x3 transform that converts camera RGB to display RGB

Illuminant-dependent

$$\text{Display} \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \text{Camera}$$

Matrix based color adjustment methods

Device-dependent diagonal

Display

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} s_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s_b \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Camera

Choose s_r and s_b, so that a neutral (gray) surface in the image is rendered as a neutral display output

You must make an educated guess about the camera RGB to a neutral surface. Example ideas:

- The average of the image is neutral
- The brightest elements of the image average to neutral
- Your idea goes here ....

Matrix based color adjustment methods

Device-independent diagonal

Display

Camera

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} & & \\ & M & \\ & & \end{pmatrix} \begin{pmatrix} s_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s_b \end{pmatrix} \begin{pmatrix} & & \\ & L & \\ & & \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Transform the device data into a calibrated space (cones, XYZ), (**L**).

Perform the diagonal transformation. again choosing s_r and s_b, so that a neutral (gray) surface in the image is rendered as a neutral signal in that space

Convert to the display representation from the calibrated space to display space (**M**)

Matrix based color adjustment methods

Device-independent linear

Display

$$\begin{pmatrix} R_1' & \cdots & R_N' \\ G_1' & \cdots & G_N' \\ B_1' & \cdots & B_N' \end{pmatrix} = \begin{pmatrix} & & \\ & C_E & \\ & & \end{pmatrix} \begin{pmatrix} R_1 & \cdots & R_N \\ G_1 & \cdots & G_N \\ B_1 & \cdots & B_N \end{pmatrix}$$

Camera

For each light condition, E, choose a matrix, C_E, that transforms multiple measurements to desirable display values.

Find these matrices for 30-50 likely lights.

The matrix C_E may be chosen by MSE or perceptual error minimization.

LUT based color adjustment methods

Display

$$\begin{pmatrix} R_1' & & R_N' \\ G_1' & \mathbf{L} & G_N' \\ B_1' & & B_N' \end{pmatrix} = \begin{array}{|c|} \hline \mathbf{E} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ \ldots \\ C_N \\ \hline \end{array} \begin{pmatrix} R_1 & & R_N \\ G_1 & \mathbf{L} & G_N \\ B_1 & & B_N \end{pmatrix}$$

Camera

Create lookup tables for each light condition, E, that map surfaces into a desired display for that light.

Such tables, which are essentially a collection of local linear transformations; they may be chosen by MSE or perceptual error minimization.

Illuminant correction summary

Estimate illuminant

E

Transform sensor data to calibrated color (XYZ)

