

# Understanding the In-Camera Image Processing Pipeline for Computer Vision

IEEE CVPR 2016

June 26, 2016

Michael S. Brown  
National University of Singapore  
(York University, Canada)

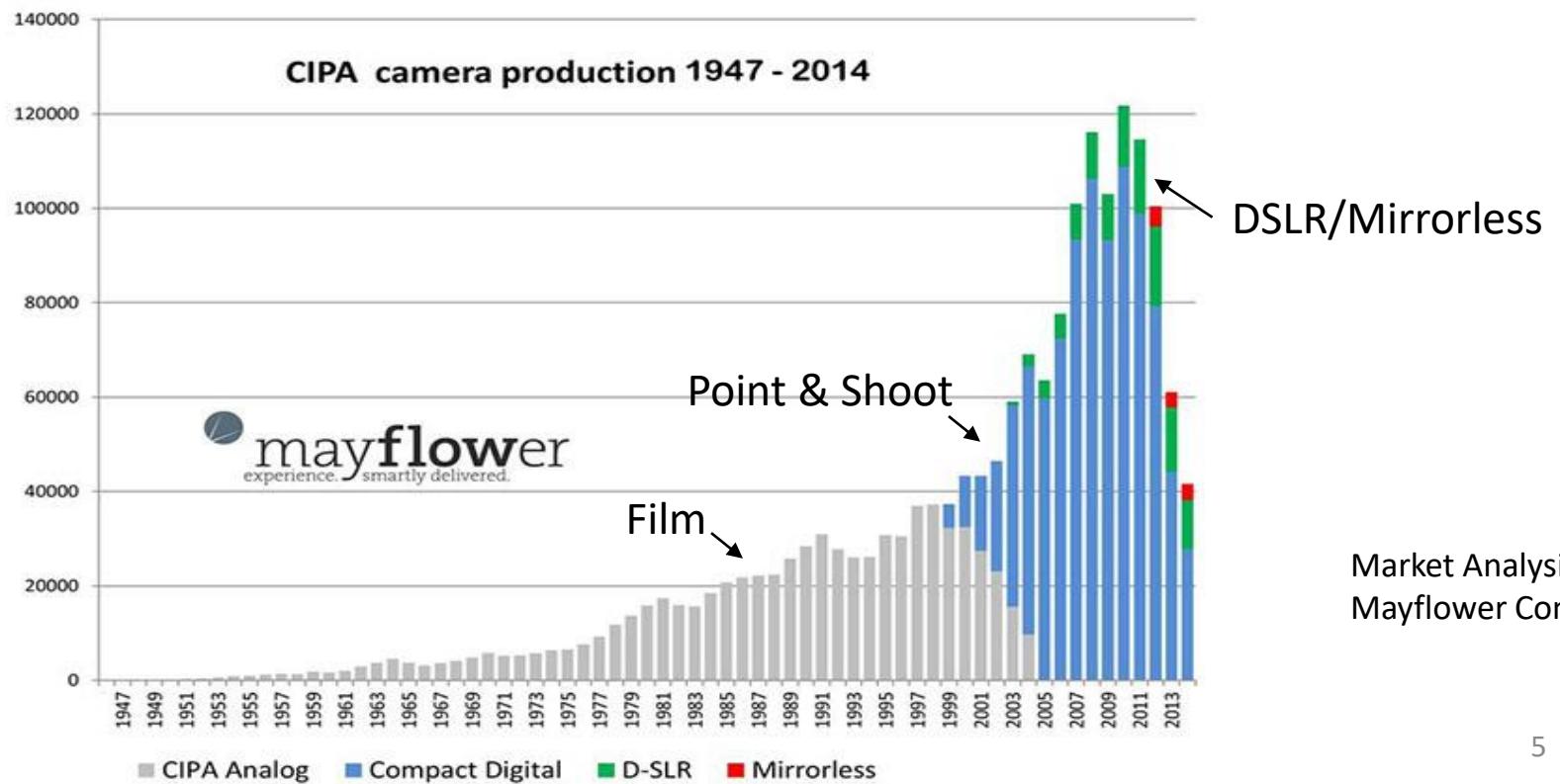
# Tutorial schedule

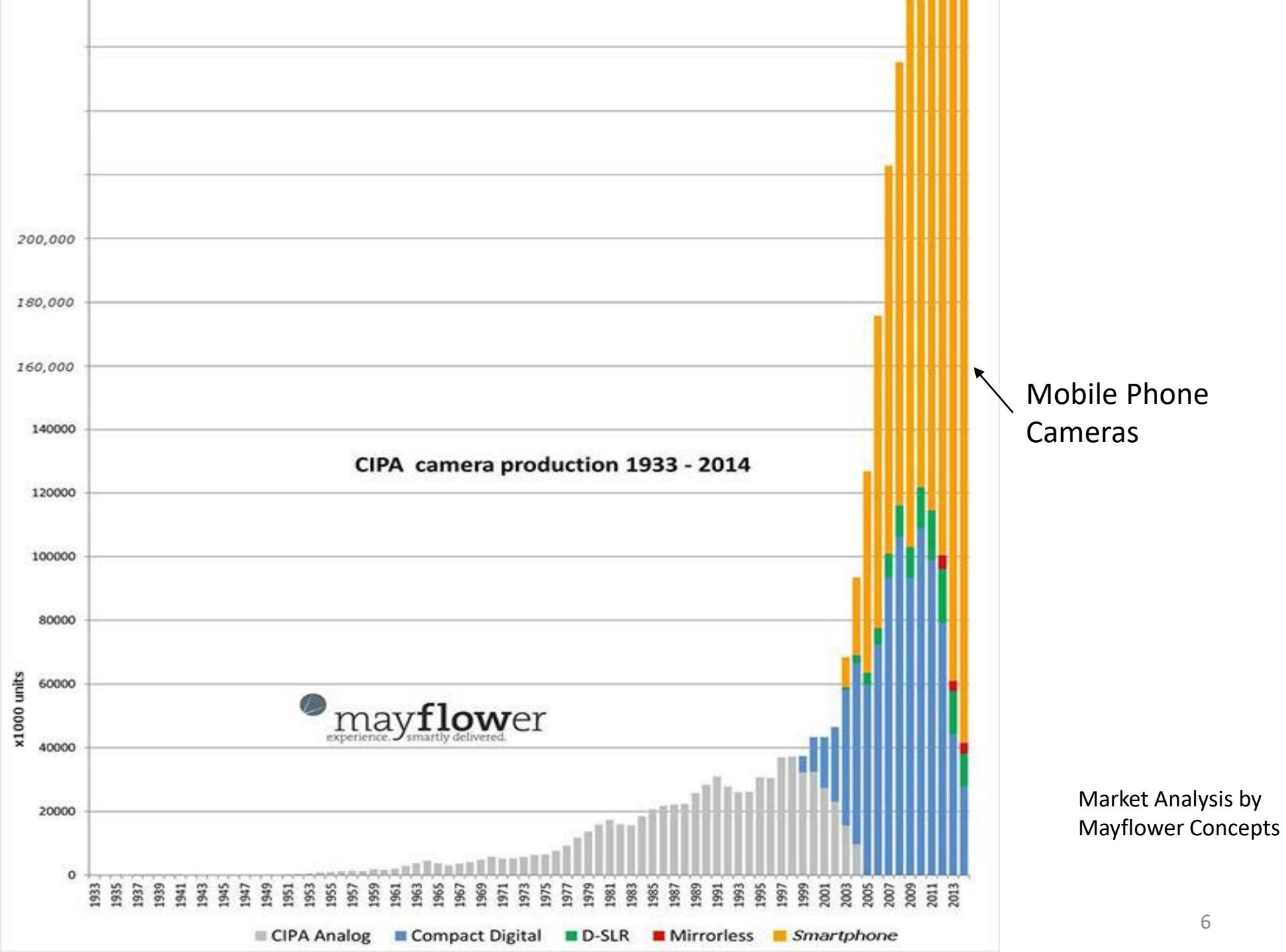
- Part 1 (General Part)
    - Motivation
    - Review of color/color spaces
    - Overview of camera imaging pipeline
  - Part 2 (Specific Part)
    - Modeling the in-camera color pipeline
    - Photo-refinishing
  - Part 3 (Wrap Up)
    - The good, the bad, and the ugly of commodity cameras and computer vision research
    - Concluding remarks
- 
- 8.30am-10.25am
- Coffee Break  
10.25am – 11am
- 11.00am-12.30pm

Motivation for  
this tutorial?

# Shifting landscape of cameras









OR



1

50+

# Image capture is mainstream



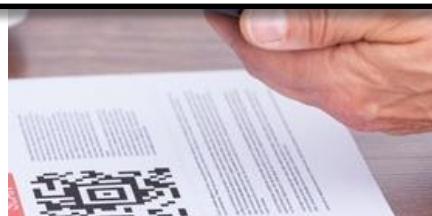
# Not always a good thing . . .



# Imaging for more than photography



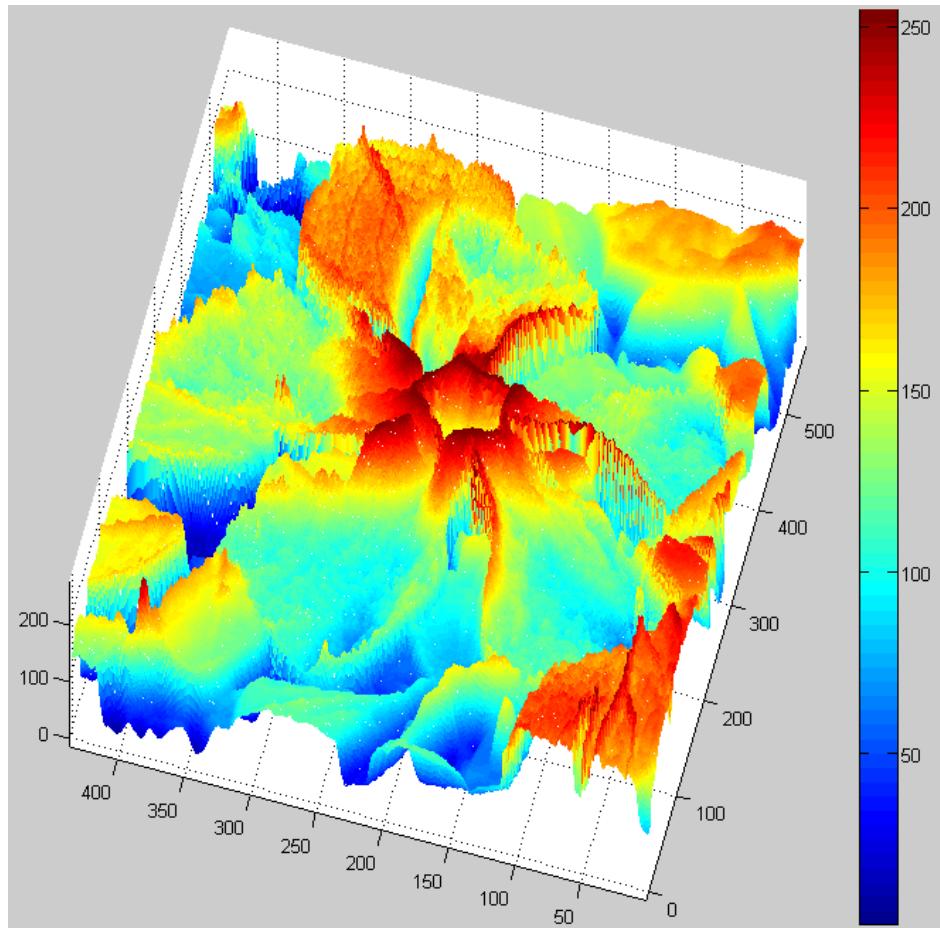
Many applications are not necessarily interested in the actual photo.



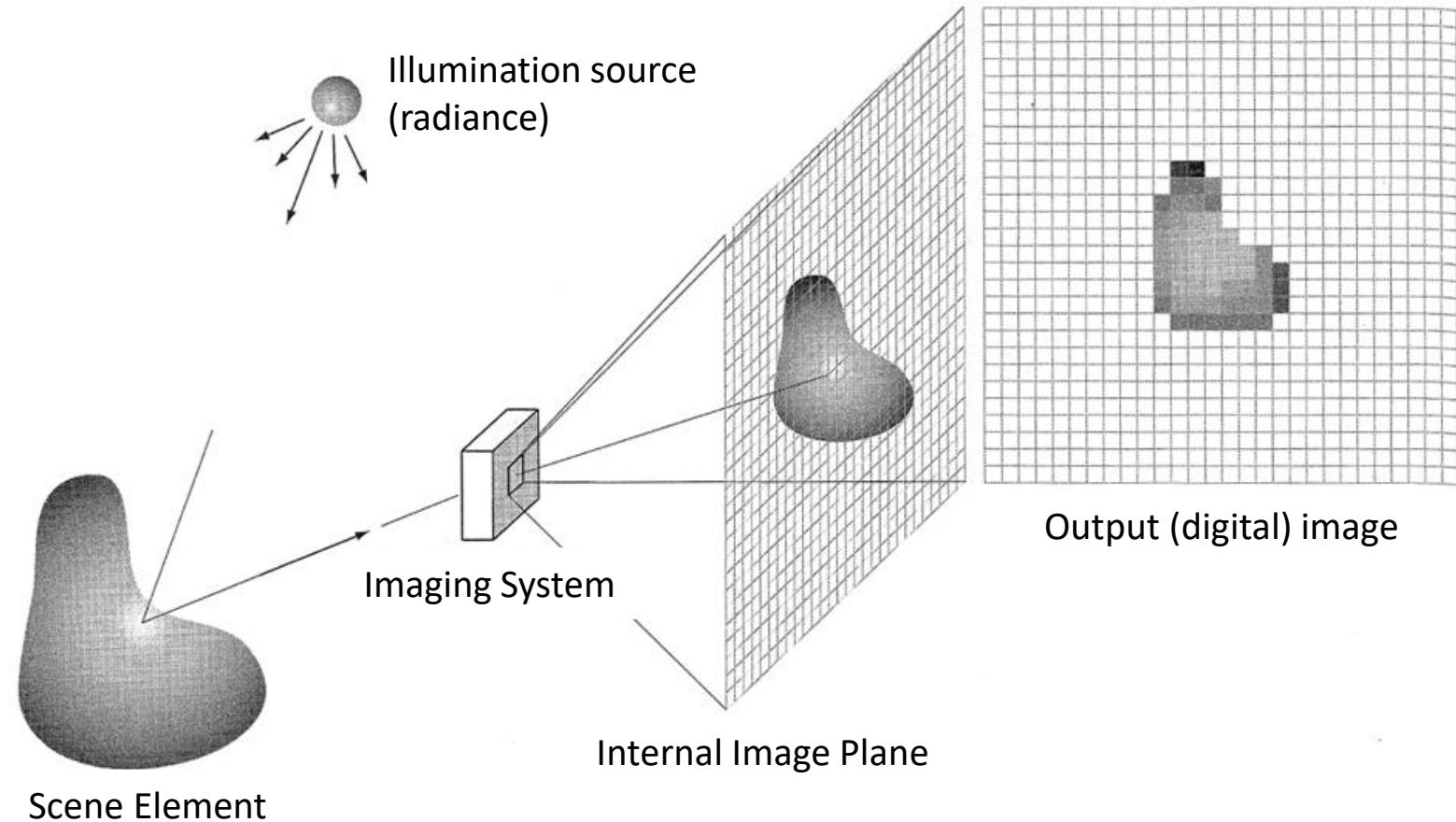
# Scientist's view of photography



# Scientist's view of photography

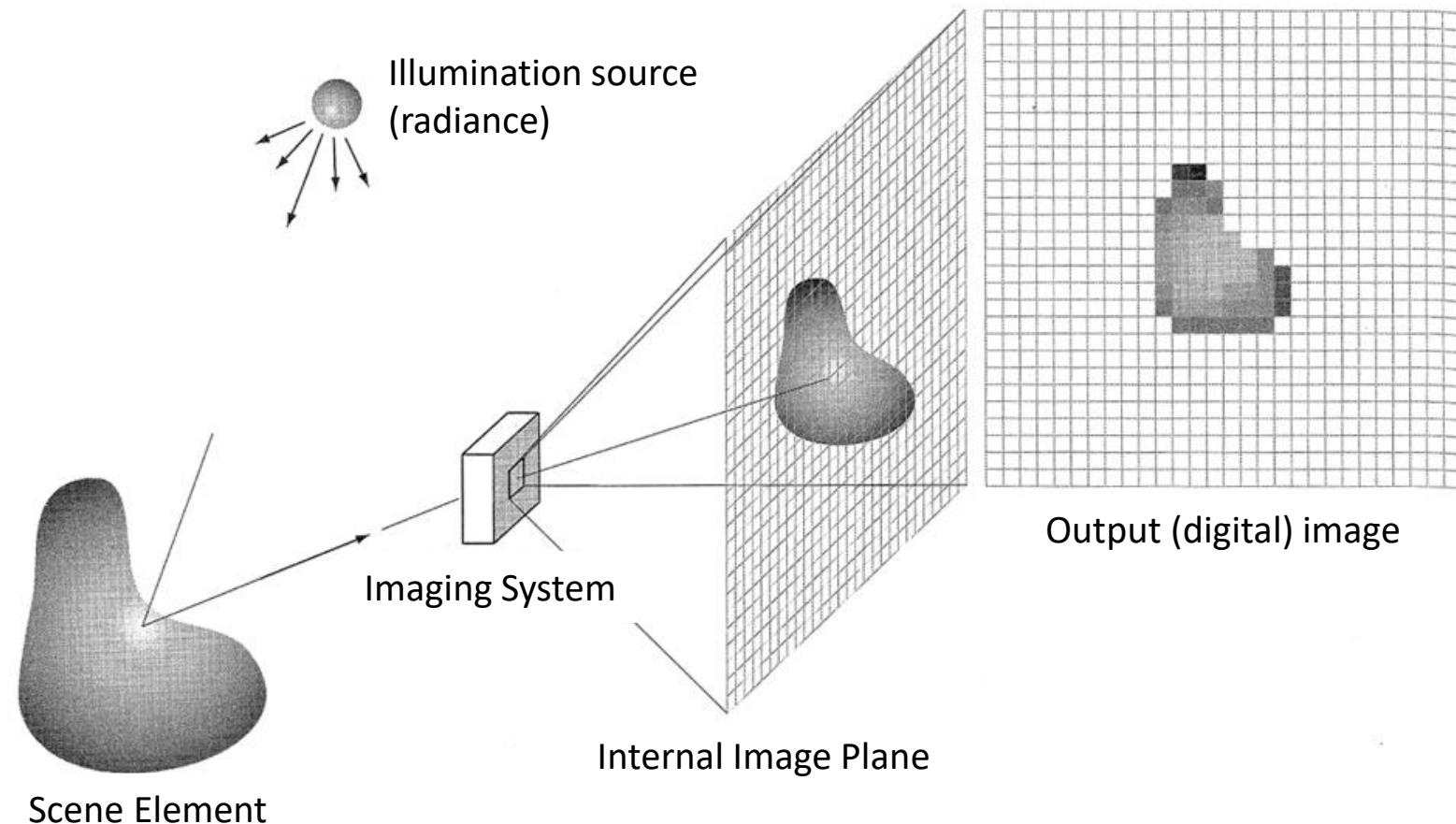


# Camera = light-measuring device



Simple models assume an image is a “quantitative measurement” of scene radiance.

# Image = radiant-energy measurement



Simple models assume an image is a “quantitative measurement” of scene radiance.

# Assumption used in many places

- Shape from shading
- HDR Imaging
- Image Matching
- Color constancy
- Etc . . .

Shape-from-shading



image of object

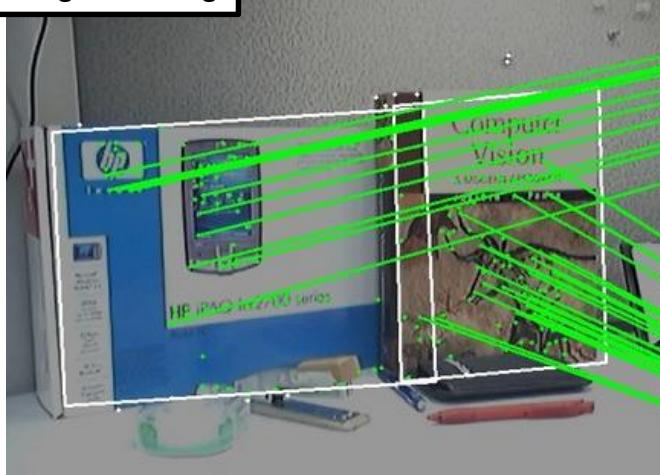


surface normals



3D model  
From Lu et al, CVPR'10

Image matching



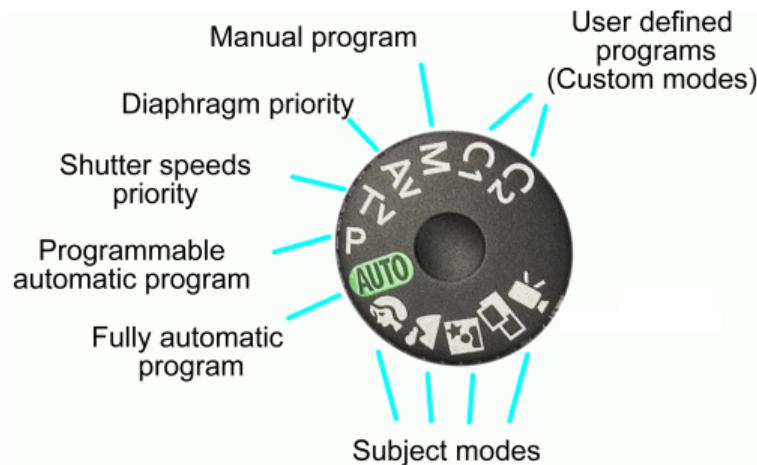
From Jon Mooser, CGIT Lab, USC

HDR imaging



From O'Reilly's digital media forum

# Camera = light-measuring device?



Portrait Mode	Soft Skin Mode	Transform Mode
Self-portrait Mode	Scenery Mode	Panorama Assist Mode
Sports Mode	Night Portrait Mode	Night Scenery Mode
Food Mode	Party Mode	Candle Light Mode
Baby Mode 1/2	Pet Mode	Sunset Mode
High Sensitivity Mode	High-speed Burst Mode	Flash Burst Mode
Starry Sky Mode	Fireworks Mode	Beach Mode
Snow Mode	Aerial Photo Mode	Pin Hole Mode
Film Grain Mode	High Dynamic Mode	Photo Frame Mode



# Light-measuring device?

Samsung Galaxy S6 edge



HTC One M9



LG G4



Google Camera App  
17  
All settings the same

# Onboard processing (photo finishing) “Secret recipe” of a camera



Photographs taken from three different cameras with the same aperture, shutter speed, white-balance , ISO, and picture style.

