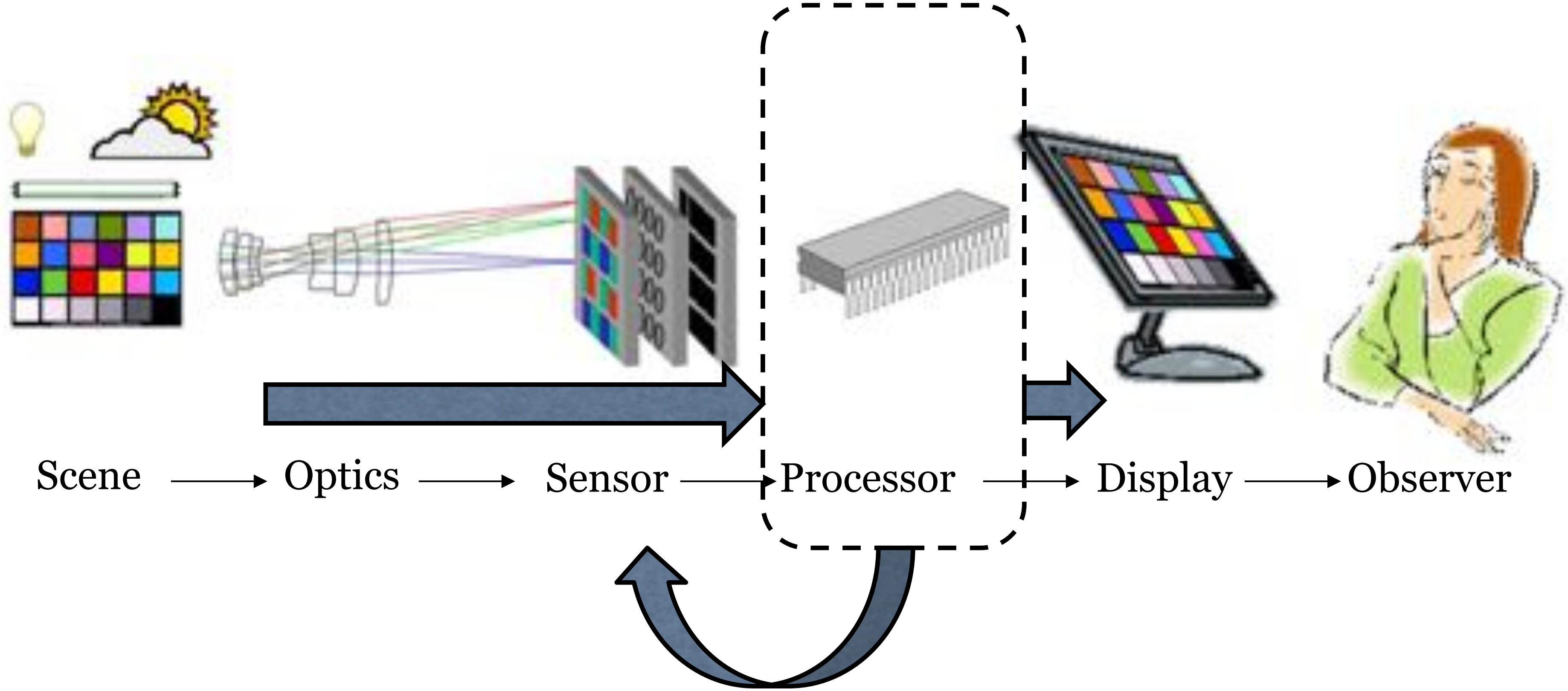
Image Processing Pipeline



Pipeline Goals

- Correct hardware imperfections
- Facilitate inter-device communication
- Address appearance (aesthetics) issues
  - Reproduce an image whose appearance matches the original up to display intensity and gamut limitations, or
  - Make a nice picture, guided by the original
  - Preserve certain image metrics (edges, geometry)

Basic Image Processing Pipeline

Black level adjustment
Lens distortion correction
Faulty pixel replacement

Exposure control

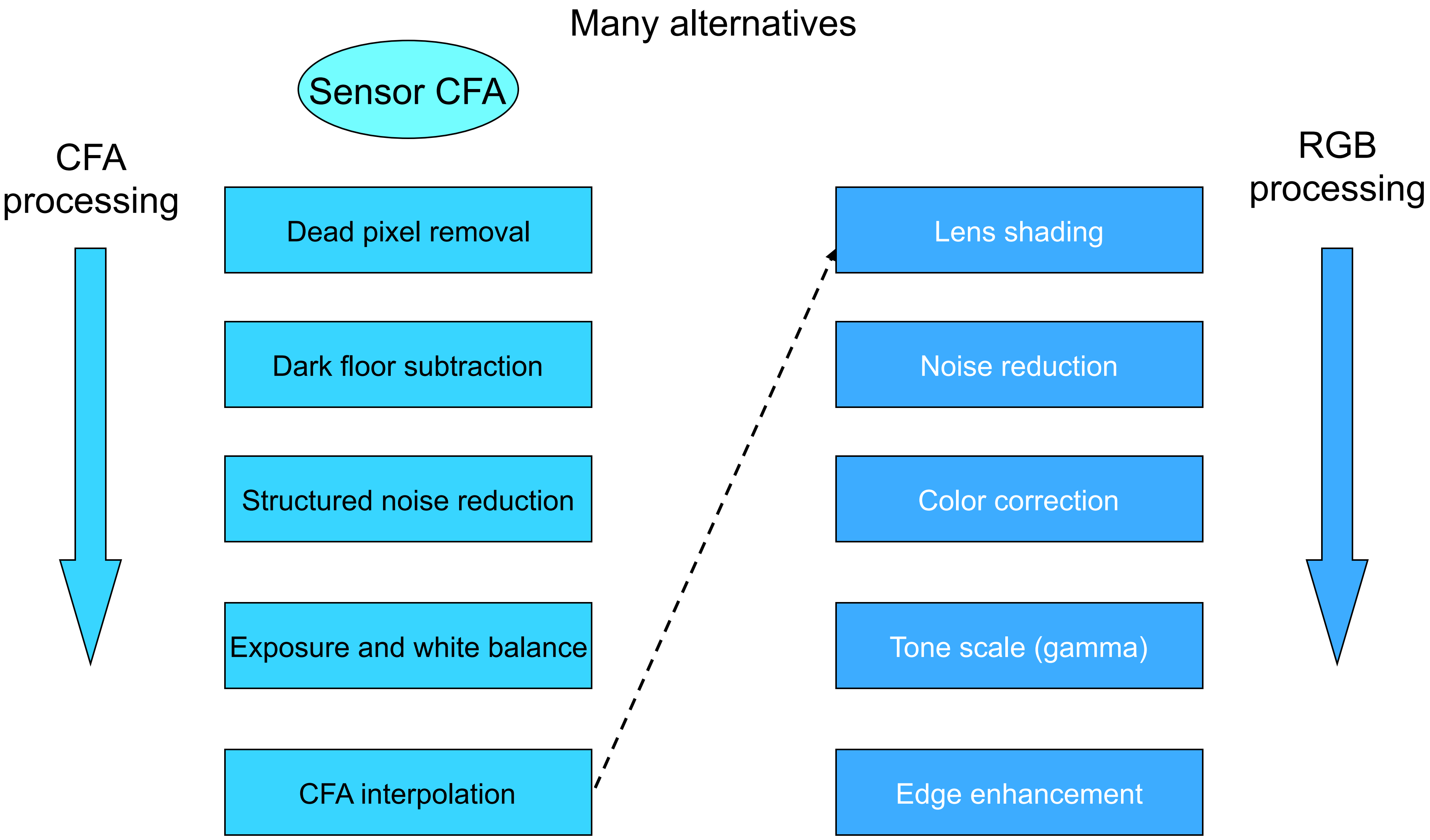
Demosaicking

Color Balancing

Compression and
Transmission (jpeg)

Reference Pipeline

(after Adams and Hamilton, in Lukas, Fig 3.1)



Reference Pipeline II

Design Considerations of Color Image Processing Pipeline for Digital Cameras

Wen-Chung Kao, *Member*, IEEE, Sheng-Hong Wang, Lien-Yang Chen, and Sheng-Yuan Lin

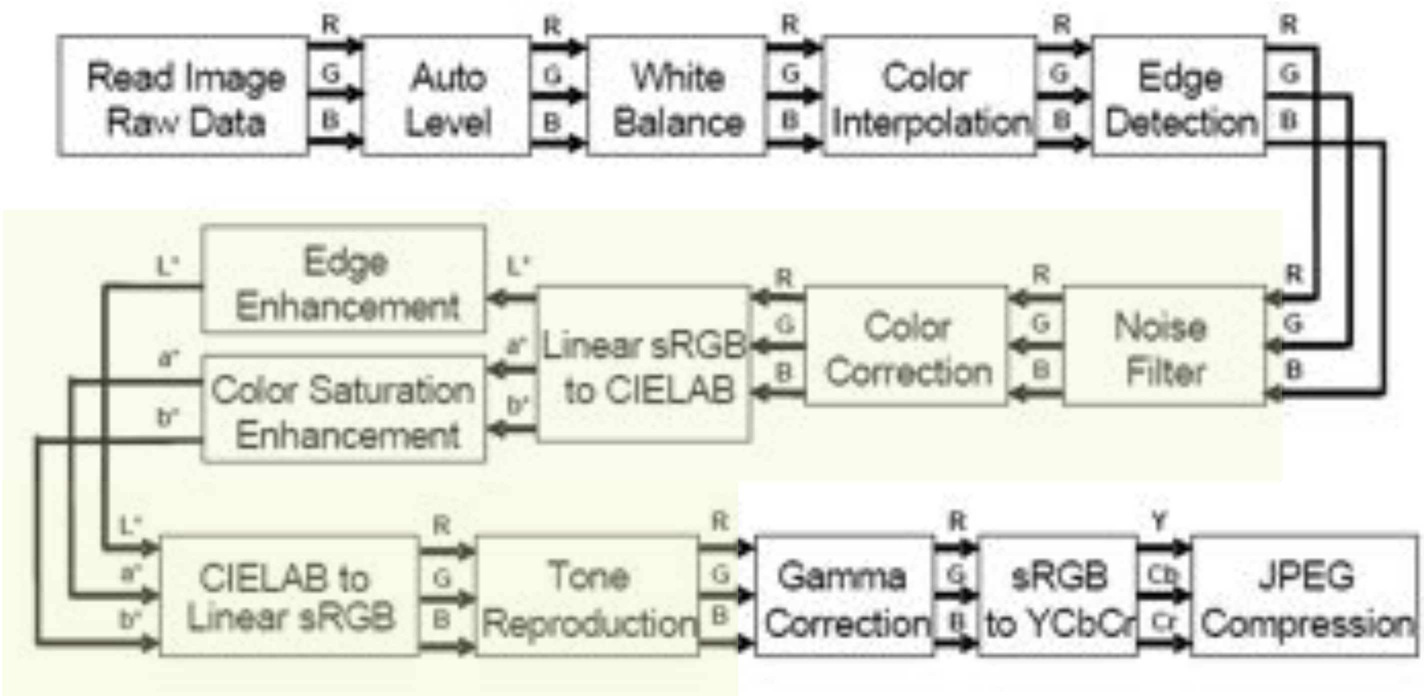
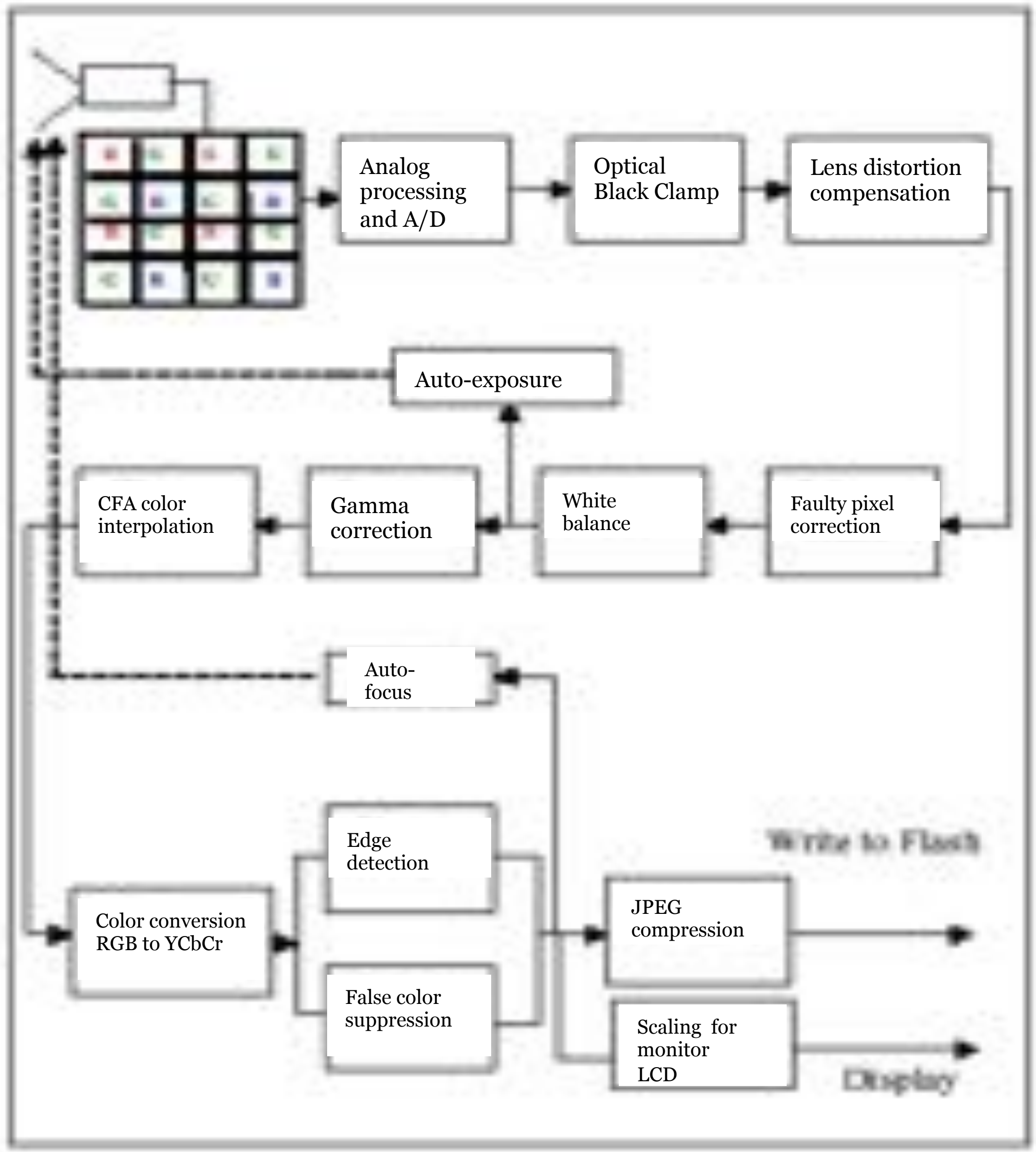


Fig. 1. The proposed image processing pipeline.

Chip controllers for the pipeline

Texas Instruments
TMS320DSC21

Programmable single chip DSP for digital cameras
2006



Basic Image Processing Pipeline

Black level adjustment
Lens distortion correction
Faulty pixel replacement

↓

Exposure control

↓

Demosaicking

Color Balancing

↓

Compression and
Transmission (jpeg)

Basic Image Processing Pipeline

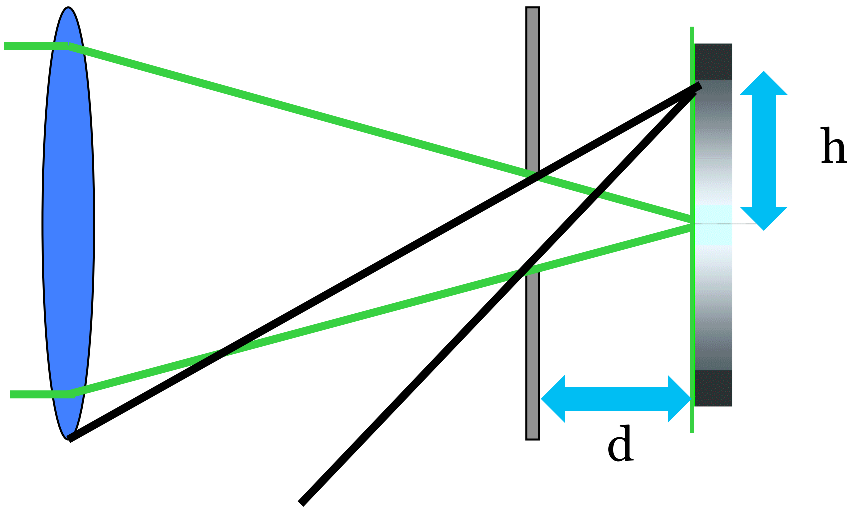
Black level adjustment
Lens distortion correction
Faulty pixel replacement

- ❑ Relative Illumination (Lens vignetting)
- ❑ Geometric distortions
- ❑ Chromatic aberration

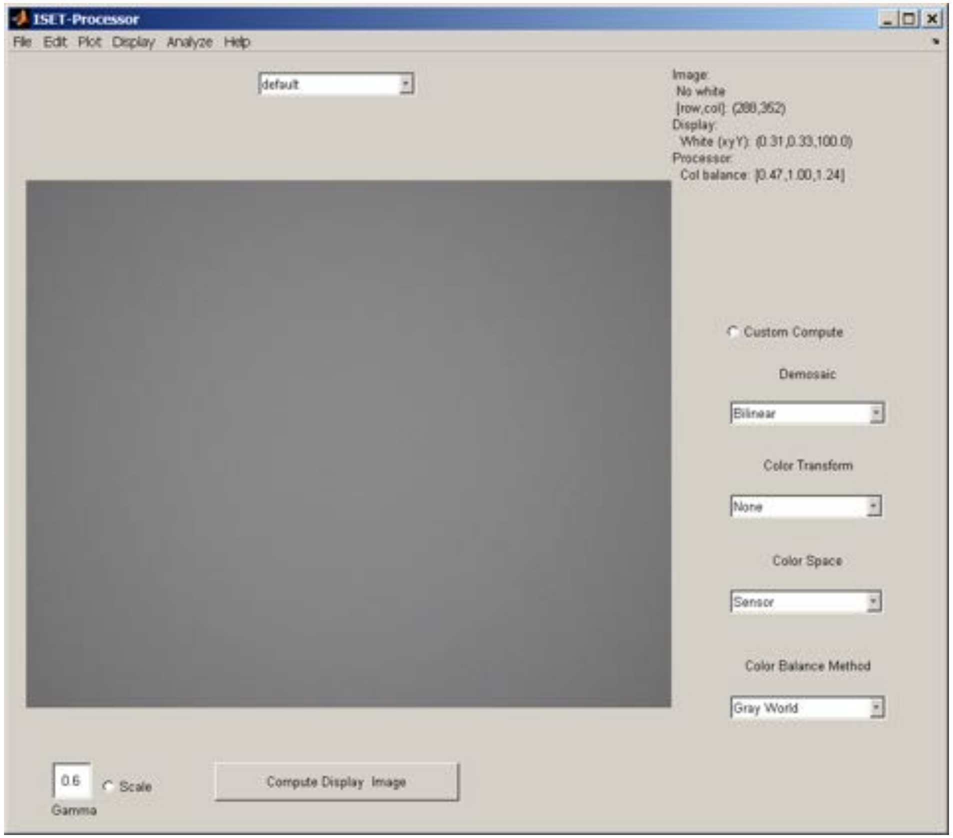
Relative illumination

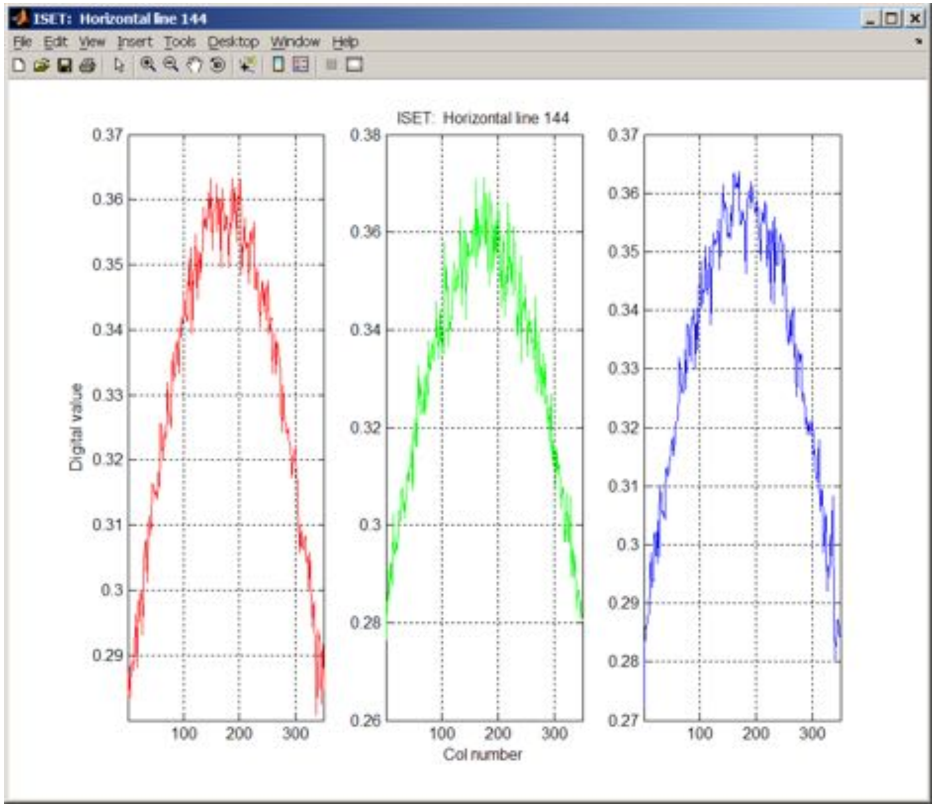
- Off-axis pixels receive a smaller bundle of incident rays compared to on-axis