Apply transformation for illuminant E

Convert to display space, adjusting for gamma and gamut

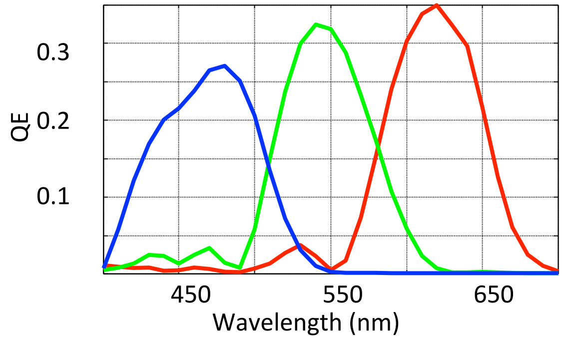
Sensor

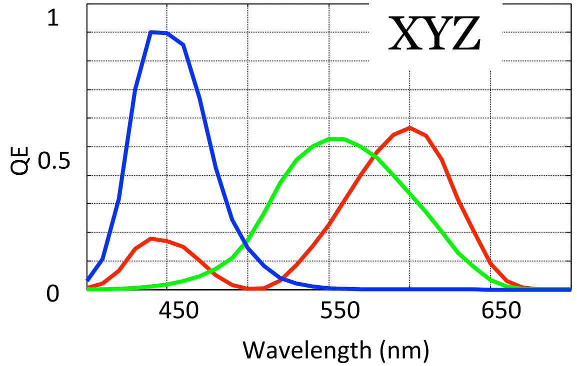
L

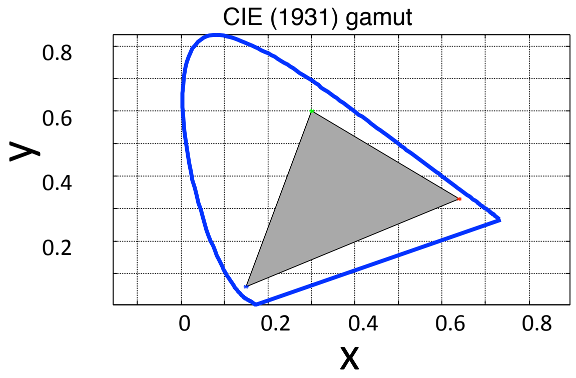
C_E

M

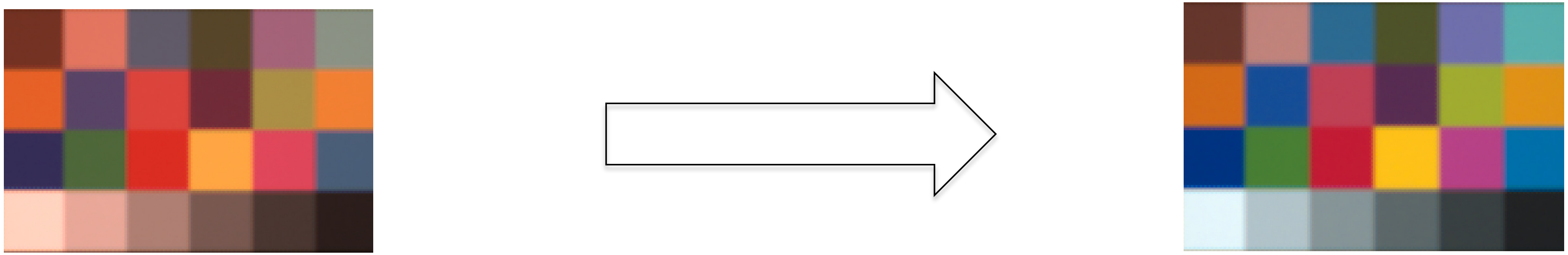
Display







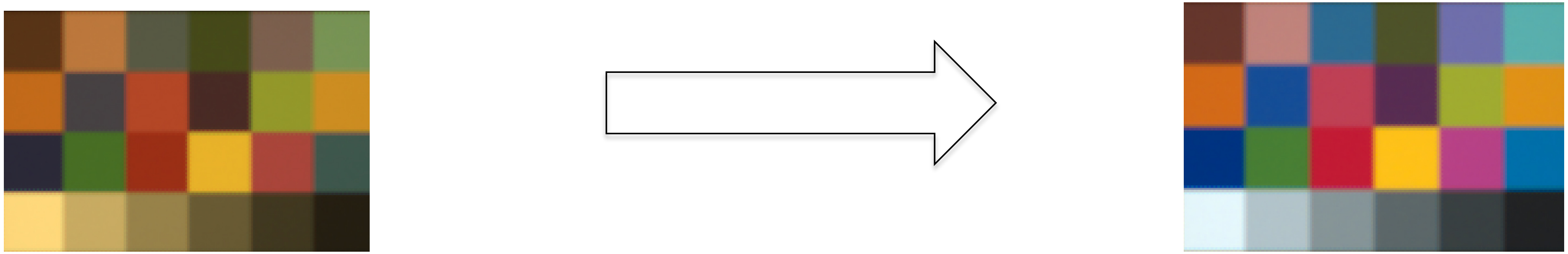
Engineering practice



1. Capture test surfaces under different lights

2. Linear regression to find 3x3 from camera to display RGB that minimizes XYZ

3. Report Delta E measure of color accuracy

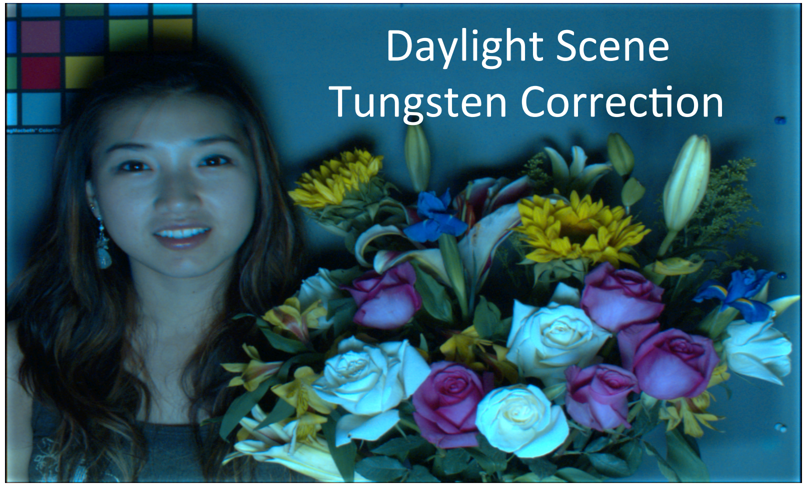


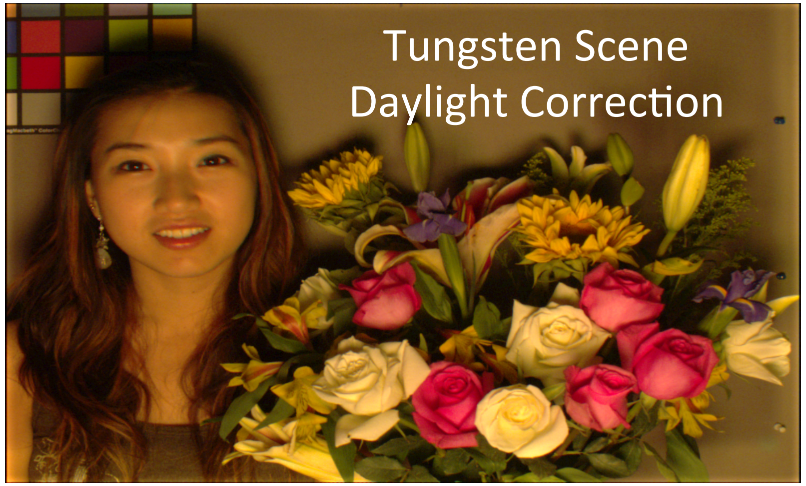
How do you estimate the Illuminant?

- Assumptions
  - Gray world
  - Image statistics

Illuminant Estimation

Bad estimate





Good estimate

Gray World

1. Illuminant Estimation
   - **Assume μ_R , μ_G , μ_B is the sensor response to the illuminant (gray surface)**
2. Color Conversion
   - Often times there is no conversion and correction is done in sensor space
   - Feng et al study shows that conversion to XYZ produces better results
3. Illuminant Transformation

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1/\mu_R & 0 & 0 \\ 0 & 1/\mu_G & 0 \\ 0 & 0 & 1/\mu_B \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

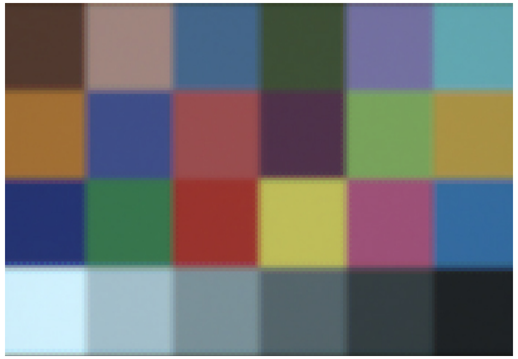
- Display Rendering
  - 3x3 matrix to convert corrected RGB to display RGB
  - Often times there is no conversion to display space (this can cause problems)
  - Apply display gamma

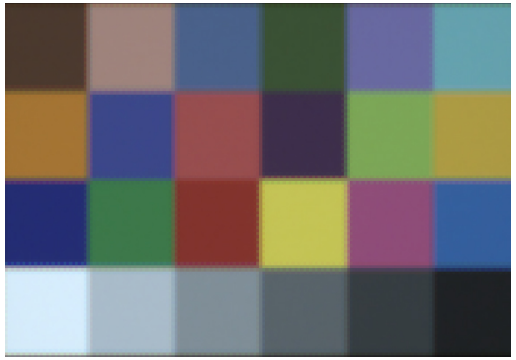
Gray World in Sensor and XYZ Space

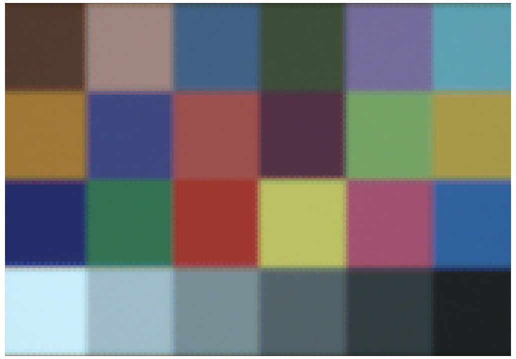
Color balancing in XYZ space produces better results

D65 | Fluorescent | Tungsten

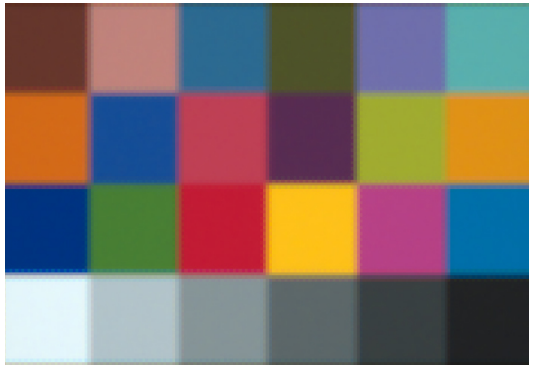
Gray World in Sensor Space

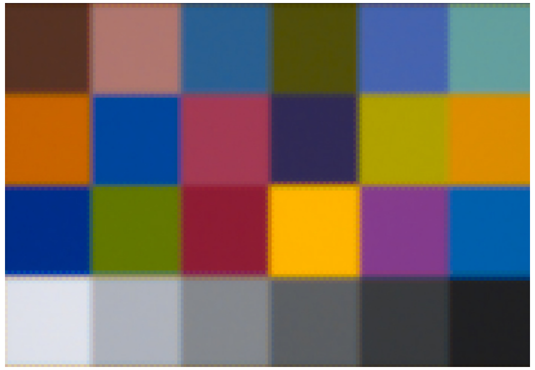


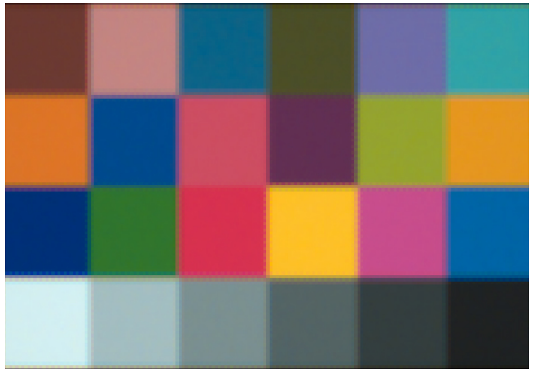




Gray World in XYZ Space







[Preferred Color Spaces for White Balancing]
F. Xiao, J. Farrell, J. DiCarlo and B. Wandell (2003).
In Proceedings of the SPIE Electronic Imaging '2003 Conference, Vol. 5017, Santa Clara, CA, January 2003 , San Jose

Why does Gray World work at all?

- Von Kries color adaptation
  - Gain change of LMS

$$\begin{pmatrix} L' \\ M' \\ S' \end{pmatrix} = \begin{pmatrix} 1/\mu_R & 0 & 0 \\ 0 & 1/\mu_G & 0 \\ 0 & 0 & 1/\mu_B \end{pmatrix} \begin{pmatrix} L \\ M \\ S \end{pmatrix}$$

1853-1928

- Illuminant correction advice
  Best Information In high intensity – black is black