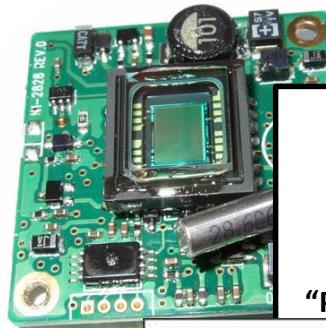
# Modern photography pipeline



Starting point:  
reality (in radiance)



**Pre-Camera**  
Lens Filter  
Lens  
Shutter  
Aperture

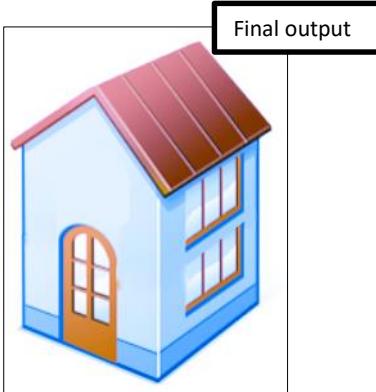


## In-Camera

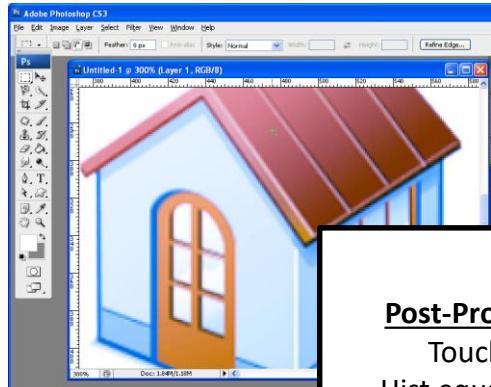
CCD response (RAW)  
CCD Demosaicing (RAW)

## "Photo-finishing Processing"

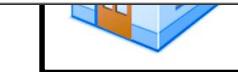
Portrait Mode	Soft Skin Mode	Transform Mode
Self-portrait Mode	Scenery Mode	Panorama Assist Mode
Sports Mode	Night Portrait Mode	Night Scenery Mode
Food Mode	Party Mode	Candle Light Mode
Baby Mode 1/2	Pet Mode	Sunset Mode
High Sensitivity Mode	High-speed Burst Mode	Flash Burst Mode
Starry Sky Mode	Fireworks Mode	Beach Mode
Snow Mode	Aerial Photo Mode	Pin Hole Mode
Film Grain Mode	High Dynamic Mode	Photo Frame Mode



Ending point:  
better than reality (in RGB)



**Post-Processing**  
Touch-up  
Hist equalization  
Spatial warping  
Etc ...



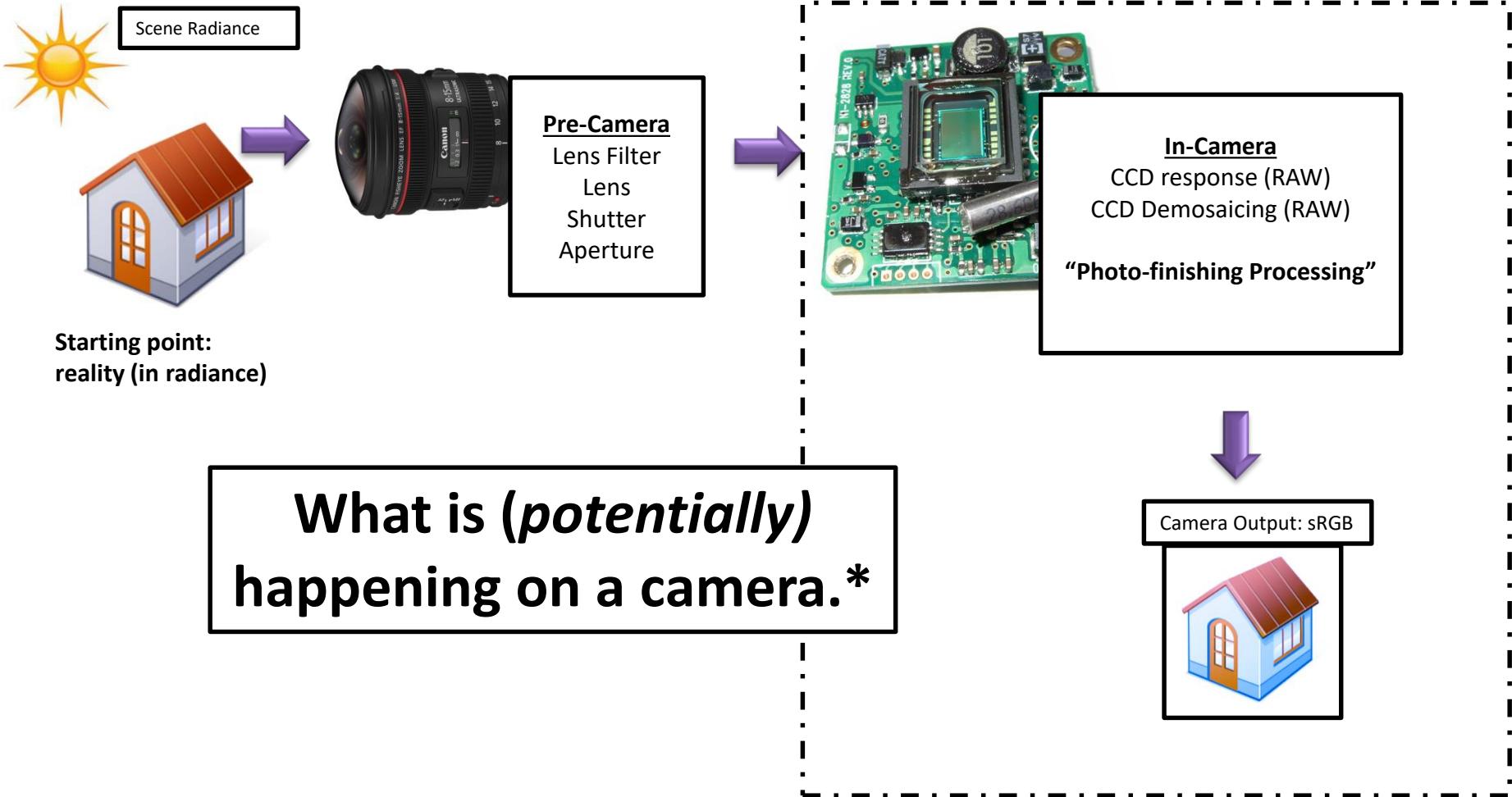
Even if we stopped here,  
the original CCD response  
potentially has had many  
levels of processing.

# Digital cameras

- Digital cameras *are far* from being light-measuring devices
- They are designed to produce visually pleasing photographs
- There is a great deal of processing (photo-finishing) happening on the camera

The goal of this tutorial is to discuss common processing steps that take place onboard consumer cameras

# This tutorial will examine



\*Camera pipelines are almost always proprietary, so knowing exactly what steps are performed for a particular make/model is not possible. This tutorial examines the most common steps likely to be found on most cameras.

# Tutorial schedule

- Part 1 (General Part)
  - Motivation
  - Review of color/color spaces
  - Overview of camera imaging pipeline
- Part 2\* (Specific Part)
  - Modeling the in-camera color pipeline
  - Photo-refinishing
- Part 3 (Wrap Up)
  - The good, the bad, and the ugly of commodity cameras and computer vision research
  - Concluding Remarks

# “Crash Course” on Color & Color Spaces

# Color

**Def** *Color* (noun): The property possessed by an object of producing different sensations on the eye as a result of the way it reflects or emits light.

*Oxford Dictionary*

# Color is perceptual

- **Color is not** a primary physical property
- Red, Green, Blue, Pink, Orange, Atomic Tangerine, Baby Pink, etc. . .
  - Adjectives we assign to “color sensations”



Which is “True Blue”?

# Subjective terms to describe color

## Hue

Name of the color  
(yellow, red, blue, green, . . . )

## Value/Lightness/Brightness

How light or dark a color is.

## Saturation/Chroma/Color Purity

How “strong” or “pure” a color is.

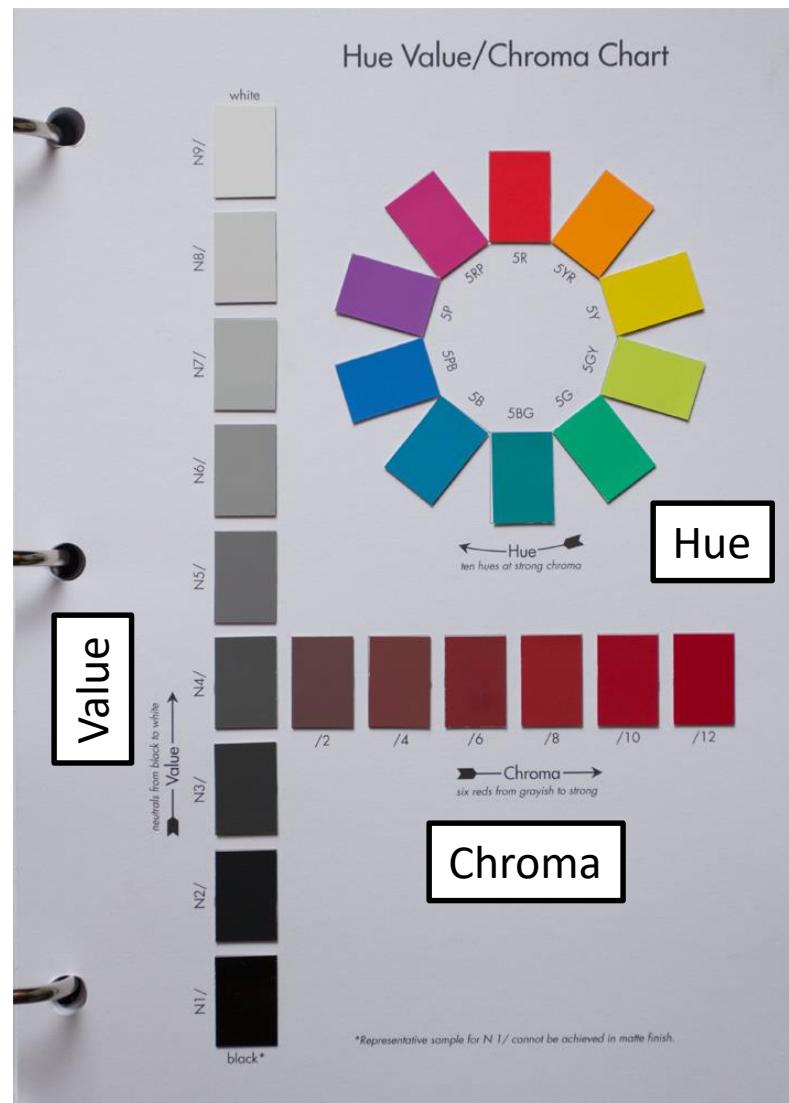
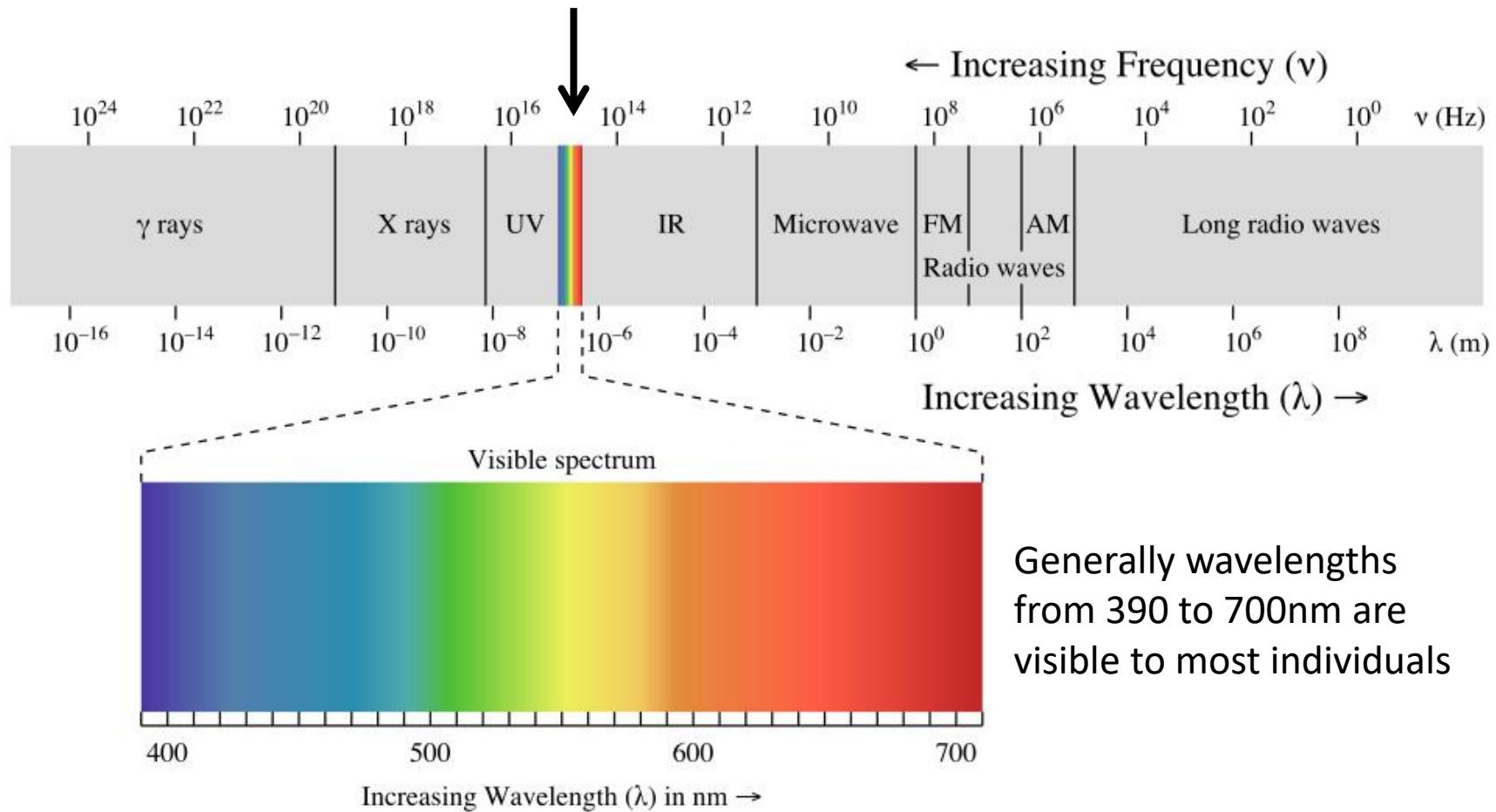


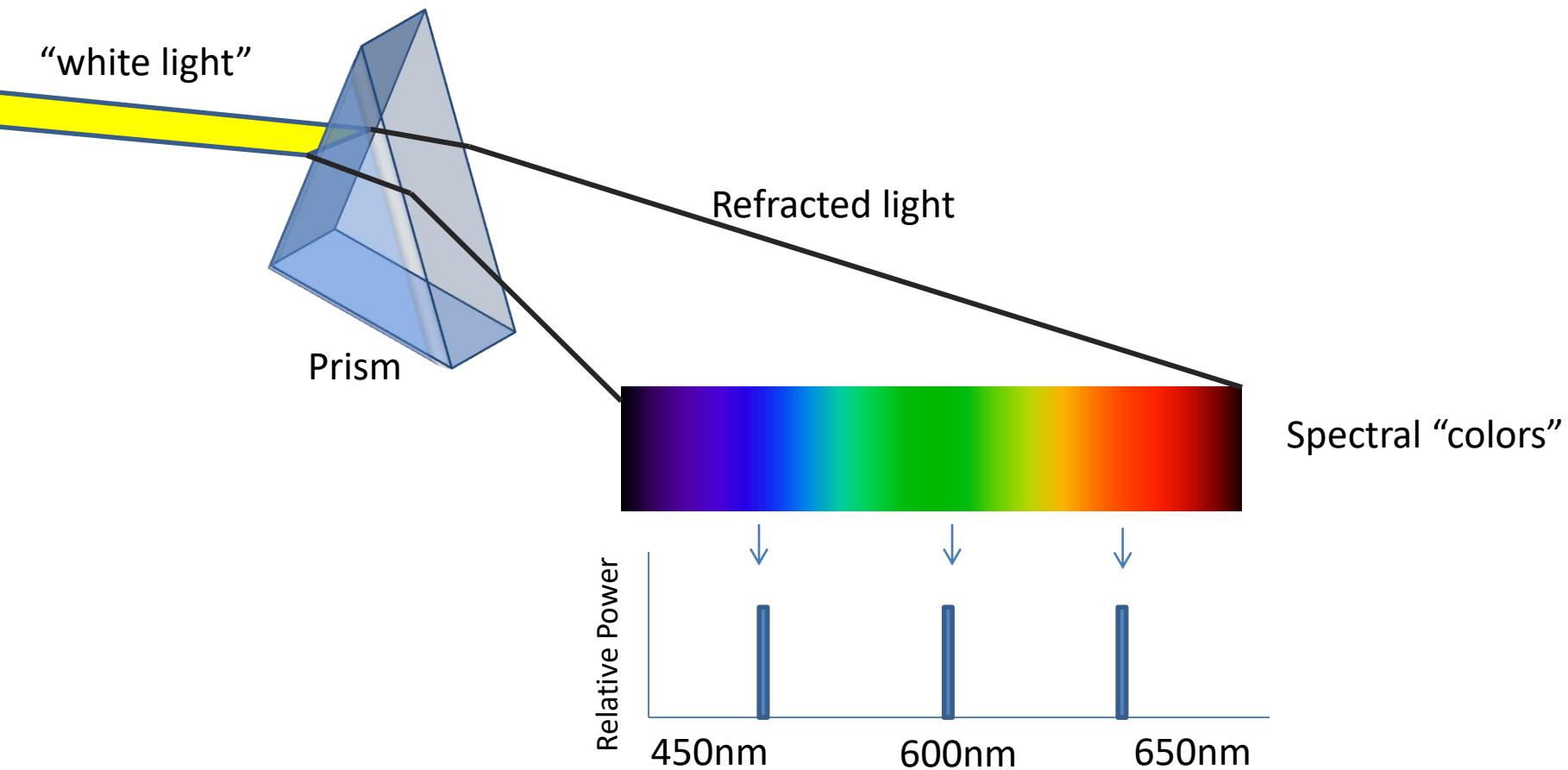
Image from Benjamin Salley  
A page from a Munsell Student Color Set

# Where do “color sensations” come from?

A *very* small range of electromagnetic radiation



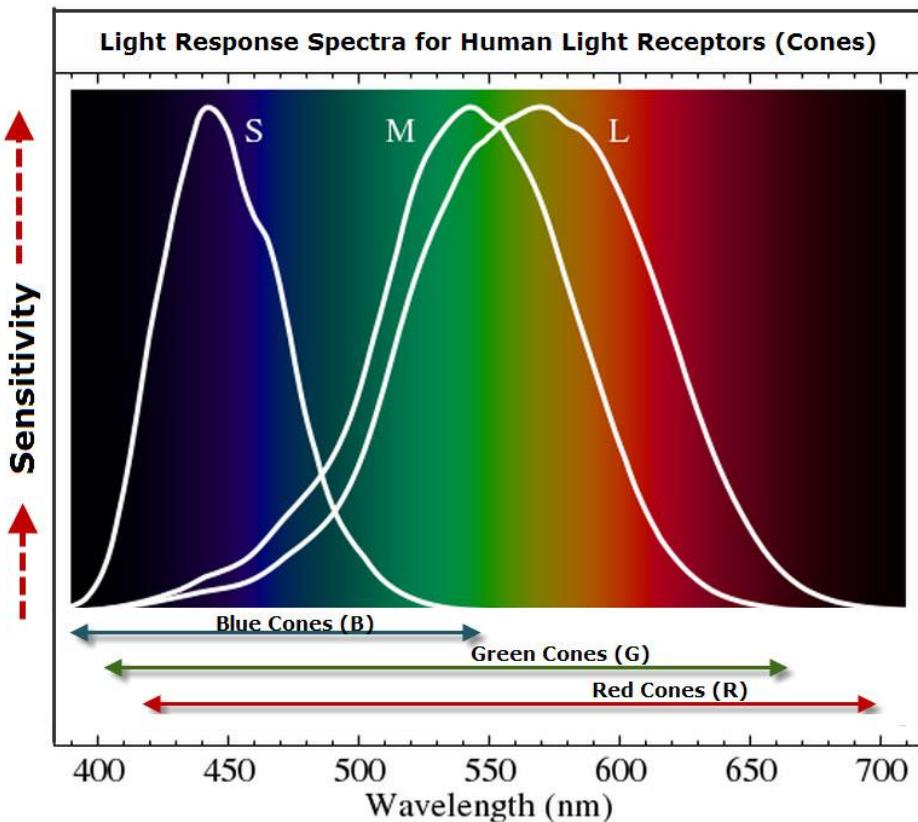
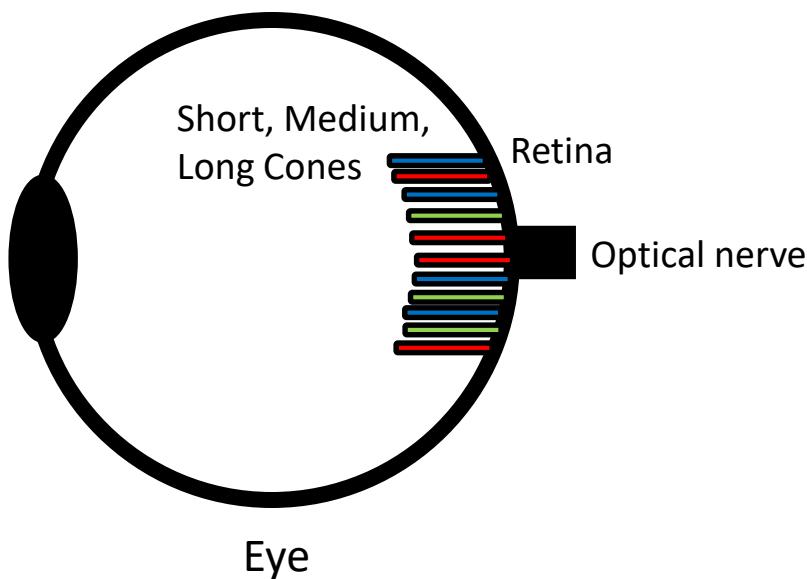
# “White light” through a prism



Light is separated into “monochromatic” light at different wave lengths.