- Relative illumination at the edge is lower

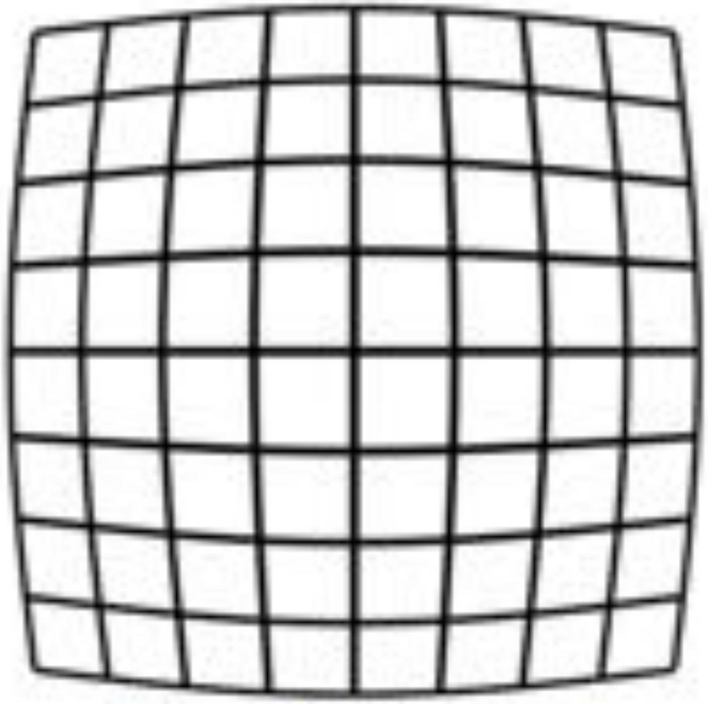


Lens Shading: Appearance

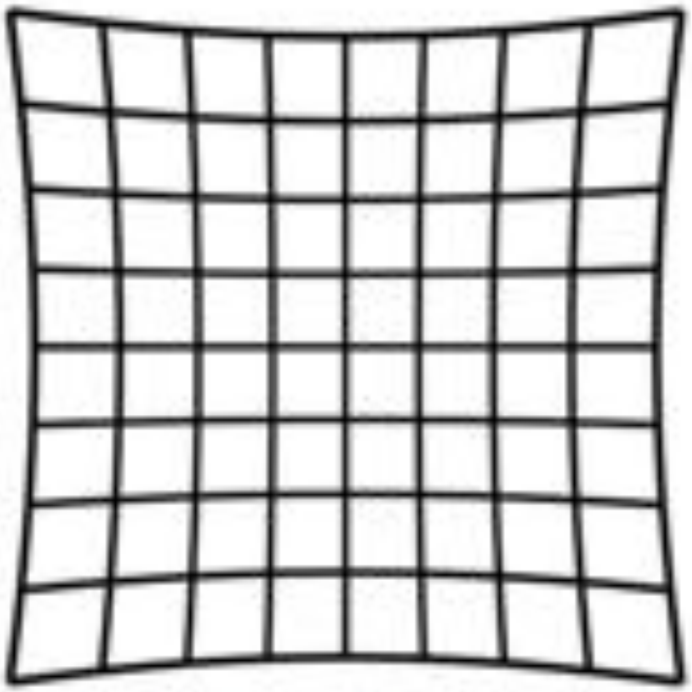




Geometric Distortions

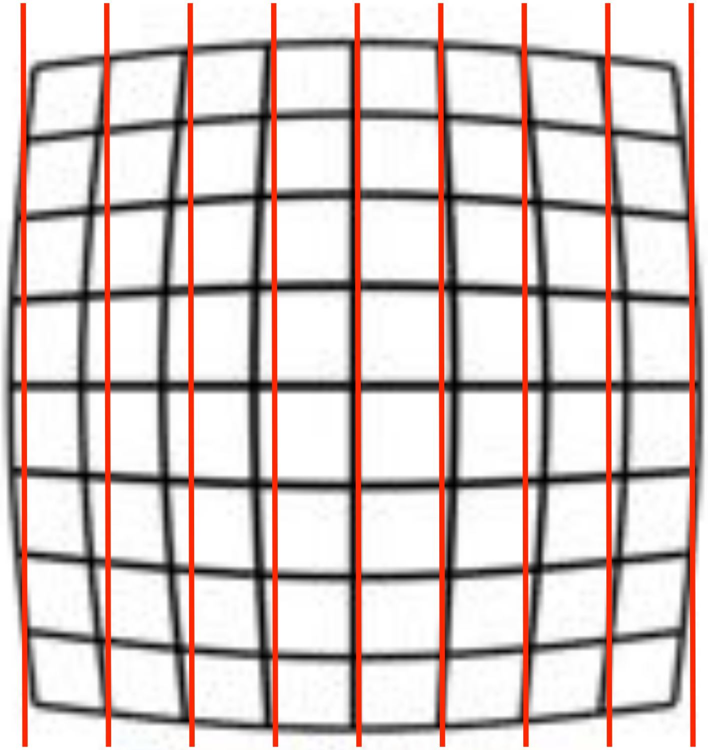


Barrel Distortion

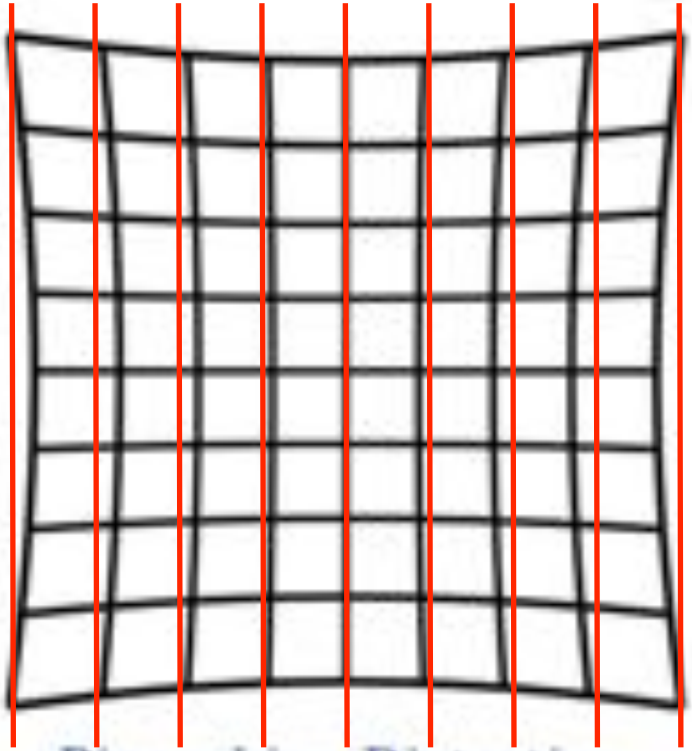


Pincushion Distortion

Geometric Distortions

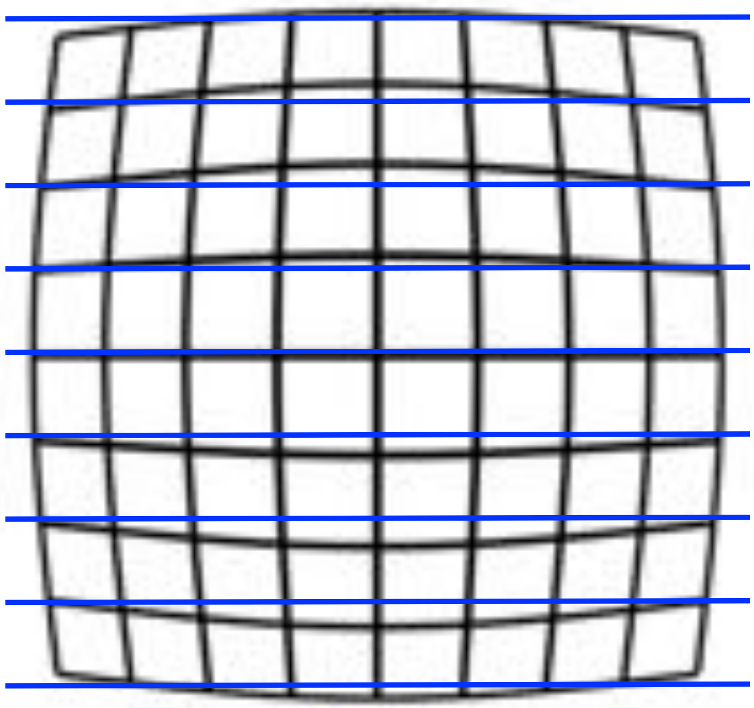


Barrel Distortion

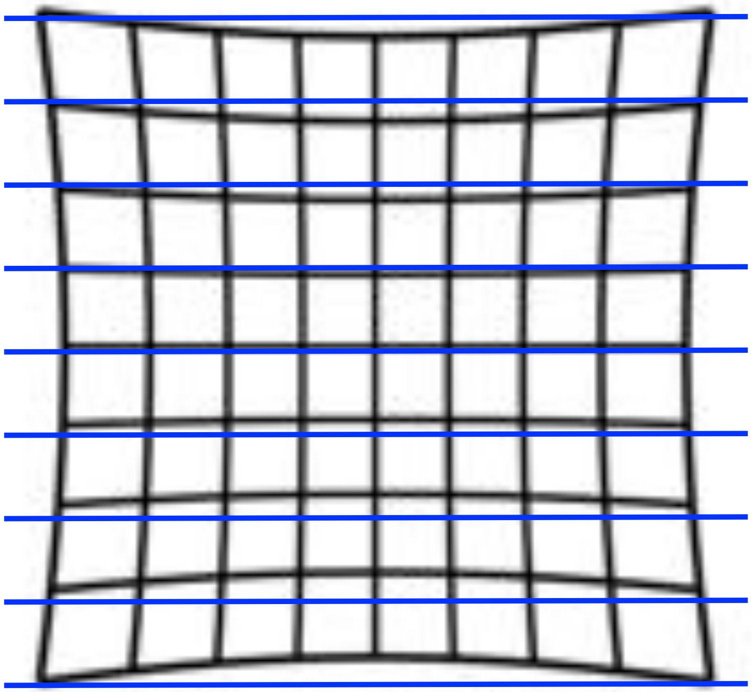


Pincushion Distortion

Geometric Distortions



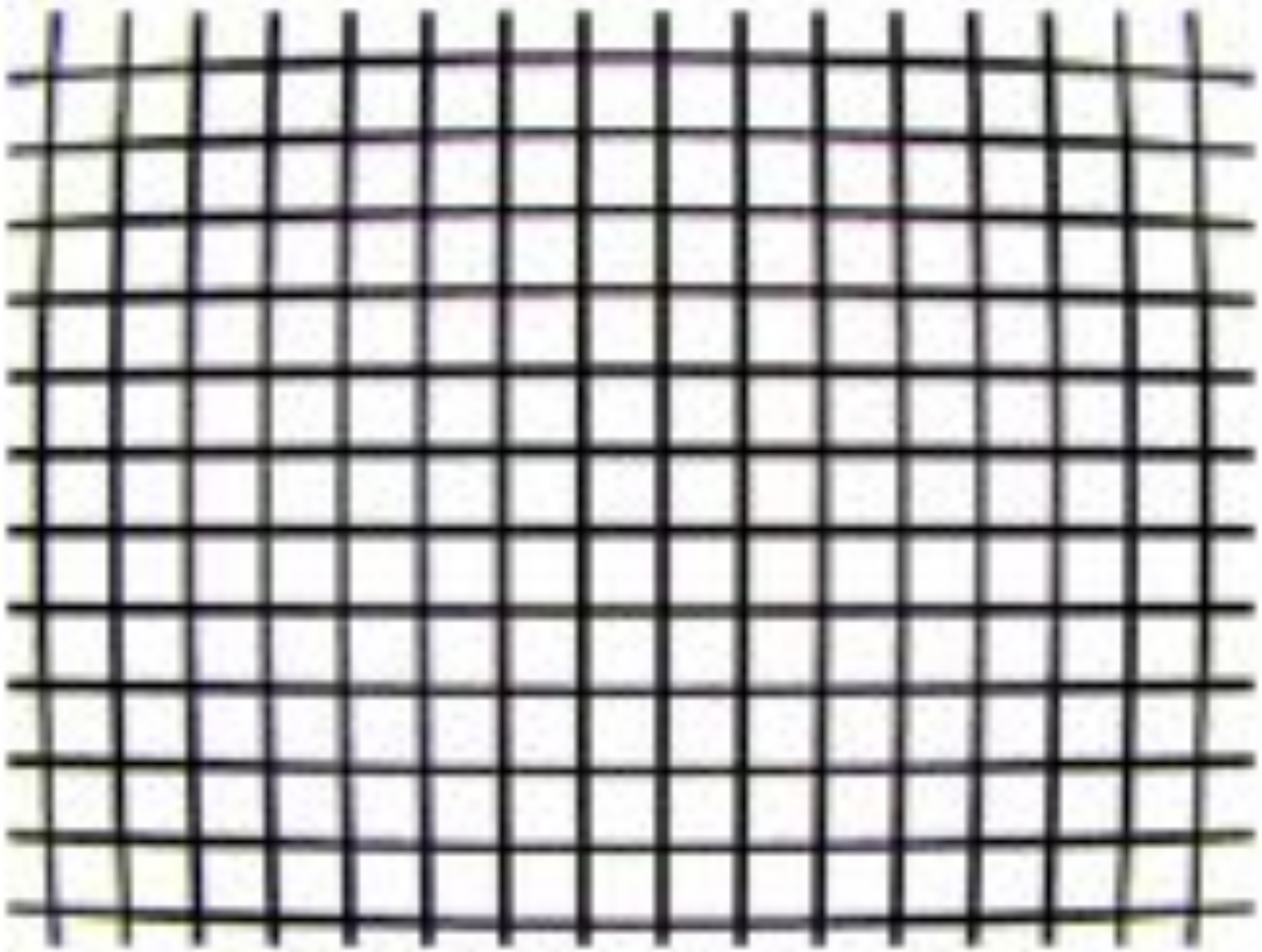
Barrel Distortion



Pincushion Distortion

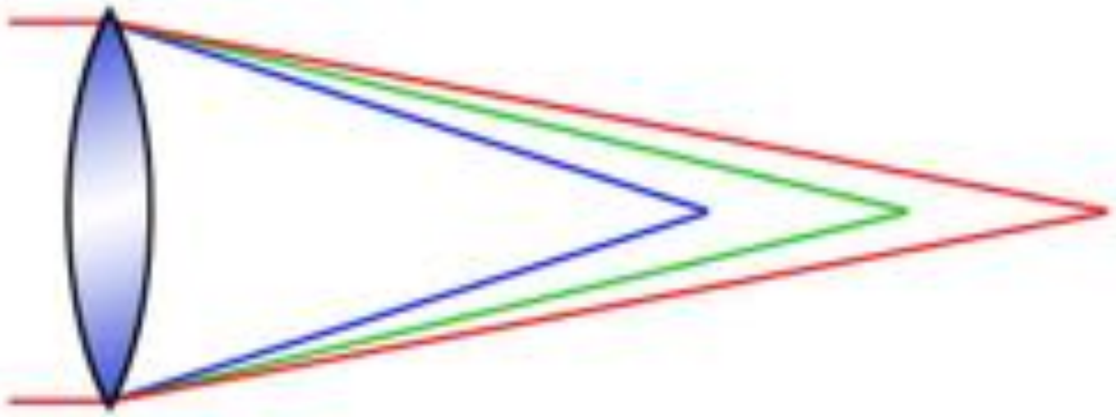
Lens Distortion

Chromatic Aberration



Chromatic Aberration

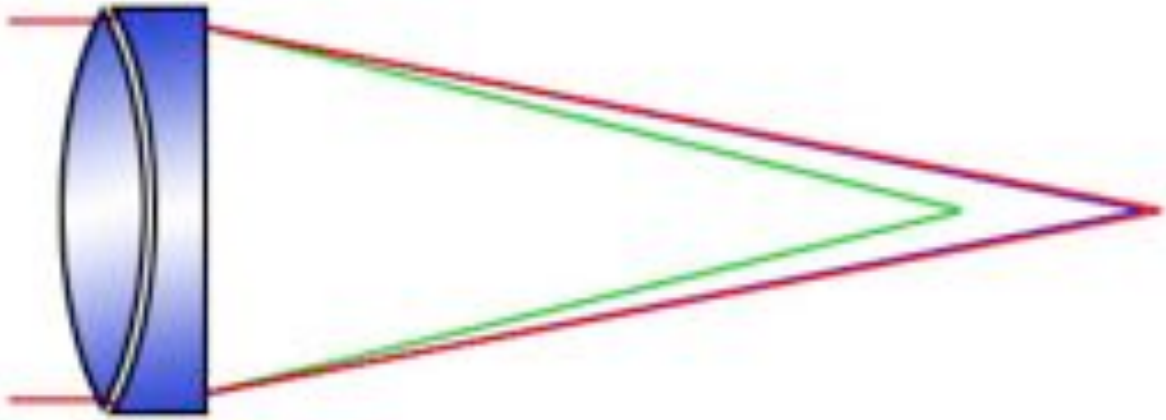
Longitudinal chromatic aberration: different wavelengths focus at different distances from the sensor



vs. Lateral chromatic aberrations: different wavelengths focus at different positions on the sensor

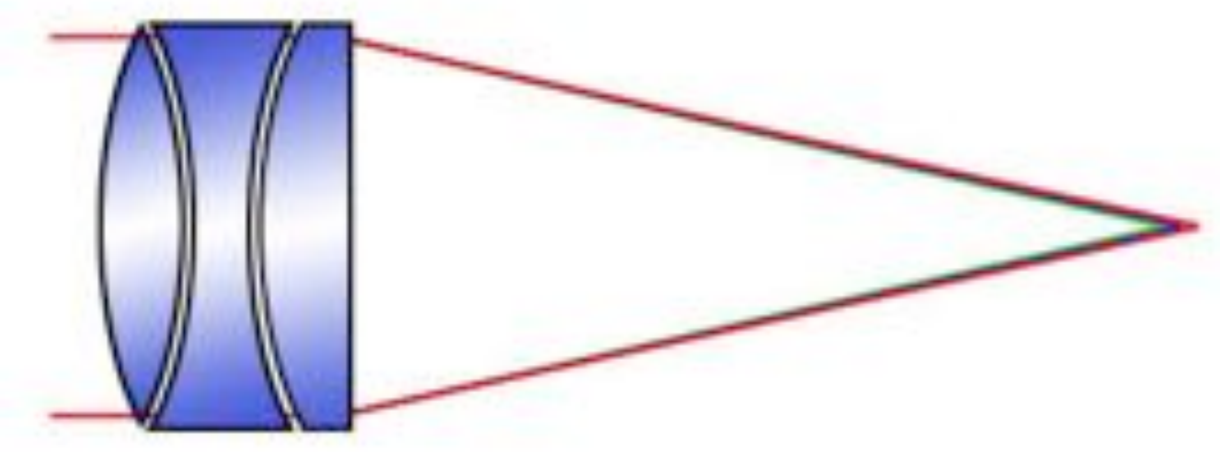
Chromatic Aberration

Add another optical element



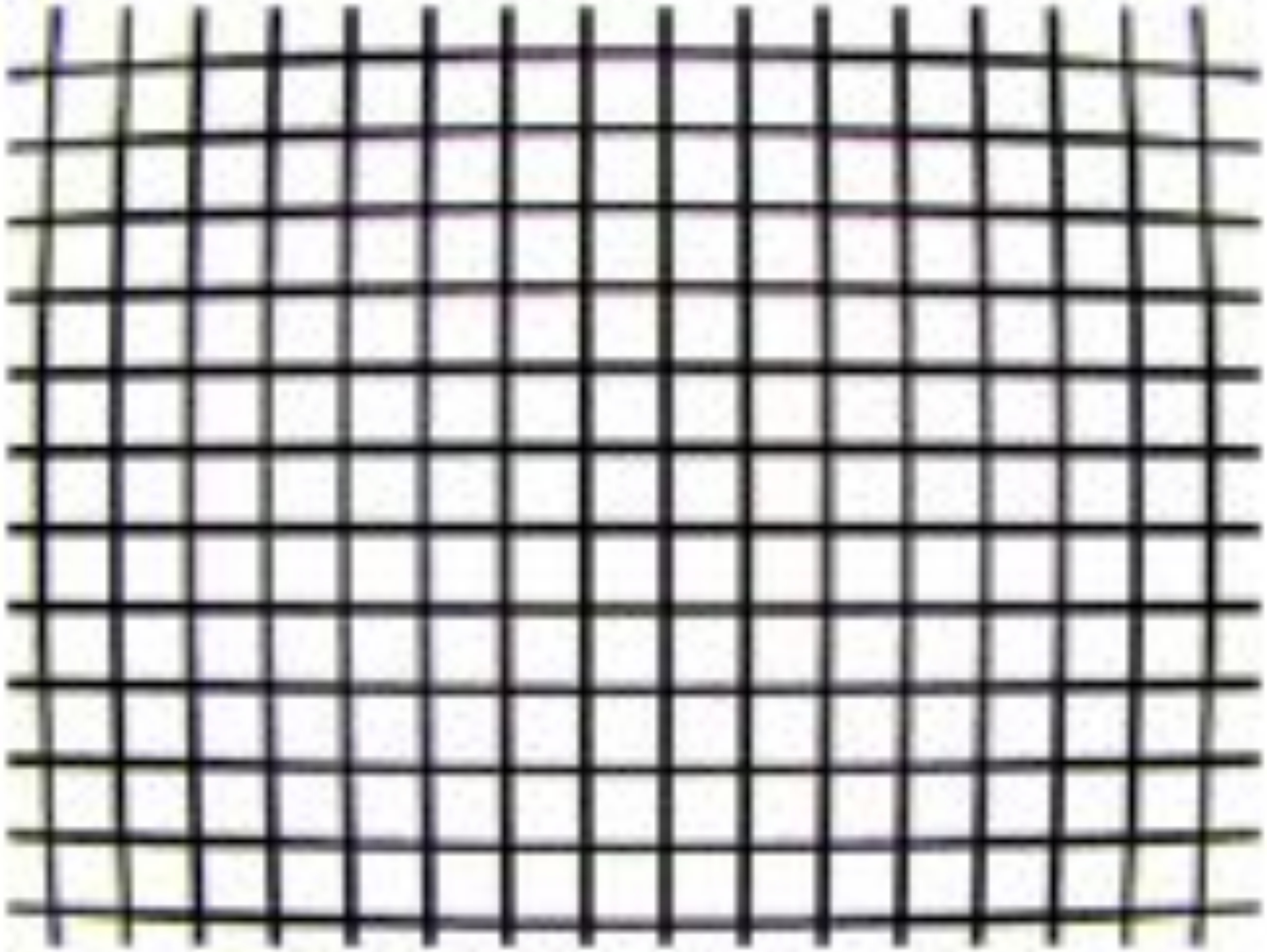
Chromatic Aberration

And another optical element



Optical designs minimize chromatic aberrations at the center of the sensor

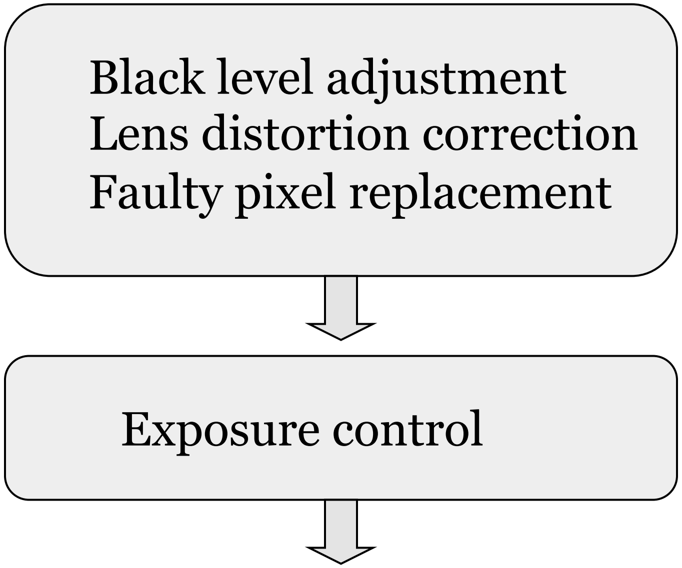
Chromatic Aberration



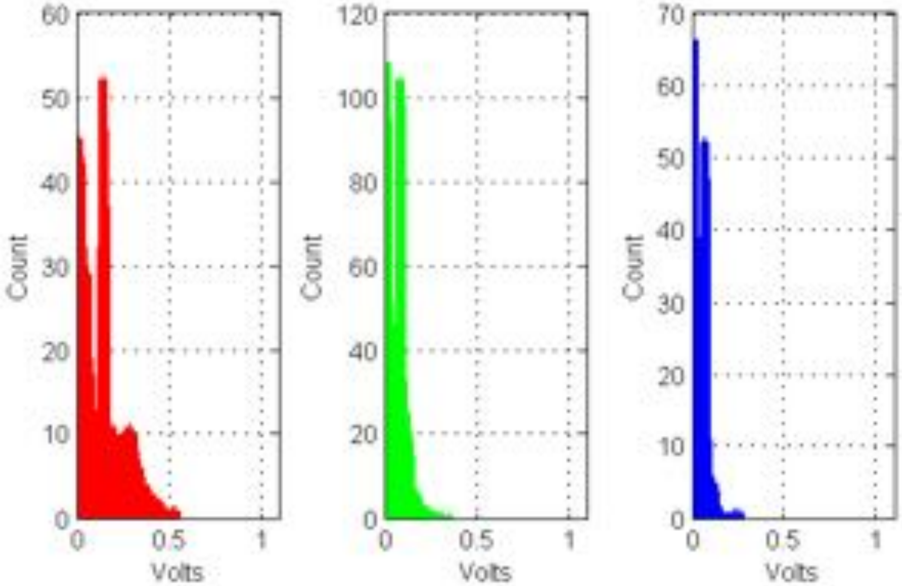
Chromatic Aberration

http://www.cambridgeincolour.com/TUTORIALS

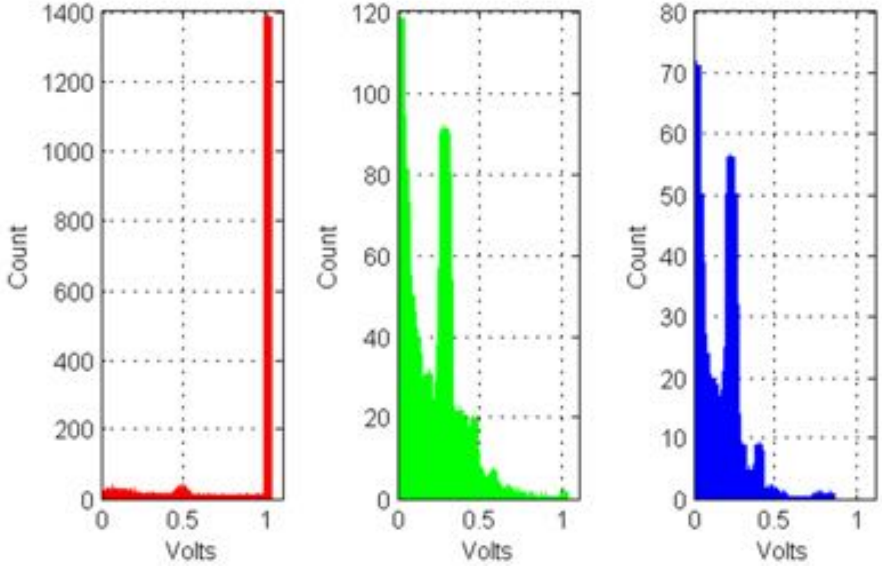
Basic Image Processing Pipeline



Underexposed



Overexposed



Exposure Control

Increase or decrease the amount of light by varying

1. Aperture size
   - When focal length is fixed, the f-number is determined by the diameter of the aperture

     F = focal length/aperture diameter

2. Exposure duration

Set the f-number (F) and exposure time (T) so that the image pixels accumulate a charge that reaches, but does not exceed, the well-capacity of the pixels (sensor range).

Concept of Exposure Value

1. Different combinations of aperture and integration times produce the same exposure value

2. Exposure value accounts for both F-number and integration time

3. Algorithms are based on a definition of Exposure Value (EV) and estimates of the current image brightness (B).

$$EV = \log_2(\frac{F^2}{T}) = 2\log_2(F) - \log(T)$$

F : f-number

T : exposure time

Exposure Value

$$EV = \log_2(\frac{F^2}{T}) = 2\log_2(F) - \log(T)$$

F : f-number

T : exposure time

1. Smaller F-number = more light
   - The bigger the aperture, the smaller the F-number
   - F=focal length/aperture diameter
2. Larger T (exposure time) = more light
3. Hence, smaller exposure value is better

Equivalent Exposure Table

Different combinations of aperture and integration times produce the same exposure value

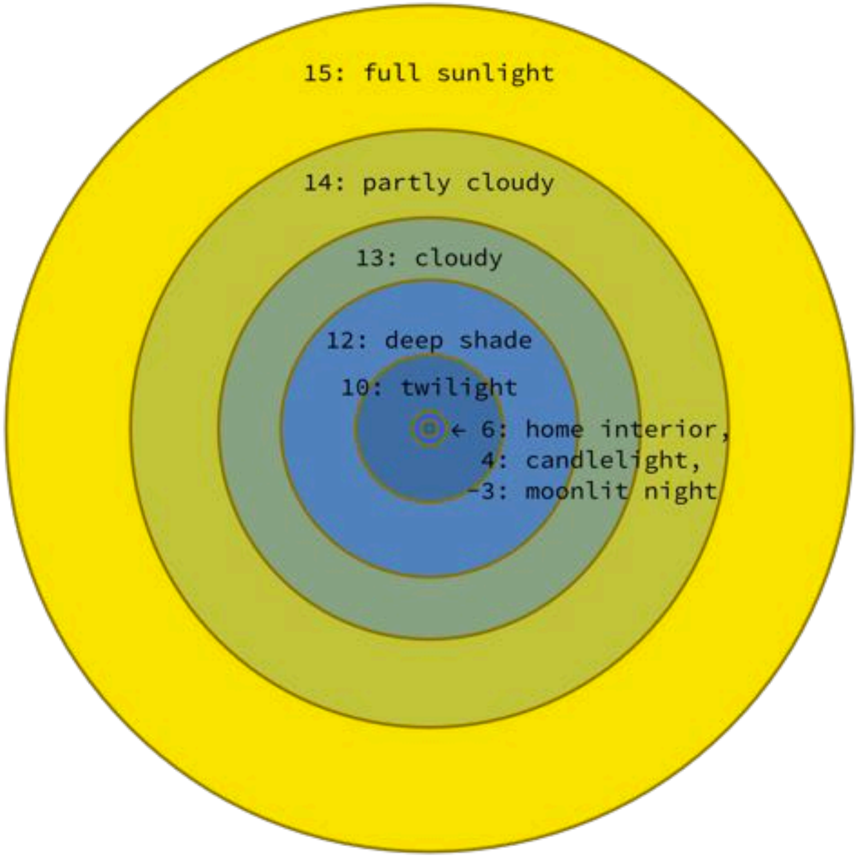
	f/1.0	f/1.4	f/2.0	f/2.8	f/4.0	f/5.6	f/8.0	f/11	f/16	f/22
1 sec	0	1	2	3	4	5	6	7	8	9
1/2	1	2	3	4	5	6	7	8	9	10
1/4	2	3	4	5	6	7	8	9	10	11
1/8	3	4	5	6	7	8	9	10	11	12
1/15	4	5	6	7	8	9	10	11	12	13
1/30	5	6	7	8	9	10	11	12	13	14
1/60	6	7	8	9	10	11	12	13	14	15
1/125	7	8	9	10	11	12	13	14	15	16
1/250	8	9	10	11	12	13	14	15	16	17
1/500	9	10	11	12	13	14	15	16	17	18

So what should the Exposure Value (EV) be?