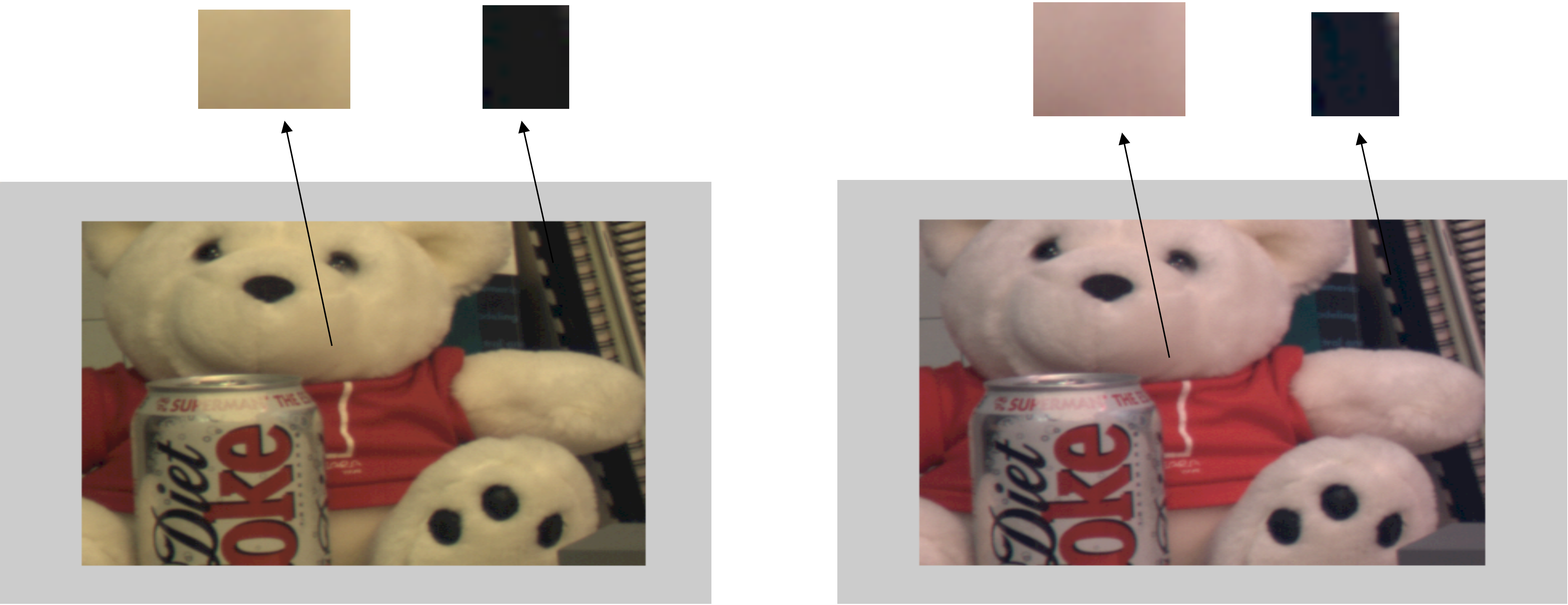
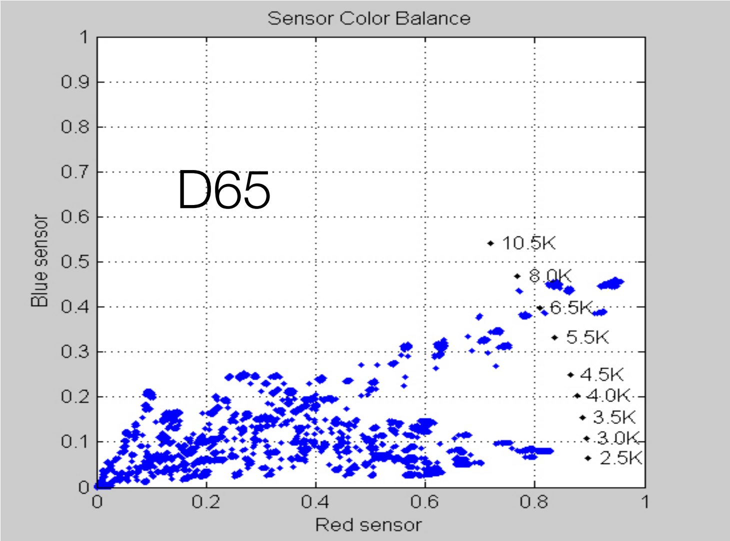
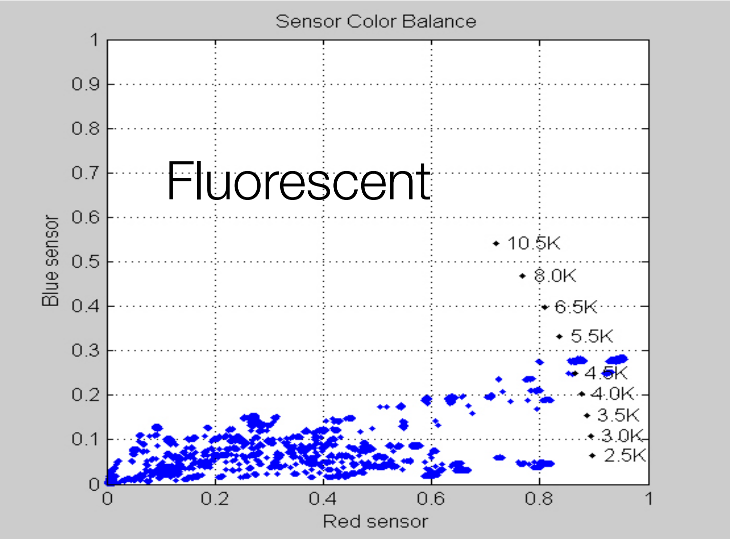


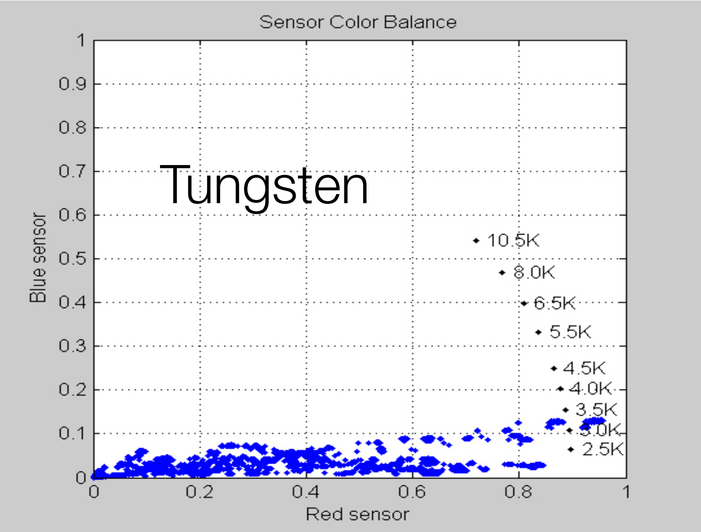
Image Statistics

1. Classify illuminant by ratio of red/blue sensor values

2. Select appropriate stored color balancing matrix
(combines color conversion, illuminant transformation and display rendering)







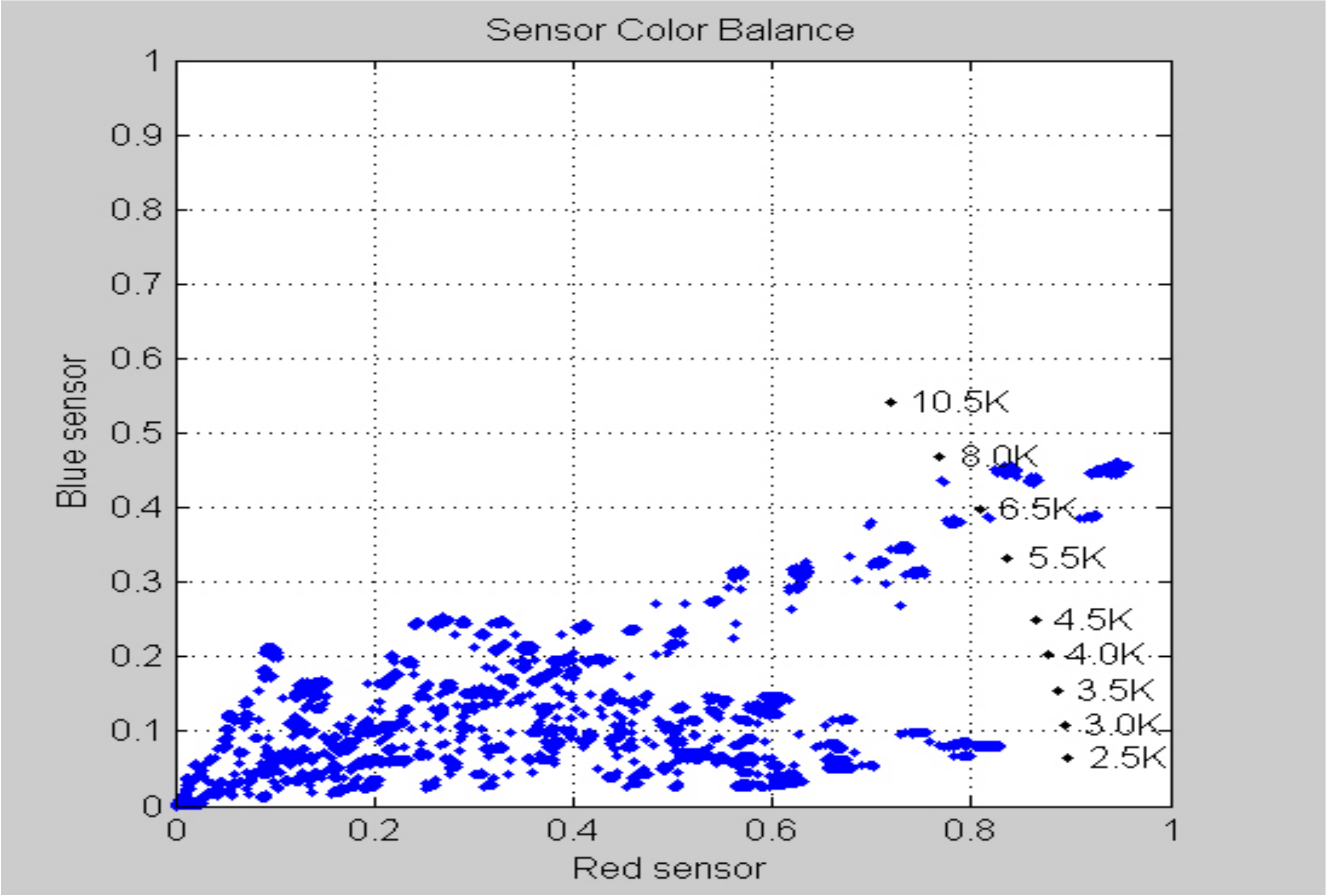
$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R_{D65} \\ G_{D65} \\ B_{D65} \end{pmatrix}$$