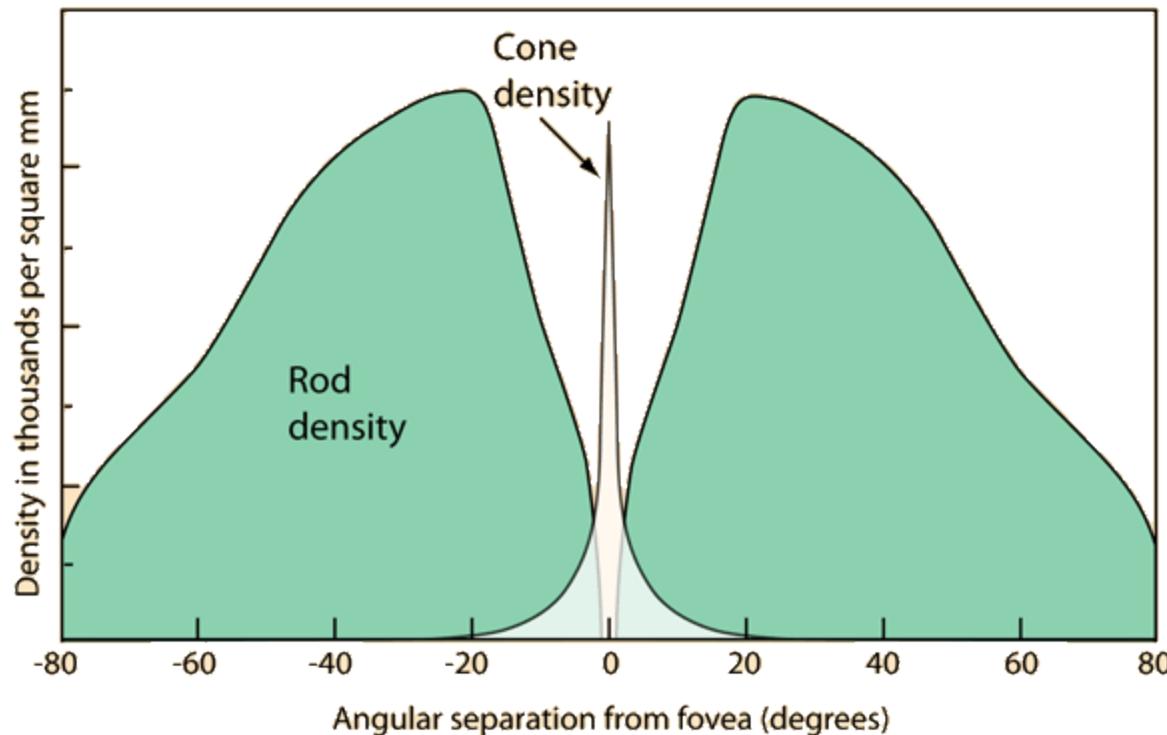
# Sensations?

- Our eye has three receptors (cone cells) that respond to visible light and give the sensation of color

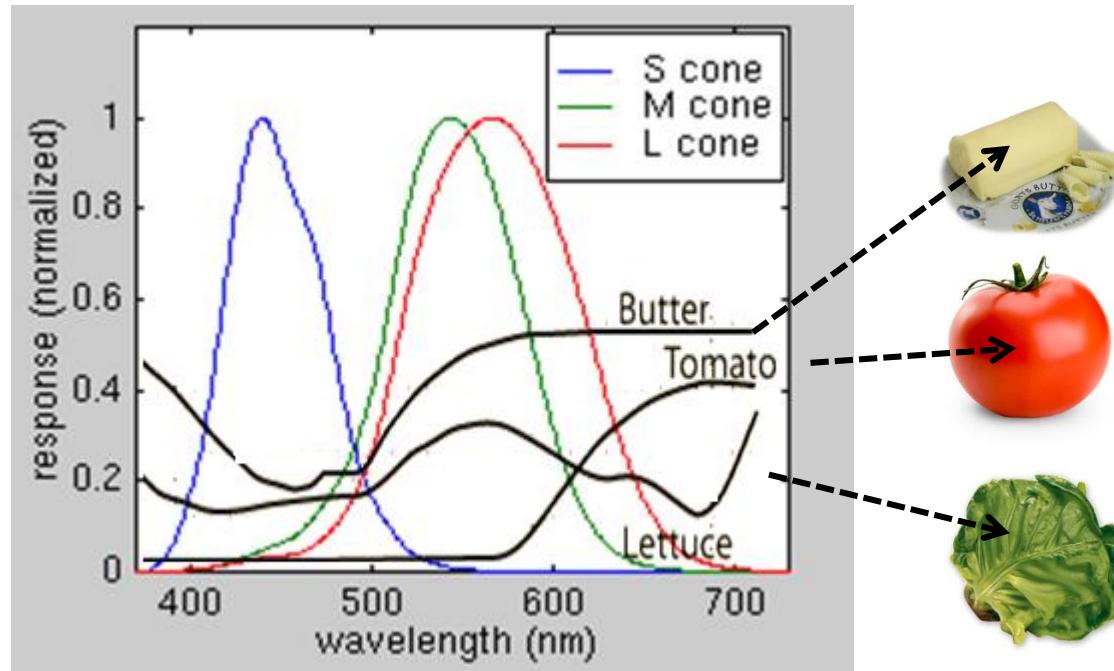


# Cones and rods

- We have additional light sensitive cells called *rods* that are not responsible for color
- Cones are most concentrated on the fovea



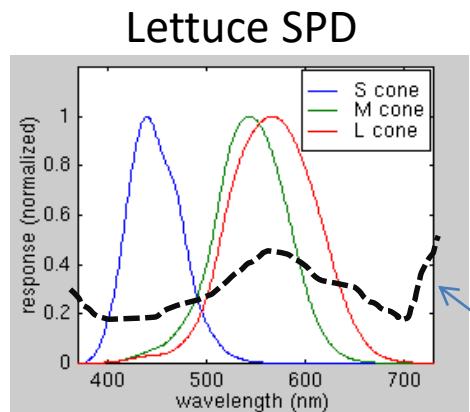
# Spectral power distribution (SPD)



We rarely see monochromatic light in real world scenes. Instead, objects reflect a wide range of wavelengths. This can be described by a *spectral power distribution* (SPD) shown above. The SPD plot shows the relative amount of each wavelength reflected over the visible spectrum.

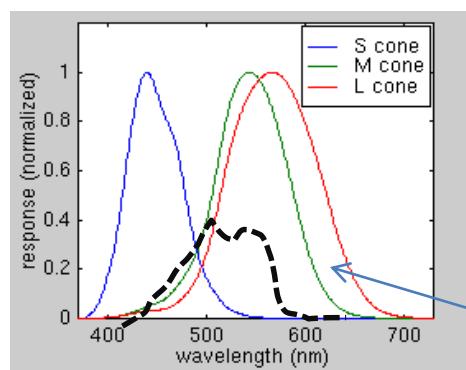
# SPD relation to color is not unique

- Due to the accumulation effect of the cones, two different SPDs can be perceived as the same color



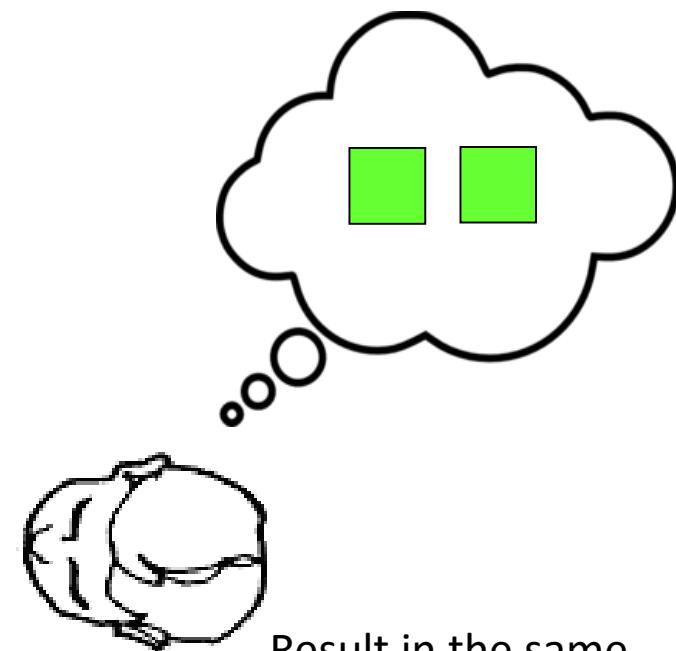
Lettuce SPD  
stimulating  
 $S=0.2$ ,  $M=0.8$ ,  
 $L=0.8$

SPD of "real lettuce"



Green ink SPD  
stimulating  
 $S=0.2$ ,  $M=0.8$ ,  
 $L=0.8$

SPD of ink in a "picture of lettuce"



# Tristimulus color theory

- Even before cone cells were discovered, it was empirically found that only three distinct colors (primaries) could be mixed to produce other colors
- Thomas Young (1803), Hermann von Helmholtz (1852), Hermann Grassman (1853), James Maxwell (1856) all explored the theory of trichromacy for human vision

# Tristimulus color theory

**Grassman's Law** states that a source color can be matched by a **linear combination** of three independent “primaries”.

$$\text{Source light 1} = R1 \text{ (red bar)} + G1 \text{ (green bar)} + B1 \text{ (blue bar)}$$

Three primaries and the weights ( $R1, G1, B1$ ) of each primary needed to match the source light #1 perceived color.

$$\text{Source light 2} = R2 \text{ (red bar)} + G2 \text{ (green bar)} + B2 \text{ (blue bar)}$$

Same three primaries and the weights ( $R2, G2, B2$ ) of each primary needed to match the source light #2 perceived color

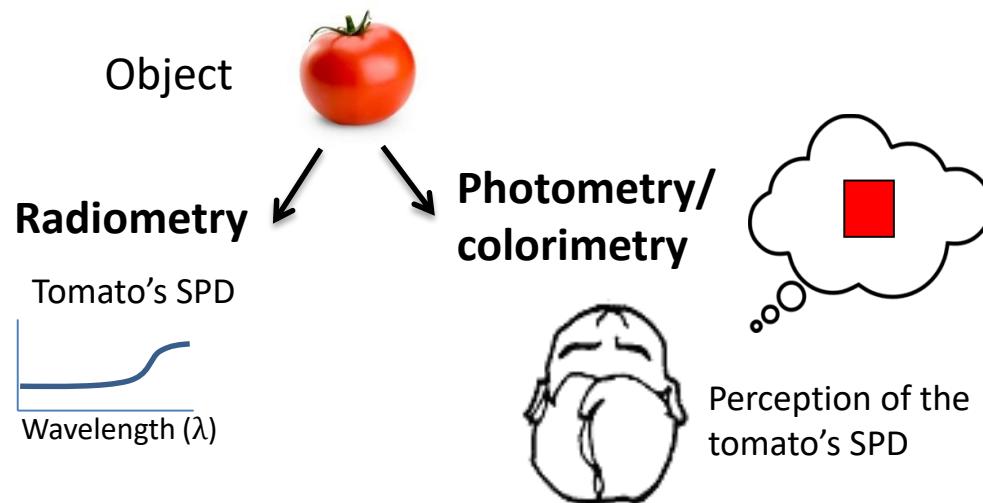
If we combined source lights 1 & 2 to get a new source light 3

The amount of each primary needed to match the new source light 3 will be the sum of the weights that matched lights 1 & 2.

$$\text{Source light 3} = (R1 + R2) \text{ (red bar)} + (G1 + G2) \text{ (green bar)} + (B1 + B2) \text{ (blue bar)}$$

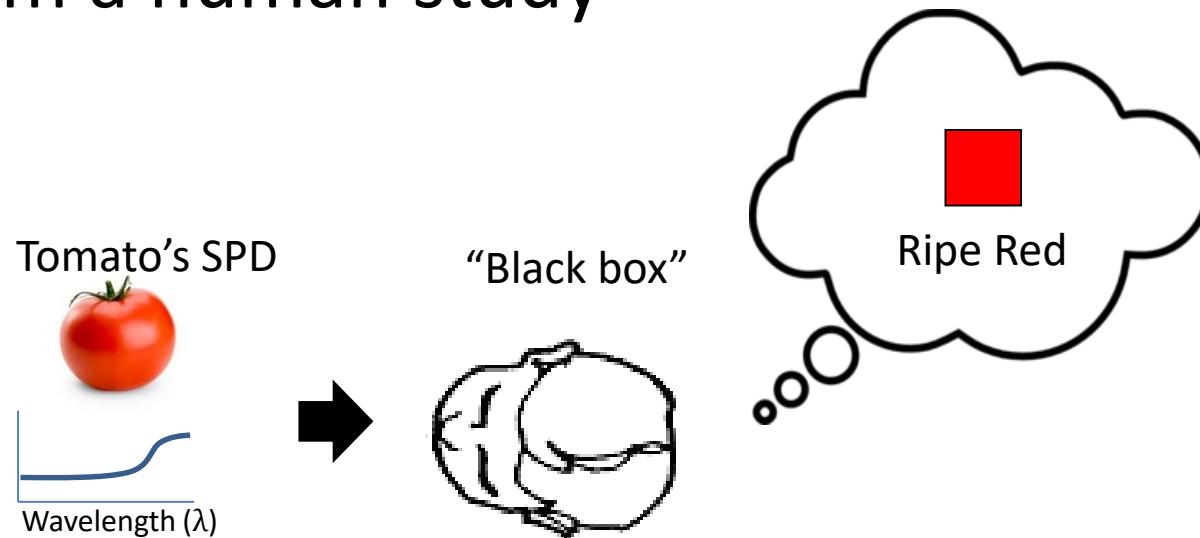
# Radiometry vs. photometry

- **Radiometry**
  - Quantitative measurements of radiant energy
  - Often shown as spectral power distributions (SPD)
  - Measures either light coming from a source (radiance) or light falling on a surface (irradiance)
- **Photometry/ colorimetry**
  - Quantitative measurement of **perceived** radiant energy based on human's sensitivity to light
  - Perceived in terms of “brightness” (photometry) and color (colorimetry)

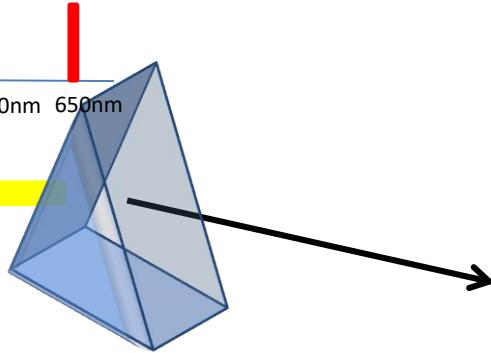


# Quantifying color

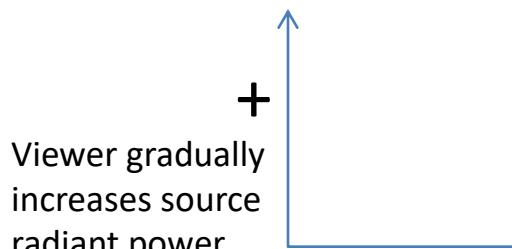
- We still need a way to quantify color & brightness
- SPDs go through a “black box” (human visual system) and are perceived as color
- The only way to quantify the “black box” is to perform a human study



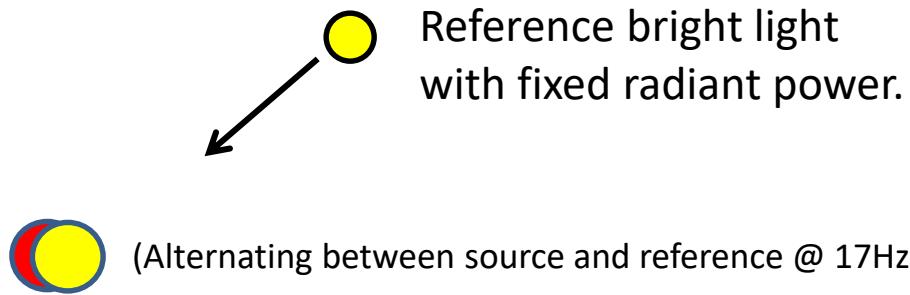
# Experiments for photometry



Chromatic **source** light at a particular wavelength and adjustable radiant power.



The “flicker photometry” experiment for photopic sensitivity.



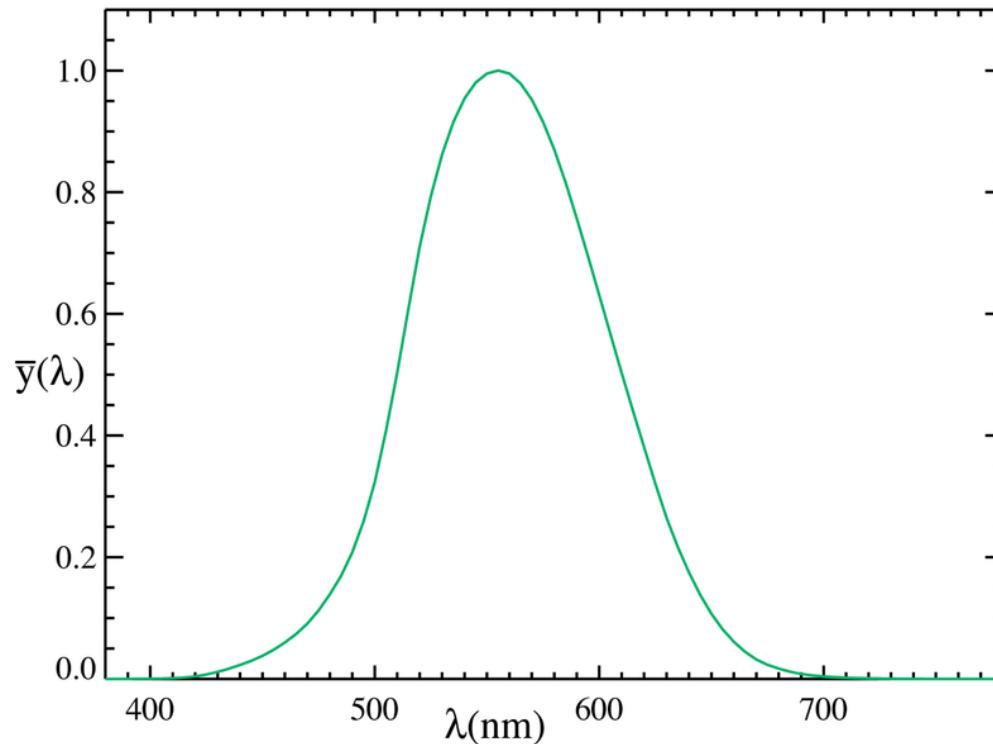
Alternate between the source light and reference light 17 times per second (17 hz). A flicker will be noticeable unless the two lights have the same perceived “brightness”.



The viewer adjusts the radiant power of the chromatic light until the flicker disappears (i.e. the lights fuse into a constant color). The amount of radiant power needed for this fusion to happen is recorded.

Repeat this flicker fusion test for each wavelength in the source light. This allows method can be used to determine the perceived “brightness” of each wavelength.

# CIE\* (1924) Photopic luminosity function

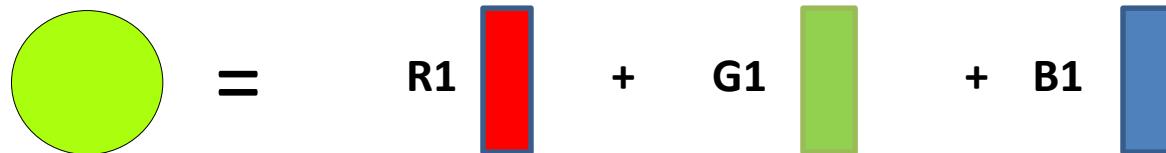


The Luminosity Function (written as  $\bar{y}(\lambda)$  or  $V(\lambda)$ ) shows the eye's sensitivity to radiant energy into luminous energy (or perceived radiant energy) based on human experiments (flicker fusion test).