Proprietary algorithms

Based on empirically determined relationship between exposure values and optimal (perceived) brightness

Could be based on image content and measures of brightness (e.g. mean, median, center-weighted, …)

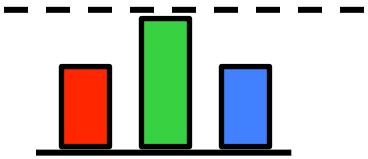


Finding the right exposure
Early film cameras

1. Set at Start
2. Select Scene Notch
3. Turn over and adjust for film speed
4. Read off Exposure for your F/Ratio

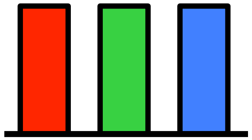
Poor exposure produces incorrect color

Raw data



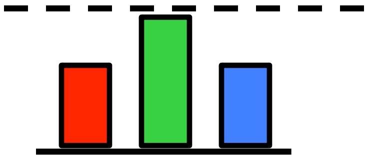
Scaled (1.2, 1.0, 1.2)
to appear neutral

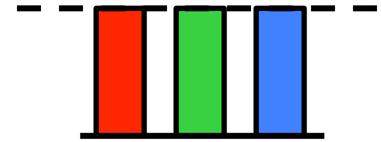
Processed data



Poor exposure produces incorrect color

Raw data

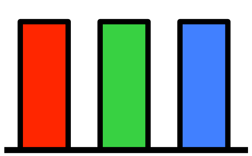




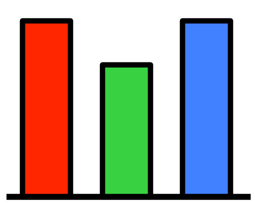
Scaled (1.2, 1.0, 1.2) to appear neutral

Scaled (1.2, 1.0, 1.2) to appear neutral

Processed data



Scaling makes the saturated image region purple



Poor exposure produces incorrect color

Basic Image Processing Pipeline

Black level adjustment
Lens distortion correction
Faulty pixel replacement

↓

Exposure control

↓

Demosaicking

Color Balancing

↓

Compression and
Transmission (jpeg)

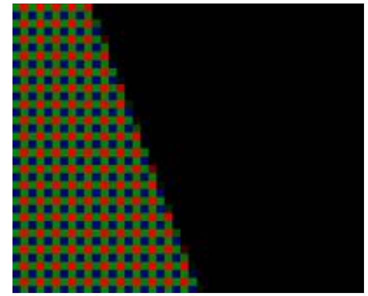
Demosaicking

- ❑ Take advantage of color correlations (between channels)
- ❑ Take into consideration local spatial structure (adaptive)
- ❑ As sensor resolution increases, differences between algorithms diminishes

(Video Tutorial)

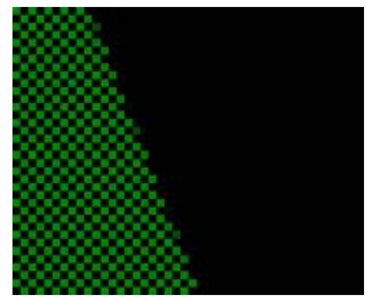
Demosaicking: The problem

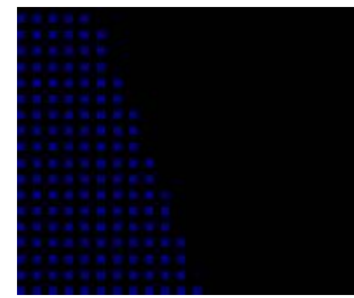
Sensor Array has Red, Green OR Blue
at every pixel



Leave measurements in place

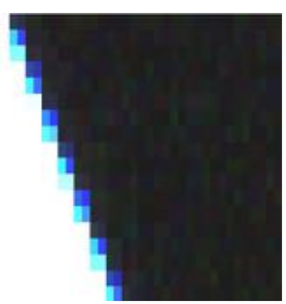
Interpolate the missing
Red, Green, and
Blue Pixel Values



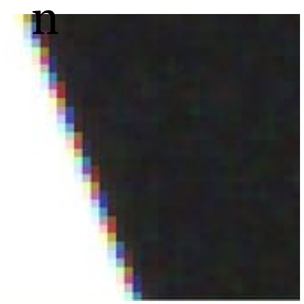


Create an image that has a
Red, Green AND Blue
at every pixel

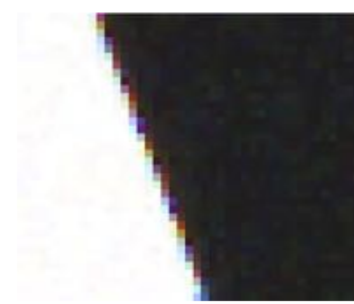
Nearest
Neighbor



Bilinear
Interpolatio
n



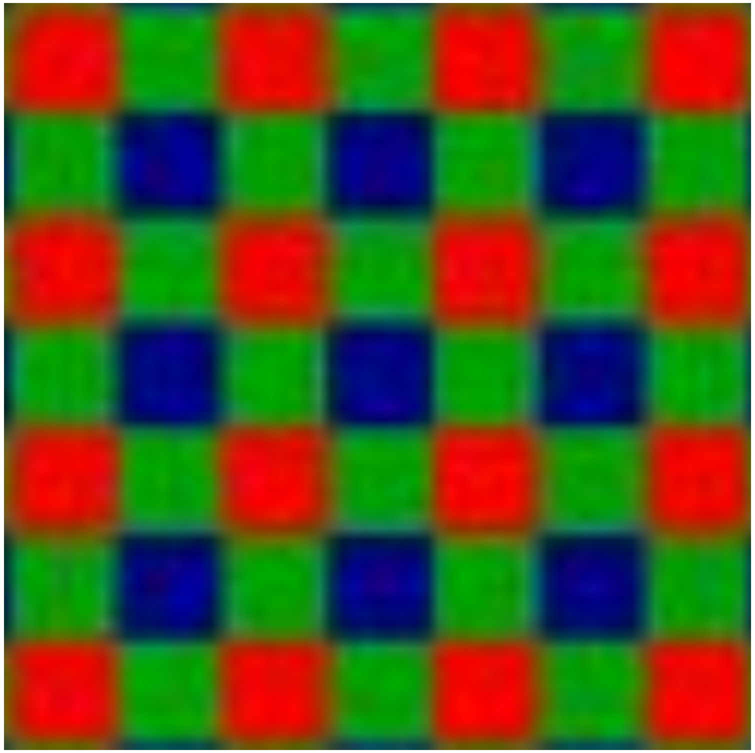
Adaptive
Laplacian



Demosaicking depends on the mosaic
The Bayer mosaic is most common

•G is present at twice the spatial resolution of the R or B

•G is associated with luminance; G is always fully interpolated

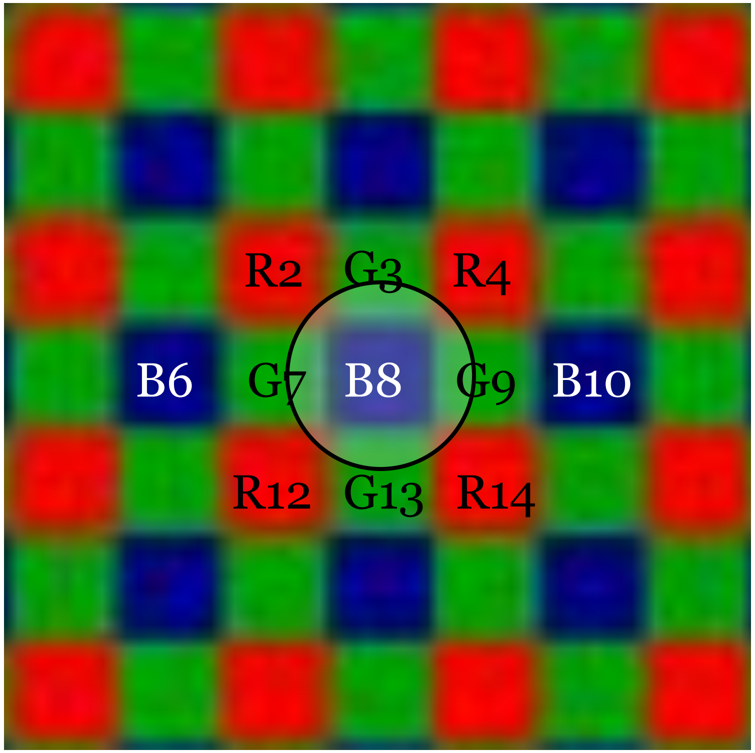


Bilinear interpolation (within channel)

Existing values are left untouched. The average of adjacent green pixel.

For example:

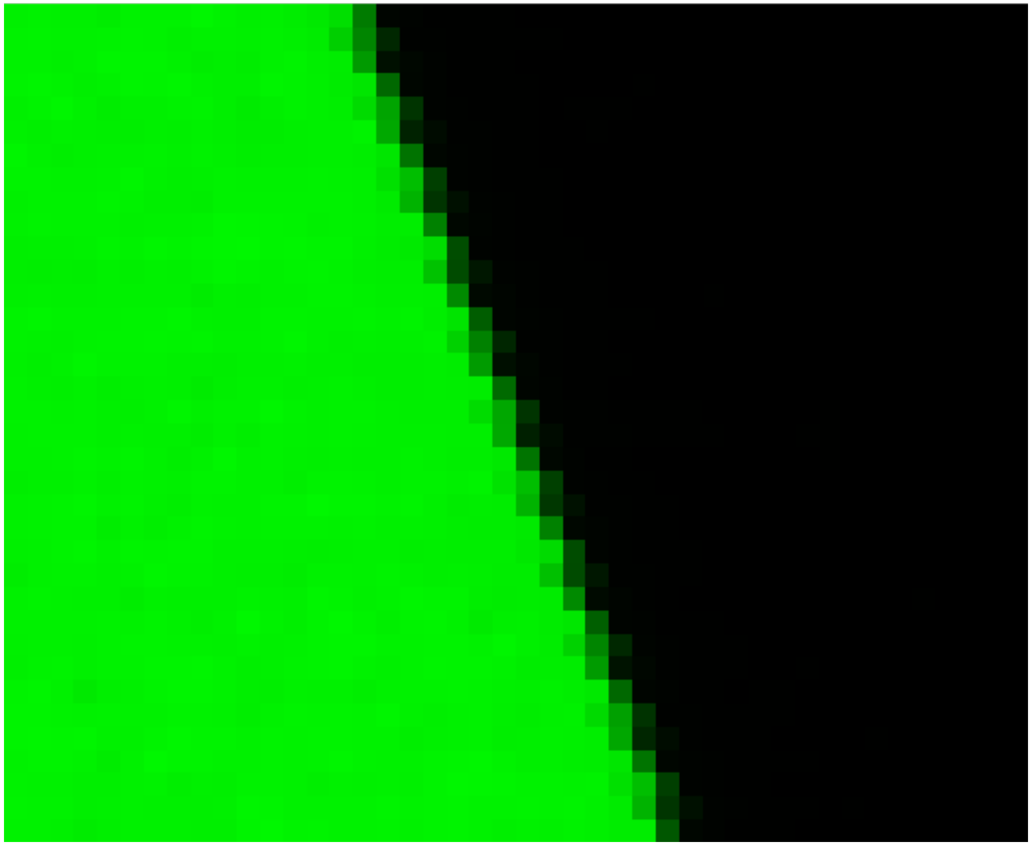
$$G8 = (G3+G7+G9+G13) / 4$$



Analysis: bilinear interpolation

Most modern algorithms

- Combine information from color channels to make a decision
- Go beyond linear interpolation and try to preserve edges

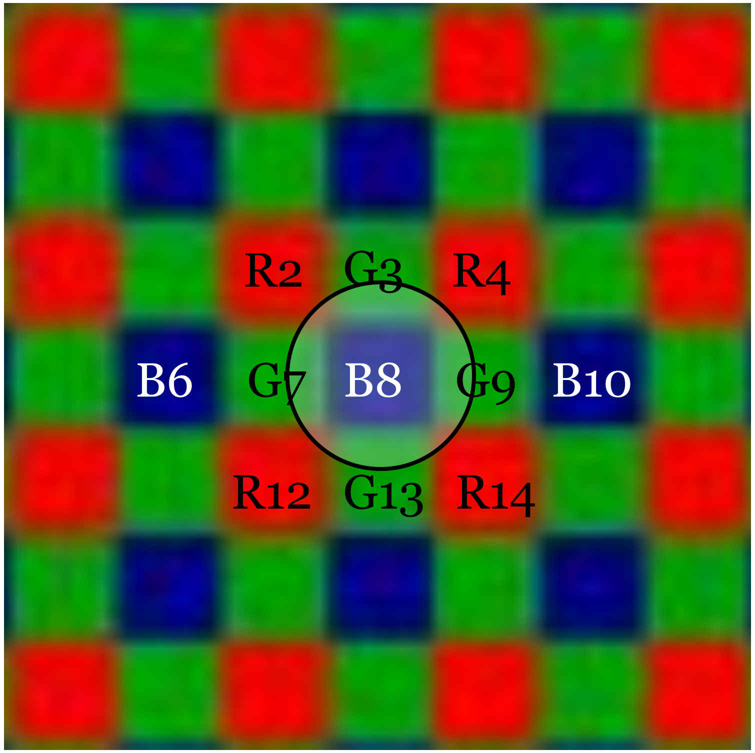


Bilinear interpolation (across channel)

Across channel uses weighted sum from other colors, in particular those at the same location – these are usually very correlated.

For example:

$$G8 = \alpha B8 + \beta \frac{G3 + G7 + G9 + G13}{4} + (1 - \alpha - \beta) \frac{R2 + R4 + R12 + R14}{4}$$

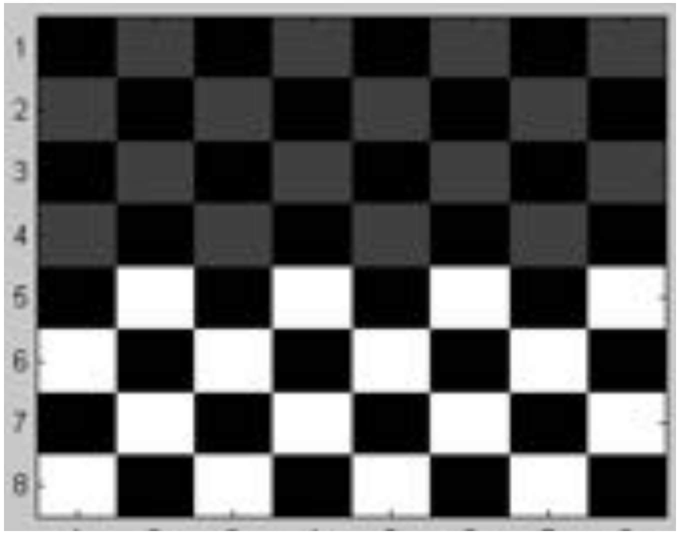


Across channel adaptive demosaicking

R G R

G B G

R G R



Goal: Don't average across edges.

Method: Before averaging, check for outliers.

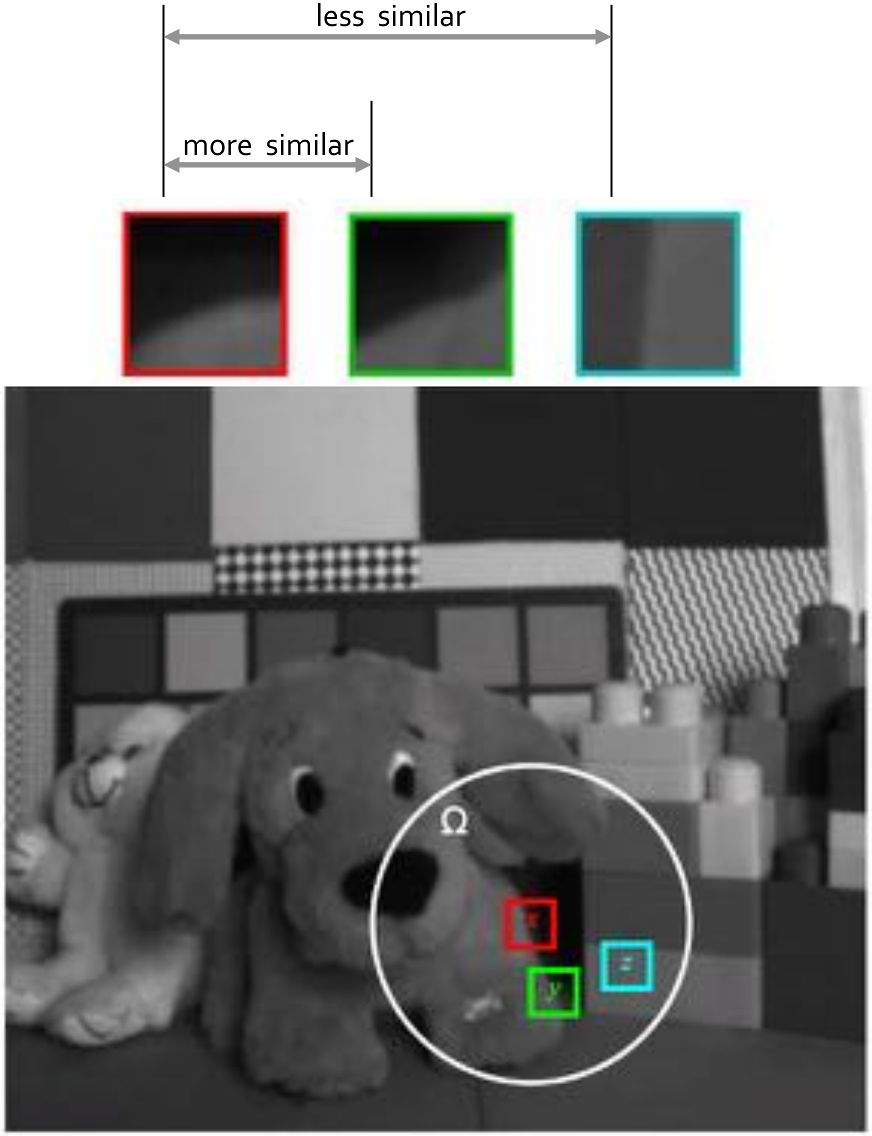
Average a G value if it is similar to others

Otherwise, exclude it from weighted sum

Pixel-similarity (nonlocal means)
(Buades et al., 2005)

http://www.mathworks.com/matlabcentral/fileexchange/13619

- Each pixel is associated with a small surrounding patch
- Pixel-similarity is determined by similarity of associated patches
- The pixel-similarity between x and y is higher than the pixel-similarity between x and z
- Hence, x and y, but not x and z, are averaged



Demosaicking

First Patents

- Bayer array: Kodak, 1976
- Demosaicking: Kodak, 1986

1976-2009: Over 100 algorithms

- http://www.danielemenon.netsons.org/top/demosaicking-list.php

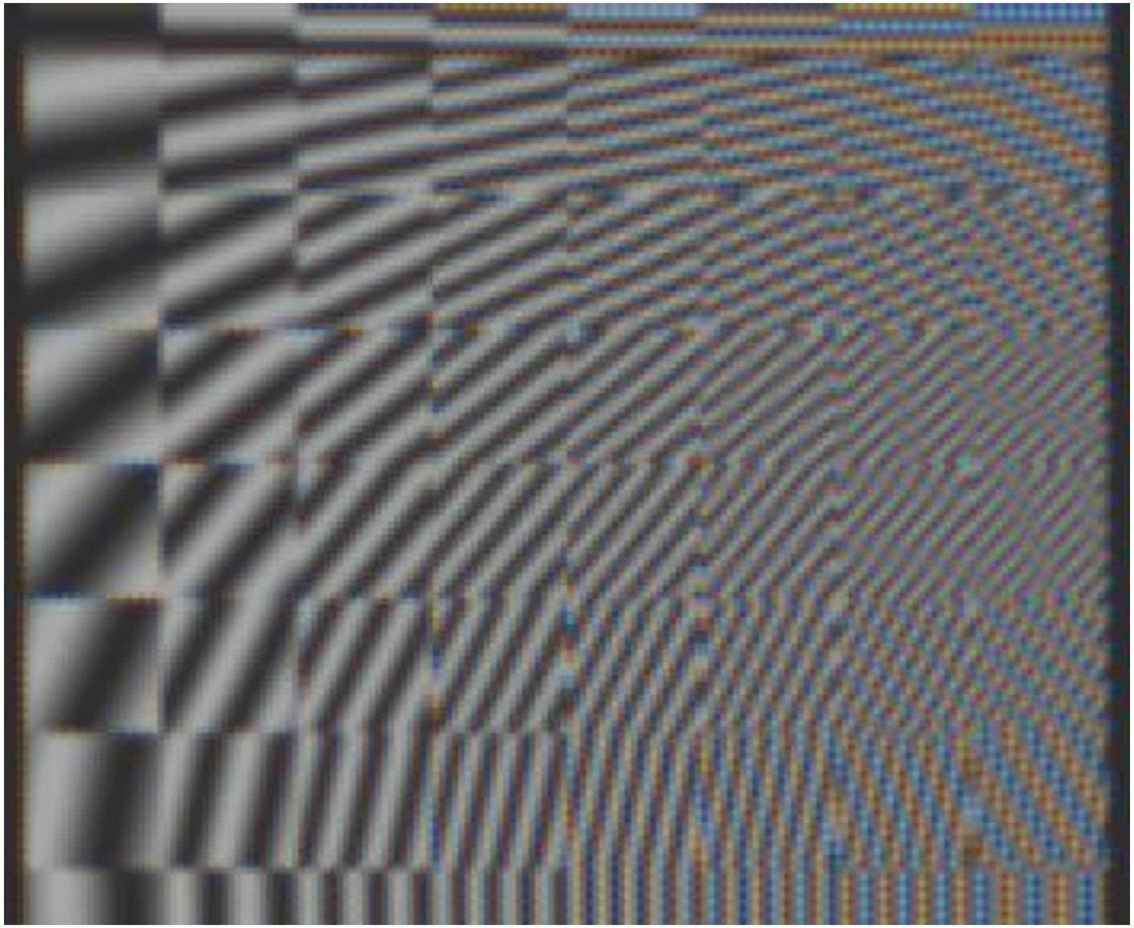
Nice review article: Gunturk, Glotzbach, Altunbasak, Schafer, and Mersereau, "Demosaicking:Color Filter Array Interpolation", IEEE SIGNAL PROCESSING MAGAZINE **[44]**, JANUARY 2005

- http://www.ece.gatech.edu/research/labs/MCCL/pubs/dwnlds/bahadir05.pdf

Demosaic comparisons
Bilinear and adaptive

Bilinear

Blurred

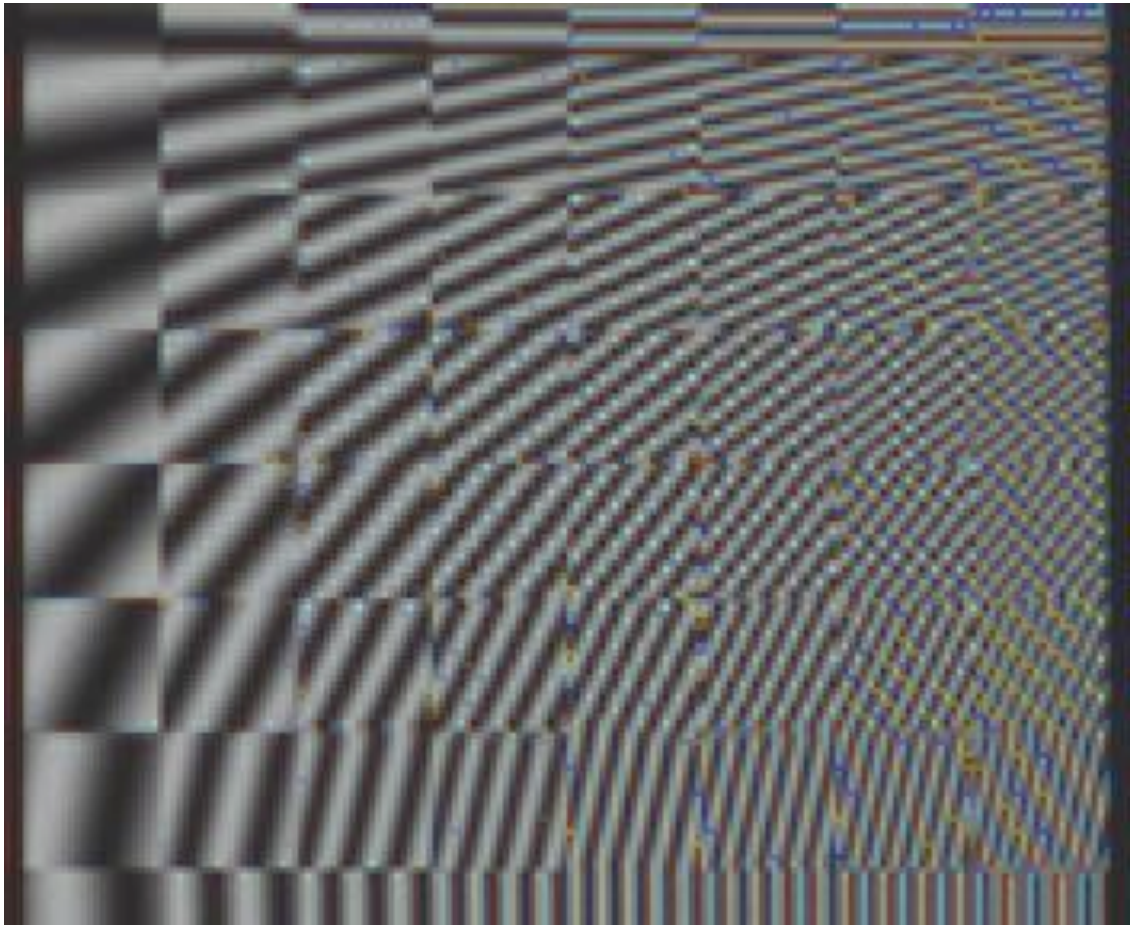


Chromatic artifacts

Chromatic artifacts

Demosaic comparisons
Bilinear and adaptive

Adaptive Laplacian



Chromatic artifacts

Sharper

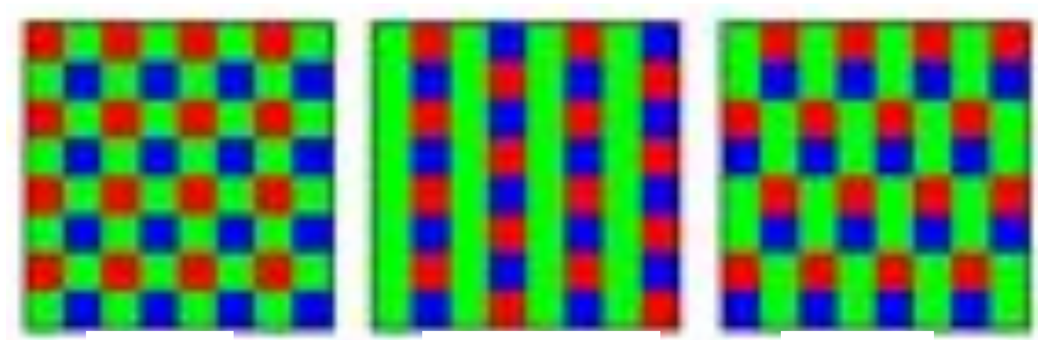
Chromatic artifacts

Demosaicking

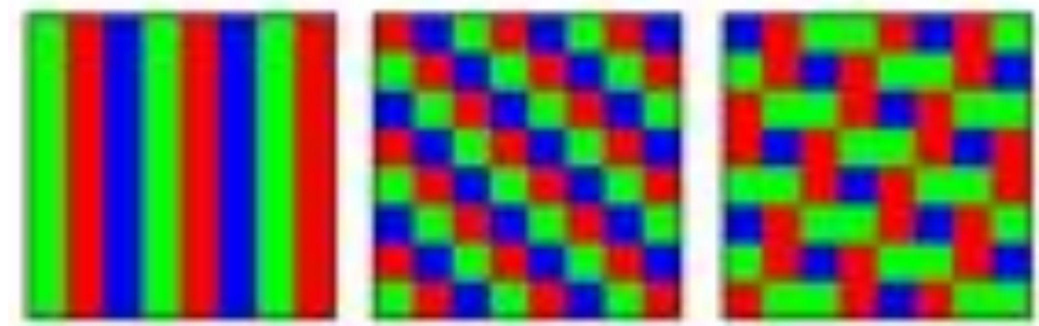
- Take advantage of color correlations (between channels)
- Take into consideration local spatial structure (adaptive)
- **As sensor resolution increases, differences between algorithms diminishes**

New algorithms for new CFAs

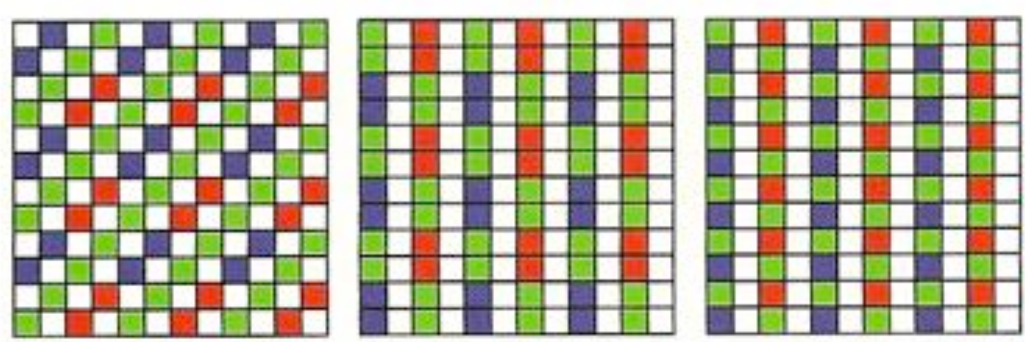
Most widely used



Bayer Yamanaka Lukac



Striped Diagonal striped Holladay



L^3 pipeline

RAW image

Classify pixel | Apply transform

Rendered image

Standard pipeline

L^3 pipeline

- Standard pipeline (Bayer CFA)
  - Requires multiple algorithms
  - Each algorithm requires optimization

- L^3 pipeline (all CFAs)
  - Multiple algorithms are integrated into one transform
  - Machine learning optimizes transform
  - Rendering is simple, fast and low-power

Watch the video

Learning the image processing pipeline

Brian Wandell

SCIEN talk on April 27, 2016

Go to https://talks.stanford.edu/scien/scien-colloquium-series/

Color Management

"Color management" is a process where the color characteristics for every device in the imaging chain is known precisely and utilized in color reproduction. It often occurs behind the scenes and doesn't require any intervention, but when color problems arise, understanding this process can be critical.

http://www.cambridgeincolour.com/tutorials/color-management1.htm

Color management

- Informal color processing terms
  - Used in different ways
- We use
  - Sensor correction
  - Illuminant correction

White balance

Color conversion

Color balance

Color correction

Color rendering

Illuminant transformation

Color constancy

Why do we need color management?