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R_T \\ G_T \\ B_T \end{pmatrix}$$

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R_F \\ G_F \\ B_F \end{pmatrix}$$

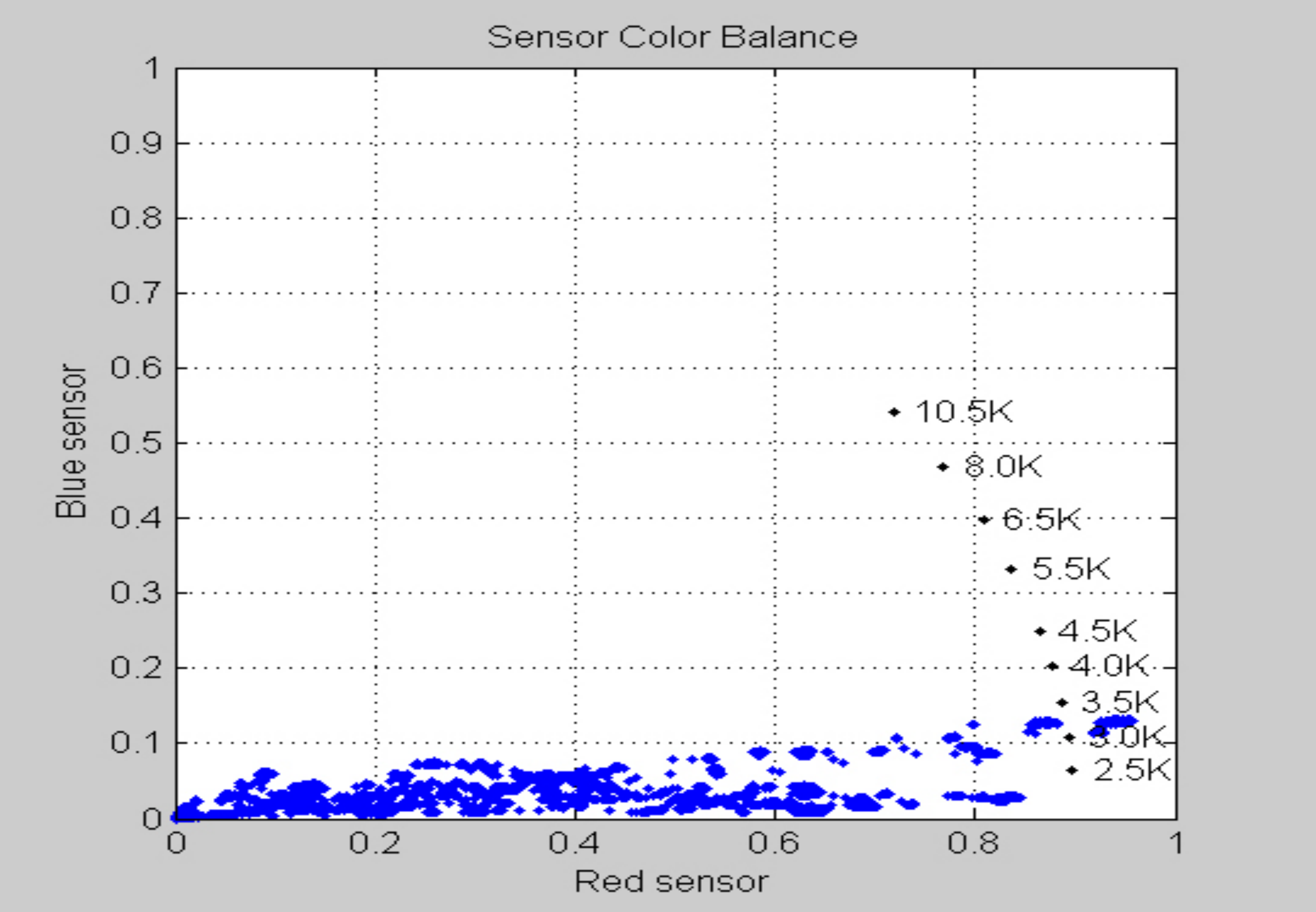
R/B Illuminant Signature

D65 Illuminant



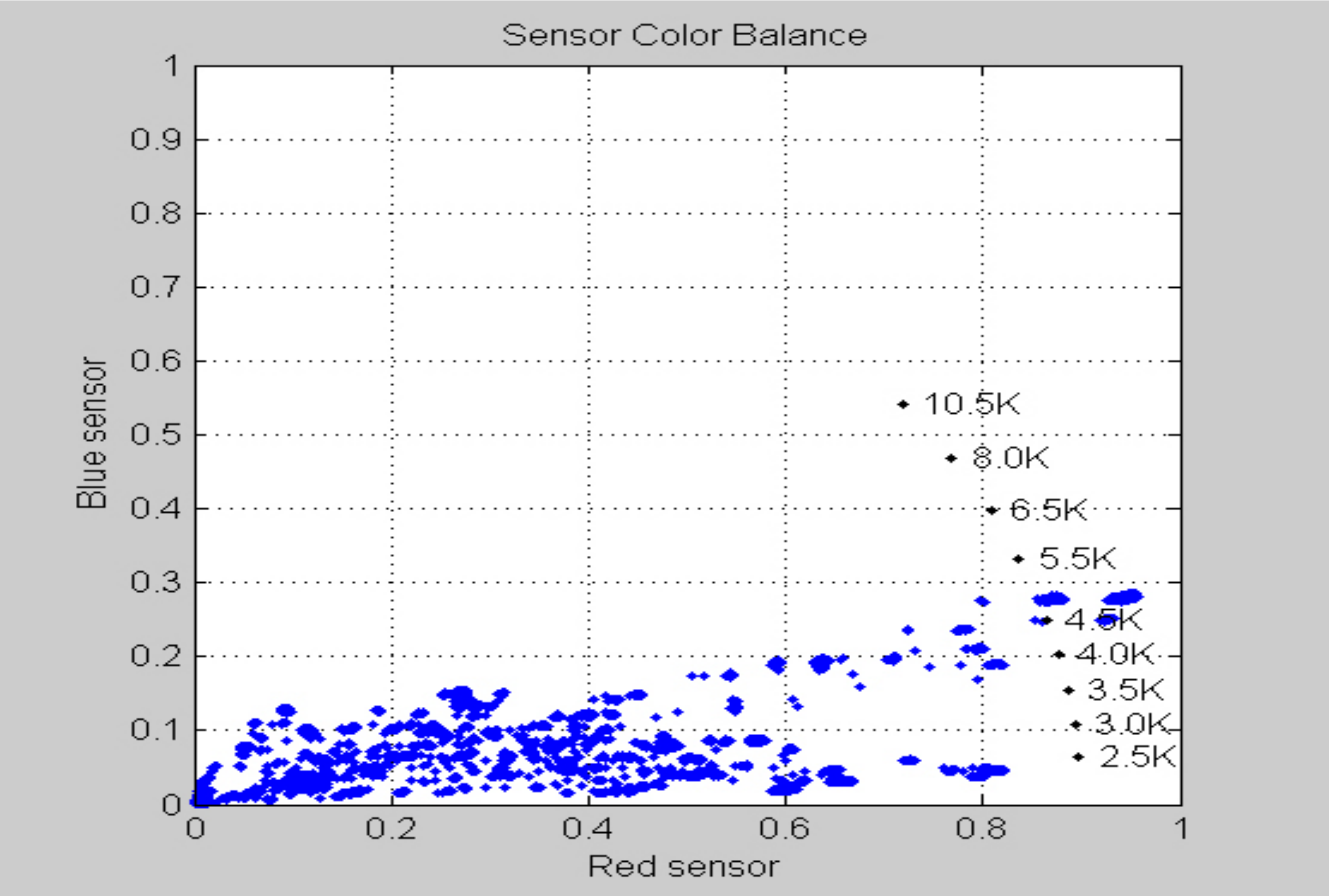
R/B Illuminant Signature

Tungsten Illuminant

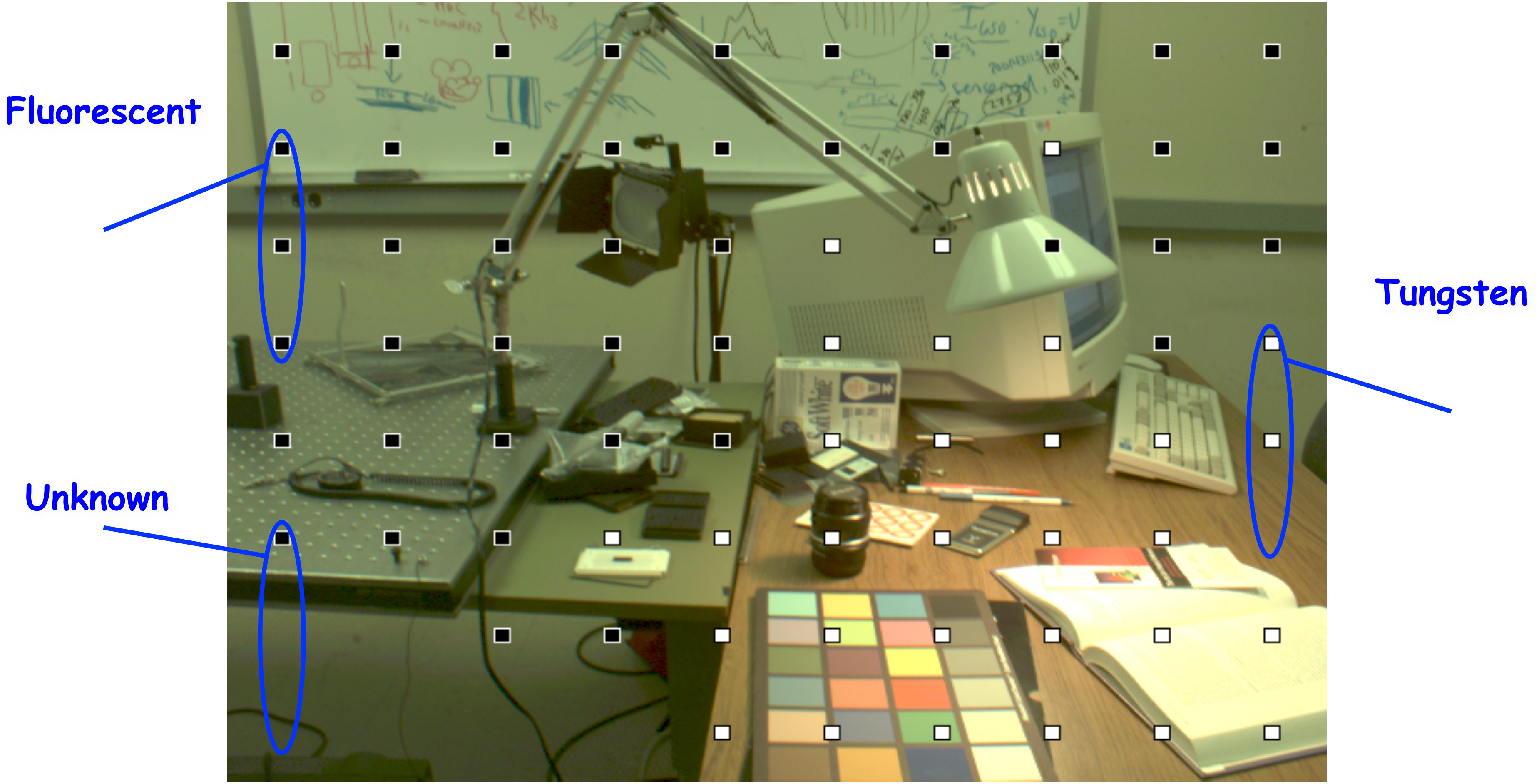


R/B Illuminant Signature

Fluorescent Illuminant

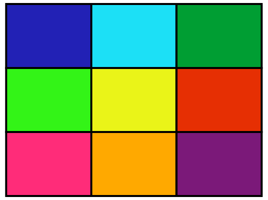


Non-uniform illumination



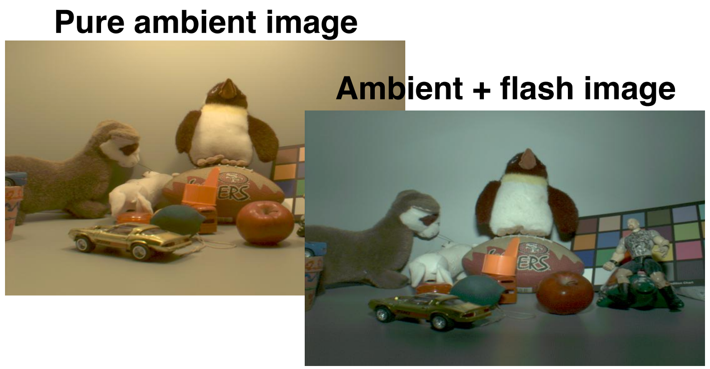
Hardware assist to Illuminant Estimation