\*International Commission on Illumination (CIE comes from the French name *Commission internationale de l'éclairage*) was a body established in 1913 as an authority on light, illumination and color . . CIE is still active today -- <http://www.cie.co.at>

# Colorimetry

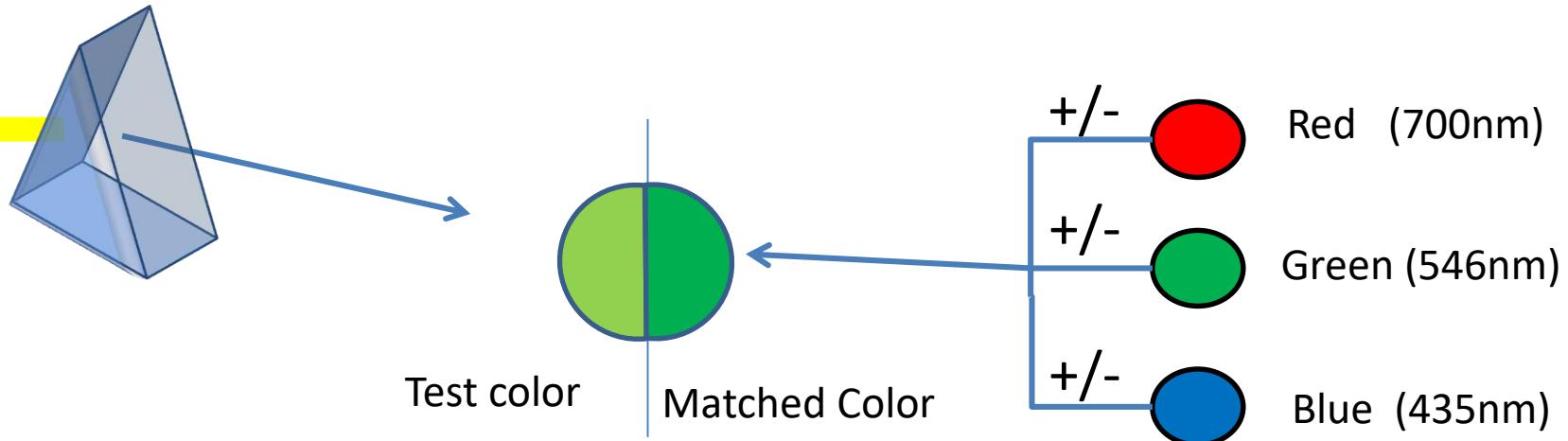
- Based on tristimulus color theory, colorimetry attempts to quantify all visible colors in terms of a standard set of primaries



Target color

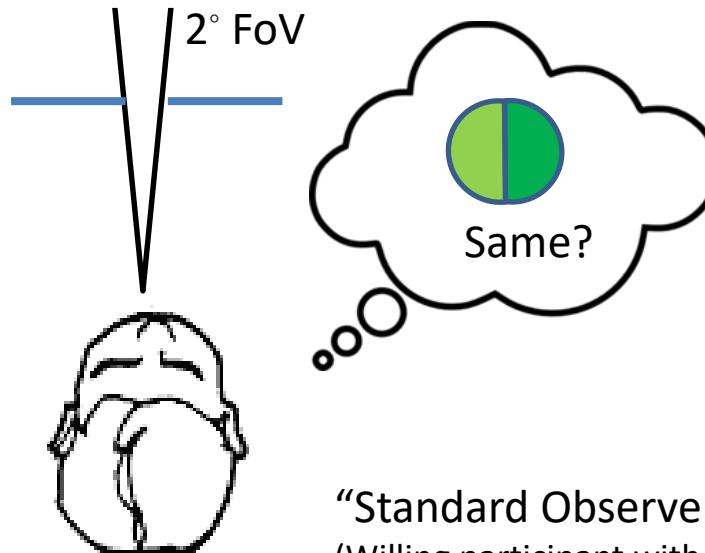
Three fixed primaries

# CIE RGB color matching



Human subjects matched test colors by add or subtracting three primaries.

Field of view was 2-degrees (where color cones are most concentrated)

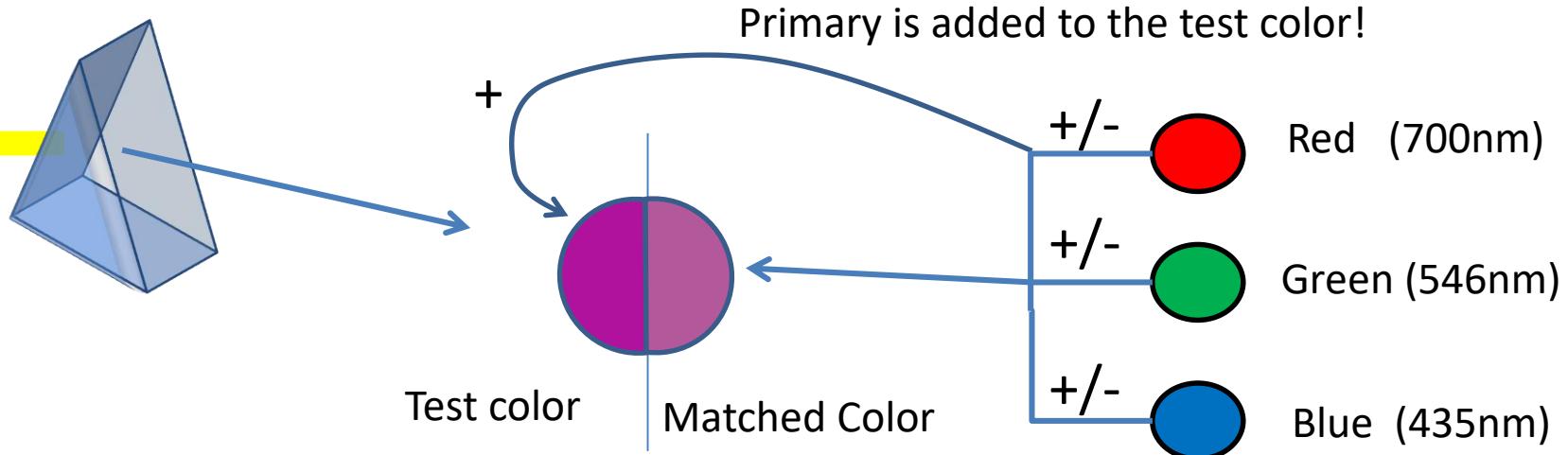


“Standard Observer”  
(Willing participant with no eye disease)

Experiments carried out by

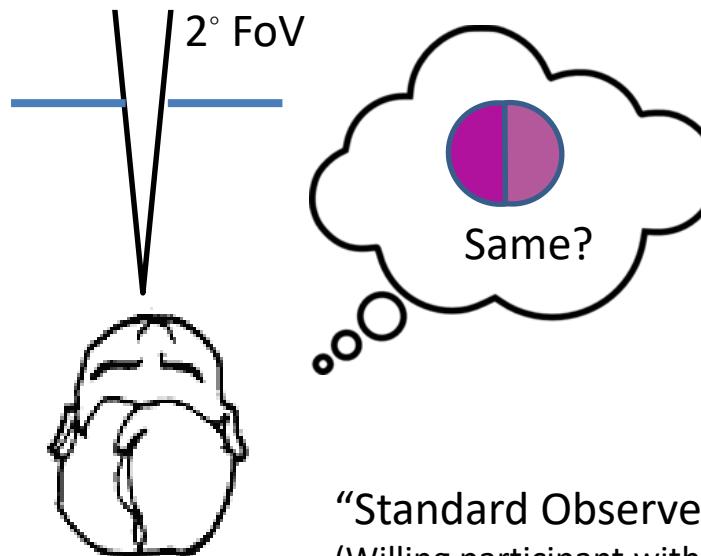
W. David Wright (Imperial College) and John Guild (National Physical Laboratory, London) – Late 1920s

# CIE RGB color matching



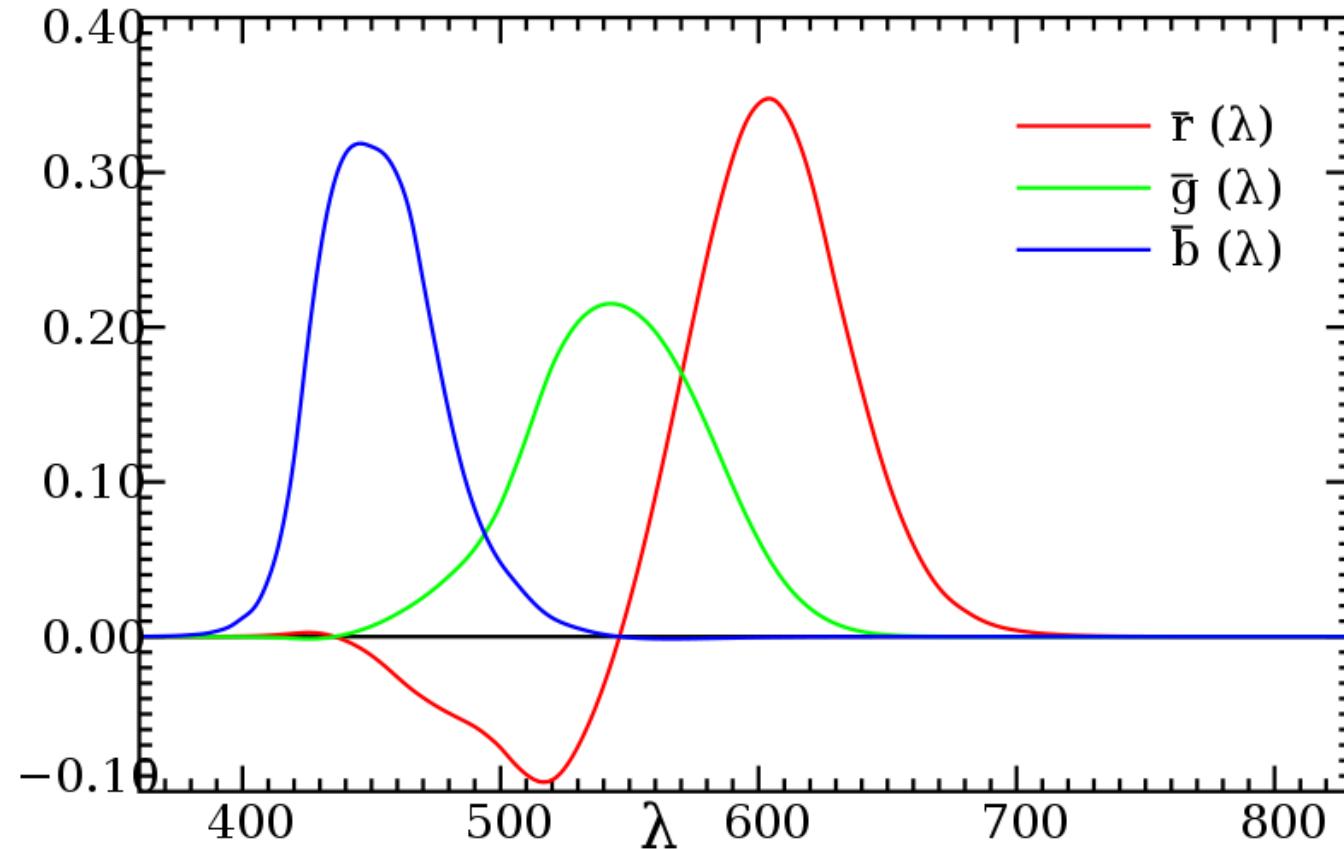
**For some test colors, no mix of the primaries could give a match!** For these cases, the subjects were asked to add primaries to the test color to make the match.

This was treated as a negative value of the primary added to the test color.



"Standard Observer"  
(Willing participant with no eye disease)

# CIE RGB results

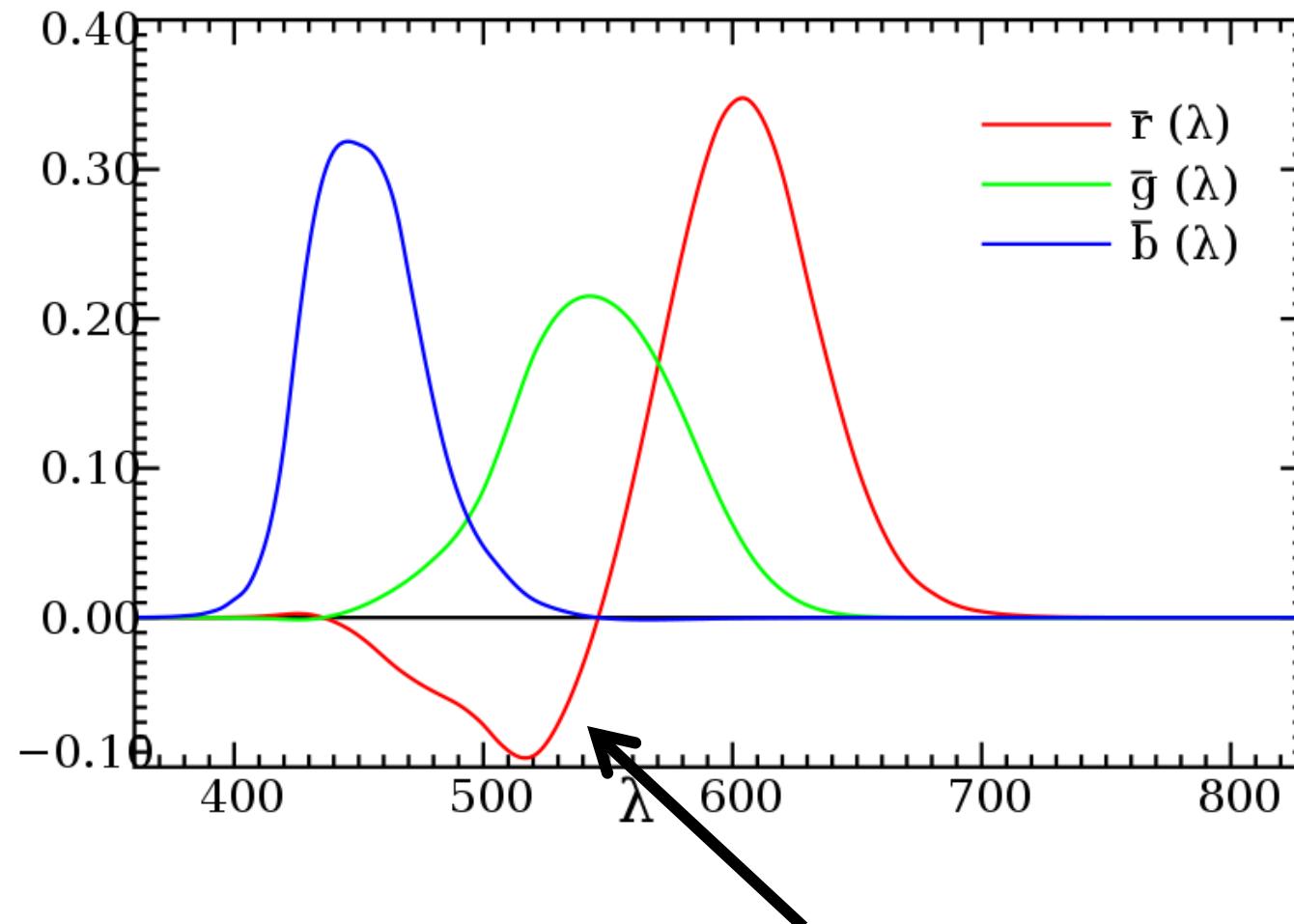


CIE RGB 2-degree Standard Observer  
(based on Wright/Guild's data)

Plots are of the mixing coefficients of each primary needed to produce the corresponding monochromatic light at that wavelength.

*Note that these functions have been scaled such that area of each curve is equal.*

# CIE RGB results



Negative points, the primary used did not span the full range of perceptual color.

# CIE 1931 - XYZ

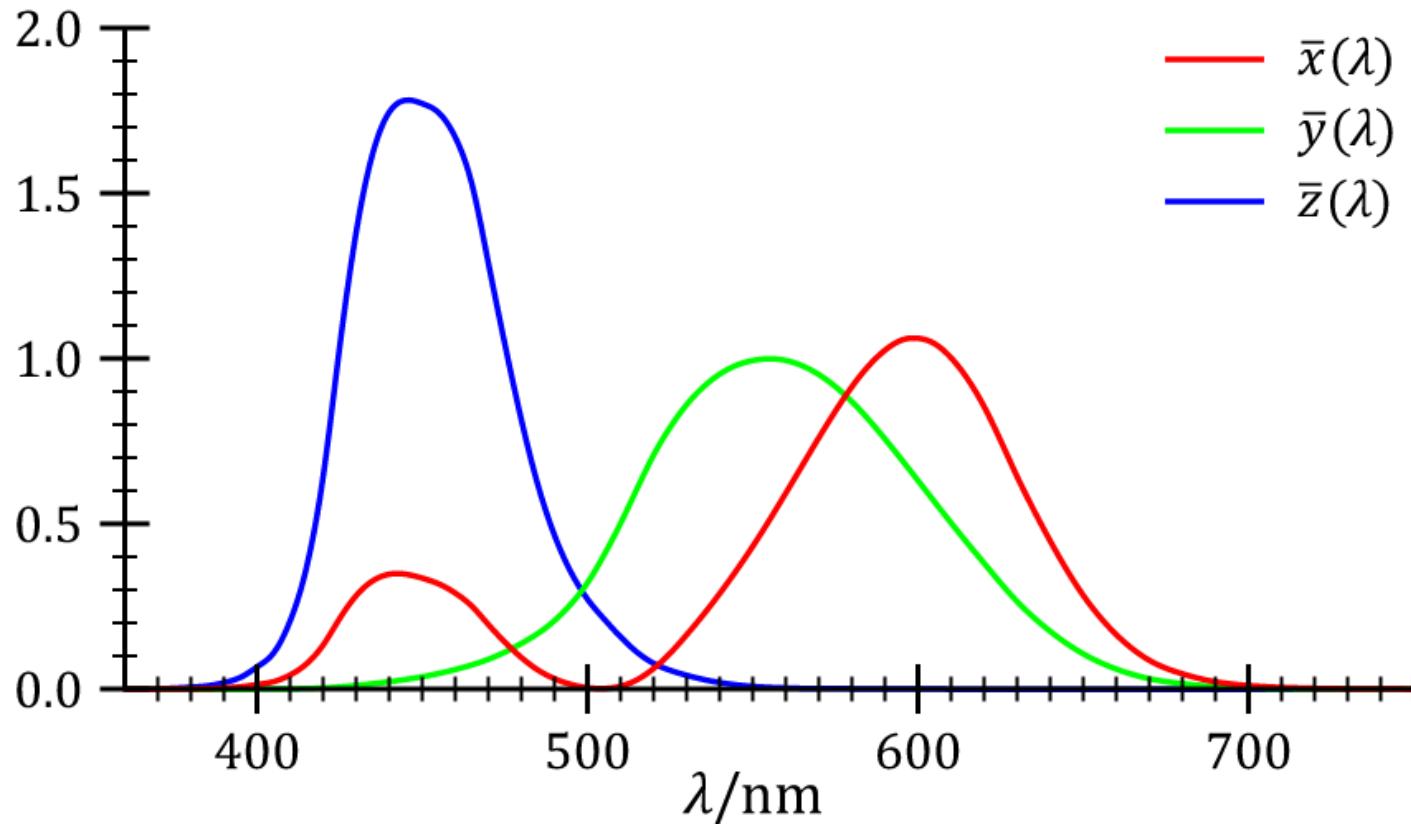
- In 1931, the CIE met and approved defining a new canonical basis, termed XYZ that would be derived from Wright-Guild's CIE RGB data
- Properties desired in this conversion:
  - White point defined at X=1/3, Y=1/3, Z=1/3
  - Y would be the luminosity function ( $V(\lambda)$ )
  - Quite a bit of freedom in selecting these XYZ basis
  - In the end, the adopted transform was:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4887180 & 0.3106803 & 0.2006017 \\ 0.1762044 & 0.8129847 & 0.0108109 \\ 0.0000000 & 0.0102048 & 0.9897952 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CIE RGB

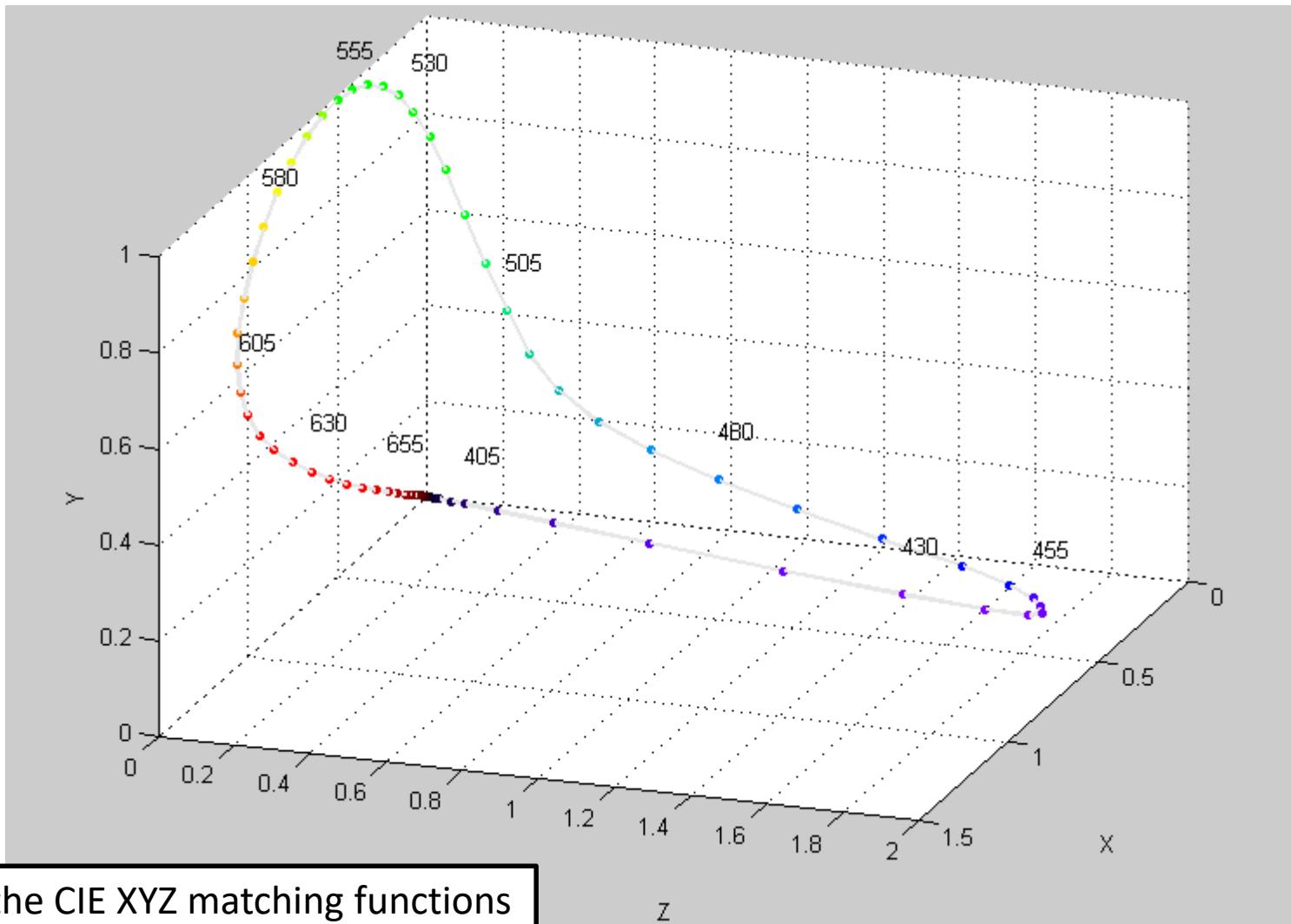
Nice article see: Fairman et al “How the CIE 1931 Color-Matching Functions Were Derived from Wright-Guild Data”, Color Research & Application, 1997

# CIE XYZ



This shows the mixing coefficients  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$  for the CIE 1931 2-degree standard observer XYZ basis computed from the CIE RGB data. Coefficients are all now positive. Note that the basis XYZ are not physical SPD like in CIE RGB, but linear combinations defined by the matrix on the previous slide.