- Sensor Correction
  - Spectral sensitivities of the camera do not match the spectral sensitivities of the human eye
  - Sensors are not "colorimetric"

- Illuminant correction
  - Camera does not adjust the gain of RGB sensors with changes in the spectral power of the illumination in the same way as the visual system adjusts the gain of the LMS sensors
  - Cameras do not have "color constancy"

- Display rendering and Color gamut mapping
  - Displays may have different color primaries and gamma
  - Range of colors a camera image processing pipeline can produce does not match the range of colors a display or printer can produce

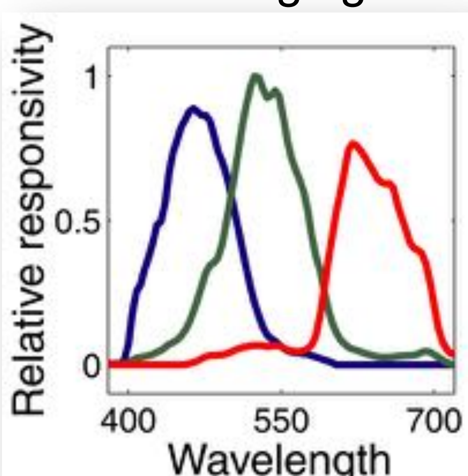
Sensor Correction

Camera sensors are not XYZ-CMFs (i.e. not “colorimetric”)

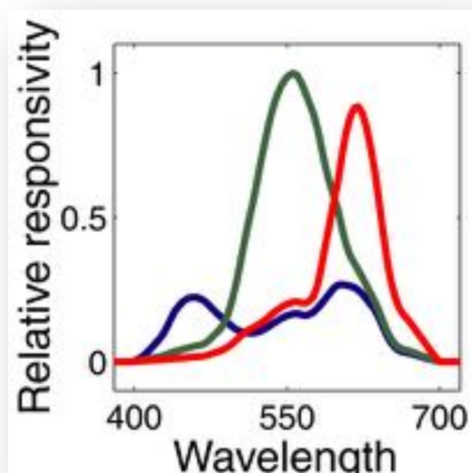
Sensor correction converts sensor RGB values to a calibrated color space (e.g. XYZ)

Sensor RGB images

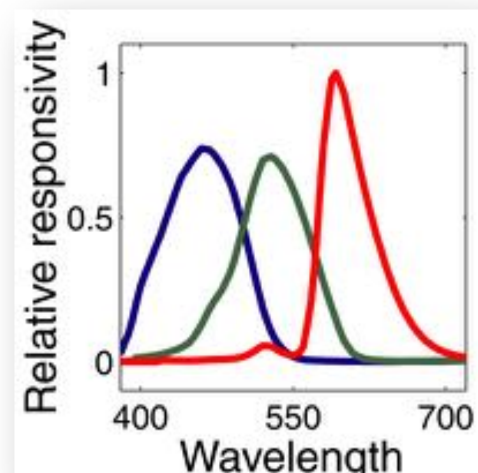
QImaging



Kodak



Nikon

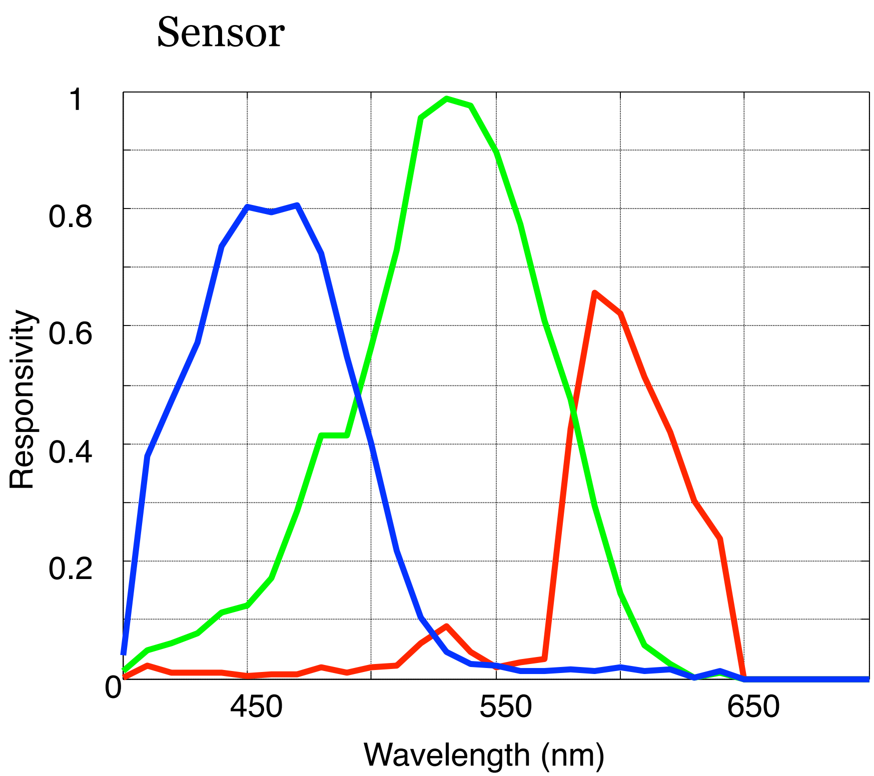


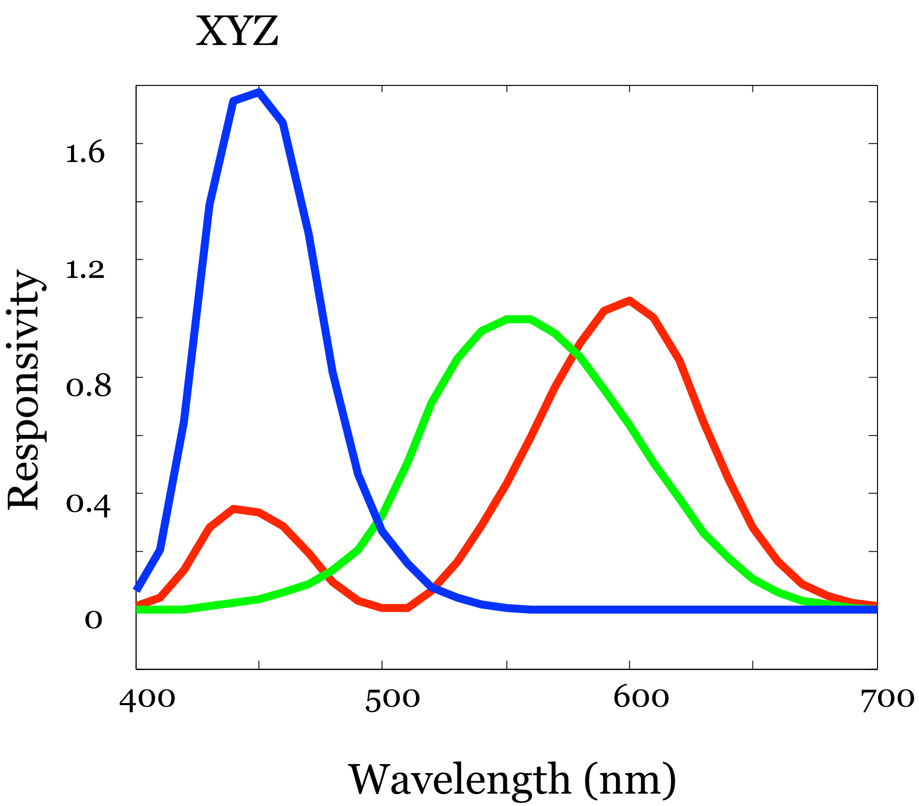
Sensor correction: RGB example

ISET: s_ipSensorConversion

$$xyz = T * sensor$$

2.2591	0.1328	0.1867
1.0159	0.9371	-0.2491
0.0156	-0.1201	1.7118





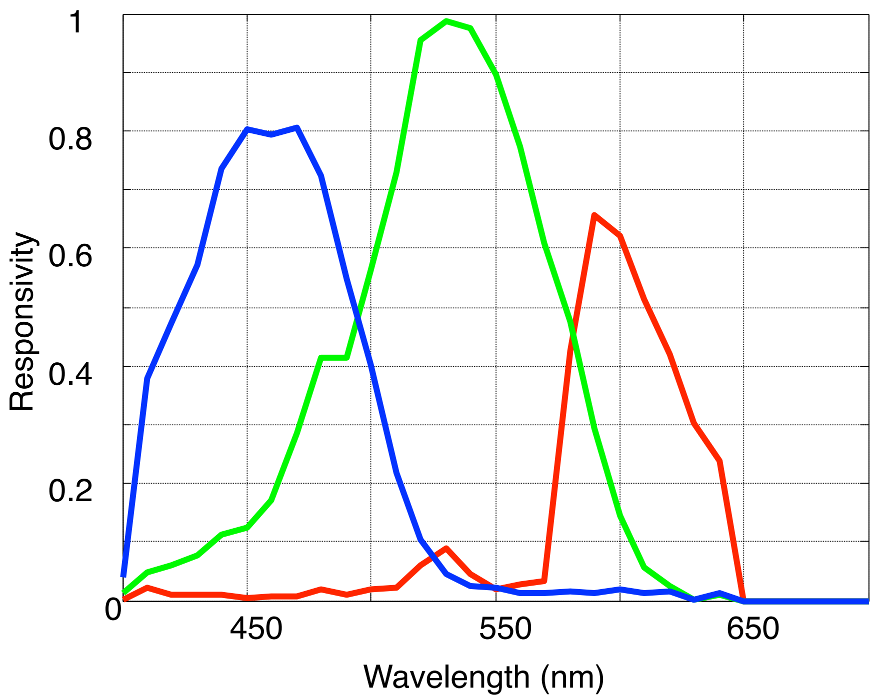
Sensor correction: RGB example

ISET: s_ipSensorConversion

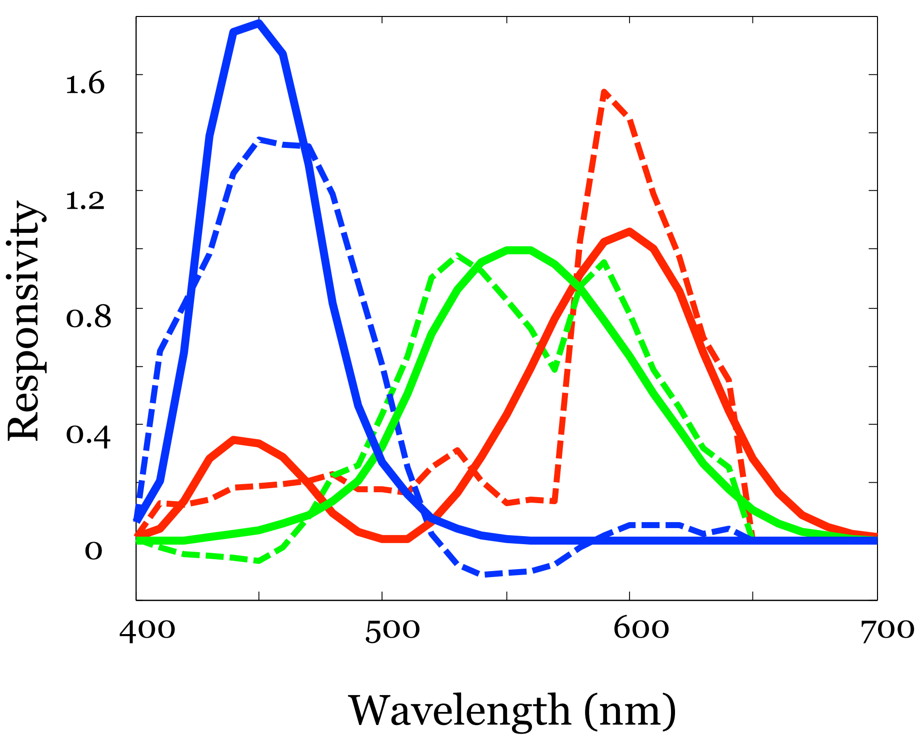
$$xyz = T * sensor$$

$$\begin{matrix} 2.2591 & 0.1328 & 0.1867 \\ 1.0159 & 0.9371 & -0.2491 \\ 0.0156 & -0.1201 & 1.7118 \end{matrix}$$

Original

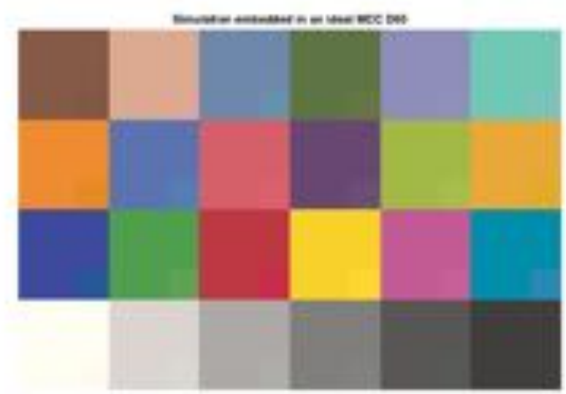


Transformed (---) compared to XYZ

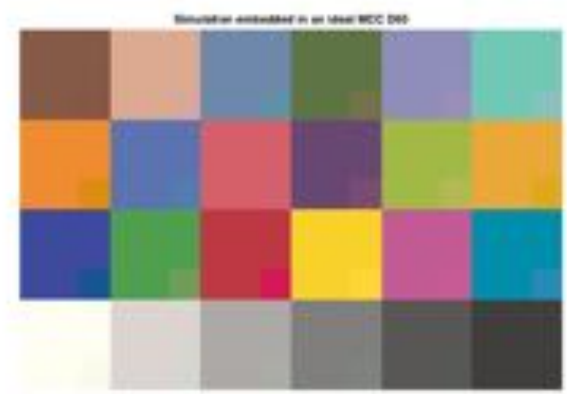


Comparison of desired and obtained XYZ values

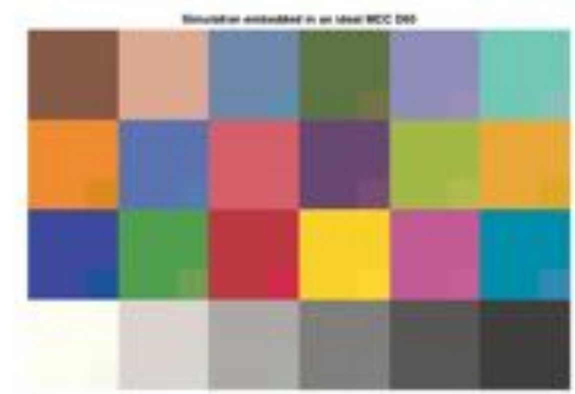
Fluorescent



Tungsten

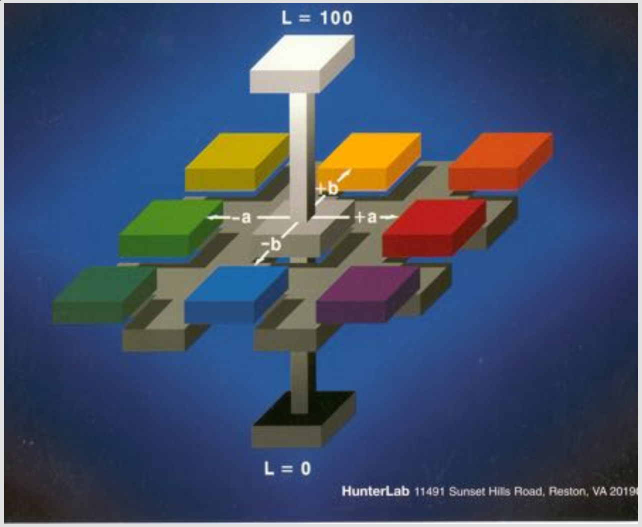


D65



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

Color Difference Metric



- Distance between two colors designed to predict color difference visibility

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

- CIE 2000 replaces this formula with a computer program

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

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

Chroma

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

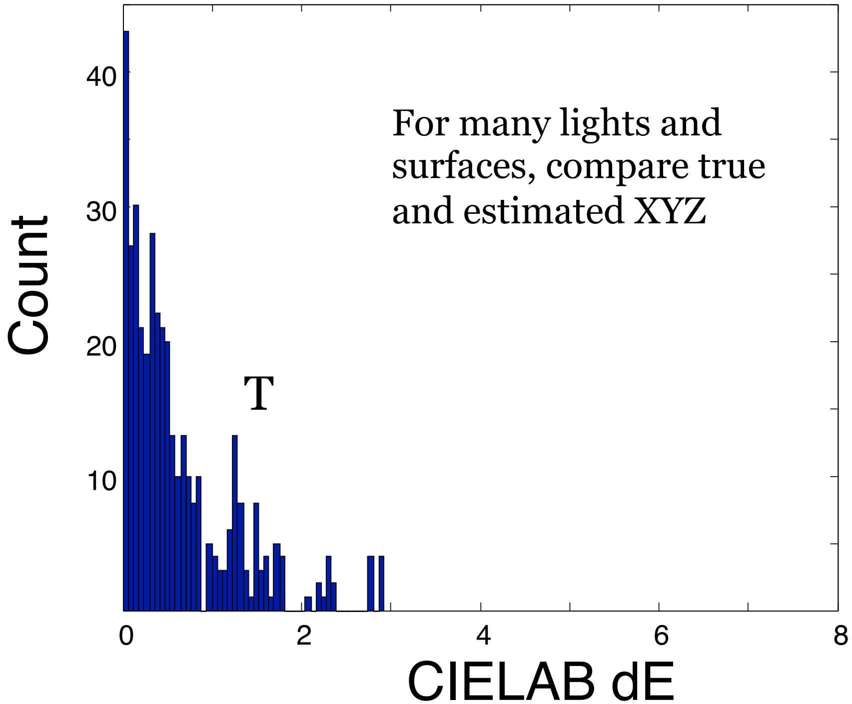
$$b^* = 200 \left\{ \left(\frac{Y}{Y_w} \right)^{1/3} - \left(\frac{Z}{Z_w} \right)^{1/3} \right\}$$

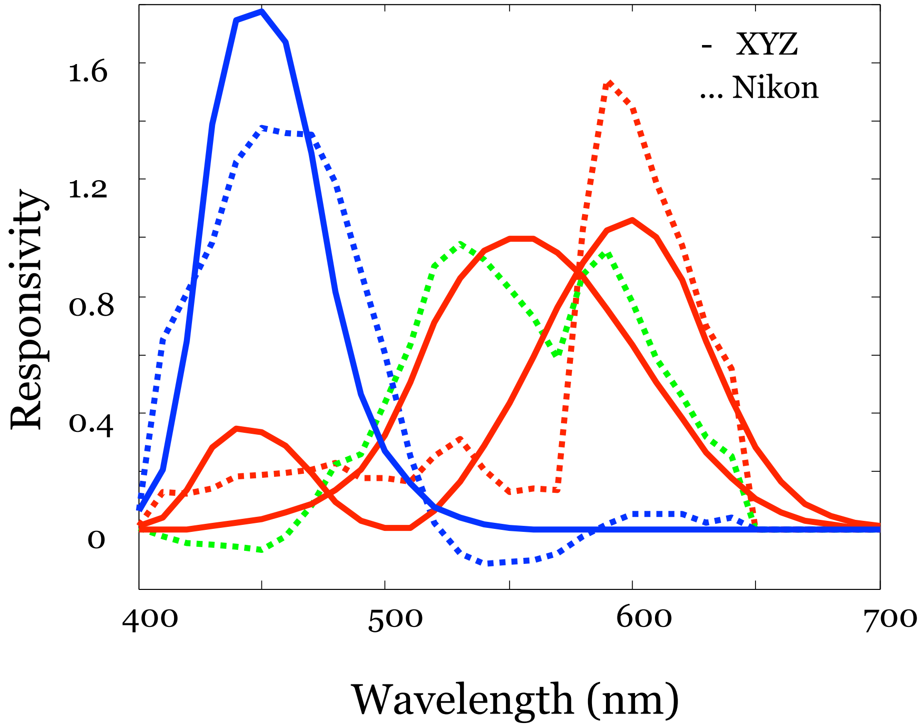
Sensor correction: RGB example

ISET: s_ipSensorConversion

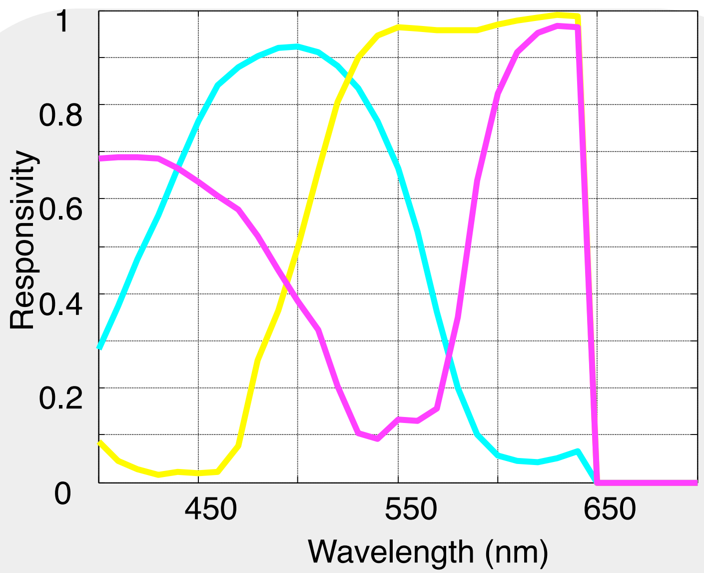
$$xyz = T * sensor$$

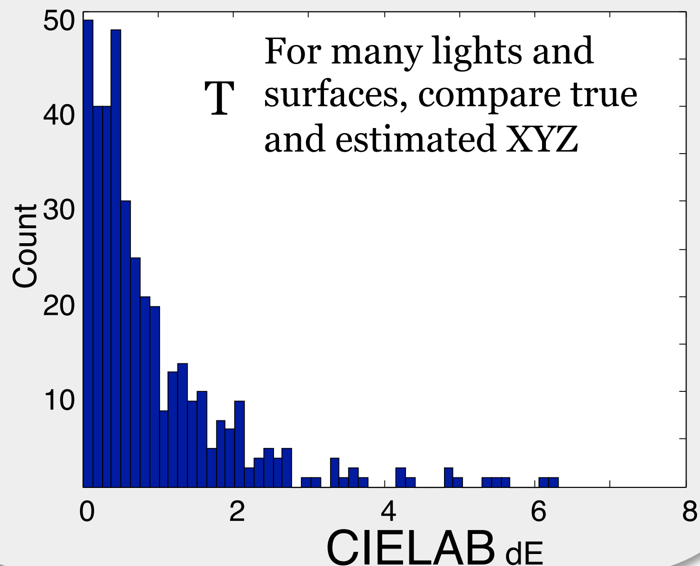
2.2591	0.1328	0.1867
1.0159	0.9371	-0.2491
0.0156	-0.1201	1.7118





Sensor correction: CMY example

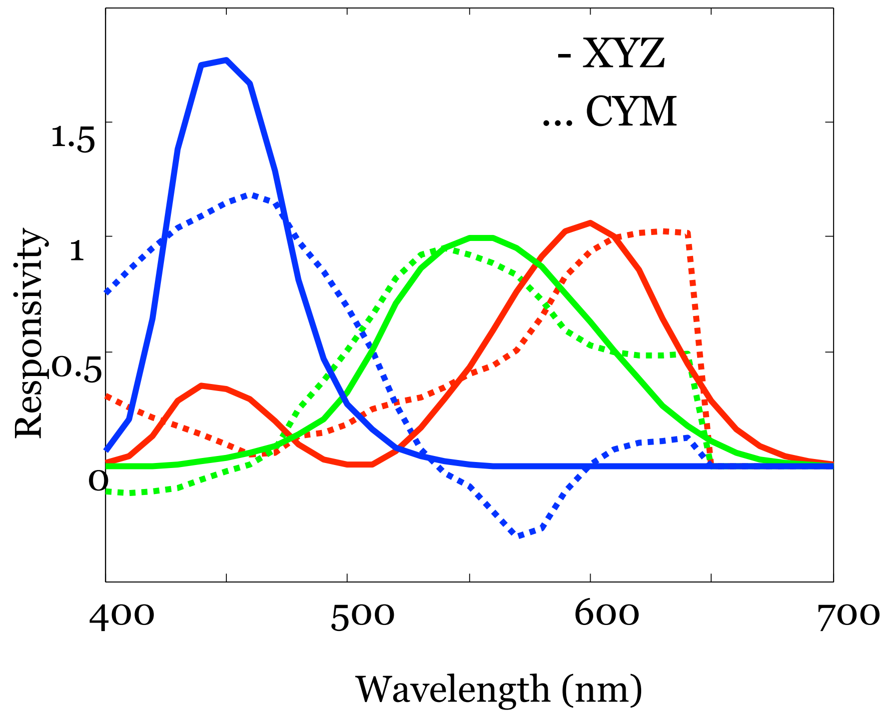




ISET: s_sensorCorrection

$$xyz = T * sensor$$

-0.2970	0.5812	0.4605
0.2581	0.8520	-0.3963
0.8259	-0.7436	0.8038



Why do we need color management?

- Sensor Correction
  - Spectral sensitivities of the camera do not match the spectral sensitivities of the human eye
  - Sensors are not "colorimetric"

- Illuminant correction
  - Camera does not adjust the gain of RGB sensors with changes in the spectral power of the illumination in the same way as the visual system adjusts the gain of the LMS sensors
  - Cameras do not have "color constancy"

- Display rendering and Color gamut mapping
  - Displays may have different color primaries and gamma
  - Range of colors a camera image processing pipeline can produce does not match the range of colors a display or printer can produce

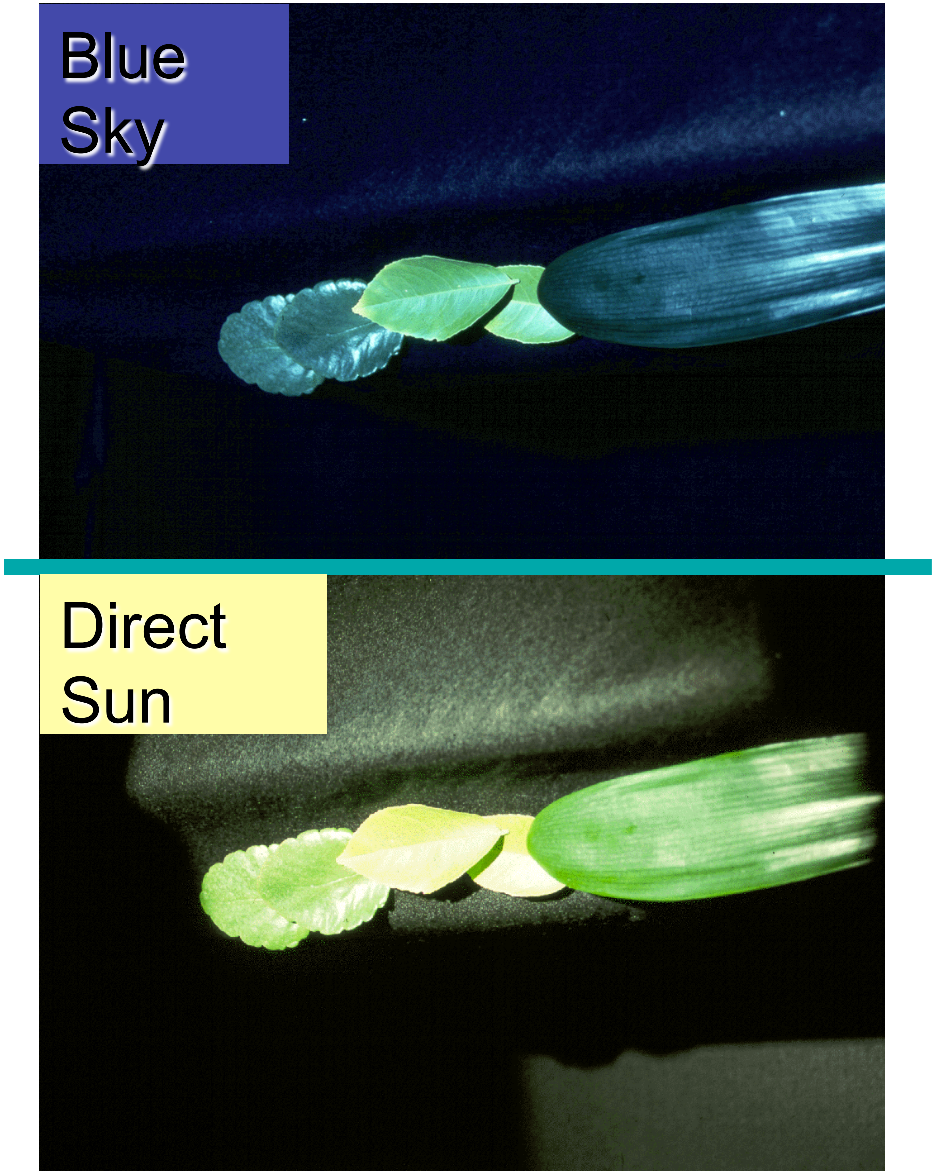
Illuminant Correction

Color Constancy: The human visual system adapts to scene illumination

Illuminant Correction: Transform camera sensor values for different scene illumination