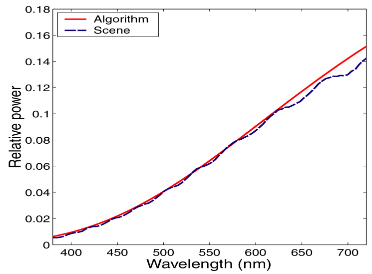
- Spectrophotometric sensor

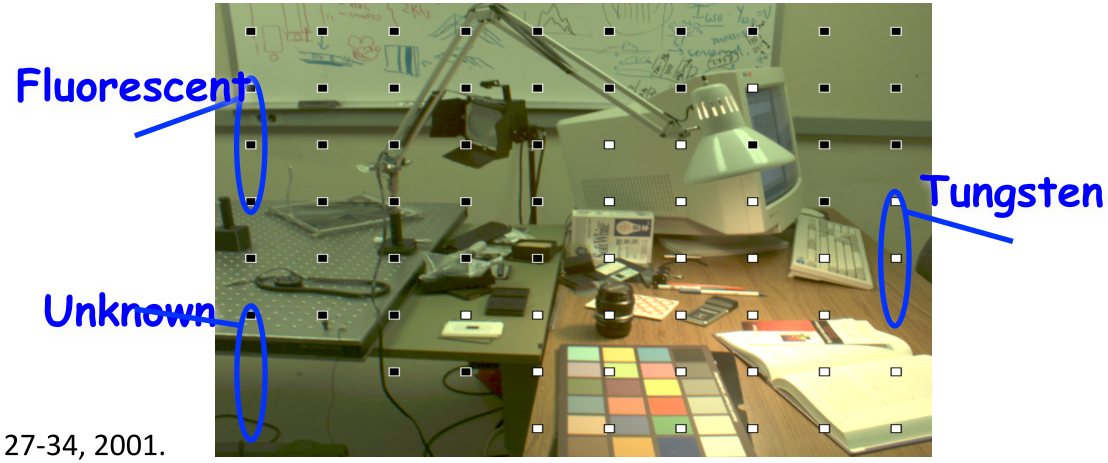


- Active Illumination Method

1. Two images: scene illuminant and (scene+ flash)
2. scene image – (scene+ flash) = pure flash image
3. Use pure flash image (known illuminant) to estimate surface reflectances
4. Use surfaces to estimate scene illuminant at different locations in the image







Illuminating Illumination.
Jeffrey M. DiCarlo, Feng Xiao and Brian A Wandell, *Ninth Color Imaging Conference*, pgs. 27-34, 2001.
(see related work from Georg Petschnigg and colleagues at Microsoft).