# CIE XYZ 3D plot



3D plot of the CIE XYZ matching functions against the XYZ axis. Note that scaling of this plot is not uniform.

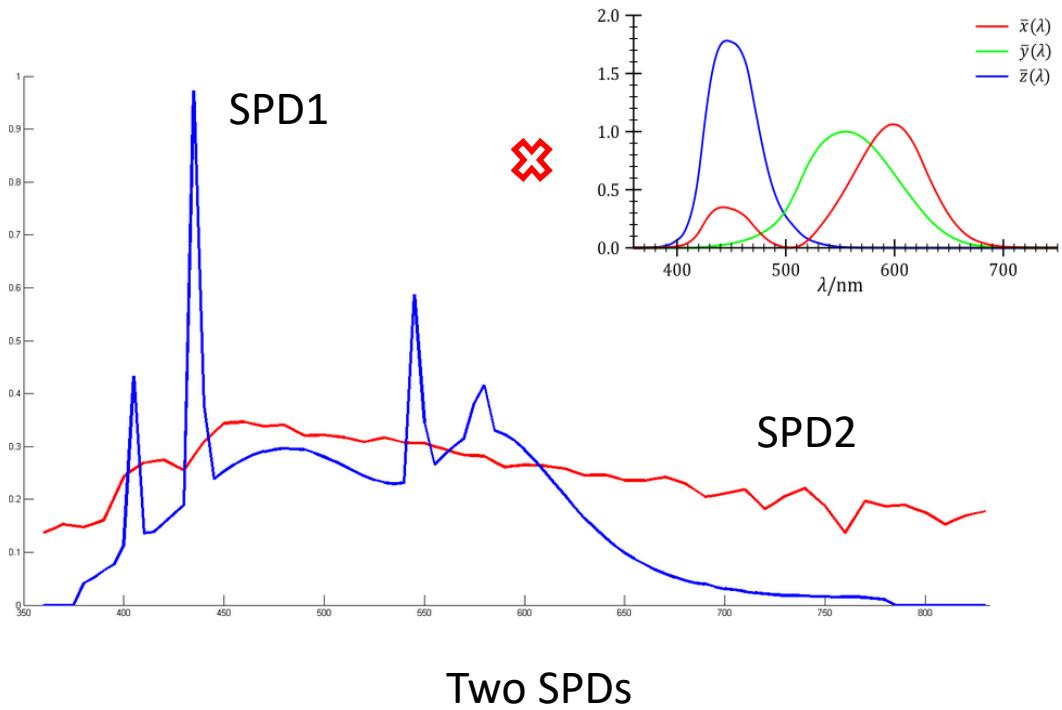
# What does it mean?

- **We now have a canonical color space to describe SPDs**
- Given an SPD,  $I(\lambda)$ , we can find its mapping into the CIE XYZ space

$$X = \int_{380}^{780} I(\lambda) \bar{x}(\lambda) d\lambda \quad Y = \int_{380}^{780} I(\lambda) \bar{y}(\lambda) d\lambda \quad Z = \int_{380}^{780} I(\lambda) \bar{z}(\lambda) d\lambda$$

- Given two SPDs, if their CIE XYZ values are equal, then they are considered the same perceived color, i.e.
  - $I_1(\lambda), I_2(\lambda) \rightarrow (X_1, Y_1, Z_1) = (X_2, Y_2, Z_2)$  [ perceived as the same color ]

# SPD to CIE XYZ example



CIE XYZ Values

SPD1

X=0.2841

Y=0.2989

Z=0.3254

SPD2

X=0.2841

Y=0.2989

Z=0.3254

Radiometric

Colorimetric

CIE XYZ gives a way to go from radiometric to colorimetric.  
Imbedded is also the photometric measurement in the Y value.

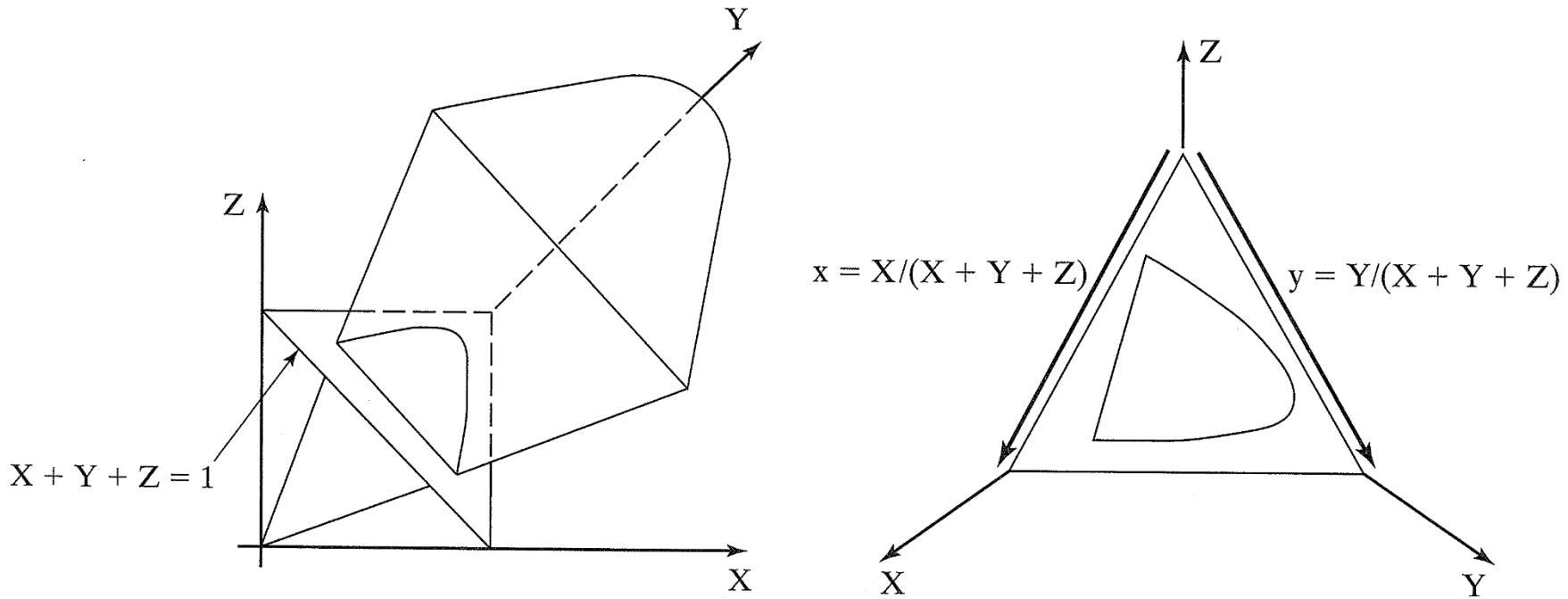
# What does it mean?

- CIE XYZ space is also considered “device independent” – the XYZ values are not specific to any device
- Devices (e.g. cameras, flatbed, scanners, printers, displays) can find mappings of their device specific values to corresponding CIE XYZ values. This provides a canonical space to match between devices (at least in theory)

# Luminance-chromaticity space (CIE xyY)

- CIE XYZ describes a color in terms of linear combination of three primaries (XYZ)
- Sometimes it is useful to discuss color in terms of luminance (perceived brightness) and chromaticity (we can think of as the hue-saturation combined)
- CIE xyY space is used for this purpose

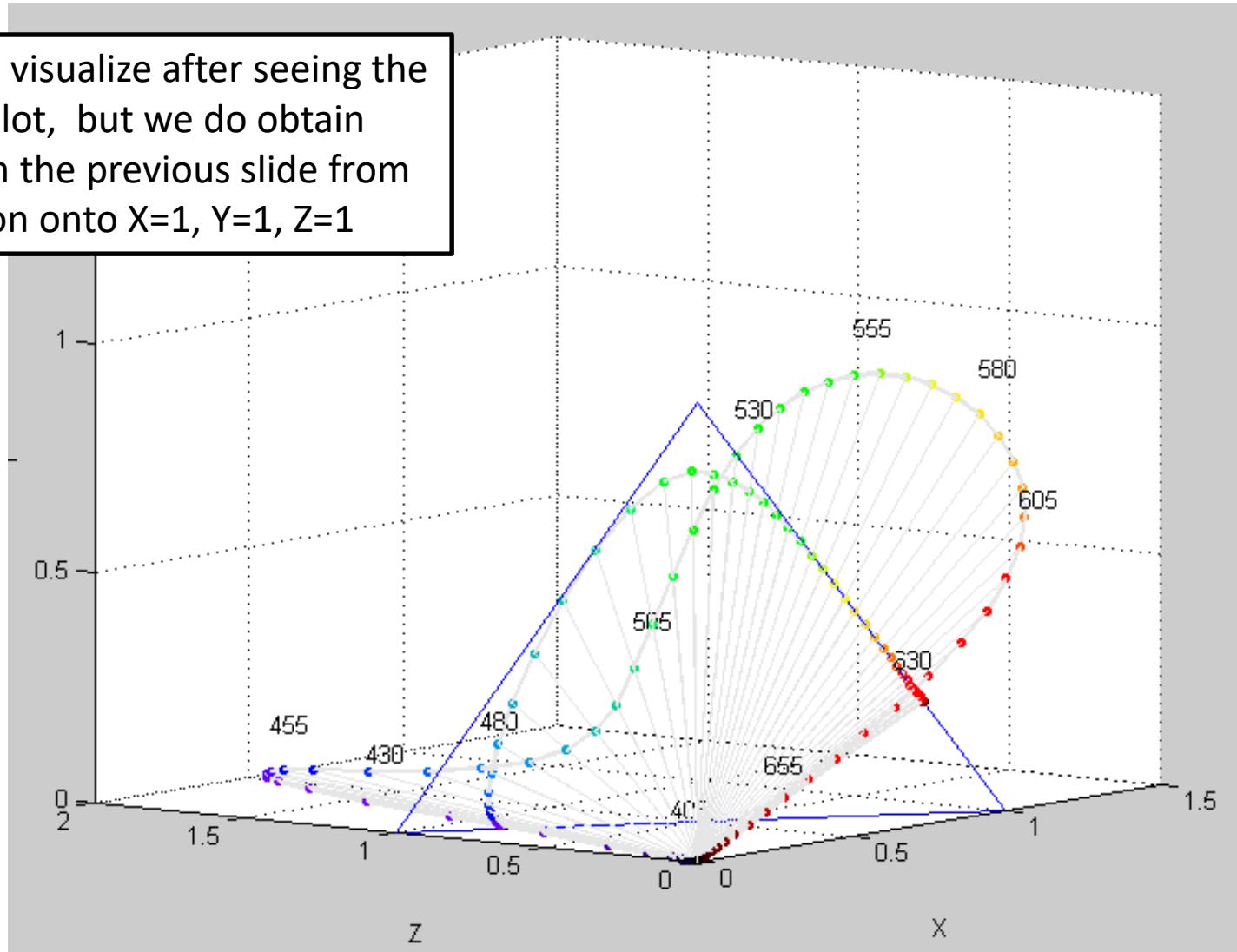
# Deriving CIE xyY



Project the CIE XYZ values onto the X=1, Y=1, Z=1 plane.

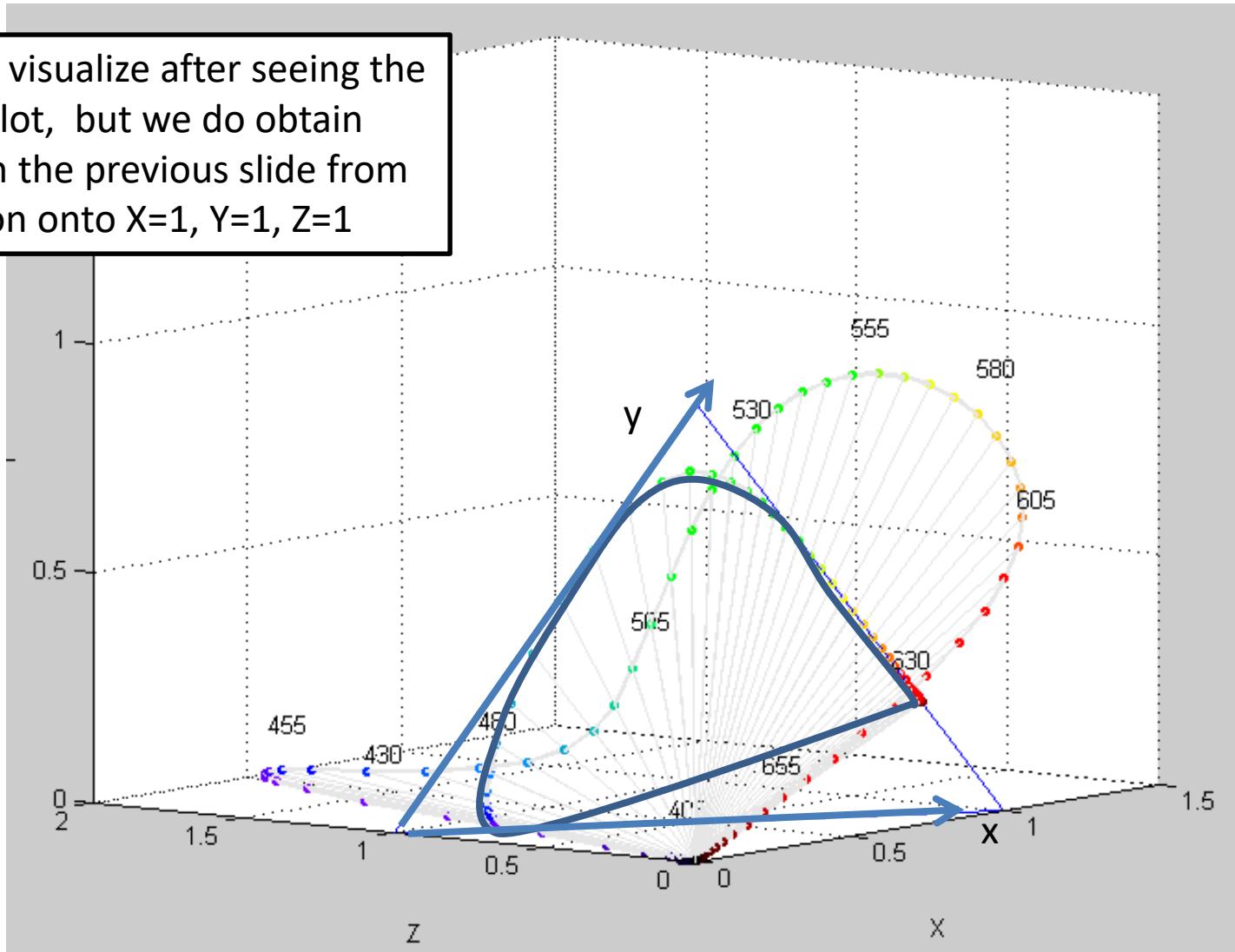
# Projection plot

A bit hard to visualize after seeing the CIE XYZ 3D plot, but we do obtain the shape on the previous slide from the projection onto  $X=1, Y=1, Z=1$



# Projection plot

A bit hard to visualize after seeing the CIE XYZ 3D plot, but we do obtain the shape on the previous slide from the projection onto X=1, Y=1, Z=1

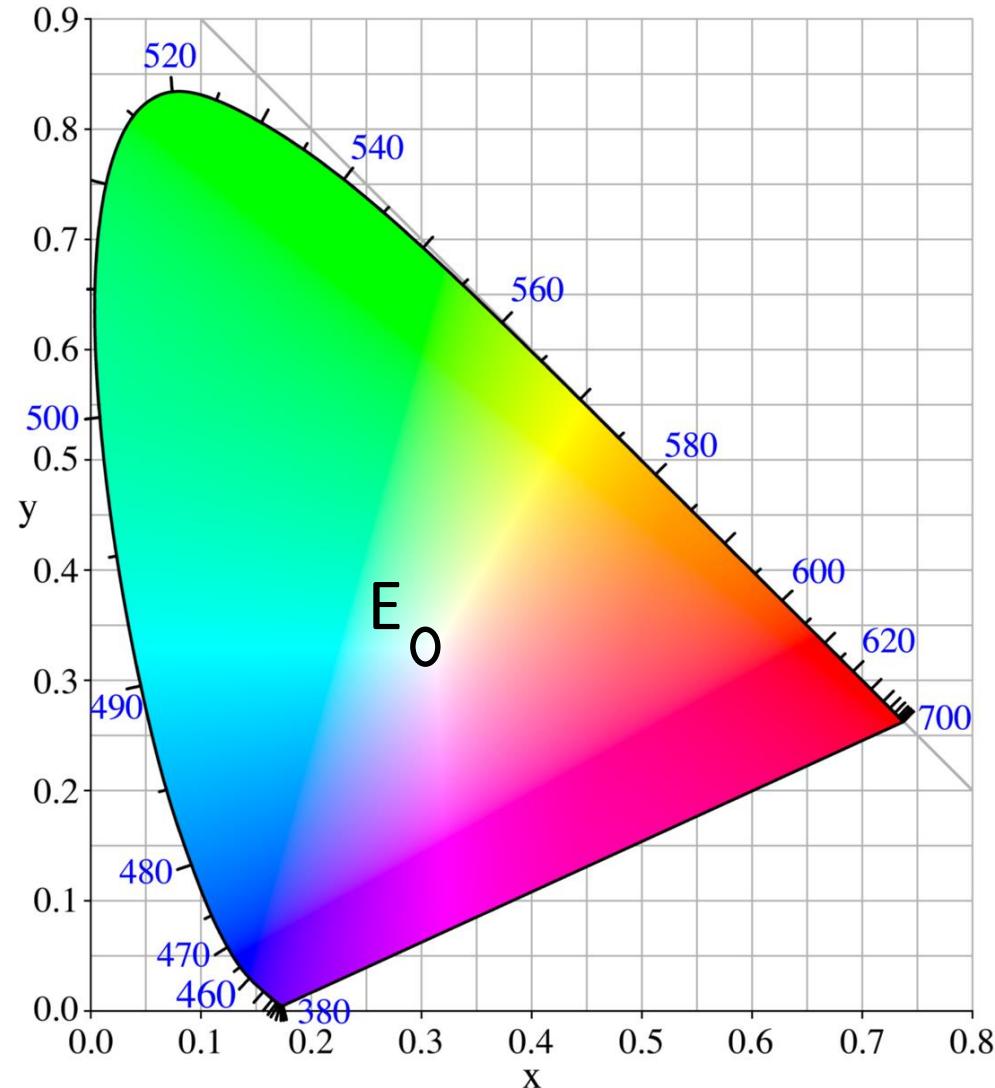


# CIE x-y chromaticity diagram

This gives us the familiar horseshoe shape of visible colors as 2D plot. Note the axis are x & y.

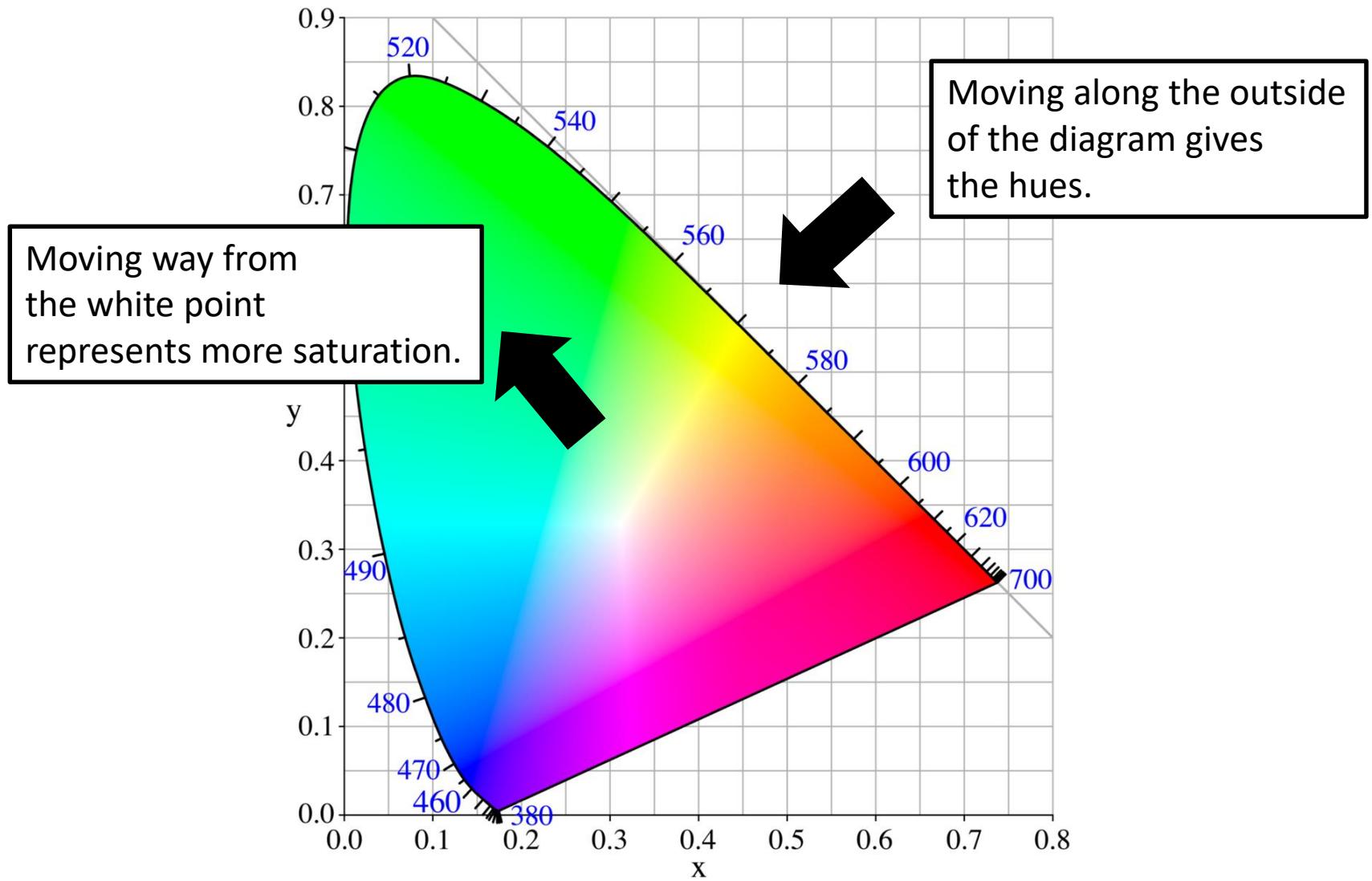
Point “E” represents where  $X=Y=Z$  have equal energy ( $X=0.33$ ,  $Y=0.33$ ,  $Z=0.33$ )

CIE XYZ “white point”



In the 1930s, CIE had a bad habit of over using the variables X, Y. Note that x, y are chromaticity coordinates,  $\bar{x}$ ,  $\bar{y}$  (with the bar above) are the matching functions, and X, Y are the imaginary SPDs of CIE XYZ.