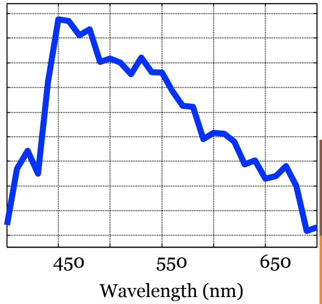
Illuminant correction

- 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

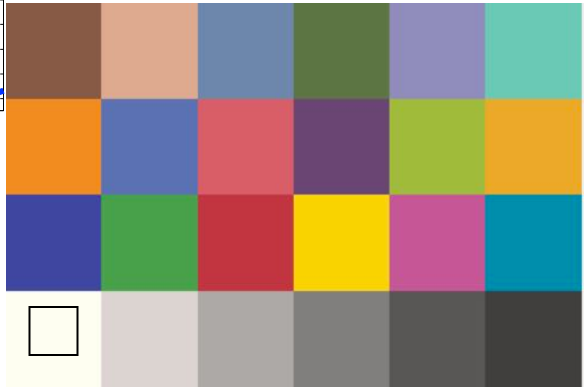


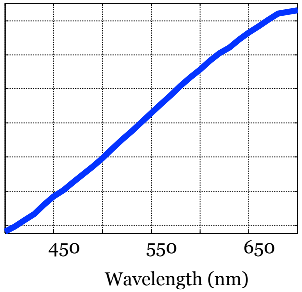
Illuminant correction

- 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

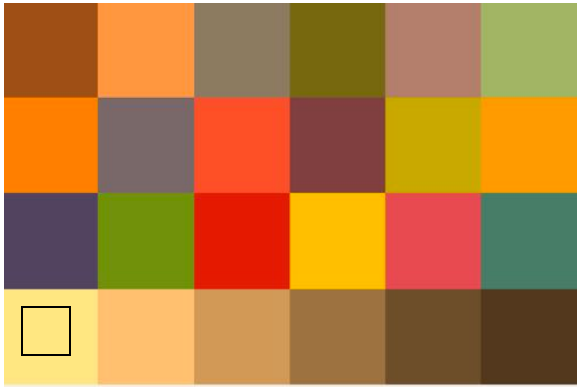


D65





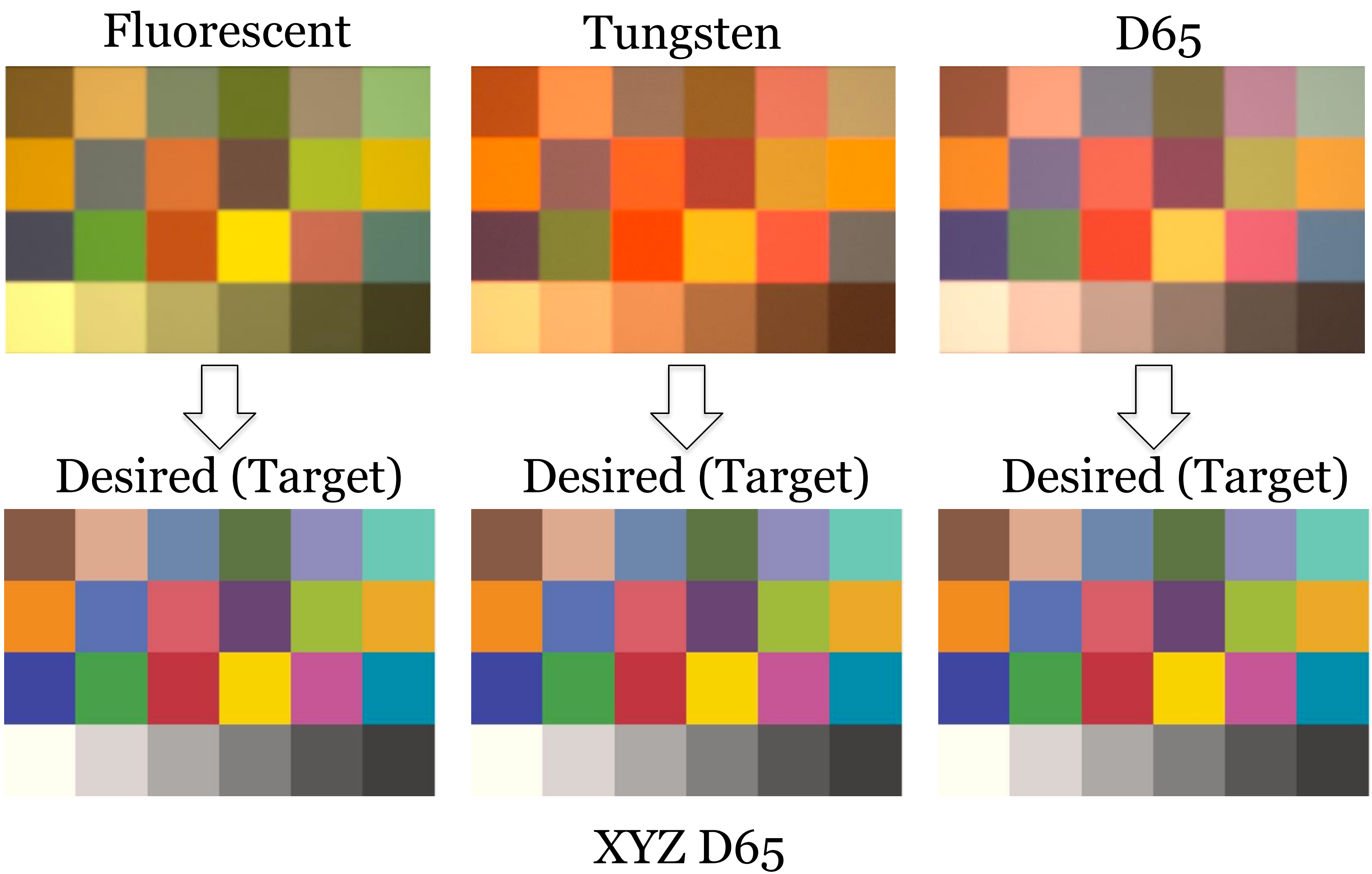
Tungsten



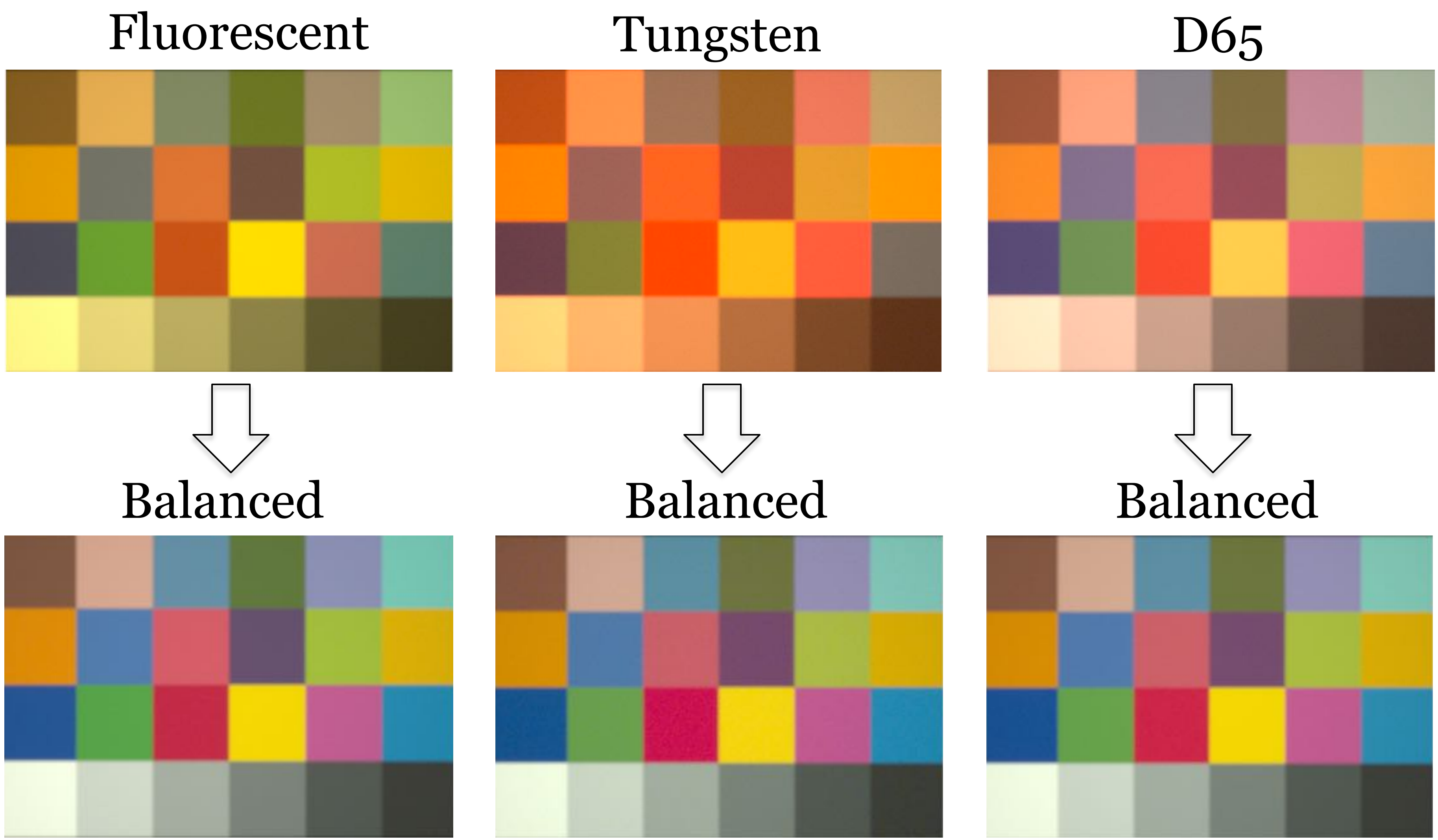
Illuminant Correction Strategy

- 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



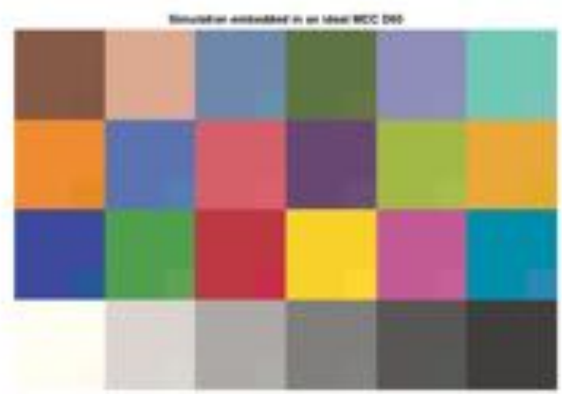
Illuminant Correction



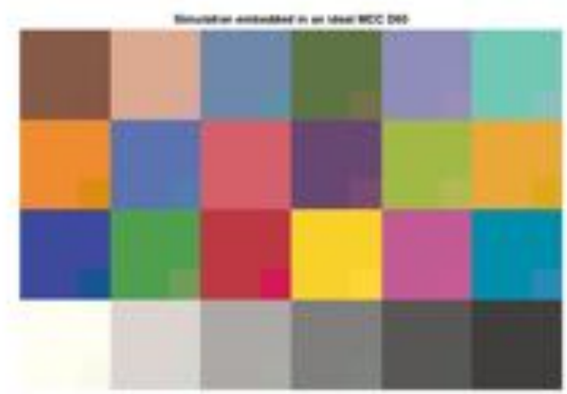
XYZ D65

Comparison of Desired and Rendered Colors

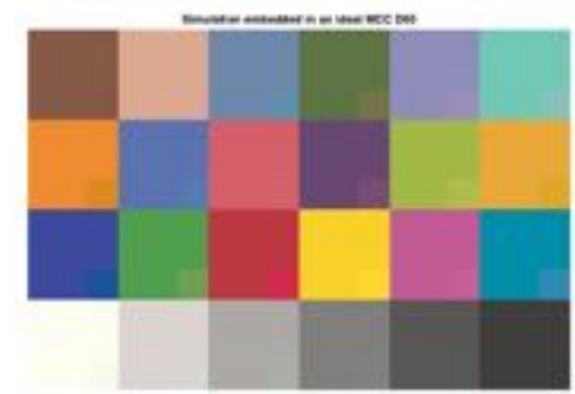
Fluorescent



Tungsten



D65

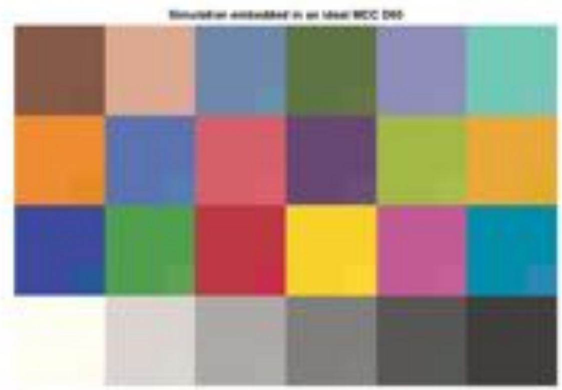


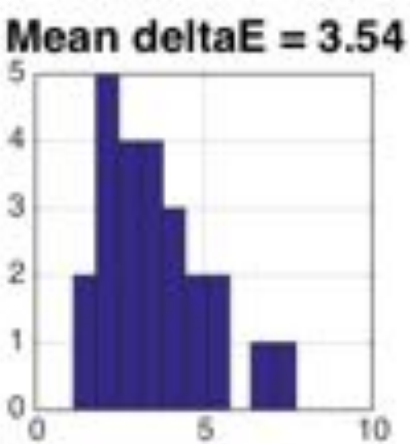
How would you quantify the visible difference between the desired and rendered colors?

Comparison of Desired and Rendered Colors

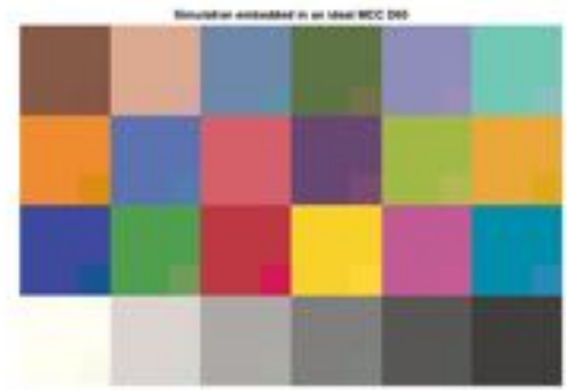
Fluorescent	Tungsten	D65

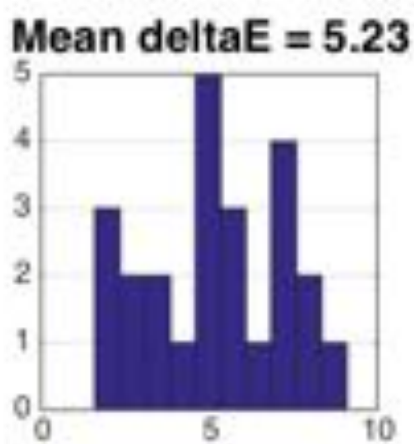
Fluorescent



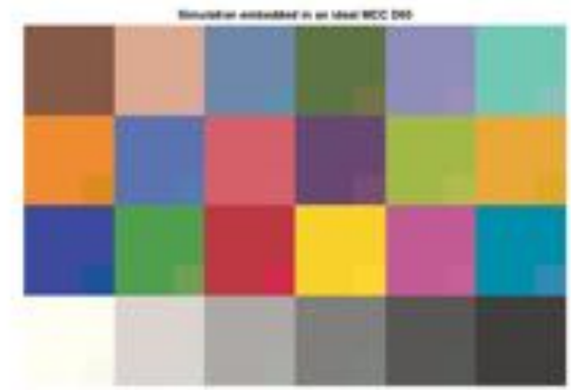


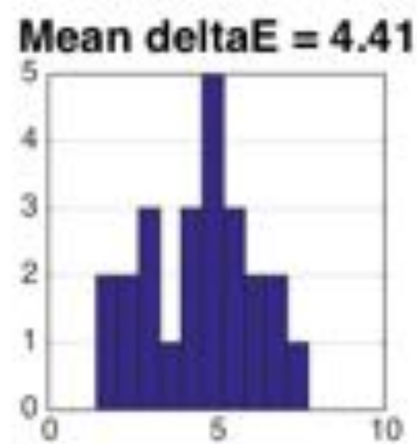
Tungsten





D65





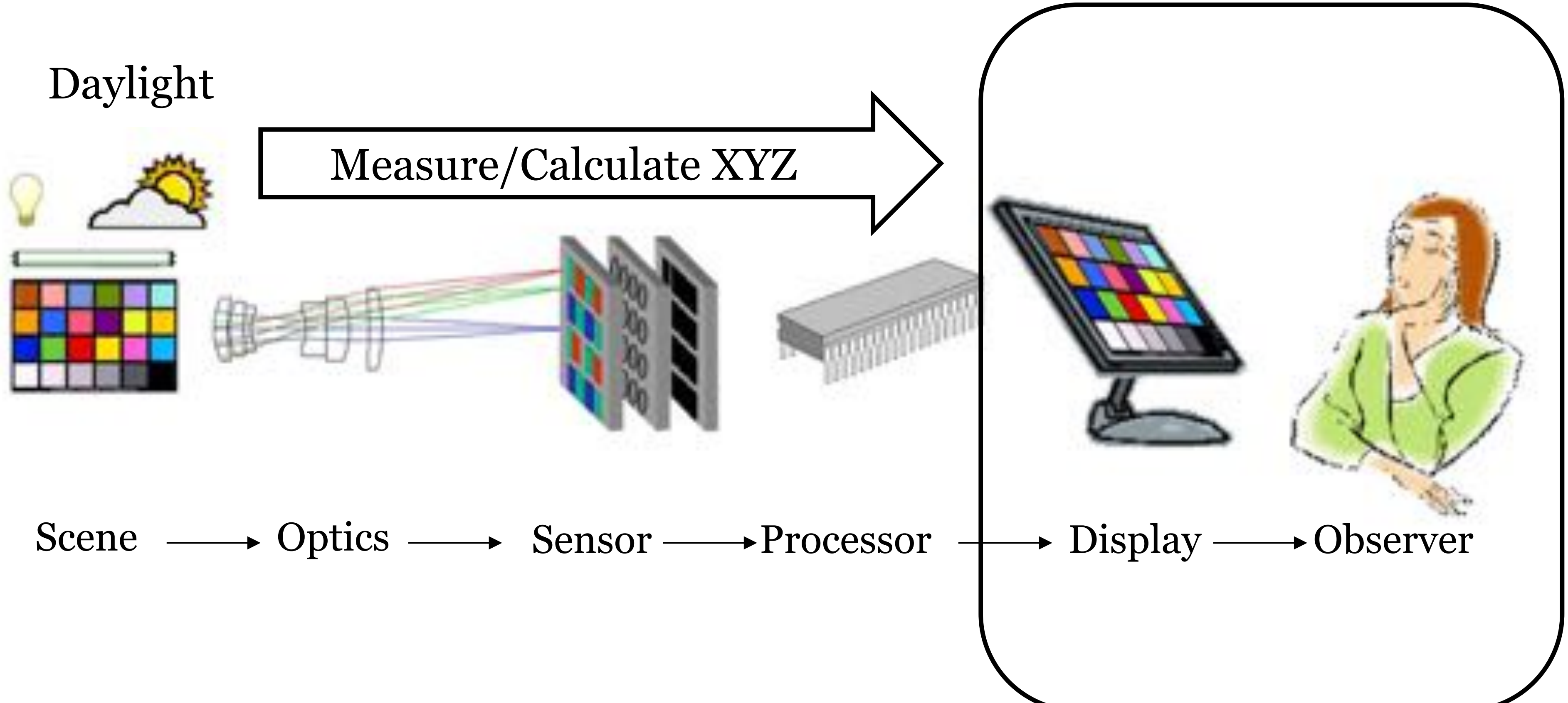
Why do we need color management?

- Sensor Correction
  - Spectral sensitivities of the camera do not match the spectral sensitivities of the human eye
  - Sensors are not "colorimetric"

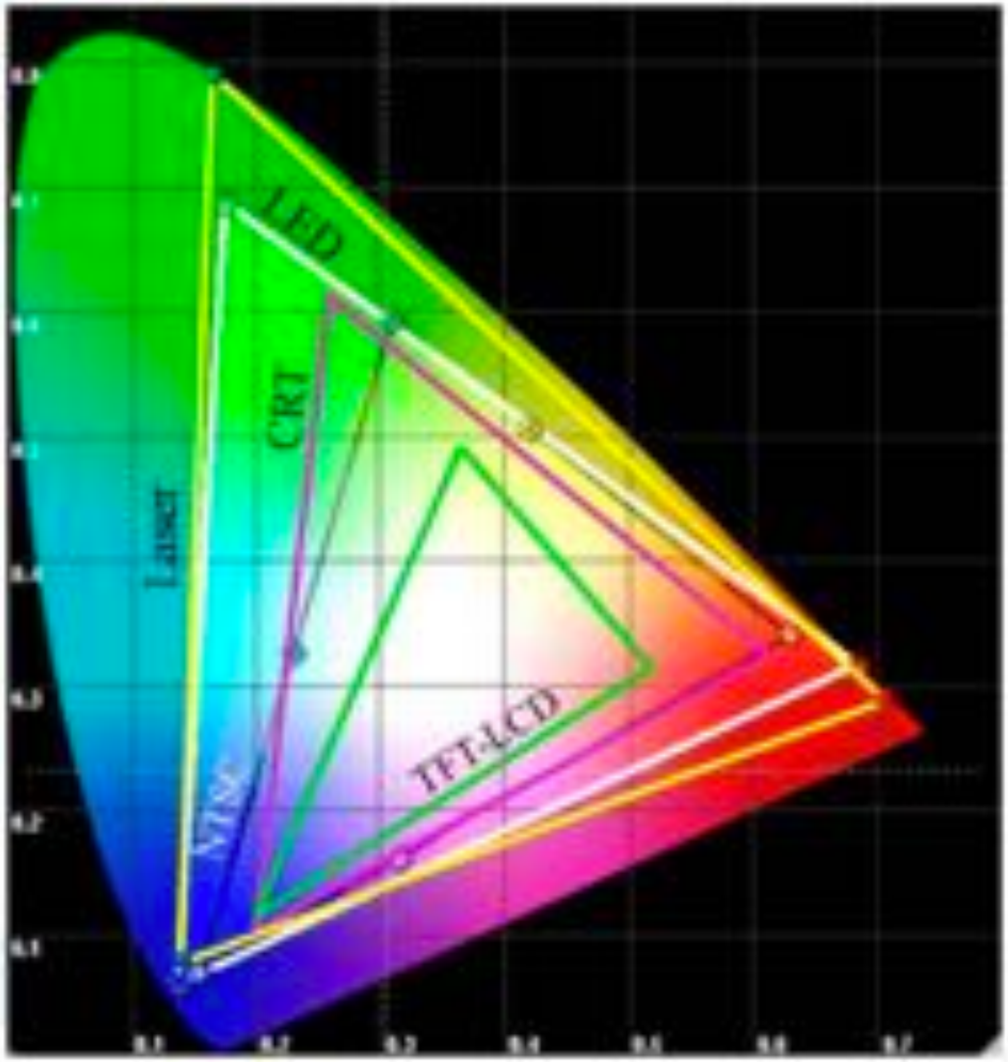
- Illuminant correction
  - Camera does not adjust the gain of RGB sensors with changes in the spectral power of the illumination in the same way as the visual system adjusts the gain of the LMS sensors
  - Cameras do not have "color constancy"

- Display rendering and Color gamut mapping
  - Displays may have different color primaries and gamma
  - Range of colors a camera image processing pipeline can produce does not match the range of colors a display or printer can produce

Display Rendering and Gamut Mapping



Displays

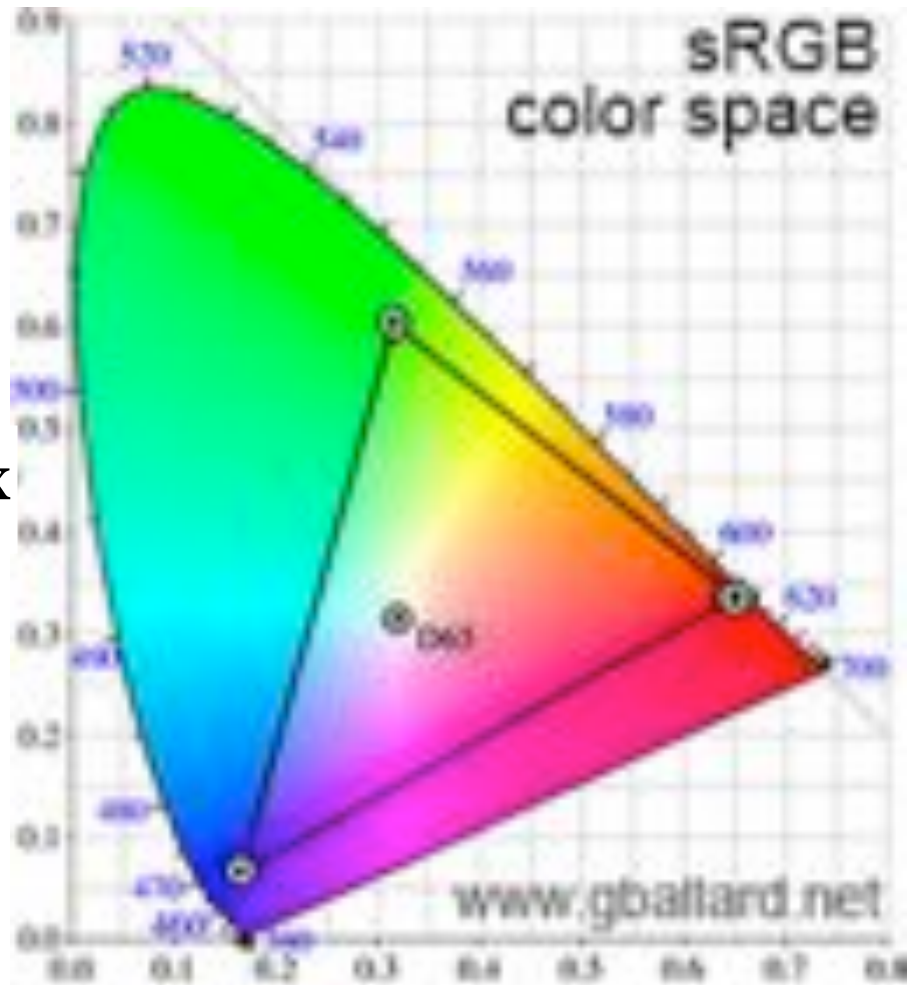


https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-18-5-4655&id=196021

Display Gamut

sRGB is the gamut for a Sony Trinitron CRT display

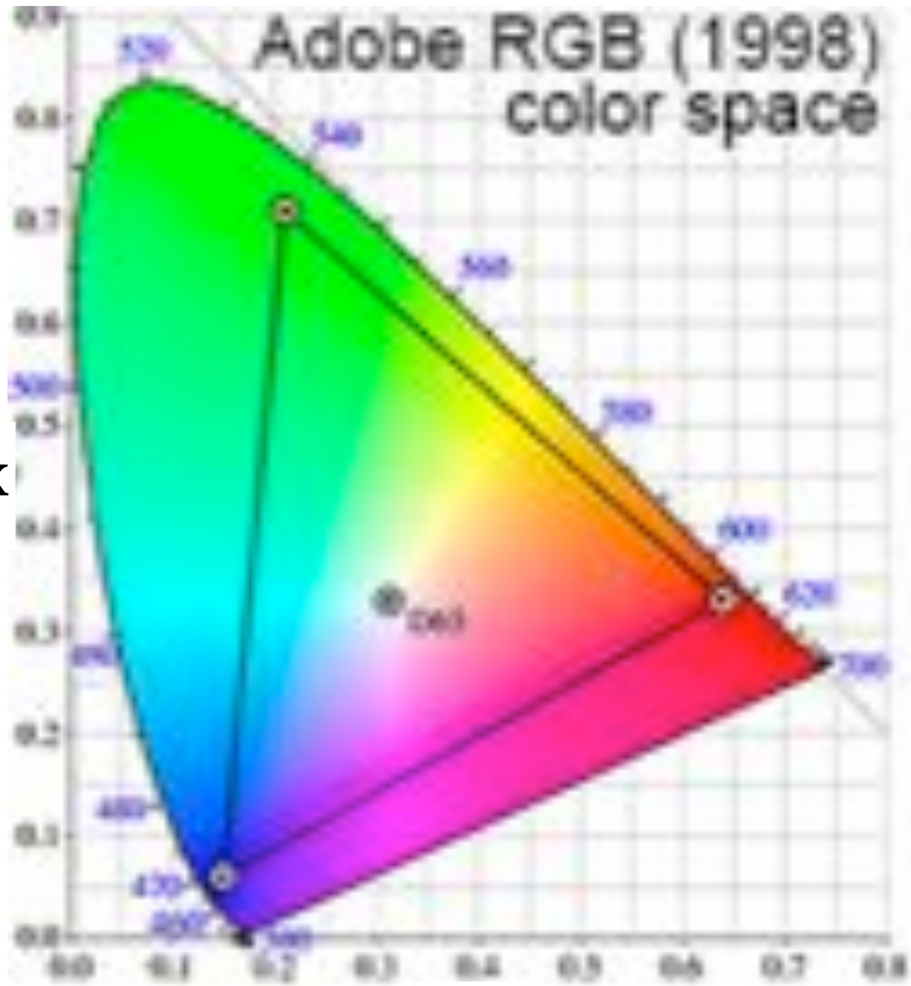
x



y

Adobe introduced a gamut that can include more displays

x



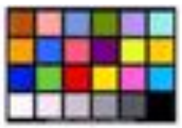
y

Display Gamut

Return to discuss this later .....