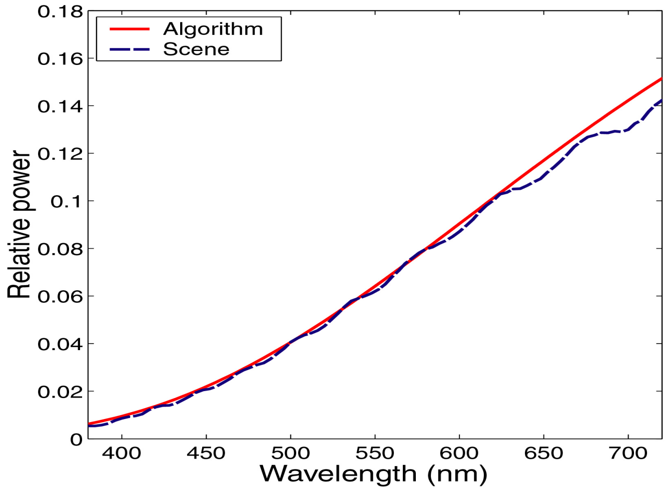
Hardware assist to Illuminant Estimation

Active Illumination Method

1. Two images: scene illuminant and (scene+ flash)
2. scene image – (scene+ flash) = pure flash image
3. Use pure flash image (known illuminant) to estimate surface reflectances
4. Use surfaces to estimate scene illuminant at different locations in the image

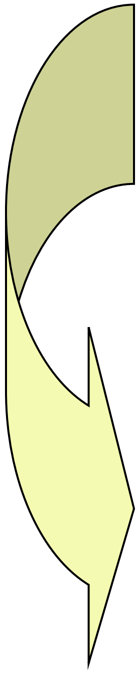
Pure ambient image

Ambient + flash image



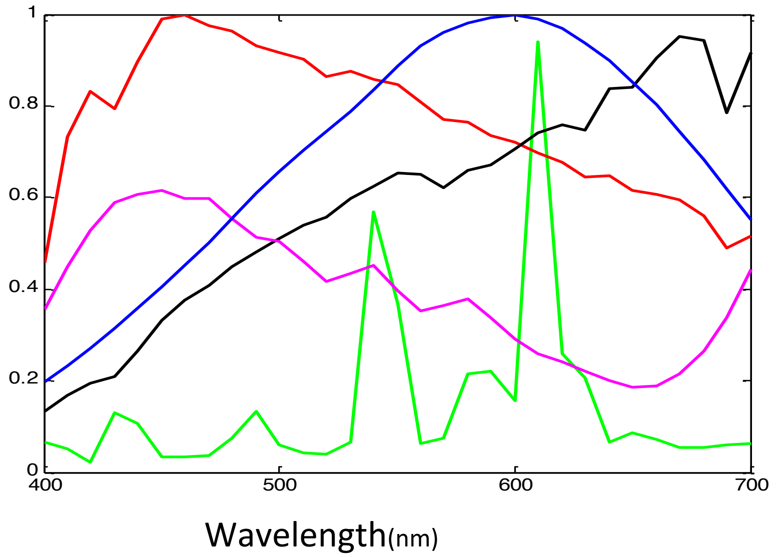
[Illuminating Illumination](#).
Jeffrey M. DiCarlo, Feng Xiao and Brian A Wandell, *Ninth Color Imaging Conference*, pgs. 27-34, 2001.
(see [related work](#) from Georg Petschnigg and colleagues at Microsoft).

Sources of Error



Illuminant estimation

Illuminant transformation



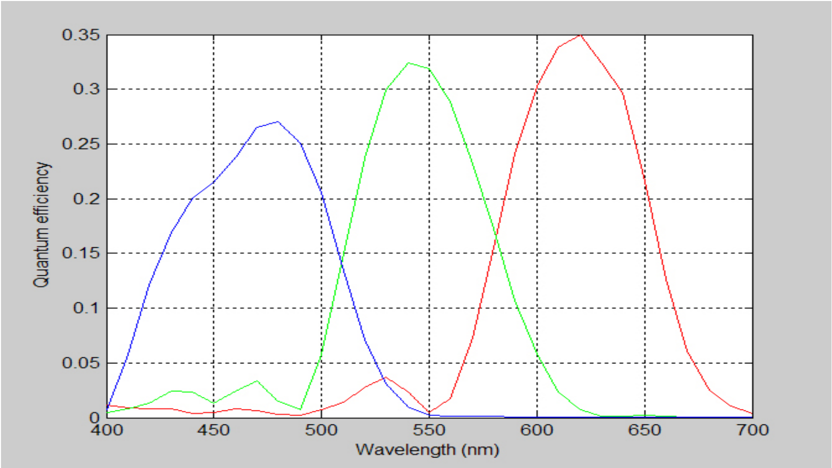
Sources of Error

1. Illuminant estimation

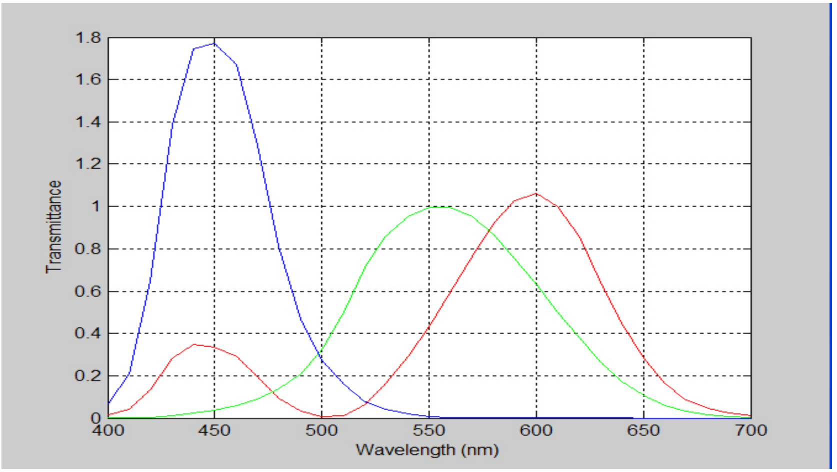
2. Conversion to calibrated color space
   - sensors are not XYZ