# CIE x-y chromaticity diagram



# CIE xyY

- Generally when we use CIE xyY, we only look at the (x,y) values on the 2D diagram of the CIE x-y chromaticity chart
- However, the Y value (the same Y from CIE XYZ) represents the perceived brightness of the color
- With values (x,y,Y) we can reconstruct back to XYZ

$$X = \frac{Y}{y}x \quad Z = \frac{Y}{y}(1 - x - y)$$

# Fast forward 80+ years

- CIE 1931 XYZ, CIE 1931 xyY (2-degree standard observer) color spaces have stood the test of time
- Many other studies have followed (most notably - CIE 1965 XYZ 10-degree standard observer), . . .
- But in the literature (and in this tutorial) you'll find CIE 1931 XYZ color space making an appearance often

# What is perhaps most amazing?

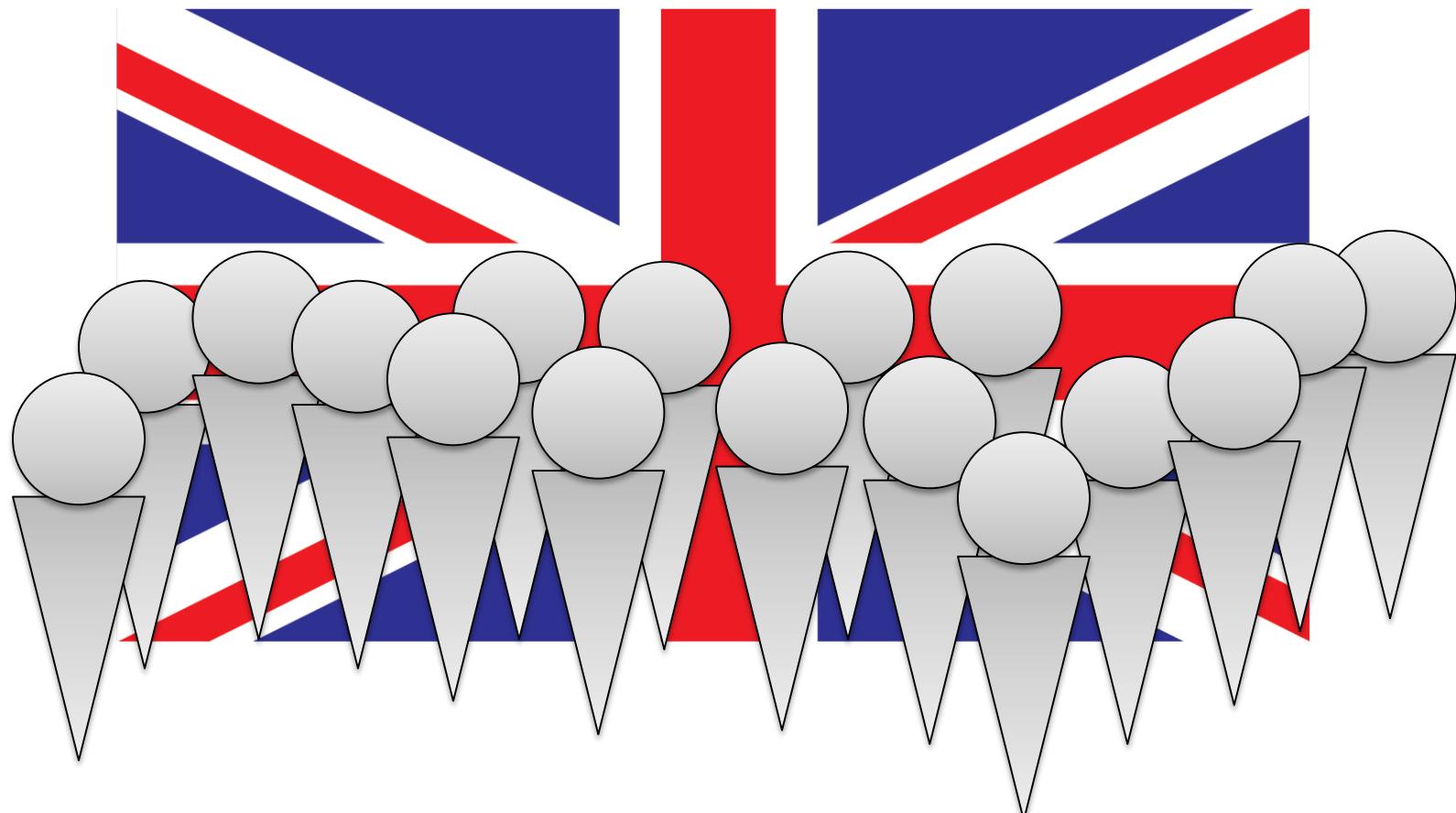
- 80+ years of CIE XYZ is all down to the experiments by the “standard observers”
- How many standard observers were used?  
100, 500, 1000?



A Standard Observer

# CIE XYZ is based on 17 Standard Observers

10 by Wright, 7 by Guild



“The Standard Observers”

# A caution on CIE x-y chromaticity

From Mark D. Fairchild book: “Color Appearance Models”

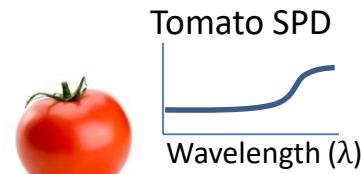
*“The use of chromaticity diagrams should be avoided in most circumstances, particularly when the phenomena being investigated are highly dependent on the three-dimensional nature of color. For example, the display and comparison of the color gamuts of imaging devices in chromaticity diagrams is misleading to the point of being almost completely erroneous.”*

We are done with color, right?  
Almost . . .

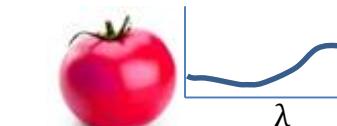
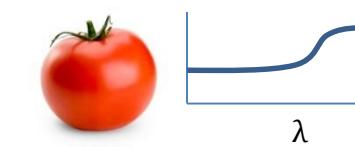
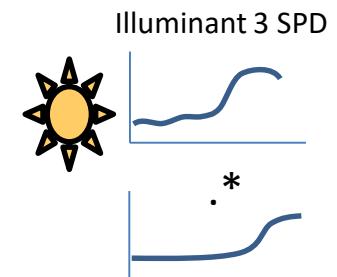
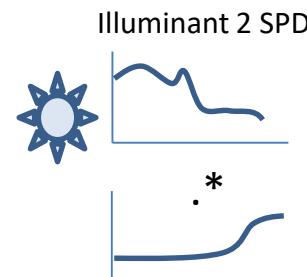
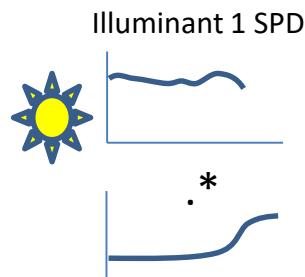
# An object's SPD

- In a real scene, an object's SPD is a combination of its reflectance properties and scene illumination

Our earlier example  
ignored illumination  
(we could assume it was pure  
white light).

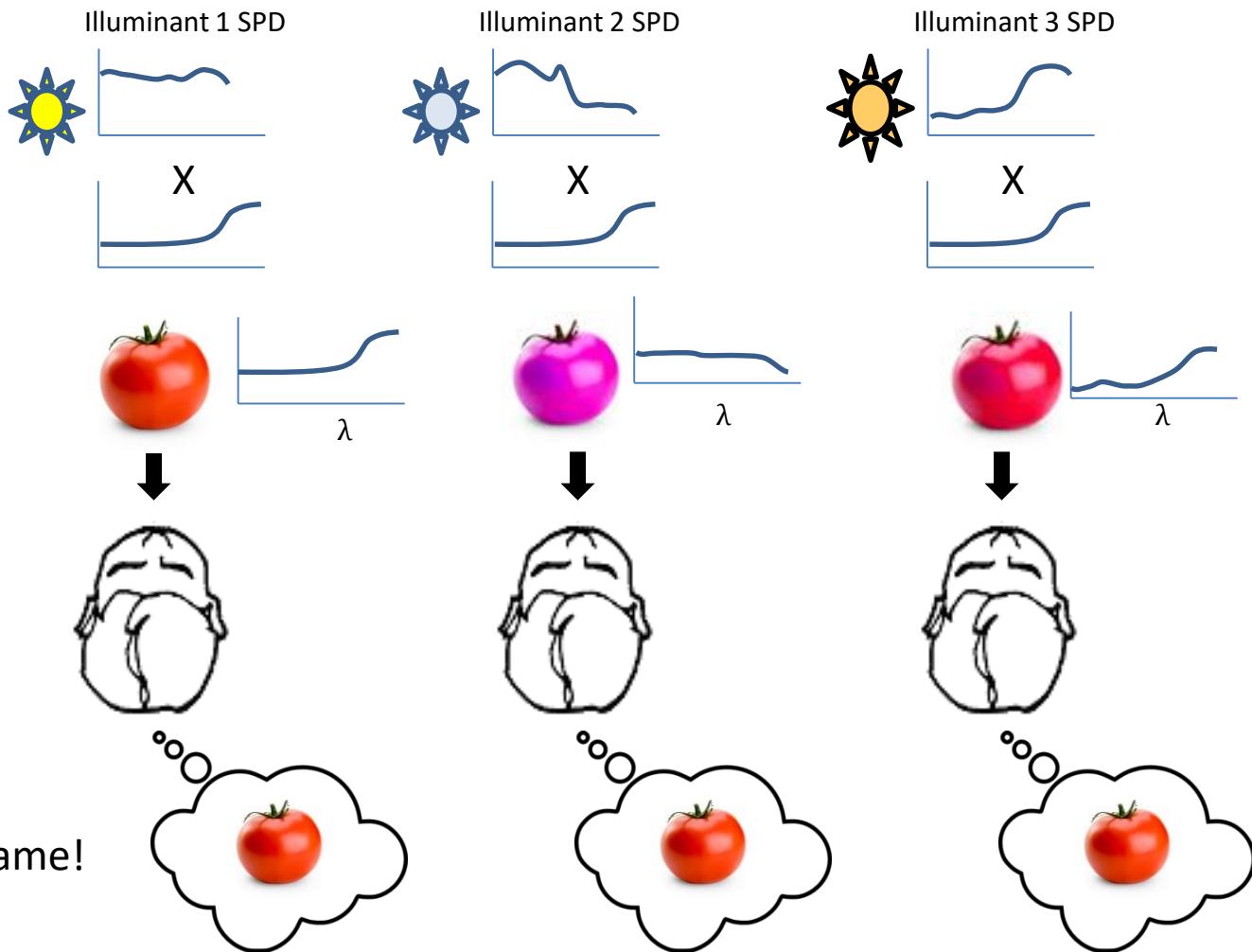


Instead, think of this  
of how the object  
reflects different  
wavelengths



# Color constancy

- Our visual system is able to compensate for the illumination



# Color constancy/chromatic adaptation

- Color constancy, or chromatic adaptation, is the ability of the human visual system to adapt to scene illumination
- This ability is not perfect, but it works fairly well
- Image sensors do not have this ability (it must be performed as a processing step, i.e. “white balance”)
- **Note:** Our eyes do not adjust to the lighting in the photograph -- we adjust to the viewing conditions of the scene we are viewing the photograph in!

# Color constancy and illuminants

- To understand color constancy, we have to consider SPDs of different illuminants

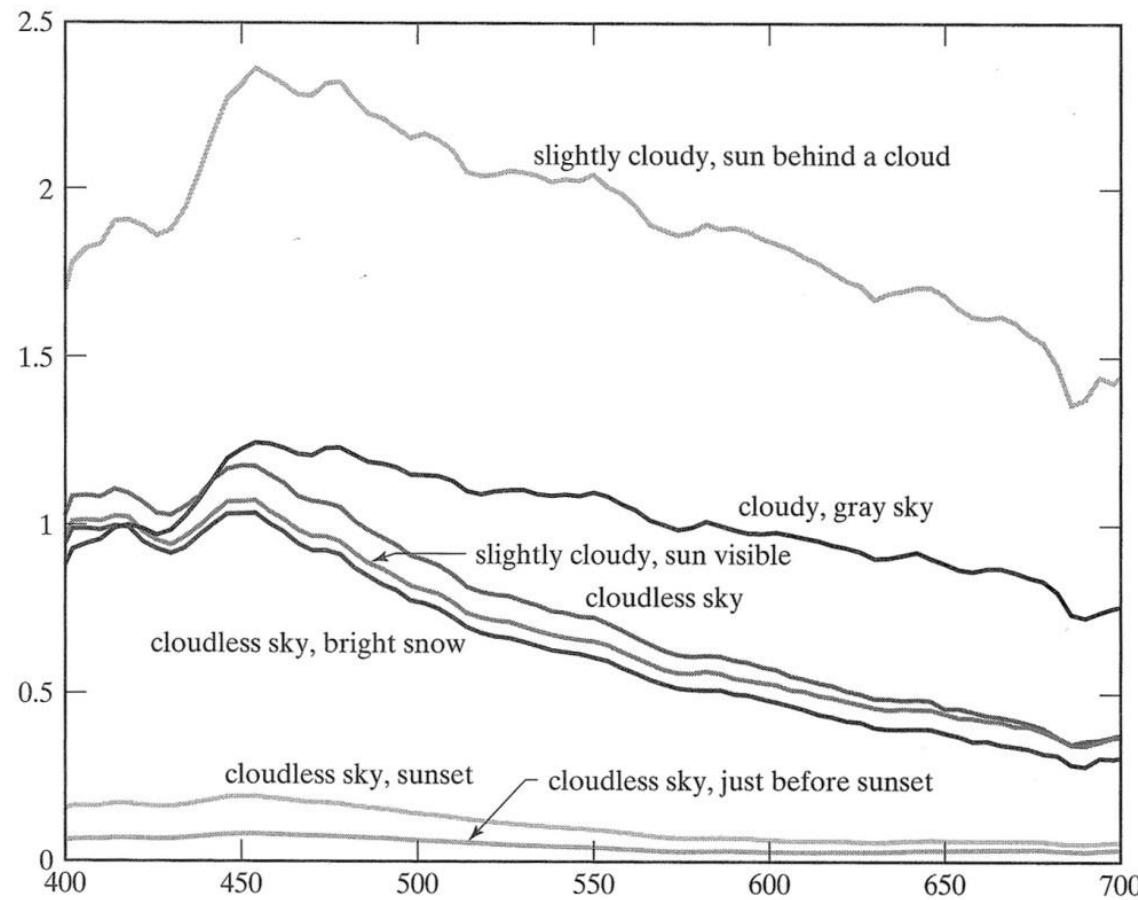
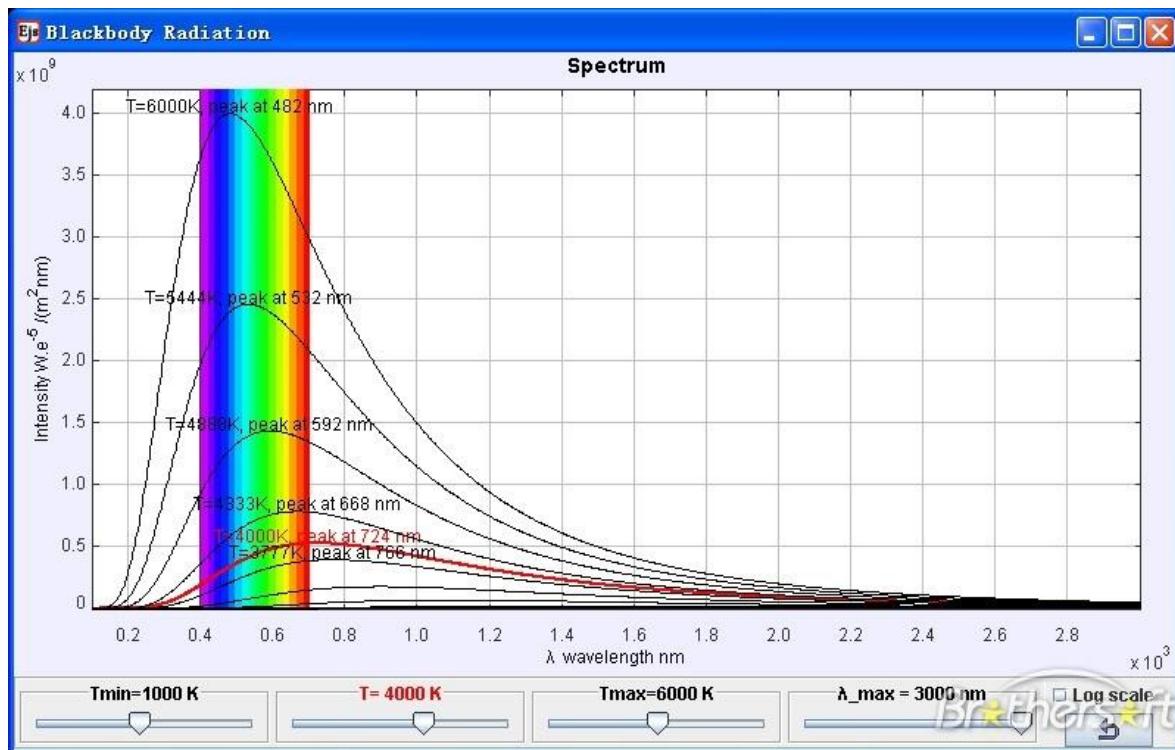


Figure from Ponce and Forsyth

# Color temperature

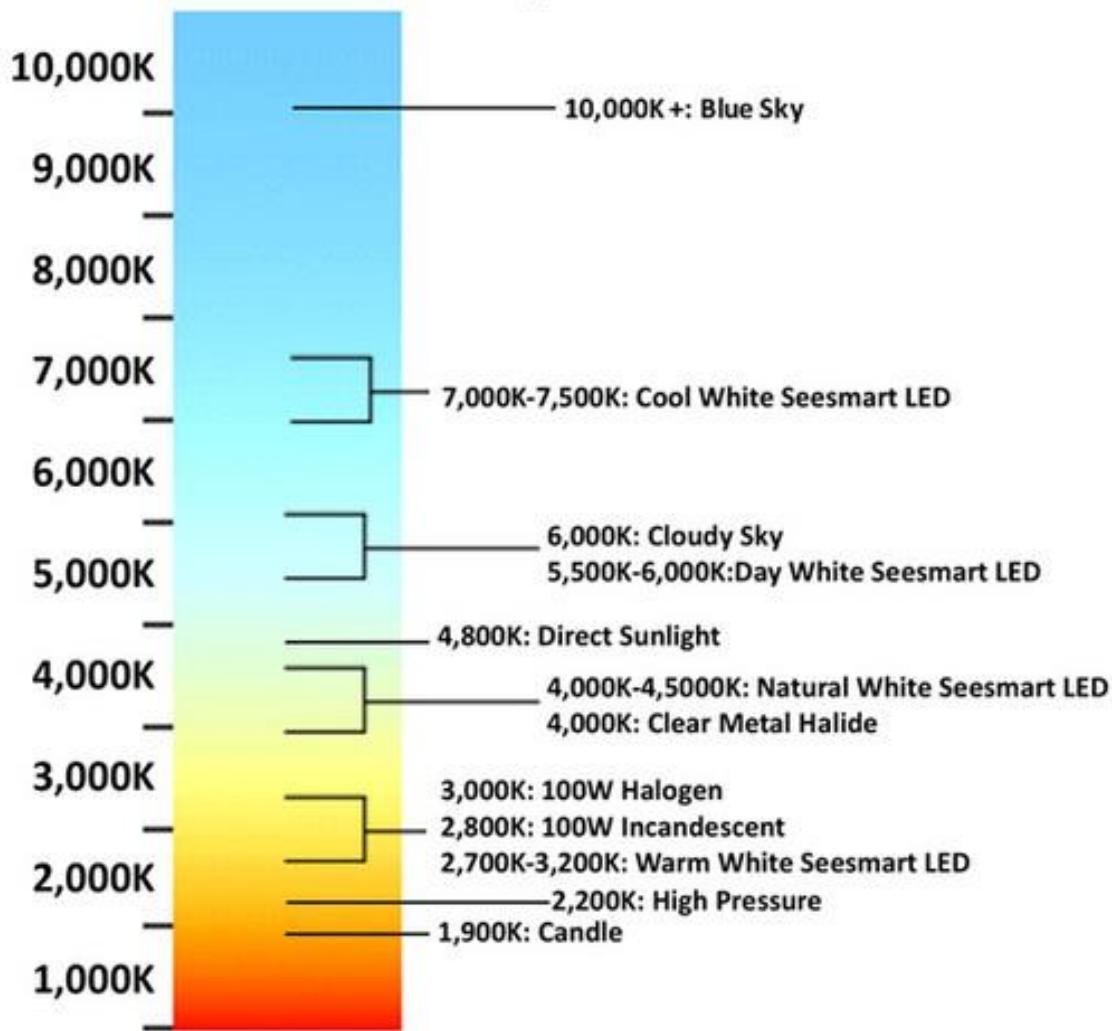
- Illuminants are often described by their color temperature
- This is associated with theoretical “blackbody radiators” that produce SPDs based on a given temperature (expressed in kelvins)
- We often map light sources (both real and synthetic) to their closest color temperature (esp in Photography/Video production)



Freeware app by, **Fu-Kwun Hwang** to generate blackbody SPDs at different temperatures, showing the range in the visible spectrum.