For now, let's assume you have an sRGB display .... We will return to discuss the history behind this decisions and the ramifications it has for color reproduction

Camera-Display Matching



Camera measures RGB and we convert to a value related to CIE-XYZ → *Estimate monitor linear intensities that match CIE-XYZ related value*

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} \text{———} \\ \text{———} \\ \text{———} \end{pmatrix}}_{\text{XYZ functions}} \begin{pmatrix} | & | & | \\ & \text{Monitor primaries} & \\ | & | & | \end{pmatrix} \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} & XM & \end{pmatrix} \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}$$

Camera-Display Matching

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} & & \\ & XM & \\ & & \end{pmatrix} \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}$$

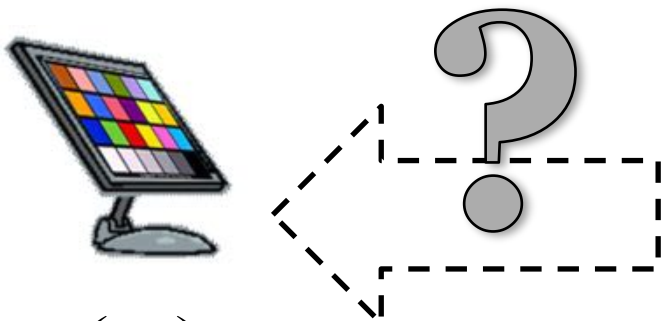
XYZ given RGB

RGB given XYZ

$$\begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix} = \begin{pmatrix} & & \\ & (XM)^{-1} & \\ & & \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Camera-Display Matching

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} & & \\ & XM & \\ & & \end{pmatrix} \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}$$



$$\begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix} = \begin{pmatrix} & & \\ & (XM)^{-1} & \\ & & \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Camera-Display Matching

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} & & \\ & XM & \\ & & \end{pmatrix} \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}$$

Correct for display gamma

$$\begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix} = \begin{pmatrix} & & \\ & (XM)^{-1} & \\ & & \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

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**
- Apply gamma to convert linear RGB to DAC values

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

Matrix coefficients depend on display

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

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

Matrix based color adjustment methods
Device-dependent diagonal

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

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-dependent 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-dependent diagonal

$$\begin{array}{c} \text{Display} \\ \\ \\ \end{array} \begin{pmatrix} R_1' & & R_N' \\ G_1' & \mathrm{L} & G_N' \\ B_1' & & B_N' \end{pmatrix} = \begin{pmatrix} & & \\ & C_E & \\ & & \end{pmatrix} \begin{pmatrix} R_1 & & R_N \\ G_1 & \mathrm{L} & G_N \\ B_1 & & B_N \end{pmatrix} \begin{array}{c} \text{Camera} \\ \\ \\ \end{array}$$

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

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

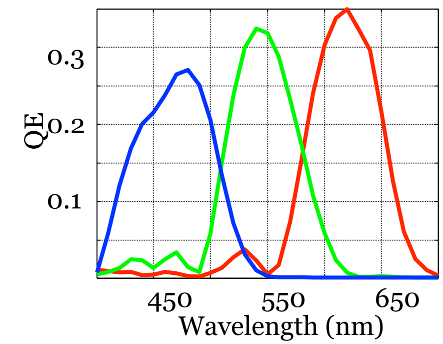
Camera

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

Sensor



Transform sensor data to calibrated color (XYZ)

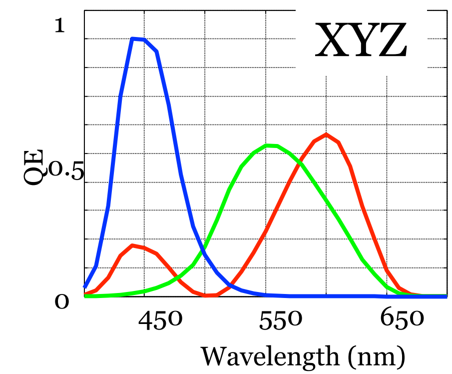
L

Estimate illuminant

E

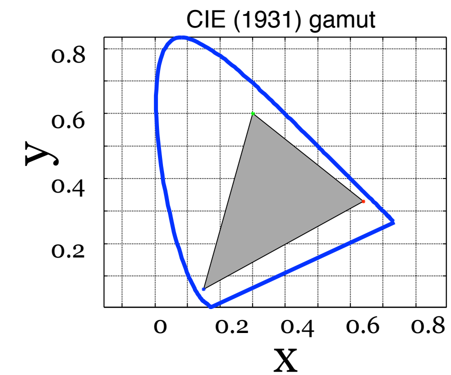
Apply transformation for illuminant E

C_E



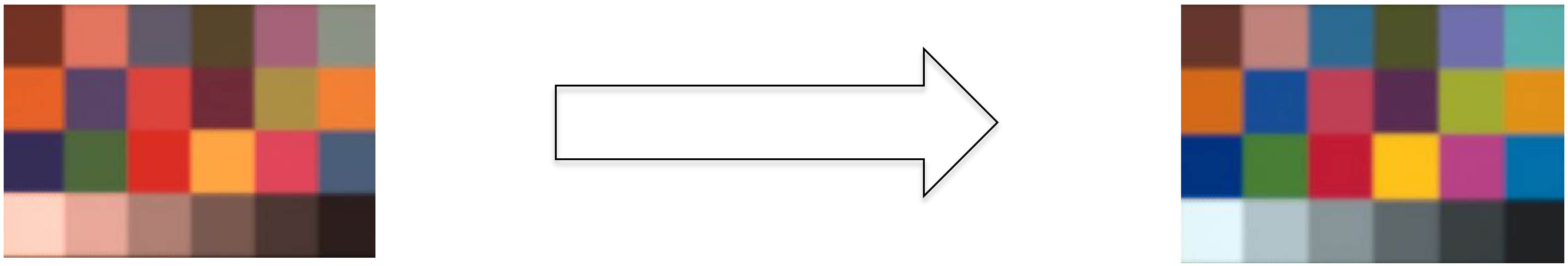
Convert to display space, adjusting for gamma and gamut

M



Display

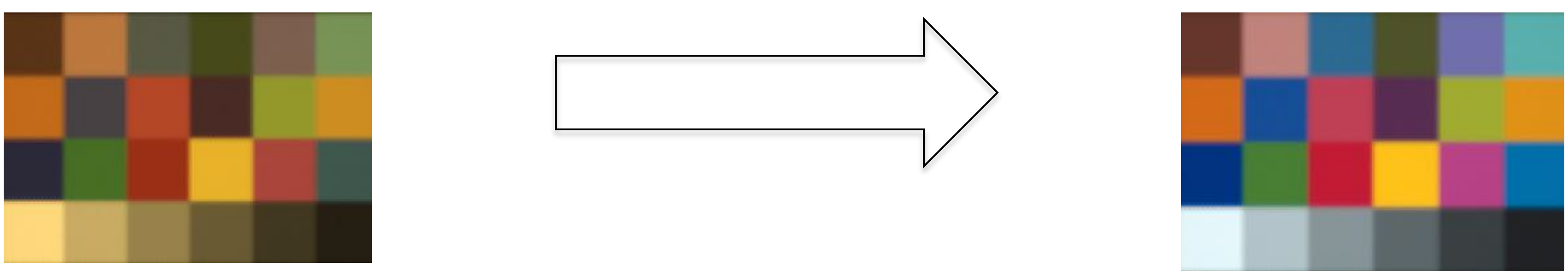
Engineering practice



1. Capture test surfaces under different lights

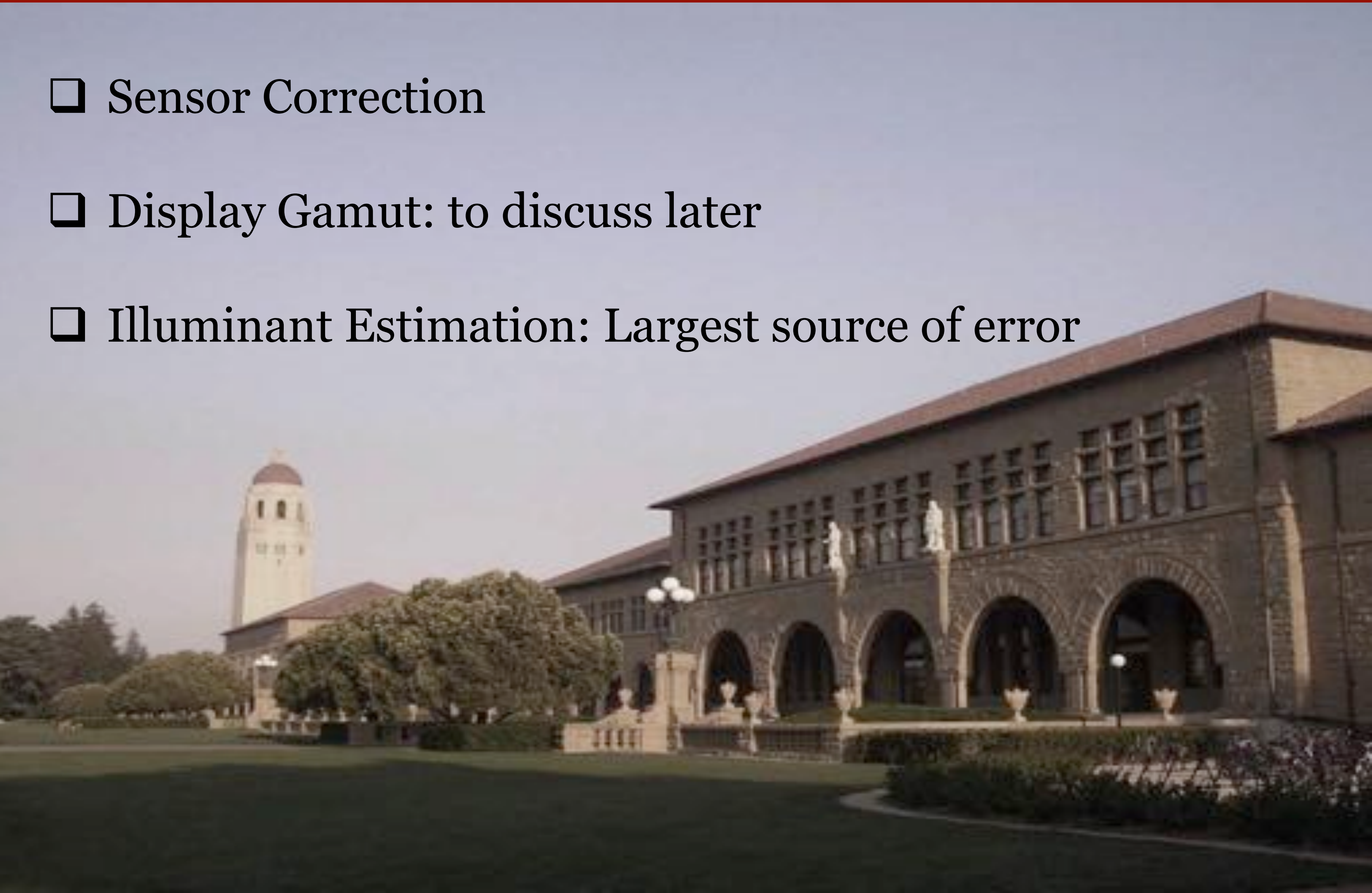
2. Linear regression to find 3x3 from camera to display RGB that minimizes XYZ

3. Report Delta E measure of color accuracy



Sources of Error

- Sensor Correction
- Display Gamut: to discuss later
- Illuminant Estimation: Largest source of error



Gray World

1. Illuminant Estimation

- Assume m_R , m_G , m_B is the sensor response to the illuminant (gray surface)

2. Color Conversion

- Skip this step? Often times there is no sensor conversion and correction is done in sensor space
- Feng et al study shows that conversion to XYZ produces better results

3. Illuminant Transformation

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

4. Display Rendering

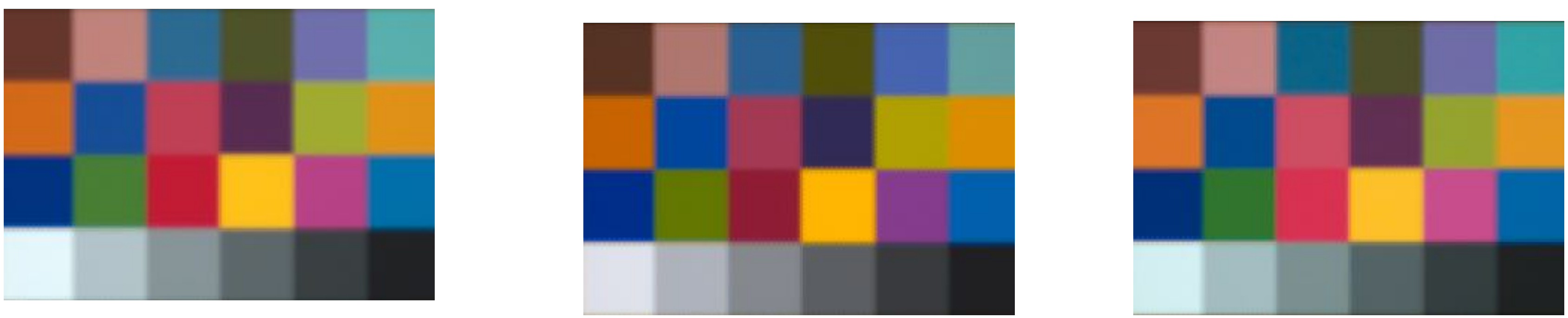
- 3x3 matrix to convert corrected RGB to display RGB
- Often times there is no conversion to display space (this can cause problems)
- Apply display gamma

Gray World in Sensor and XYZ Space

Color balancing in XYZ space produces better results

Gray World in Sensor Space (i.e no sensor conversion)

D65 Fluorescent Tungsten



Gray World in XYZ Space (i.e. sensor conversion)

[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 Estimation

Best Information about the illuminant are the bright pixels

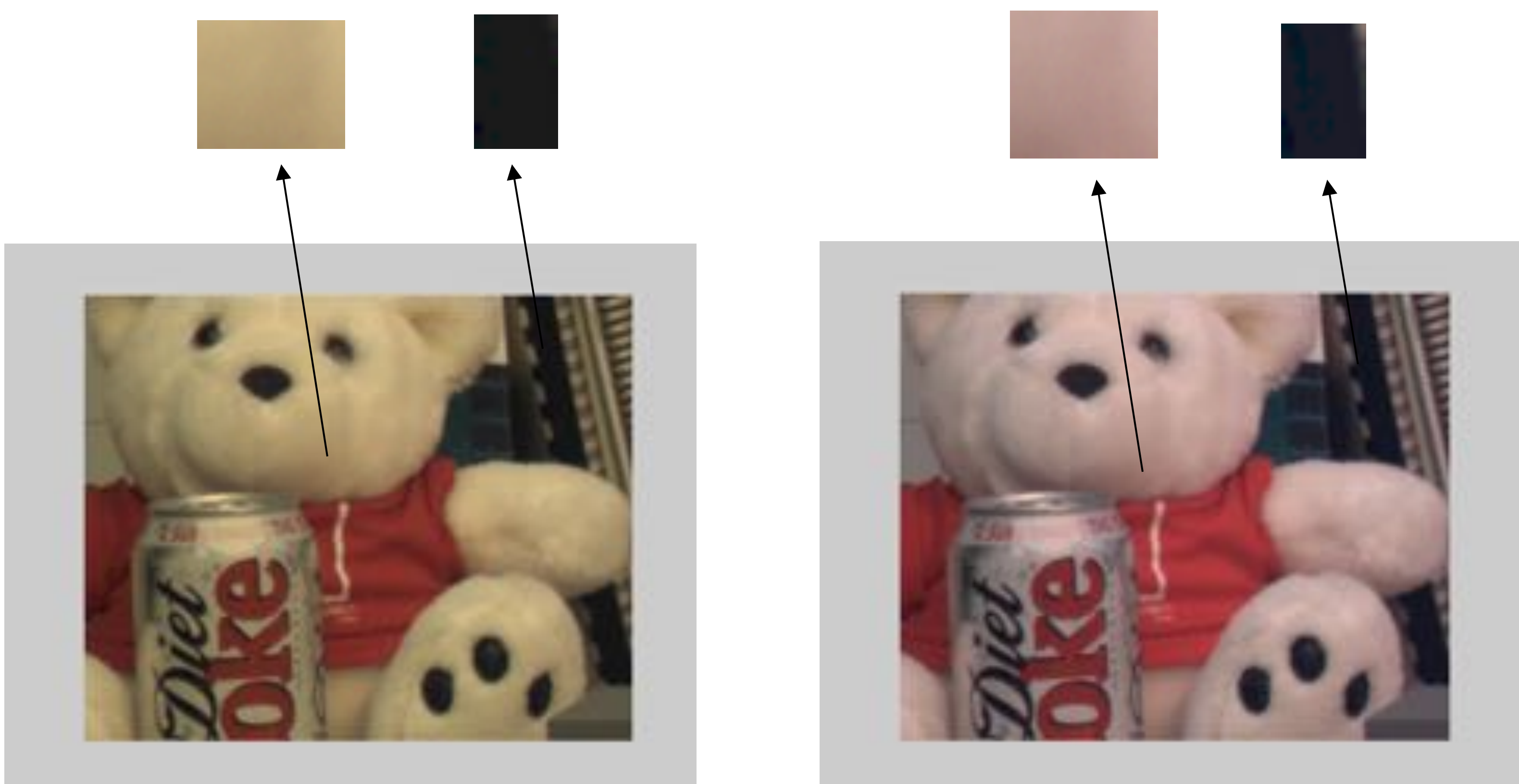
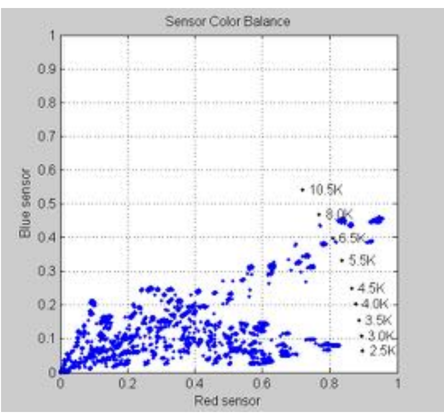


Image Statistics

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

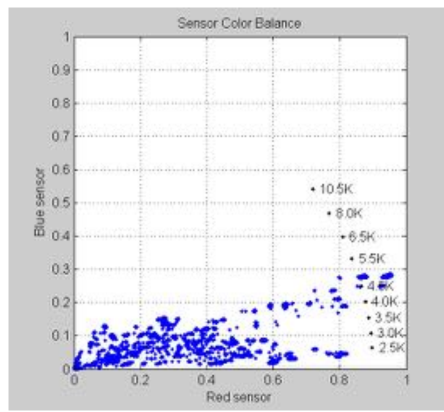
2. Select appropriate stored color balancing matrix
(combines color conversion, illuminant transformation and display rendering)

D65



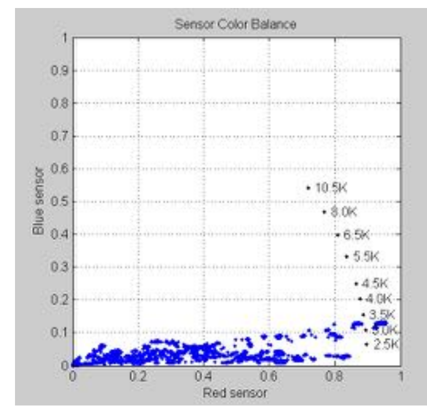
$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} \begin{pmatrix} R_{D65} \\ G_{D65} \\ B_{D65} \end{pmatrix}$$

Fluorescent



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

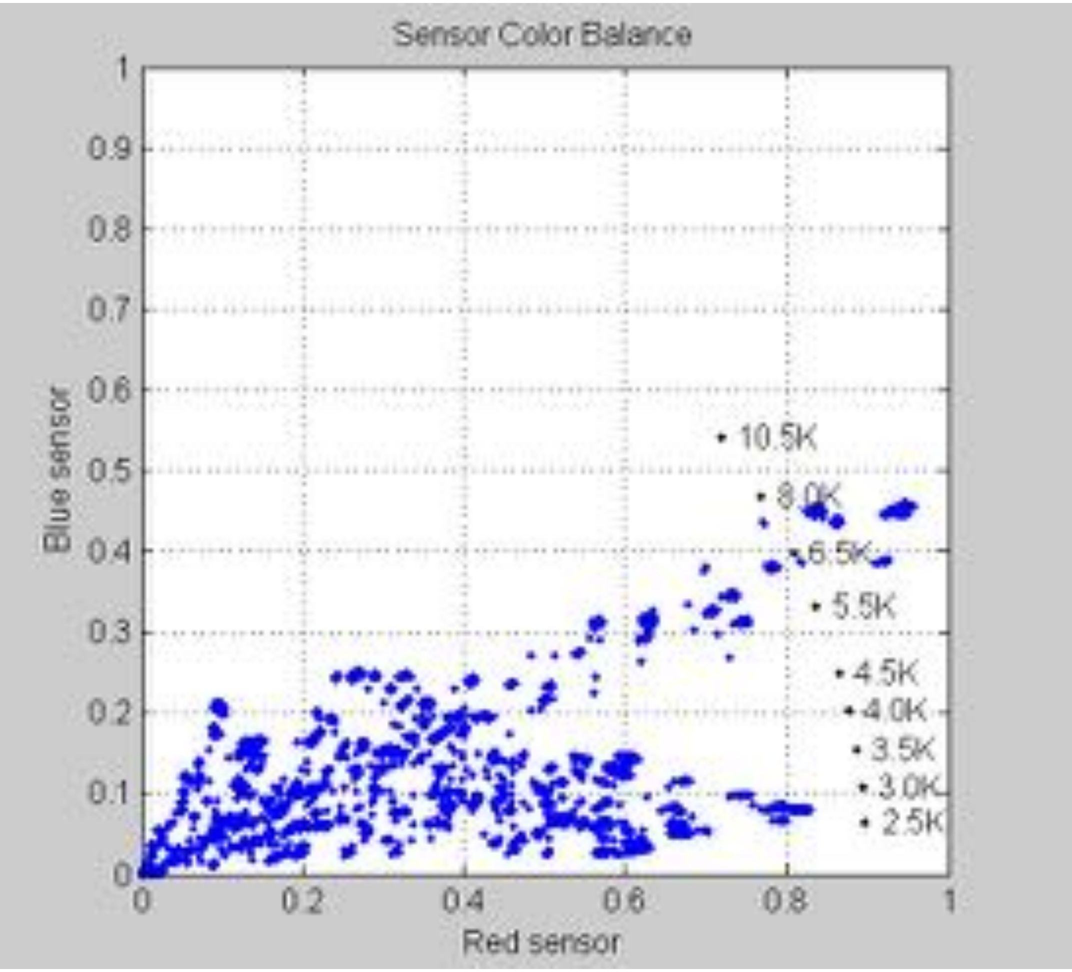
Tungsten



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

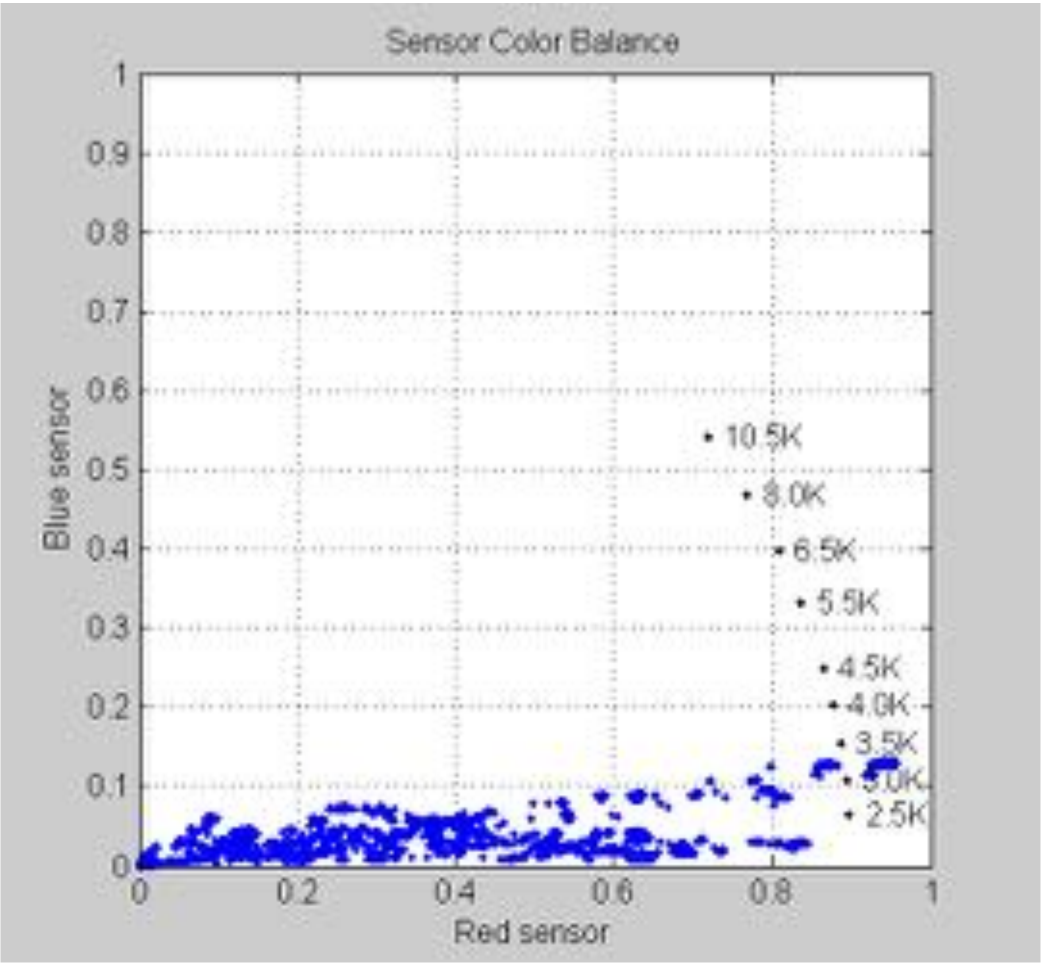
R/B Illuminant Signature

D65 Illuminant



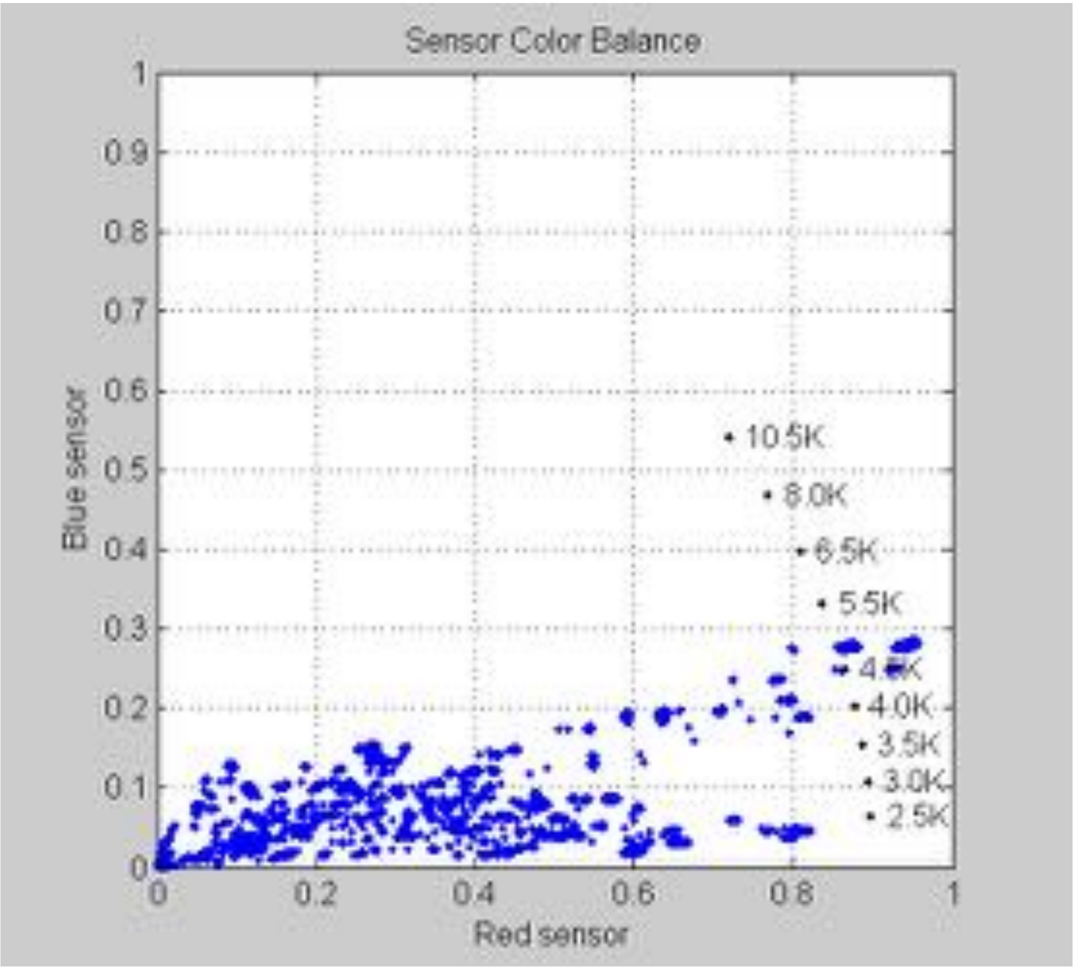
R/B Illuminant Signature

Tungsten Illuminant



R/B Illuminant Signature

Fluorescent Illuminant



Two-capture color balancing
(DiCarlo et al., CIC, 2001)

Multiple captures

Illuminant bracketing

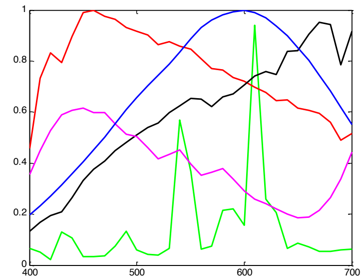
Single image

Known illuminant image

Space-varying illuminant estimation

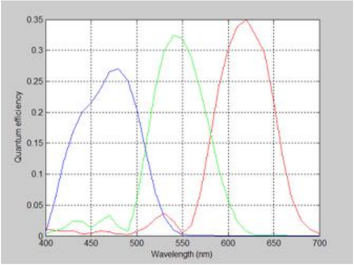
Sources of Error

1. Illuminant estimation

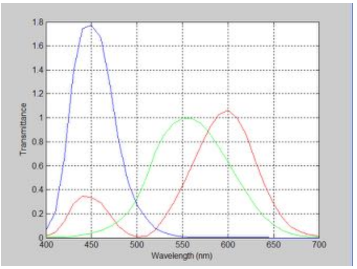


Wavelength(nm)

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



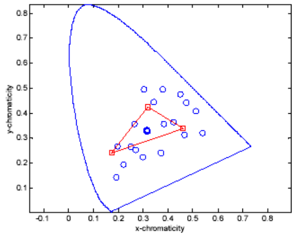
Sensors



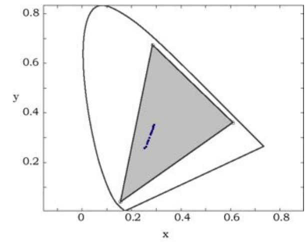
XYZ

3. Illuminant transformation

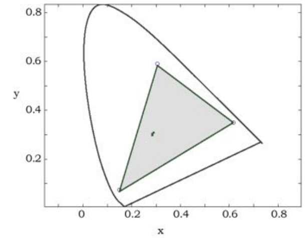
4. Display gamut



Cell phone display



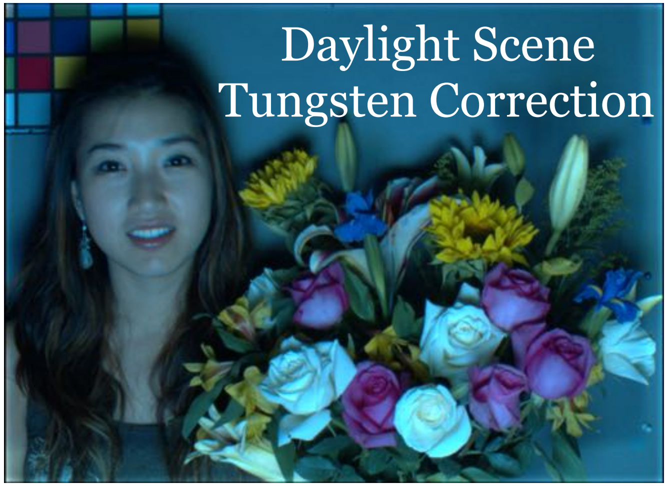
LCD



CRT

Illuminant Estimation

Bad estimate



Good estimate

Why do we need color management?