3. Illuminant transformation

Sensors



XYZ



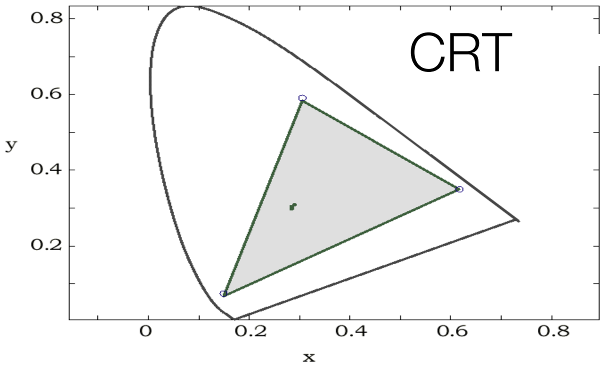
Sources of Error

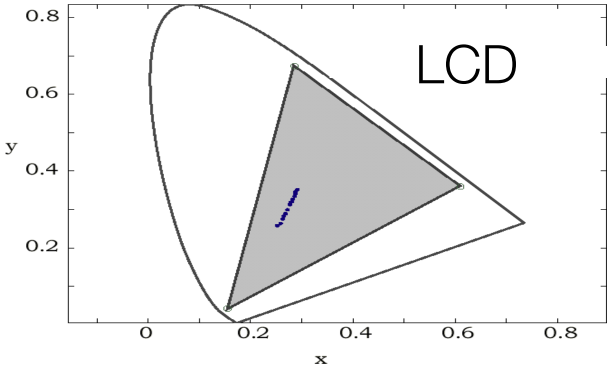
1. Illuminant estimation

2. Conversion to calibrated color space
   – sensors are not XYZ

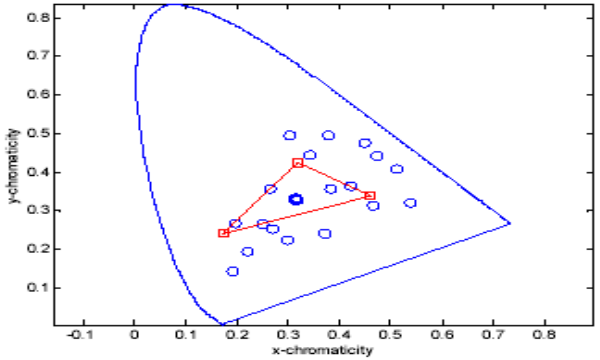
3. Illuminant transformation

4. Display gamut

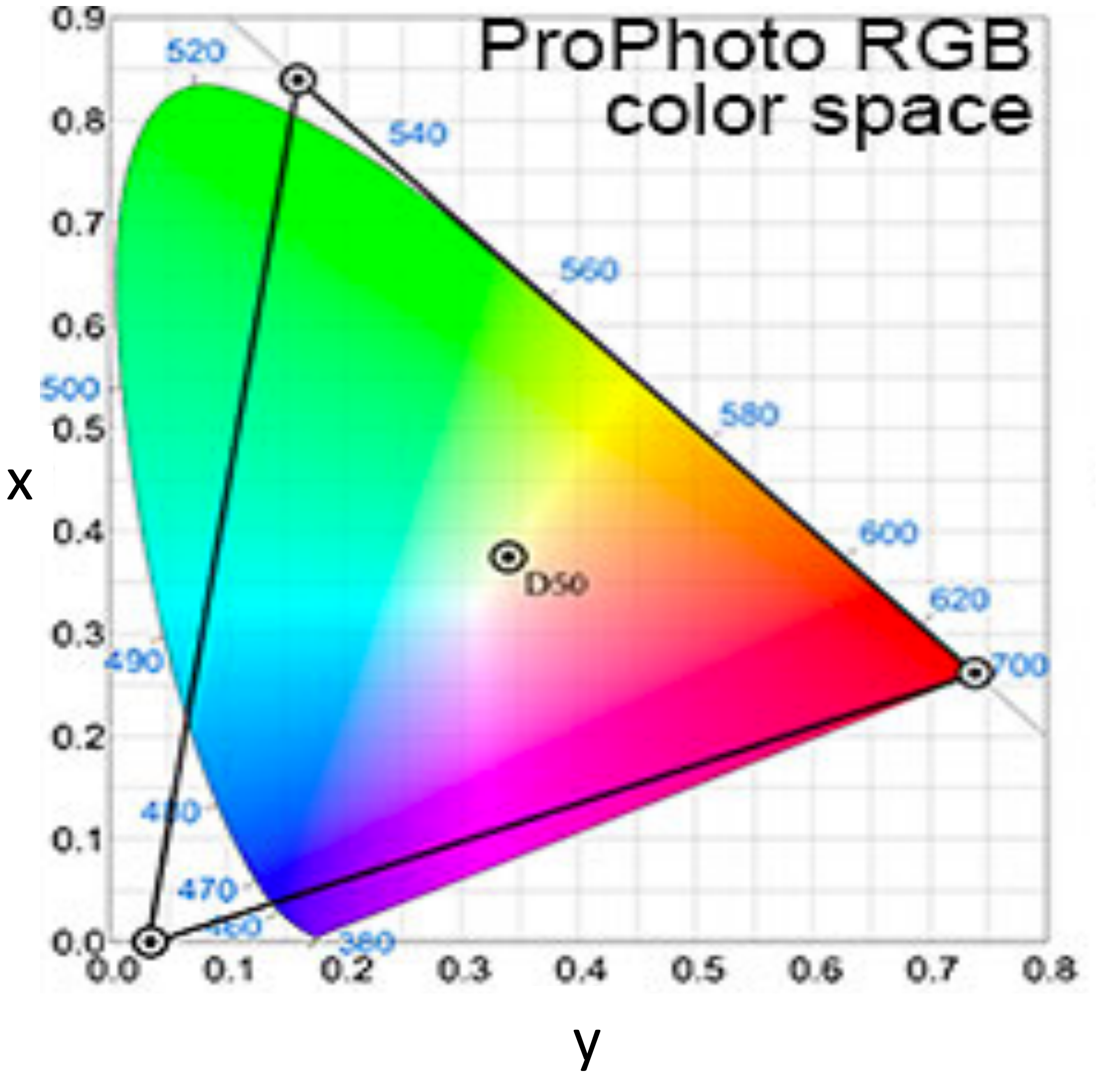




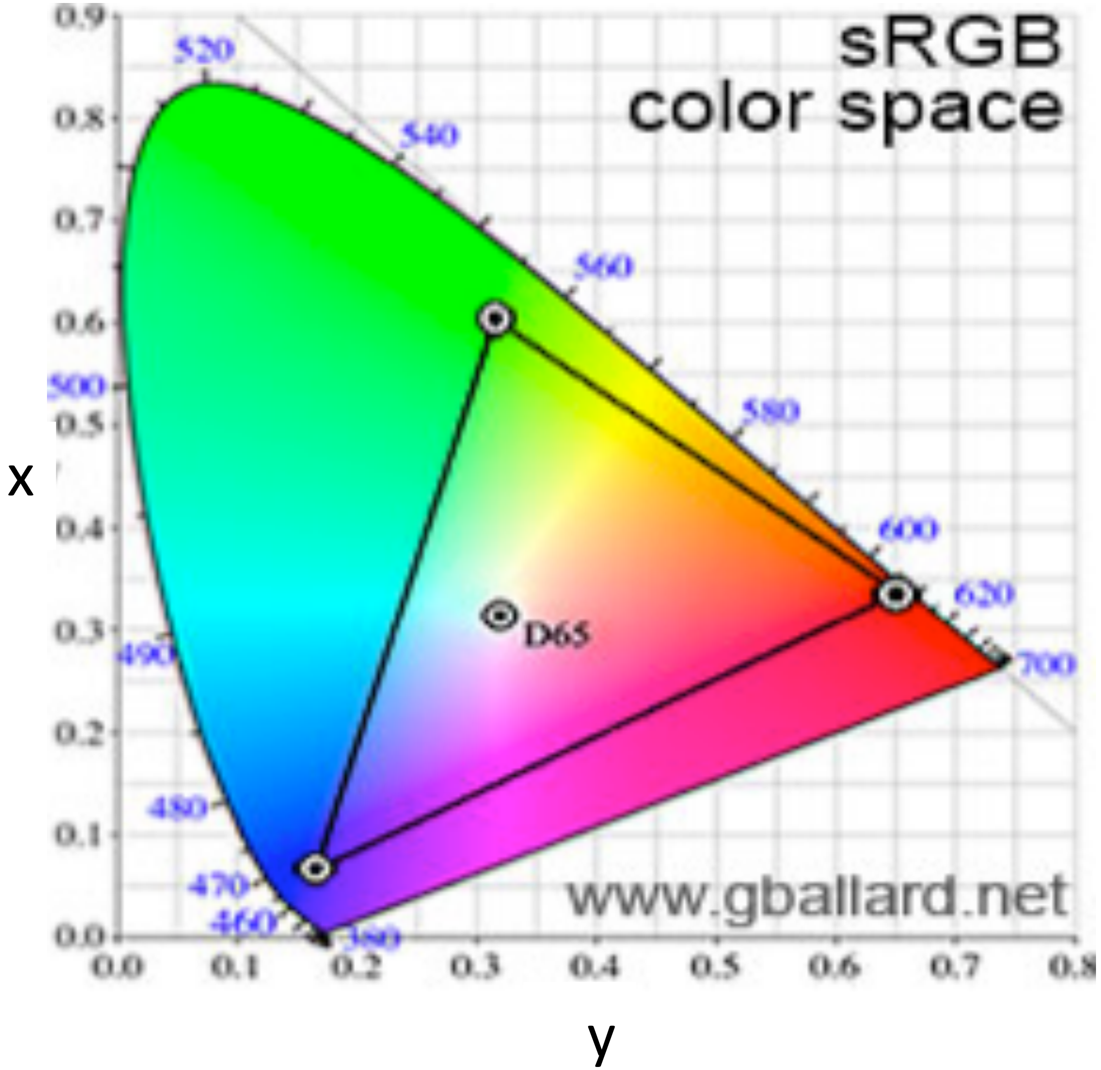
Cell phone display



Cameras can capture in “non-visible” regions of the electromagnetic spectrum

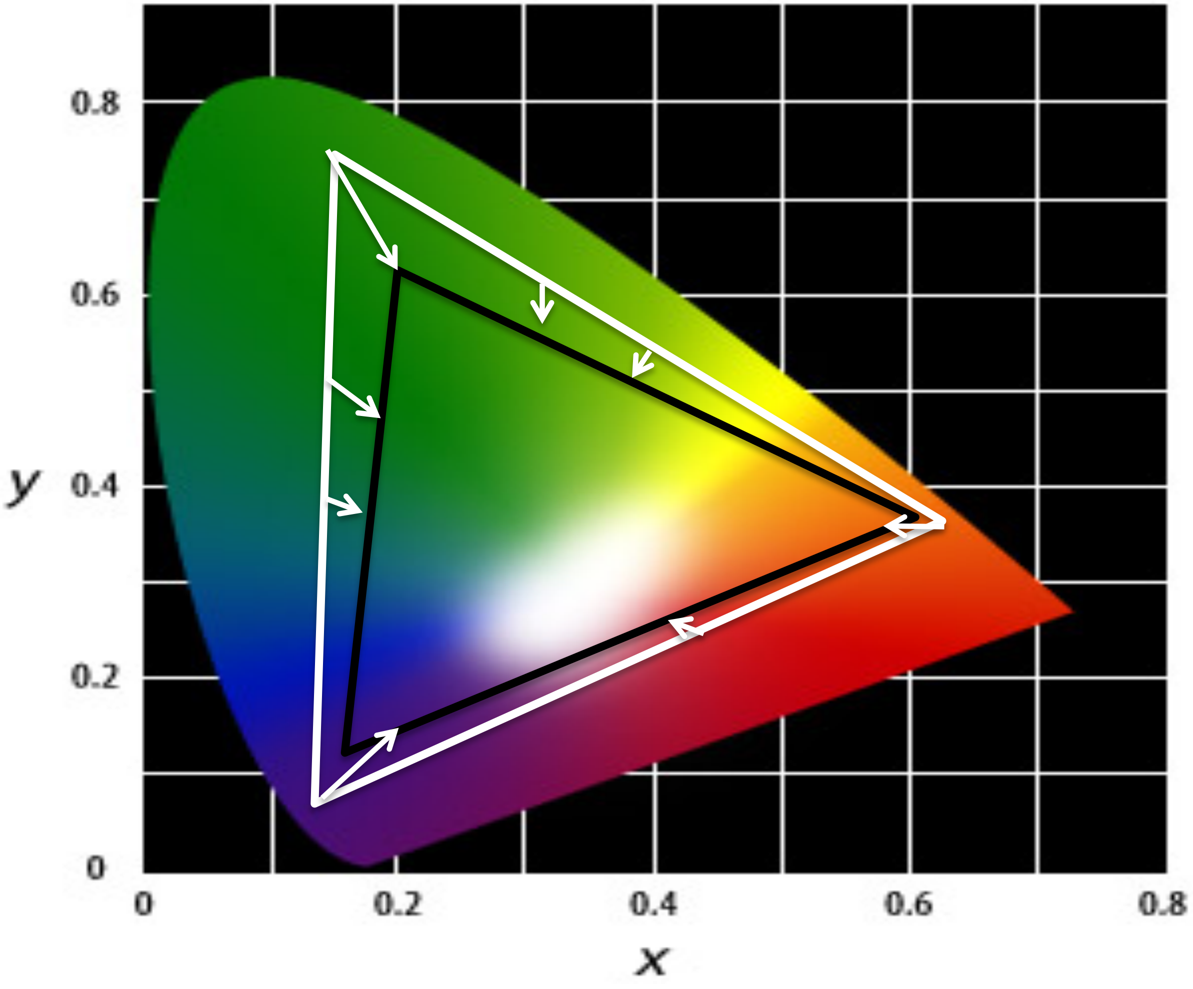


sRGB is the gamut for a Sony Trinitron CRT display