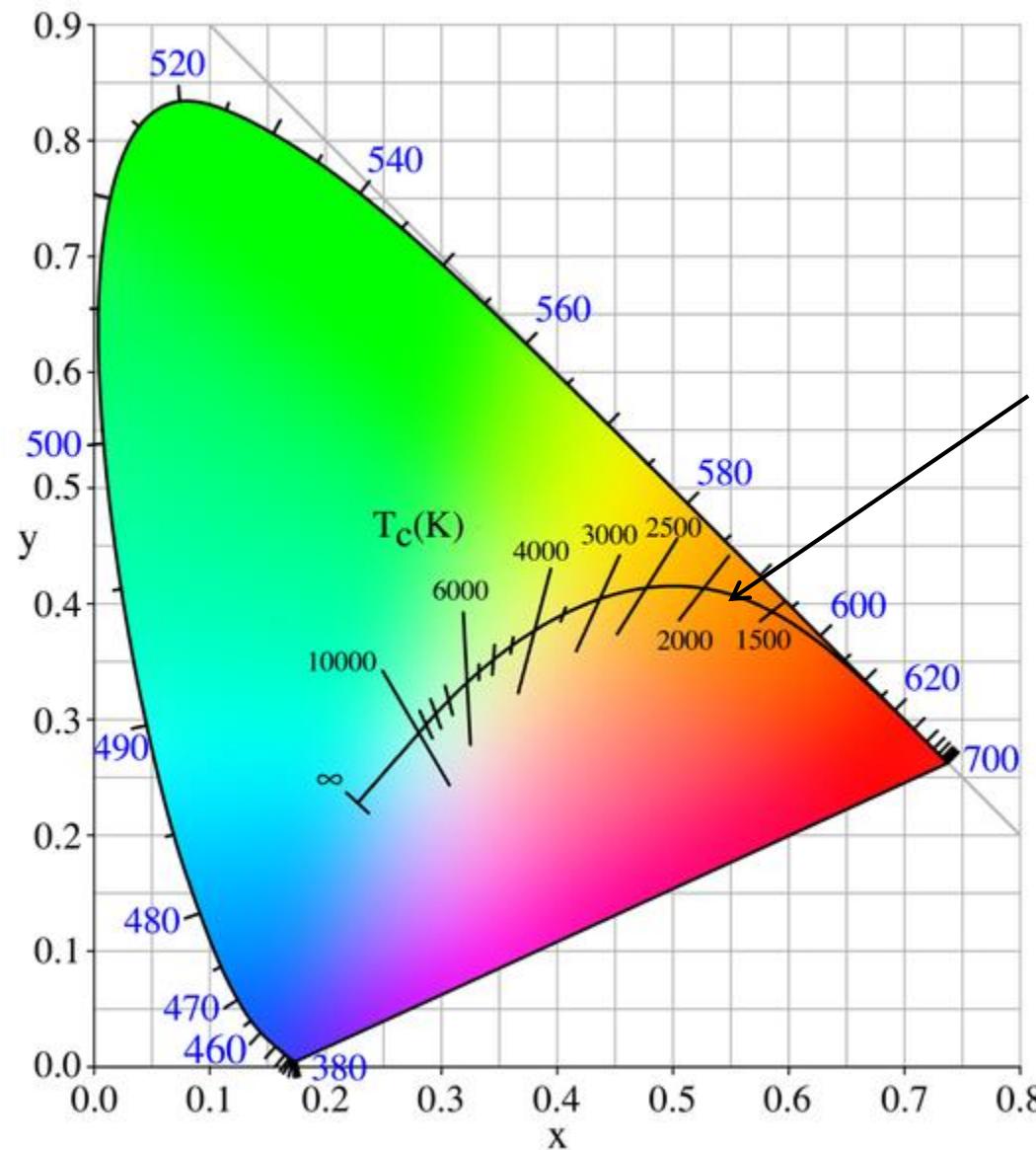
# Color temperature

## Kelvin Color Temperature Scale



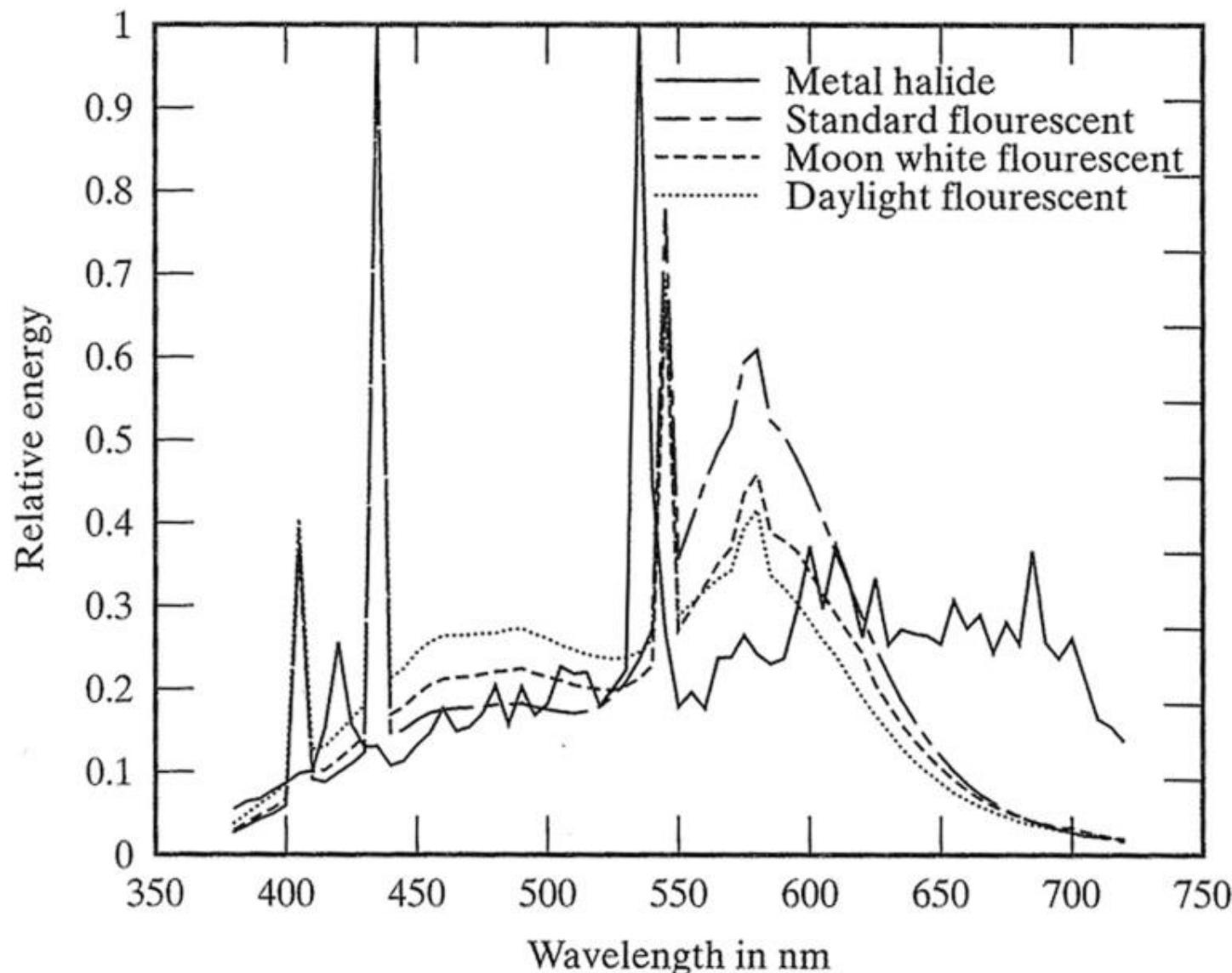
Typical description of color temperature used in photography & lighting sources.

# Plotted in CIE x-y chromaticity



Plot of color CIE xy locations of SPDs based on color temperature.

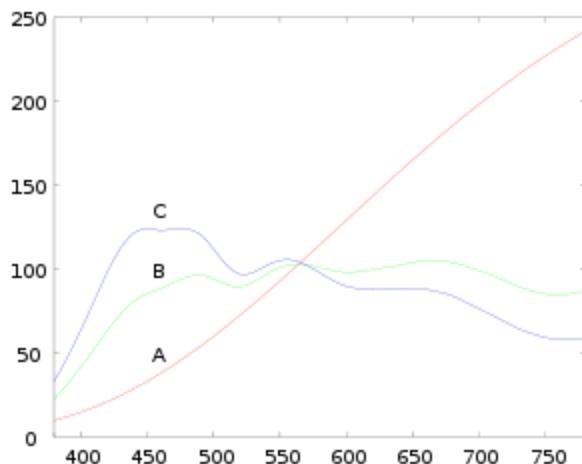
# Man made illuminants SPDs



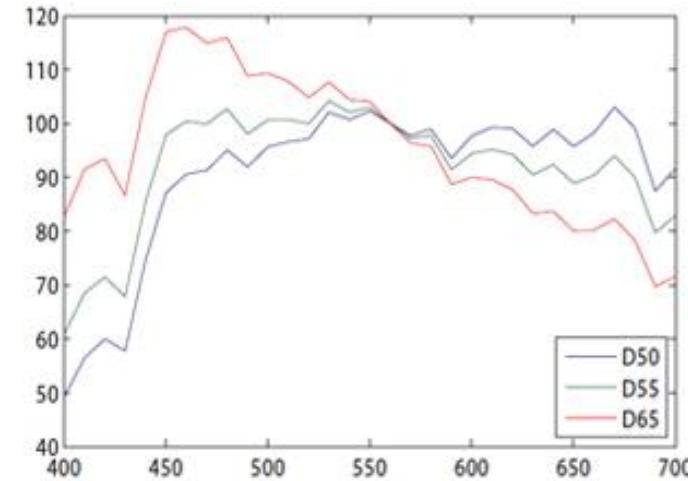
# CIE standard illuminants

- CIE established several “synthetic” SPDs that serve as proxies for common real illuminants
- Illuminant A
  - tungsten-filament lighting (i.e. a standard light-bulb)
- Illuminant B
  - noon sunlight
- Illuminant C
  - average daylight
- Illuminant D series
  - represent natural daylight at various color temps (5000K, 5500K, 6500K), generally denoted as D50, D55, D65
- Illuminant E
  - idea equal-energy illuminant with constant SPD
  - does not represent any real light source, but similar to D55
- Illuminant F series
  - emulates a variety of fluorescents lamps (12 in total)

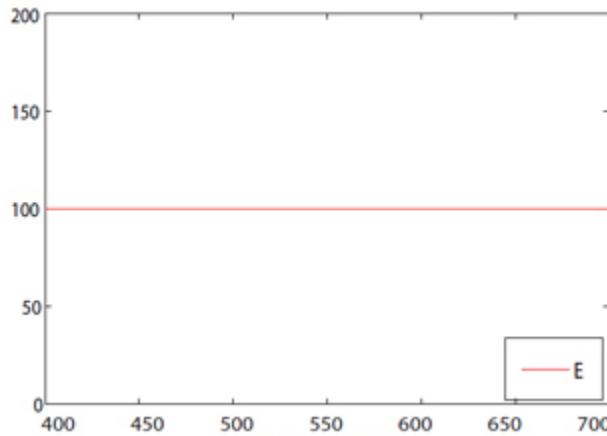
# CIE standard illuminants



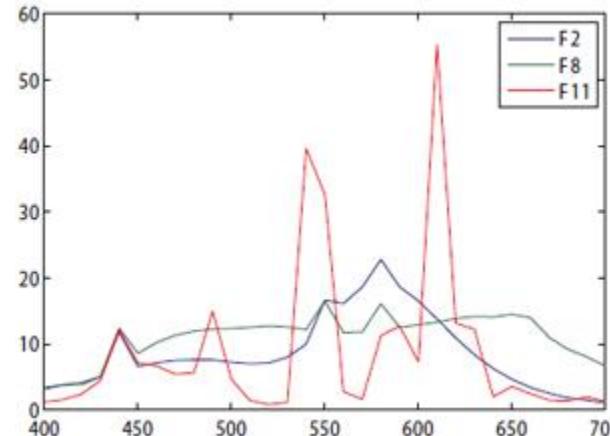
SPDs for CIE standard illuminant A, B, C



SPDs for CIE standard illuminant D50, D55, D65



SPDs for CIE standard illuminant E

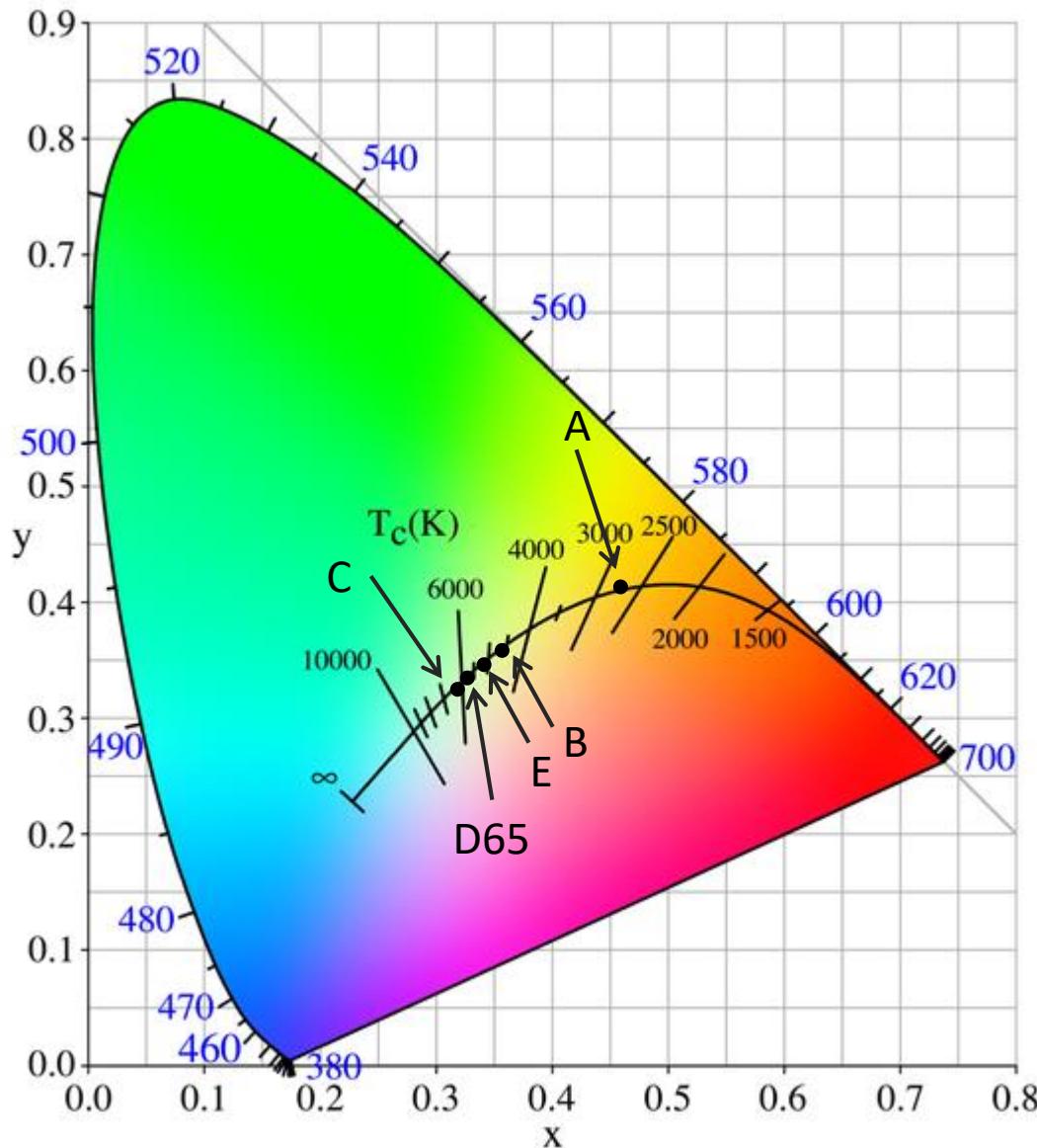


SPDs for CIE standard illuminants F2, F8, F11

# White point

- A white point is a CIE XYZ or CIE xyY value of an ideal “white target” or “white reference”
- This is essentially an illuminants SPD in terms of CIE XYZ/CIE xyY
  - We can assume the white reference is reflecting the illuminant
- The idea of chromatic adaptation is to make white points the same between scenes

# White points in CIE x-y chromaticity



## CIE Illuminants

A, B, C, D65, E in terms of CIE x-y

CIE	x	,	y
A	0.44757	,	0.40745
B	0.34842	,	0.35161
C	0.31006	,	0.31616
D65	0.31271	,	0.32902
E	0.33333	,	0.33333

# Color constancy (at its simplest)

- (Johannes) Von Kries transform
- Compensate for each channel corresponding to the L, M, S cone response

$$\begin{bmatrix} L_2 \\ M_2 \\ S_2 \end{bmatrix} = \begin{bmatrix} 1/L_{1w} & 0 & 0 \\ 0 & 1/M_{1w} & 0 \\ 0 & 0 & 1/S_{1w} \end{bmatrix} \begin{bmatrix} L_1 \\ M_1 \\ S_1 \end{bmatrix}$$

$L_2, M_2, S_2$  is the new LMS response with the illuminant divided “out”. In this case white is equal to [1,1,1]

$L_{1w}, M_{1w}, S_{1w}$  is the LMS response to “white” under this illuminant

$L_1, M_1, S_1$  are the input LMS space under an illuminant.

# Illuminant to illuminant mapping

- More appropriate would be to map to another illuminant's LMS response (e.g. in the desired viewing condition)
- $(LMS)_1$  under an illuminant with white-response  $(L_{1w}, M_{1w}, S_{1w})$
- $(LMS)_2$  under an illuminant with white-response  $(L_{2w}, M_{2w}, S_{2w})$

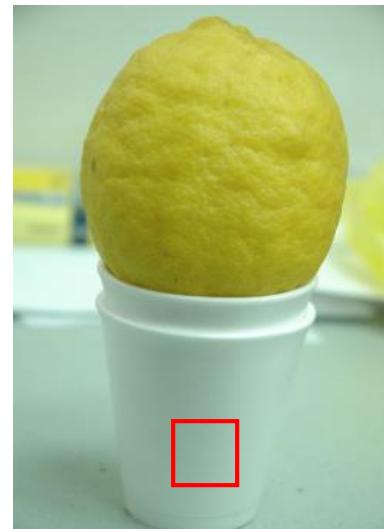
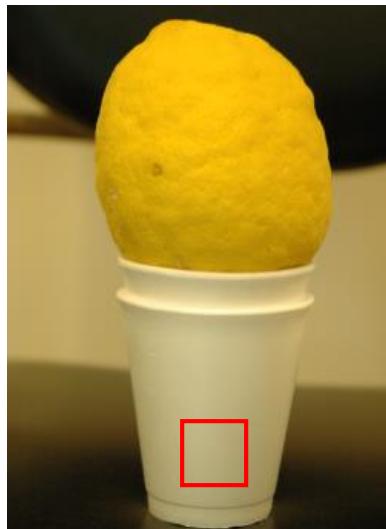
$$\begin{bmatrix} L_2 \\ M_2 \\ S_2 \end{bmatrix} = \begin{bmatrix} L_{2w}/L_{1w} & 0 & 0 \\ 0 & M_{2w}/M_{1w} & 0 \\ 0 & 0 & S_{2w}/S_{1w} \end{bmatrix} \begin{bmatrix} L_1 \\ M_1 \\ S_1 \end{bmatrix}$$

$L_2, M_2, S_2$  is the new LMS response with the illuminant divided "out" and scaled to LMS<sub>2</sub> illuminant

$L_{1w}, M_{1w}, S_{1w}$  is the LMS response to "white" the input illuminant,  $L_{2w}, M_{2w}, S_{2w}$  response to "white" of output illuminant

$L_1, M_1, S_1$  are the input LMS space under an illuminant.

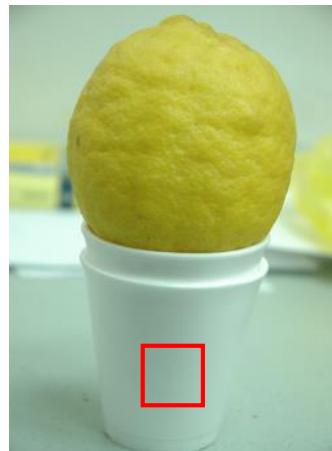
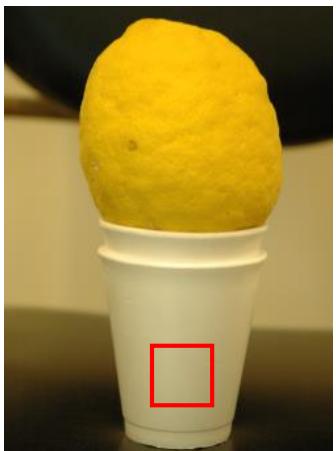
# Example



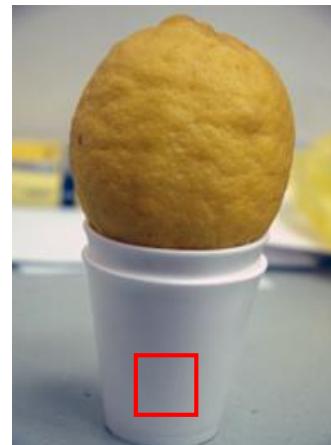
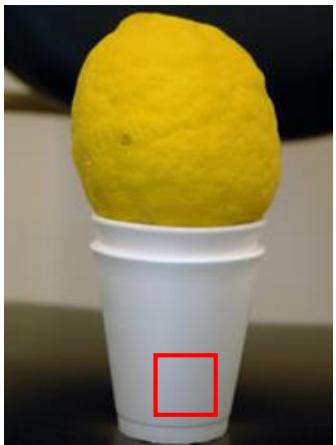
*Simulation of different “white points” by photographing a “white” object under different illumination.*

# Example

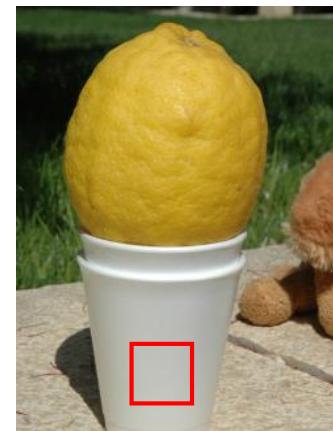
Input



Adapted to  
“target”  
illuminant



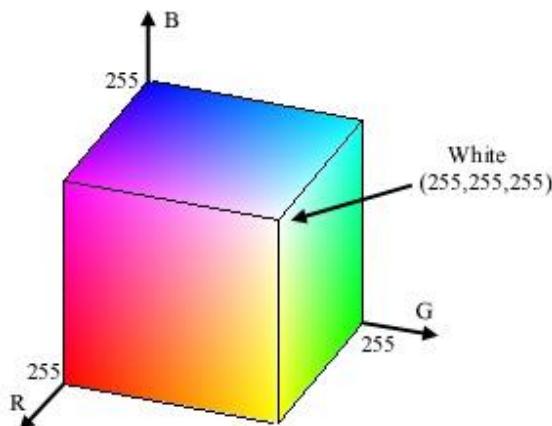
Here, we have mapped the two input images to one below to mimic chromatic adaptation. The “white” part of the cup is shown before and after to help show that the illumination falling on white appears similar after the “chromatic adaptation”.