- Sensor Correction
    - Spectral sensitivities of the camera do not match the spectral sensitivities of the human eye
    - Sensors are not "colorimetric"

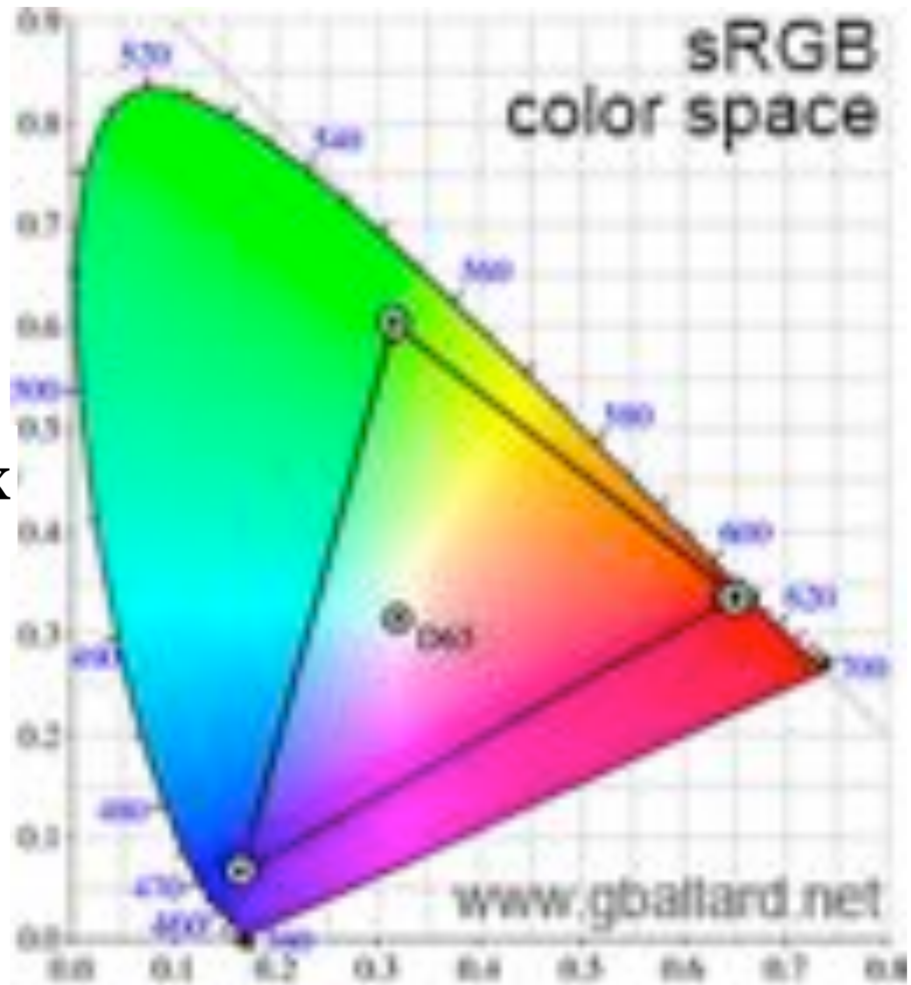
- Illuminant correction
    - Camera does not adjust the gain of RGB sensors with changes in the spectral power of the illumination in the same way as the visual system adjusts the gain of the LMS sensors
    - Cameras do not have "color constancy"

- Display rendering and Color gamut mapping
    - Displays may have different color primaries and gamma
    - Range of colors a camera image processing pipeline can produce does not match the range of colors a display or printer can produce

Display Gamut

sRGB is the gamut for a Sony Trinitron CRT display

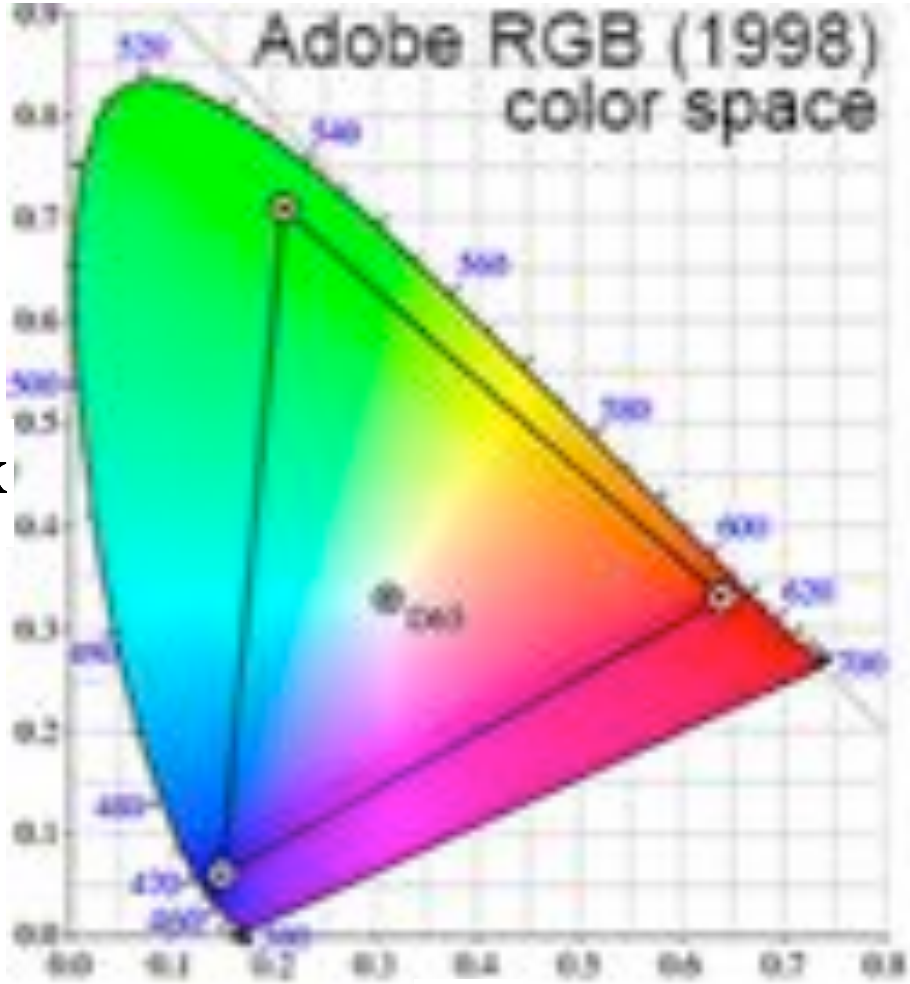
x



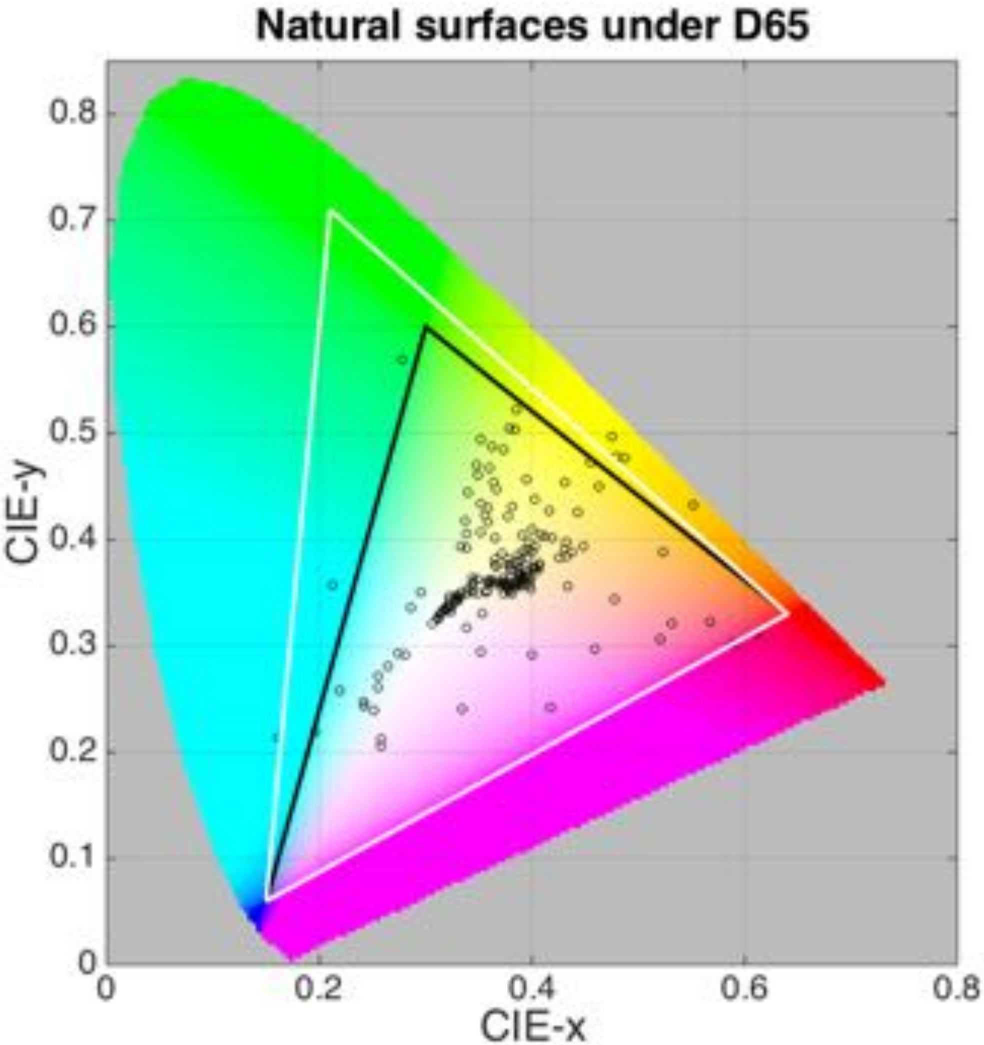
y

Adobe introduced a gamut that can include more displays

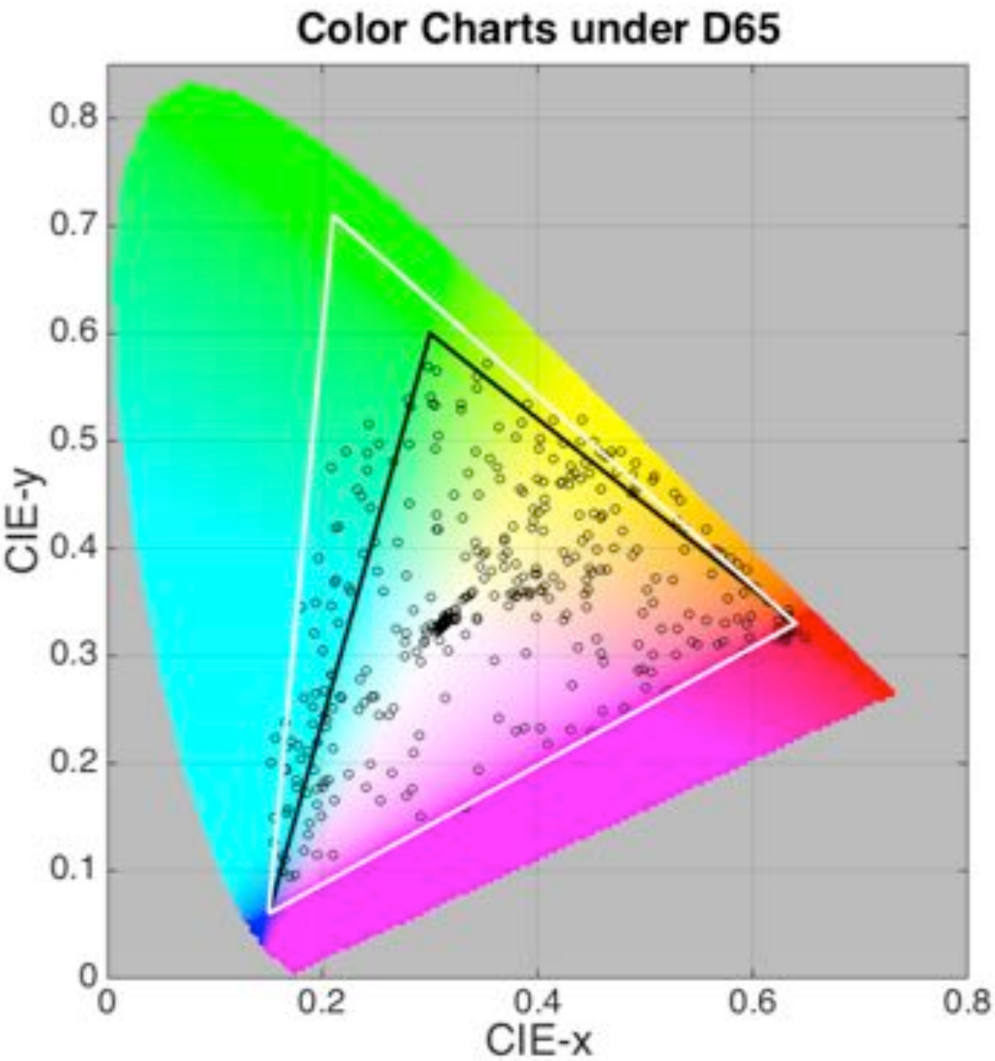
x



y

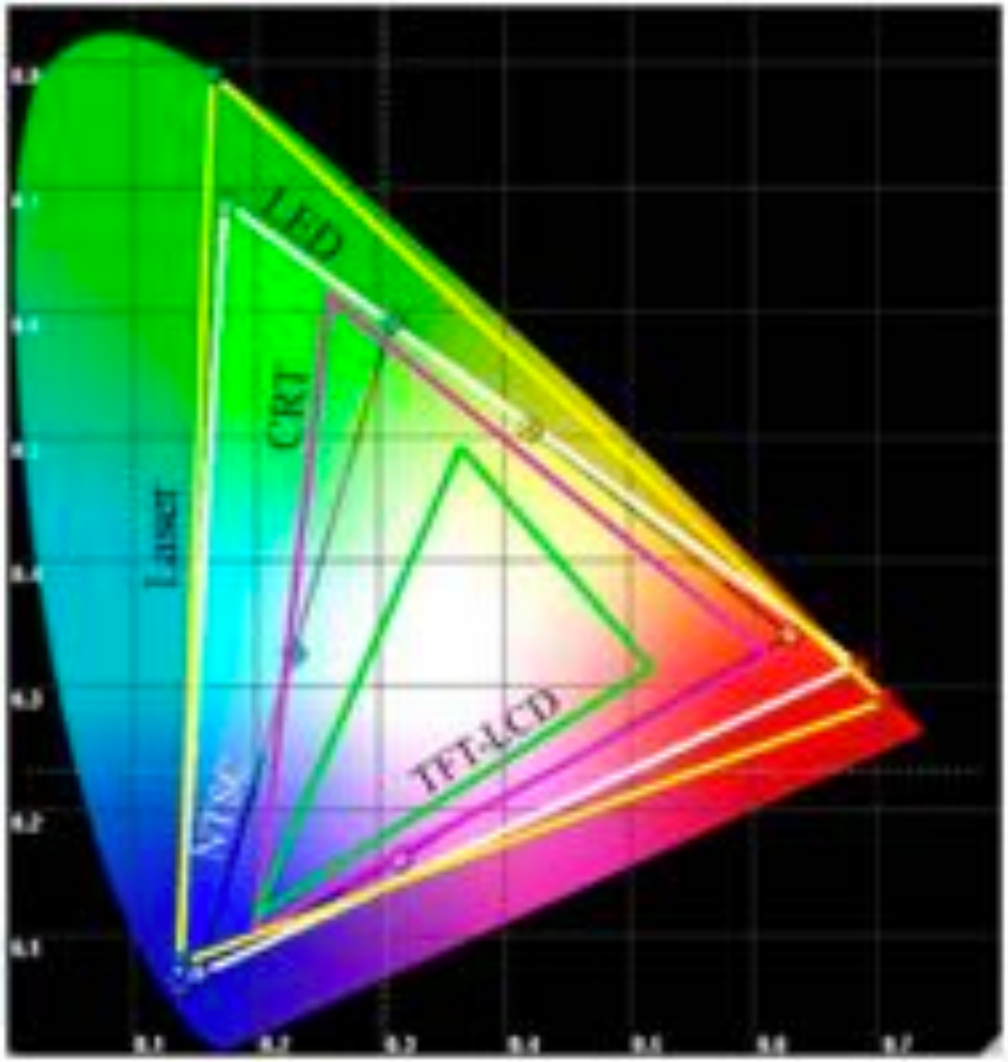


Natural surfaces under D65
CIE-y
CIE-x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
0.2
0.4
0.6
0.8



Color Charts under D65
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
CIE-y
0
0.2
0.4
0.6
0.8
CIE-x

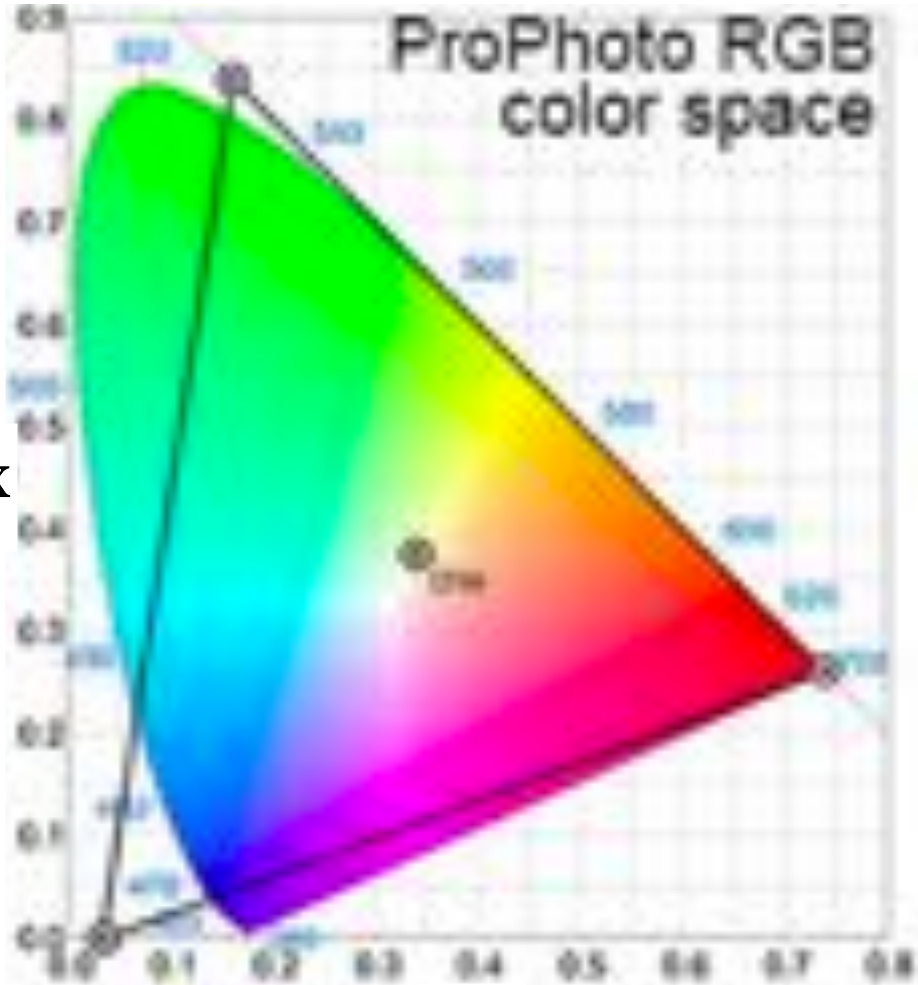
Displays



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

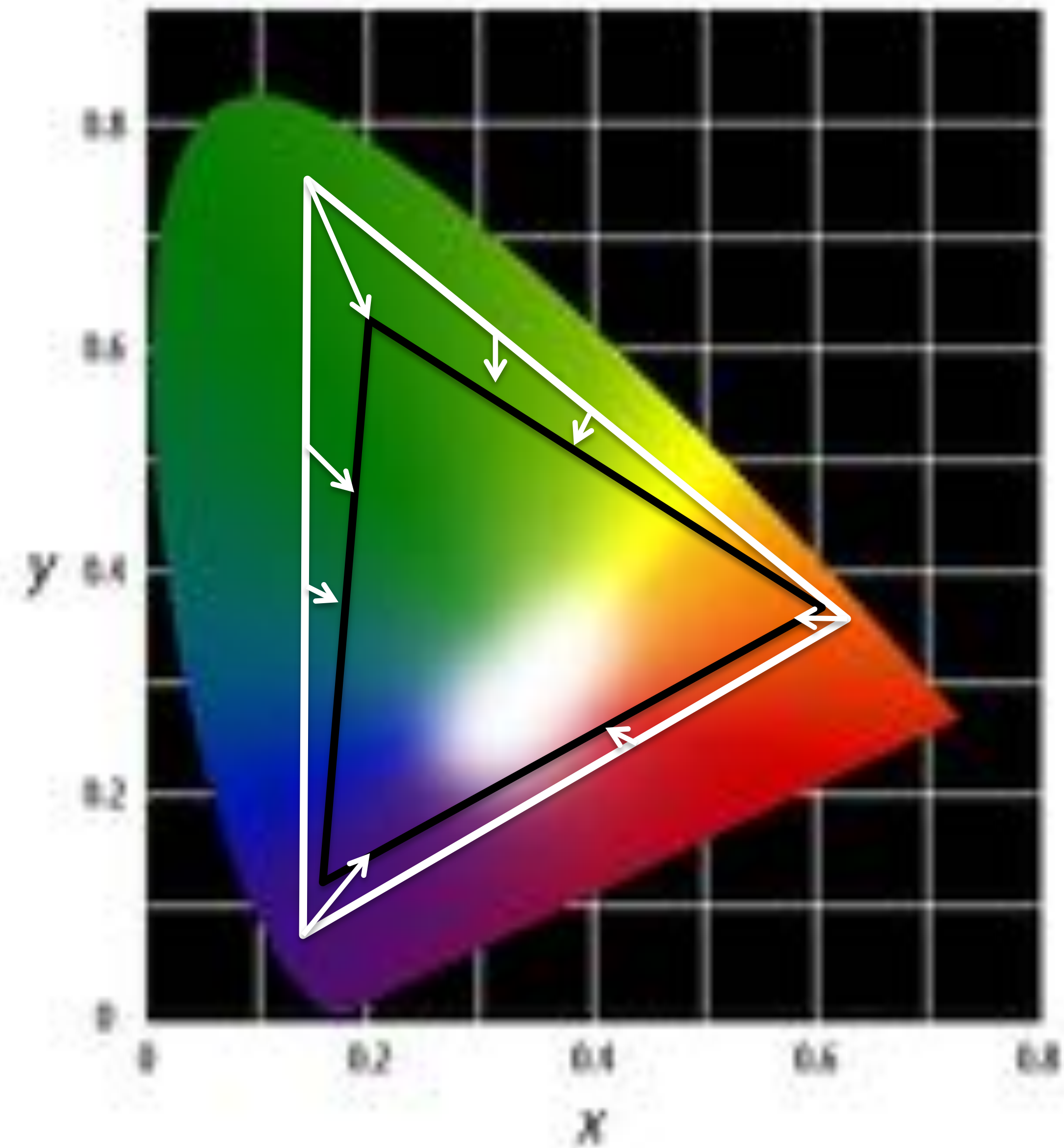
Cameras can capture in “non-visible” regions of the electromagnetic spectrum

x

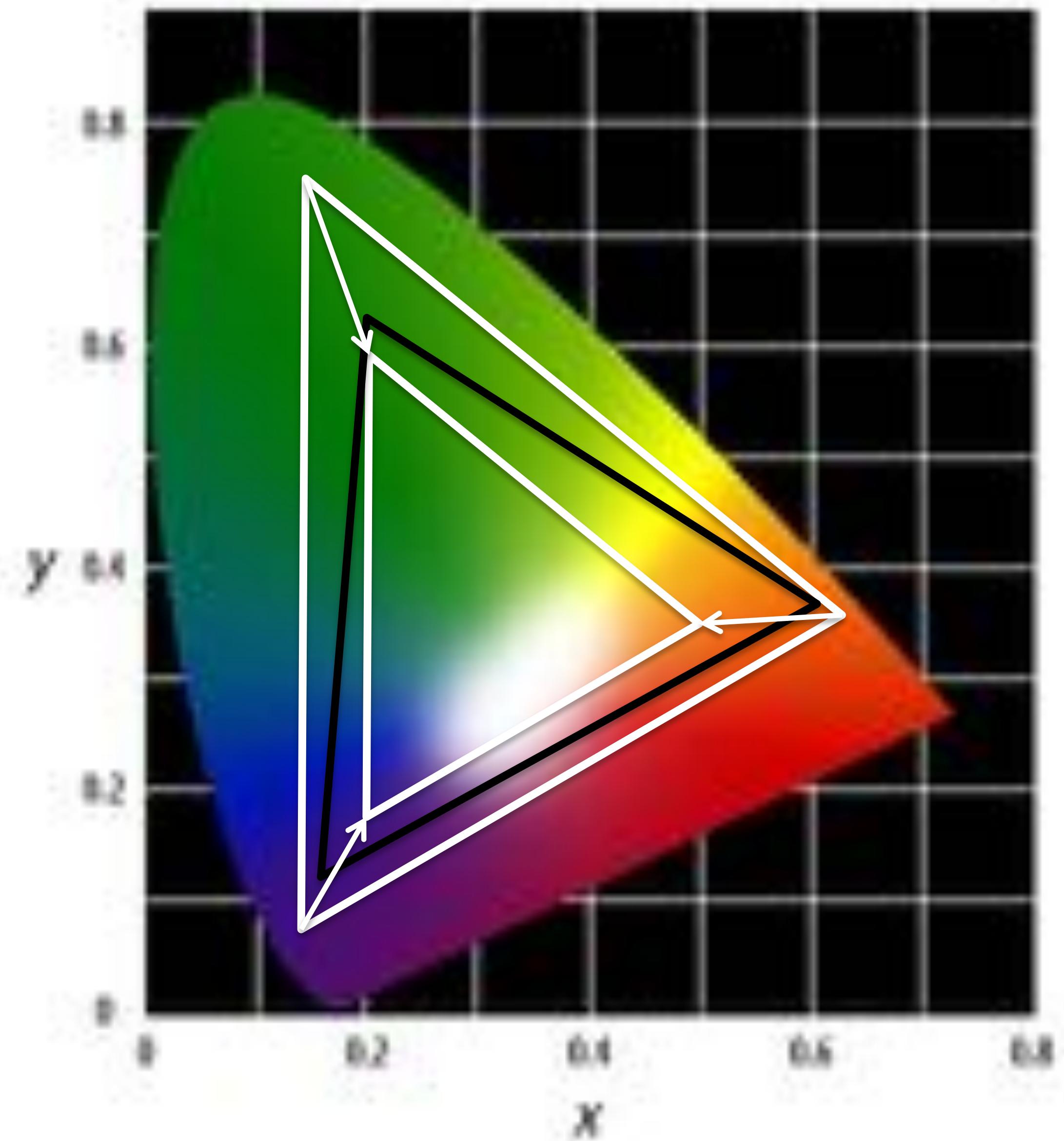


y

Gamut Clipping ("Colorimetric")