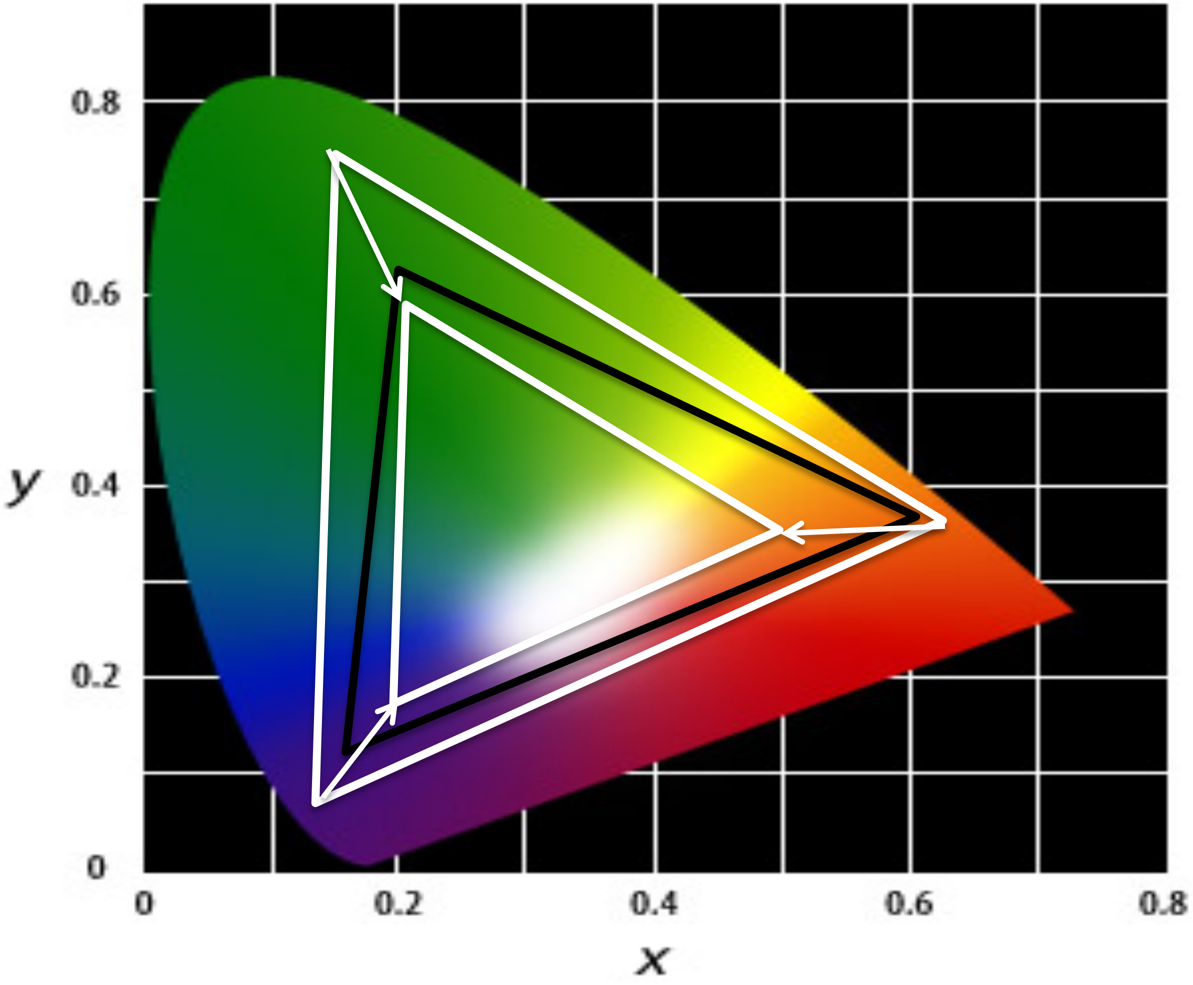


Gamut Mapping

Gamut Clipping ("Colorimetric")

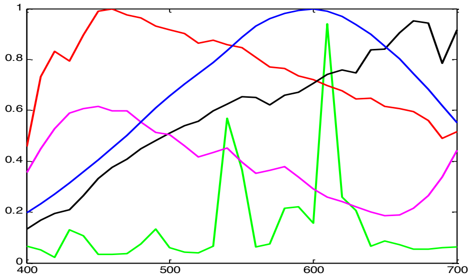


Gamut Compression ("Perceptual")



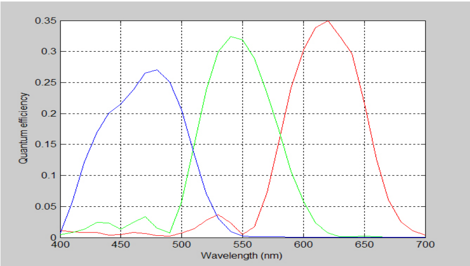
Sources of Error

1. Illuminant estimation

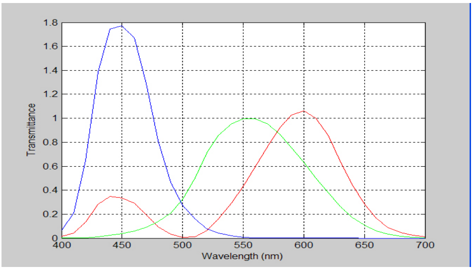


Wavelength(nm)

2. Conversion to calibrated color space
   – sensors are not XYZ



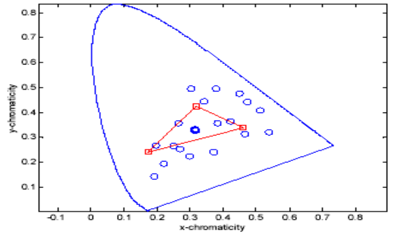
Sensors



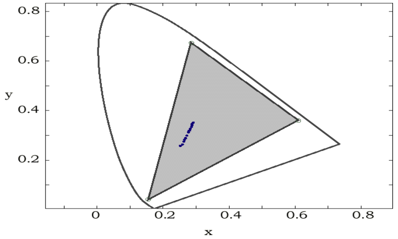
XYZ

3. Illuminant transformation

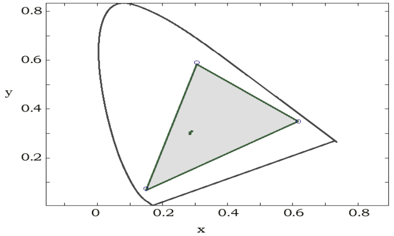
4. Display gamut



Cell phone display



LCD



CRT