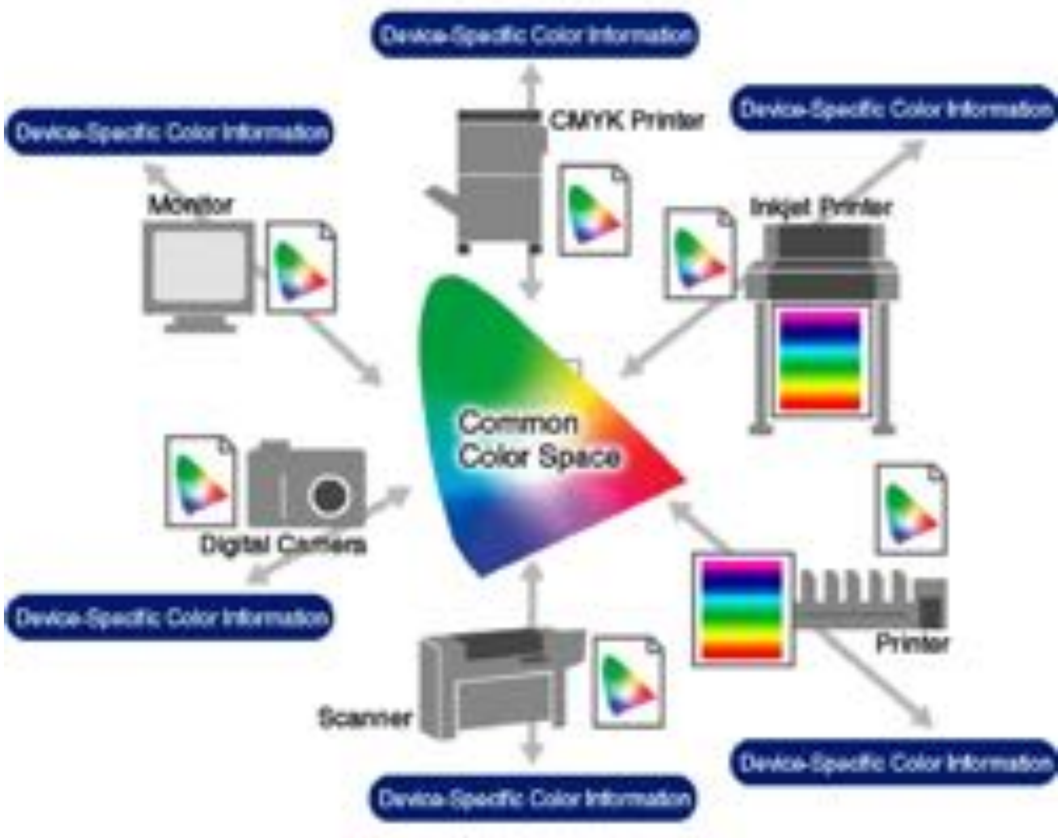


Gamut Compression ("Perceptual")



Color spaces

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

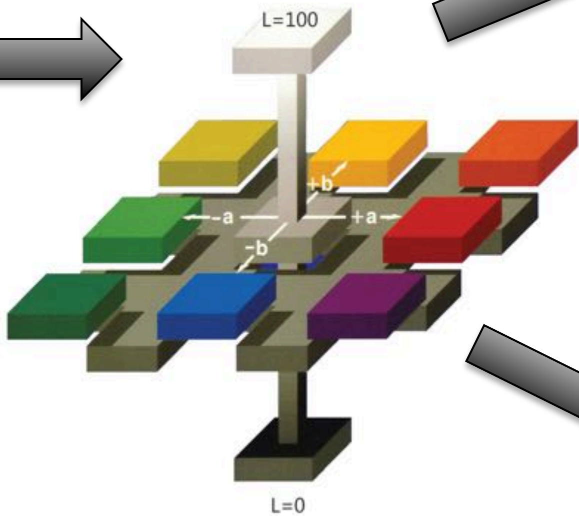


http://www.eizo.com/library/management/cms/

History of Color Management

Early 1990s

Sony Trinitron CRT



Lab Color Space

Printers

History of Color Management

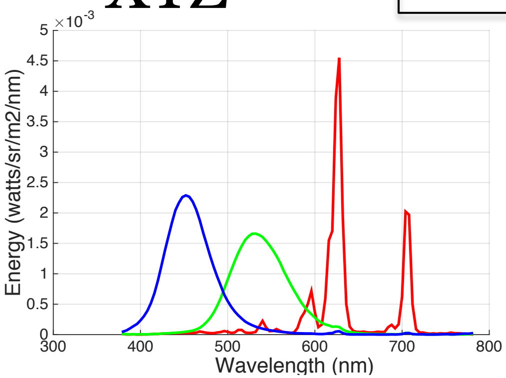
Calculate XYZ

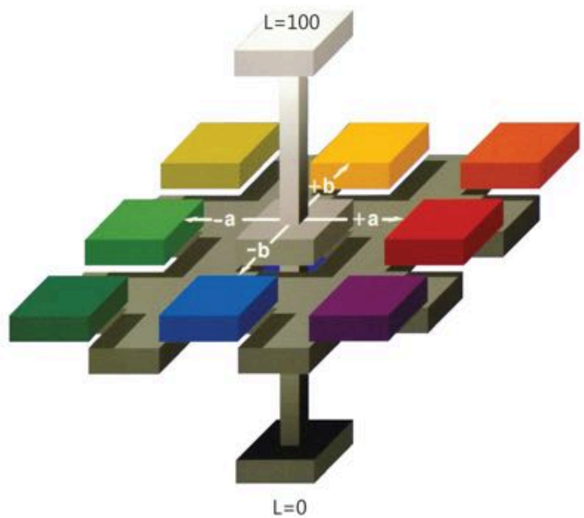
Convert to

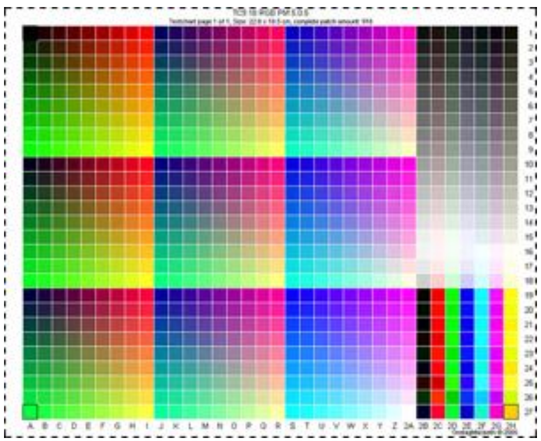
Lab LUT

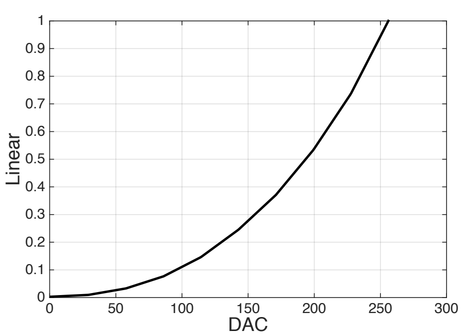
Convert to

Measure XYZ









Early 1990s

History of Color Management

Late 1990s

Sony Trinitron CRT

Printers

Device-dependent color management

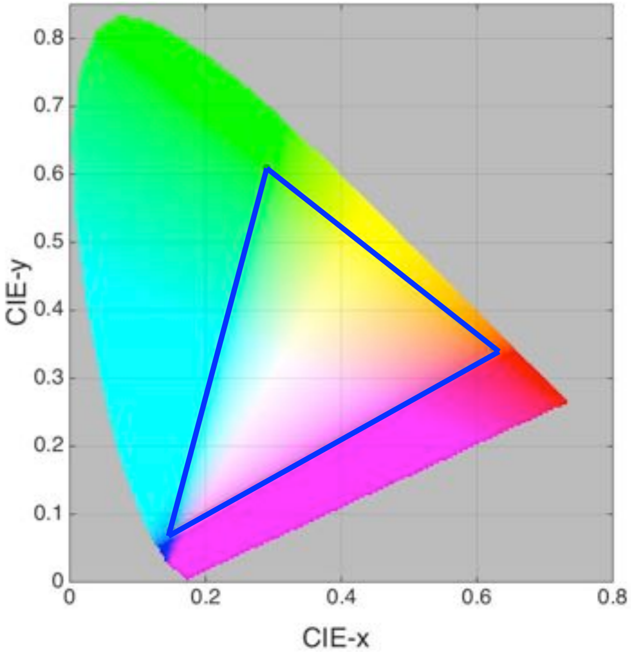
sRGB is the result of a committee … made up of engineers from HP and Microsoft in 1996

“sRGB is based on – are you ready for this? – sRGB is based on the display characteristics of consumer-grade CRT monitors manufactured in the early 1990’s”

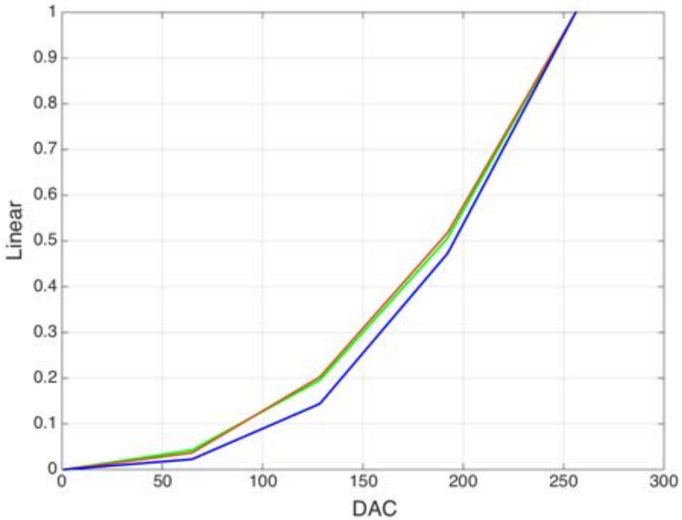
“And what about you, today, sitting in front of your LCD monitor? …. sRGB is probably not a good monitor profile for your LCD monitor”

http://ninedegreesbelow.com/photography/srgb-history.html

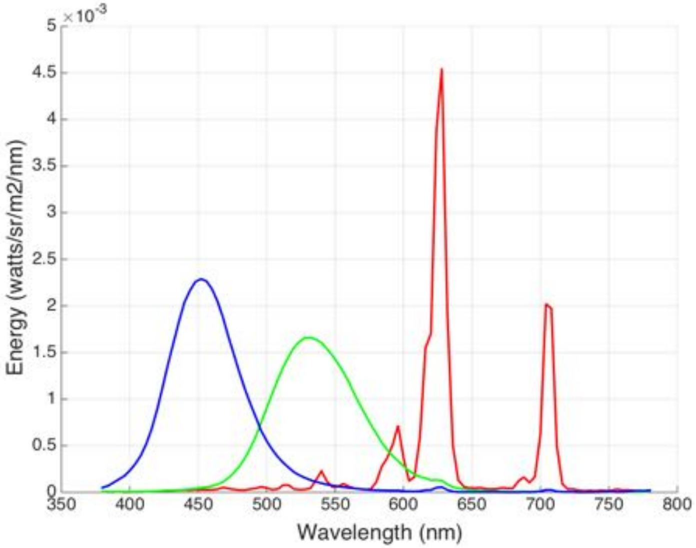
CRT



CIE-y
CIE-x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

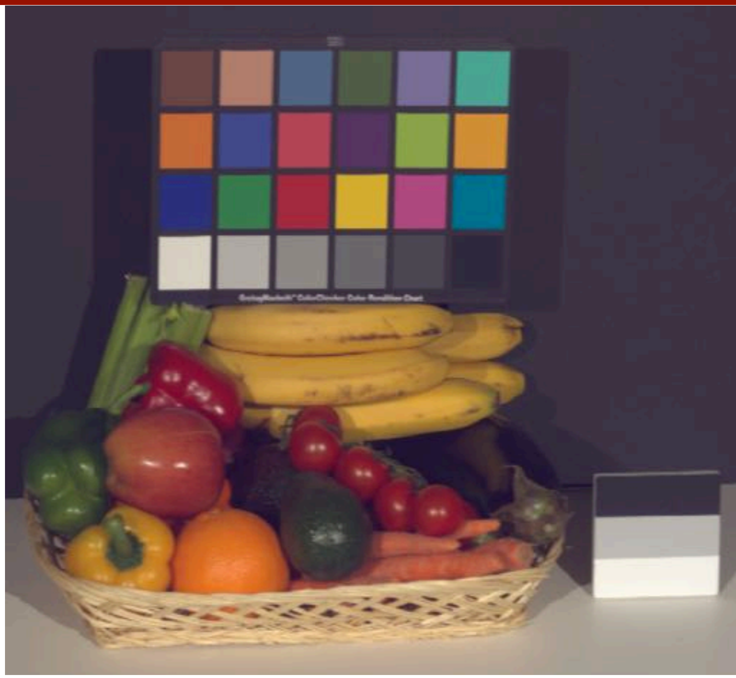


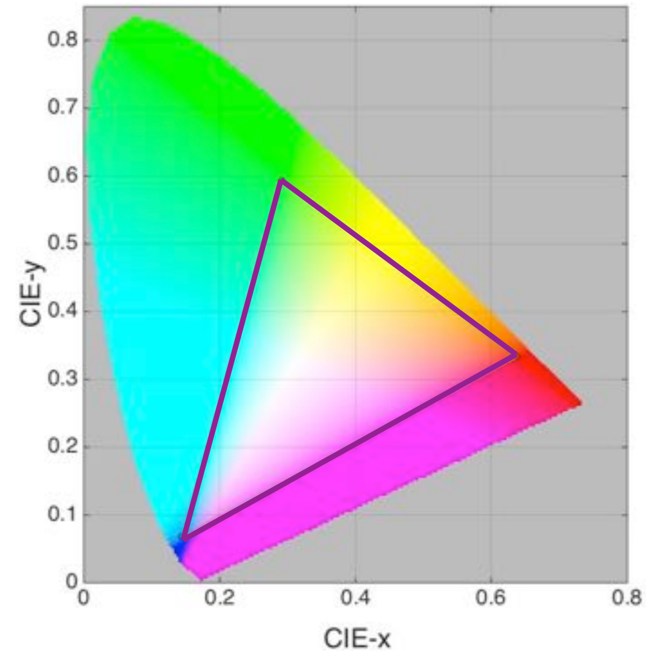
Linear
DAC
0
50
100
150
200
250
300

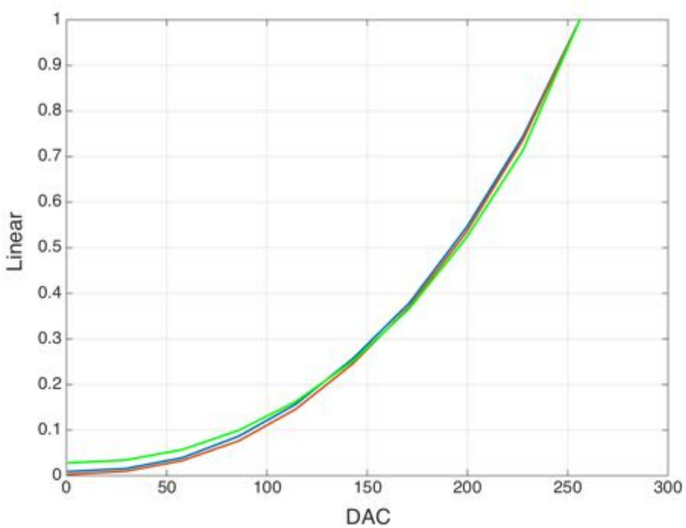


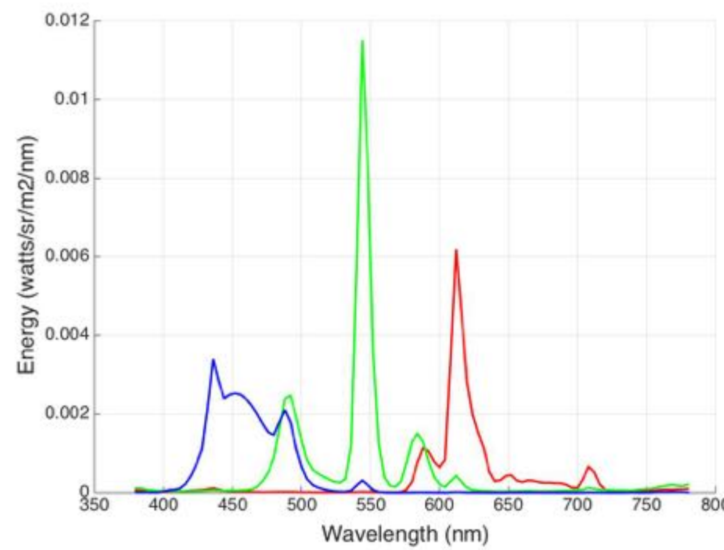
$\times 10^{-3}$
Energy (watts/sr/m2/nm)
Wavelength (nm)
350
400
450
500
550
600
650
700
750
800

LCD

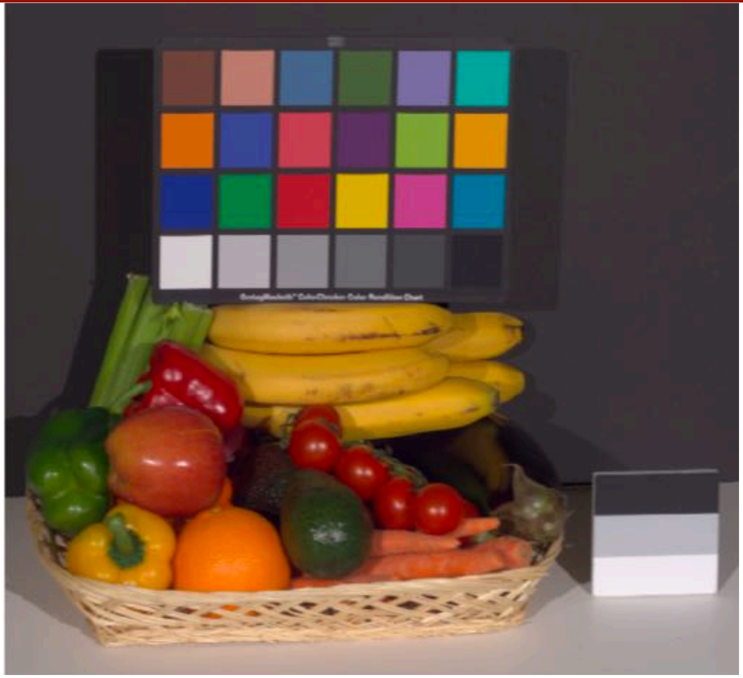


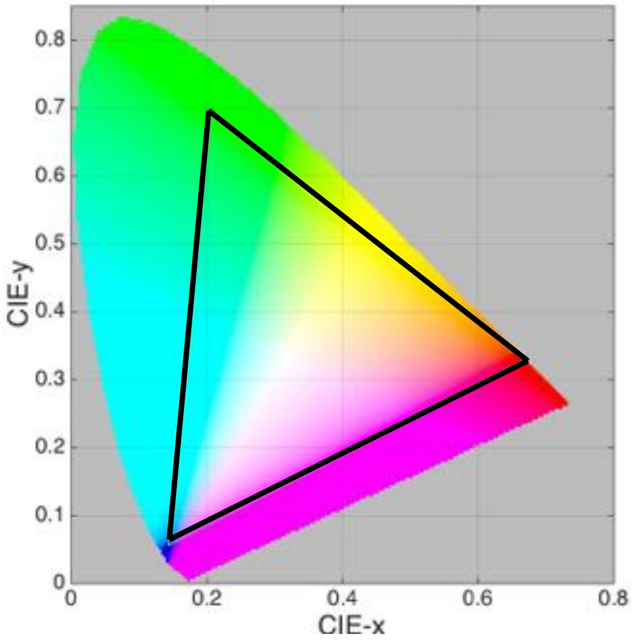

CIE-y
CIE-x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

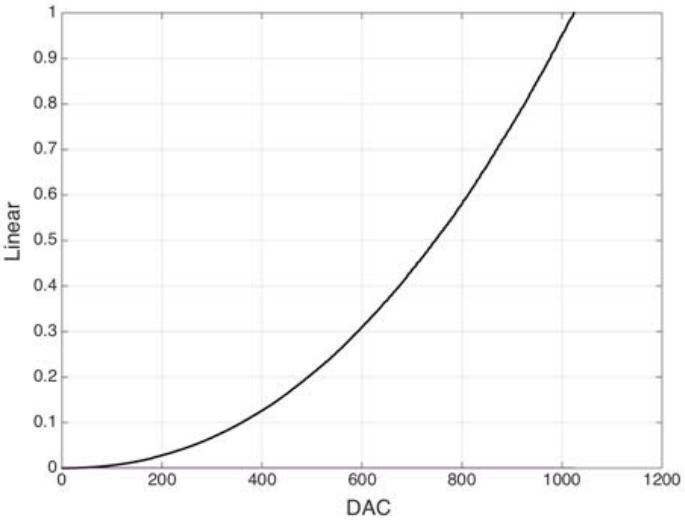

Linear
DAC
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
50
100
150
200
250
300

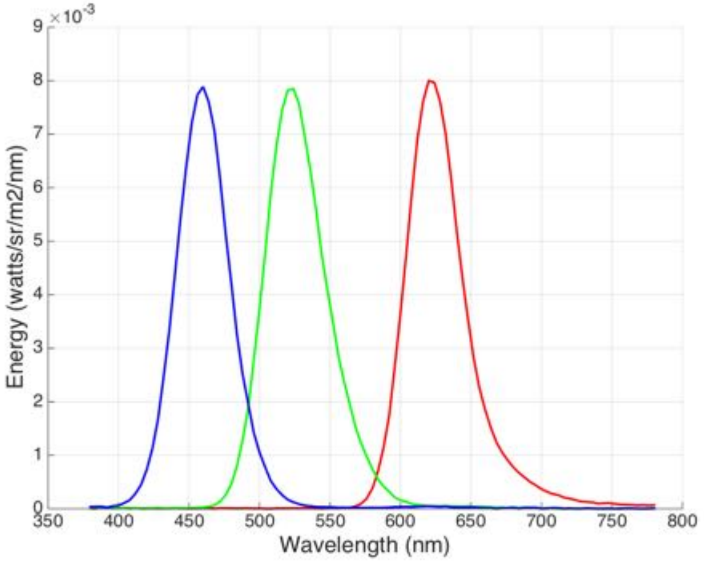

Energy (watts/sr/m2/nm)
Wavelength (nm)
0
0.002
0.004
0.006
0.008
0.01
0.012
350
400
450
500
550
600
650
700
750
800

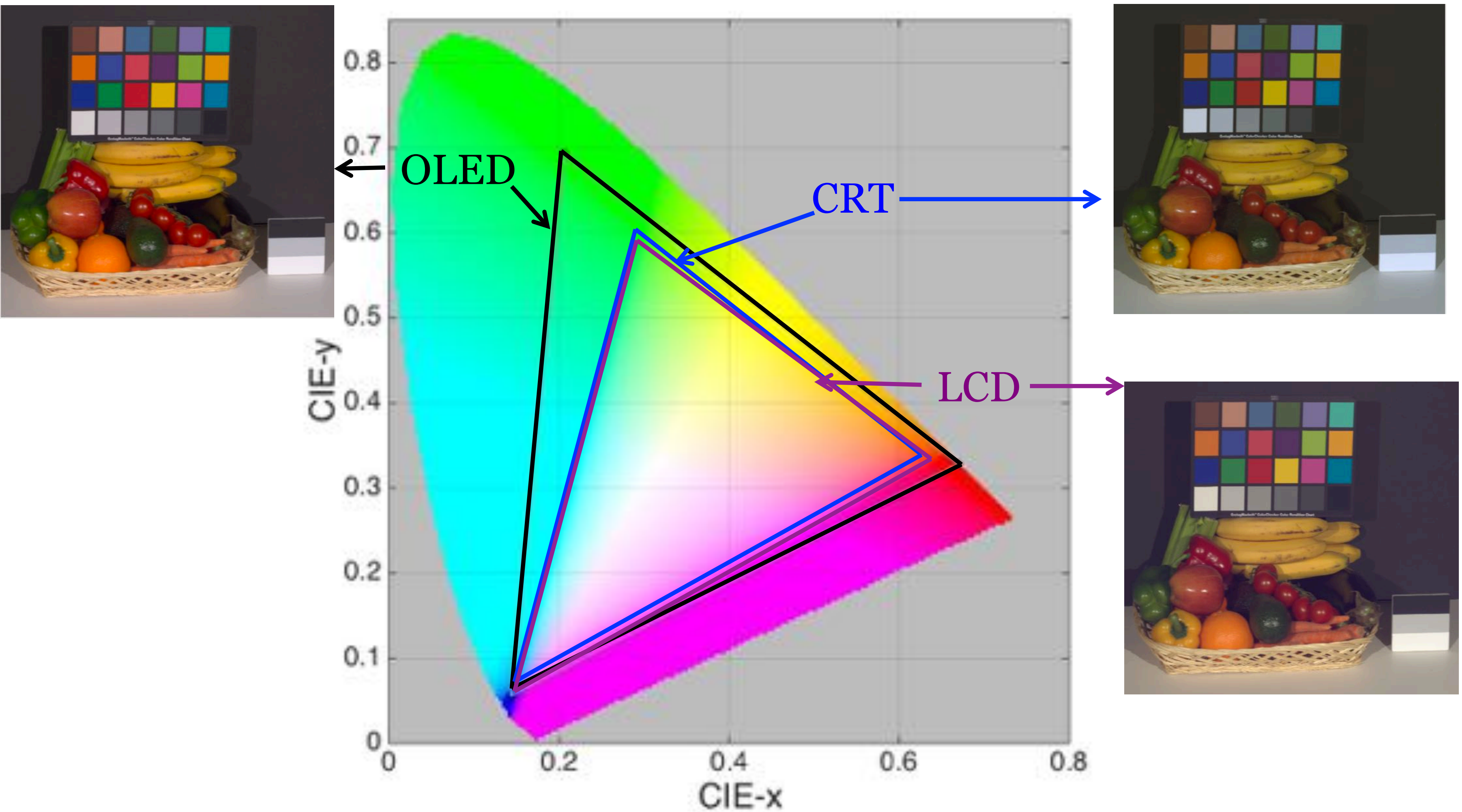
OLED




CIE-y
CIE-x
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8


Linear
DAC
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
200
400
600
800
1000
1200


Energy (watts/sr/m2/nm)
Wavelength (nm)
$\times 10^{-3}$
9
8
7
6
5
4
3
2
1
0
350
400
450
500
550
600
650
700
750
800


OLED
CRT
LCD
CIE-y
CIE-x
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8