Now we are finally done with color?  
Almost (really) . . .

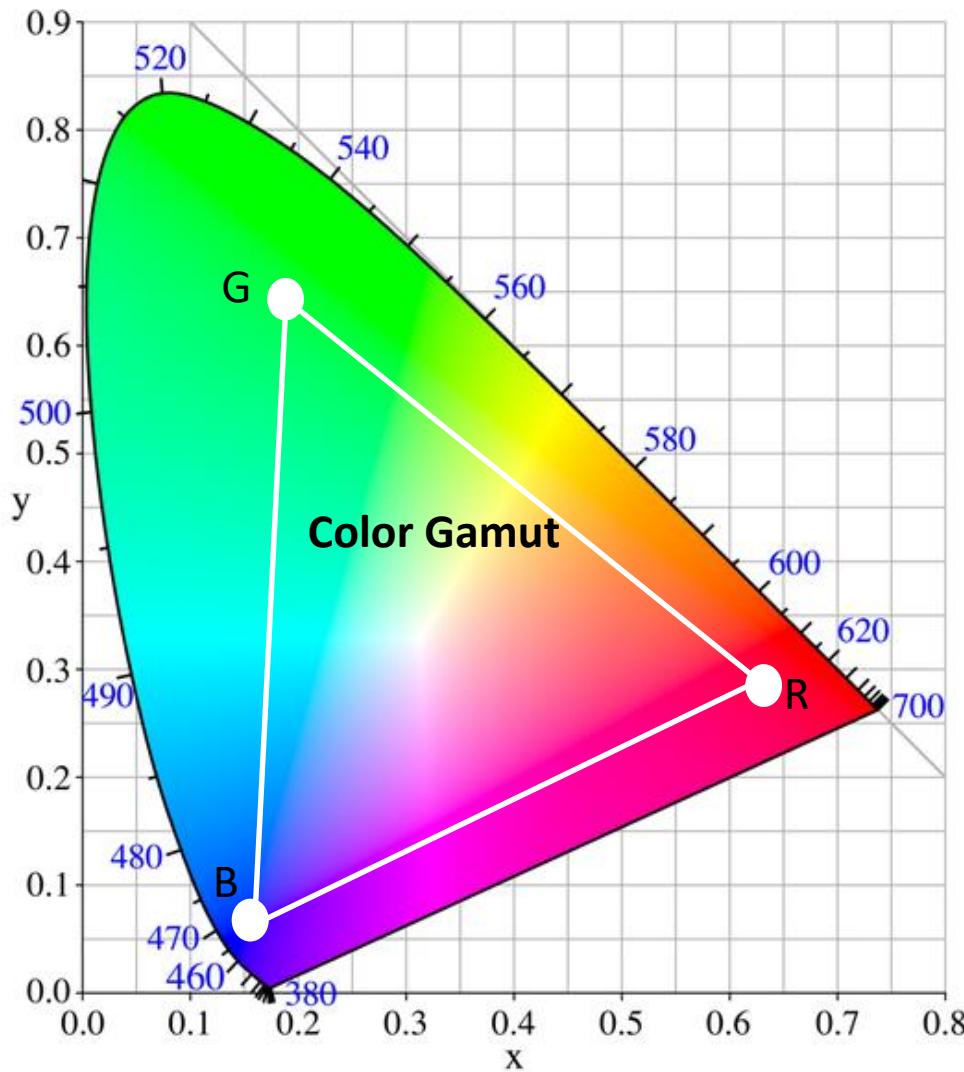
# CIE XYZ and RGB

- While CIE XYZ is a canonical color space, images/devices rarely work directly with XYZ
- XYZ are not real primaries
- RGB primaries dominate the industry
- We are all familiar with the RGB color cube



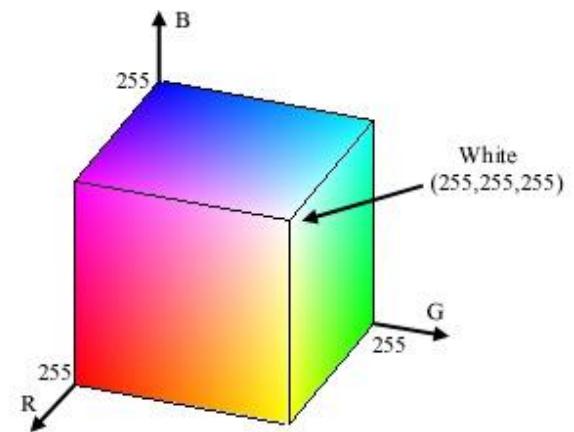
But by now, you should realize that Red, Green, Blue have no quantitative meaning. We need to know their corresponding SPDs or CIE XYZ values

# Device specific RGB values

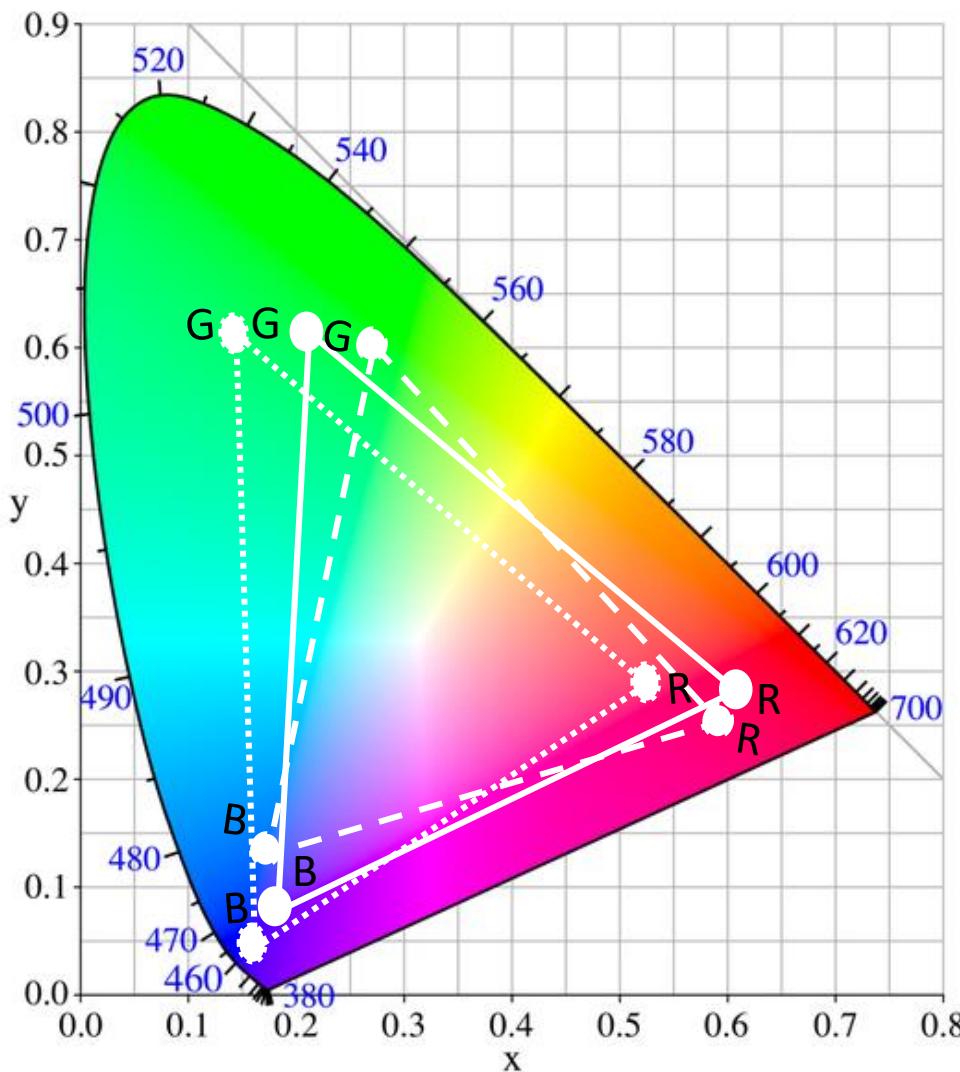


The RGB values span a subspace of CIE-XYZ to define the devices gamut.

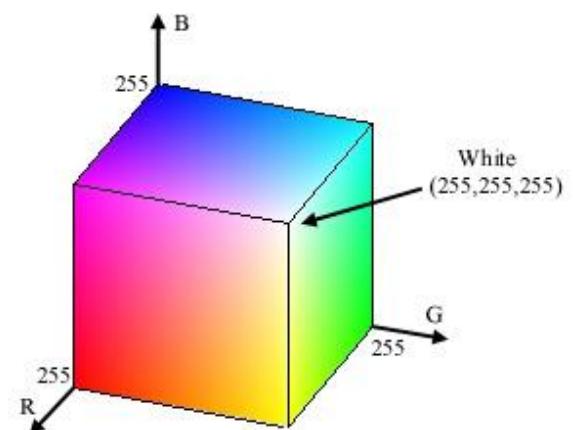
If you have RGB values, they are specific to a particular device .



# Trouble with RGB

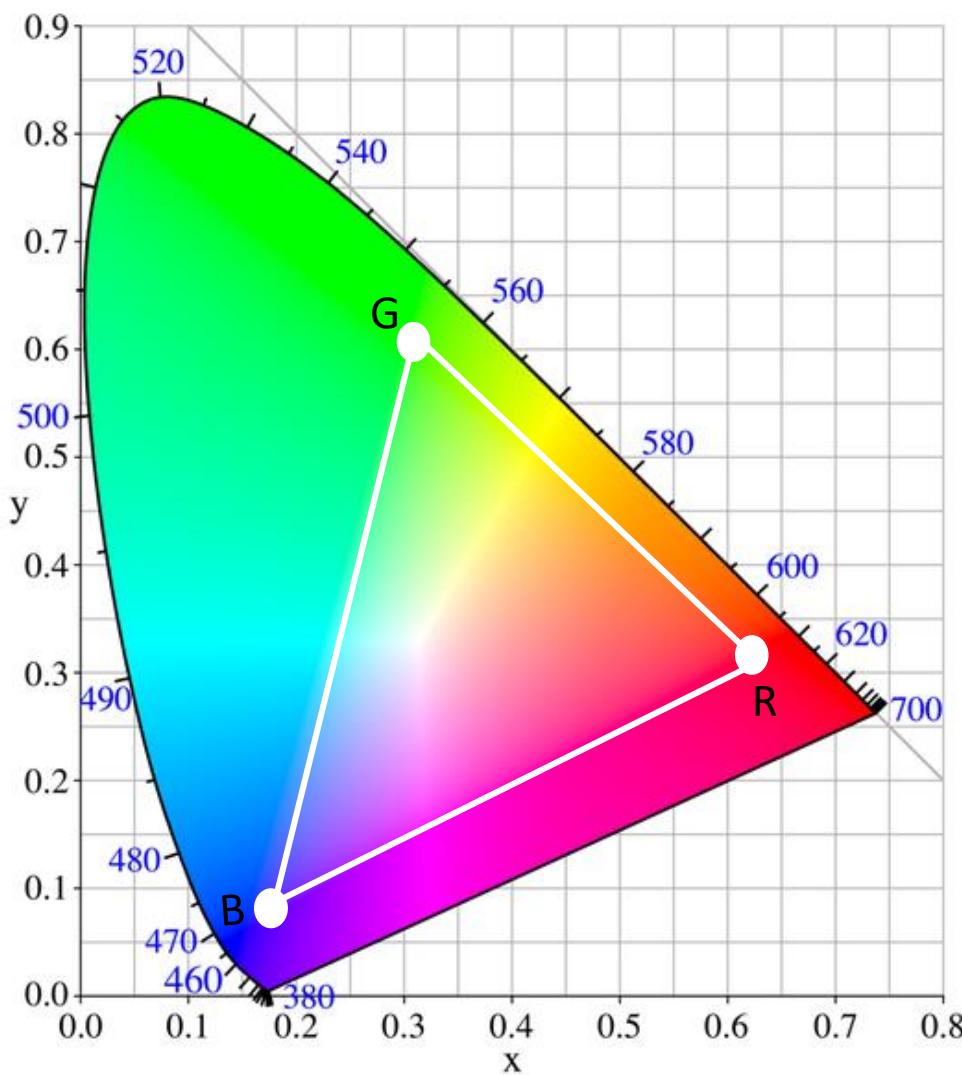


Device 1 —  
Device 2 .....  
Device 3 - -



RGB values have no meaning if the primaries between devices are not the same! This is a **huge** problem for color reproduction from one device to the next.

# Standard RGB (sRGB)



In 1996, Microsoft and HP defined a set of “standard” RGB primaries.

$$R = \text{CIE } xyY (0.64, 0.33, 0.2126)$$

$$G = \text{CIE } xyY (0.30, 0.60, 0.7153)$$

$$B = \text{CIE } xyY (0.15, 0.06, 0.0721)$$

This was considered an RGB space achievable by most devices at the time.

White point was set to the D65 illuminant. **This is an important thing to note.** It means sRGB has built in the assumed viewing condition (6500K daylight).

# CIE XYZ to sRGB conversion

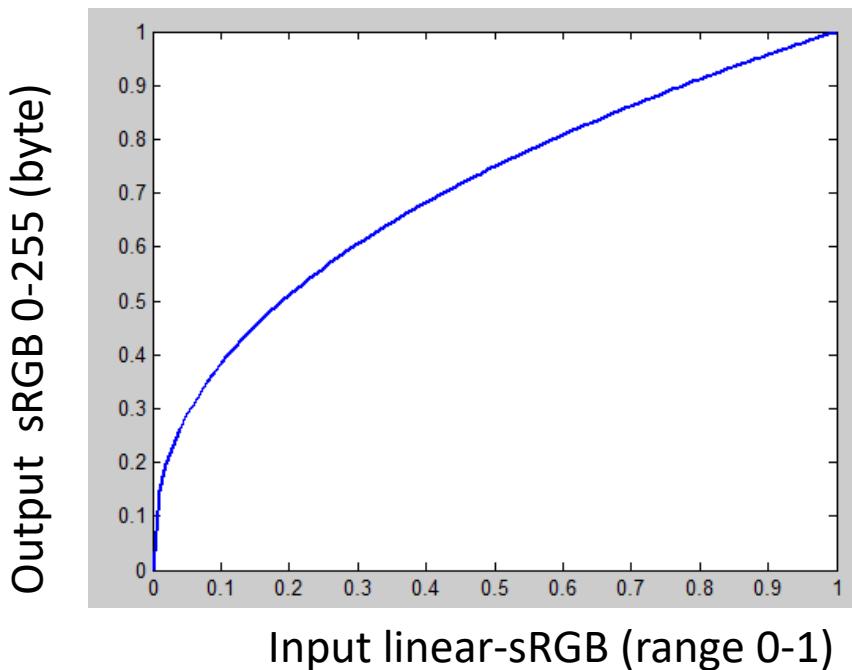
Matrix conversion:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2404542 & -1.5371385 & 0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Linearized sRGB (D65)    CIE XYZ

- R=G=B=1 is defined as illuminant D65 is in CIE XYZ\*
- This is the linear-sRGB space
- sRGB also specifies a gamma correction of the values

# sRGB gamma curve

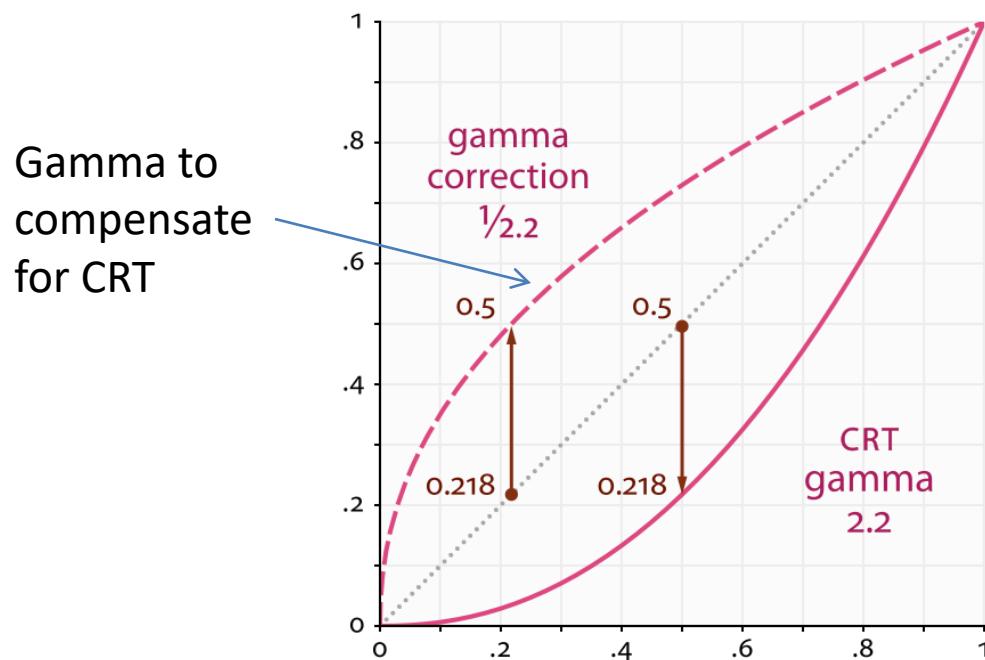


This is a close approximation of the actual sRGB gamma

Actual formula is a bit complicated, but effectively this is gamma ( $I' = 255 * I^{(1/2.2)}$ ), where  $I'$  is the output intensity and  $I$  is the linear sRGB ranged 0-1, with a small linear transfer for linearized sRGB values close to 0 (not shown in this plot).

# Gamma justification

- Gamma encoding is used in hardware to compensate for the non-linear characteristics of cathode ray tubes (CRT).
- The application of the  $\text{gamma}=1/(2.2)$  will be undone by the CRT's non-linear responsible to voltage to beam intensity to produce the desired result

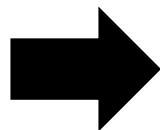
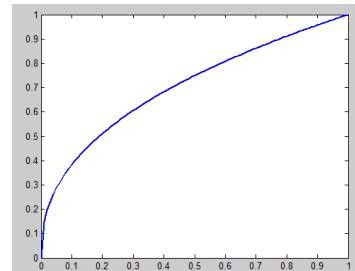


# Before (linear sRGB) & after (sRGB)



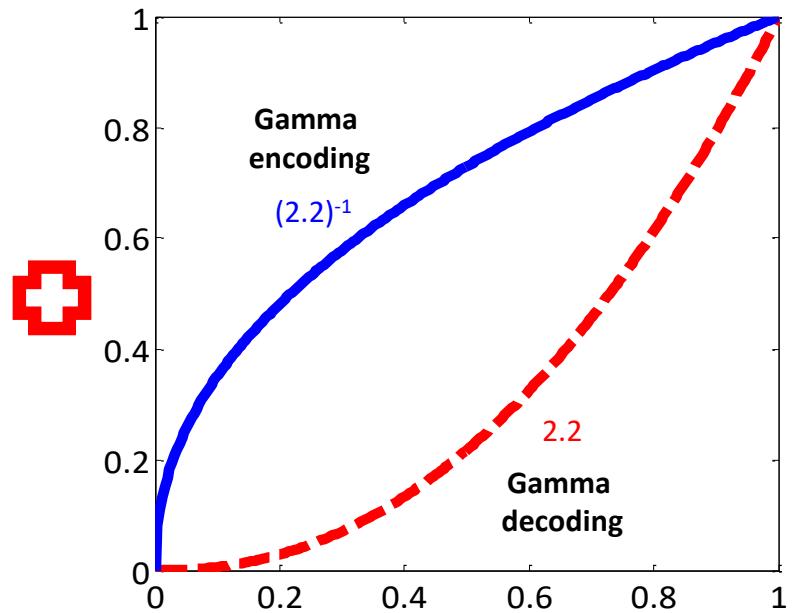
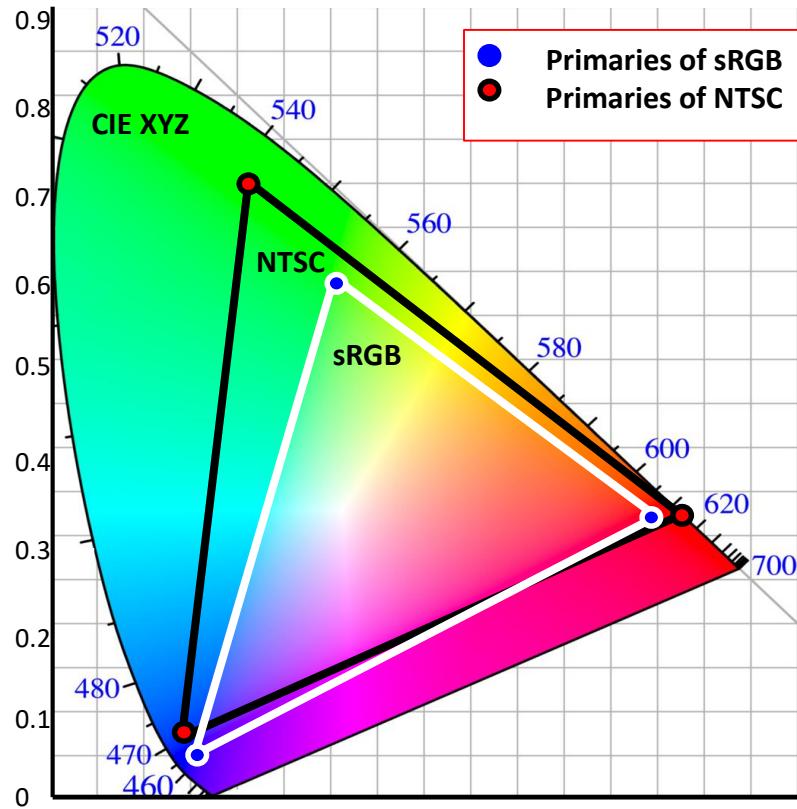
Linear sRGB

Final sRGB



Final sRGB

# Standardization isn't new - NTSC/PAL



# CIE XYZ $\leftrightarrow$ NTSC/sRGB (know your color space!)



It is important to  
know which color space  
your image is in.

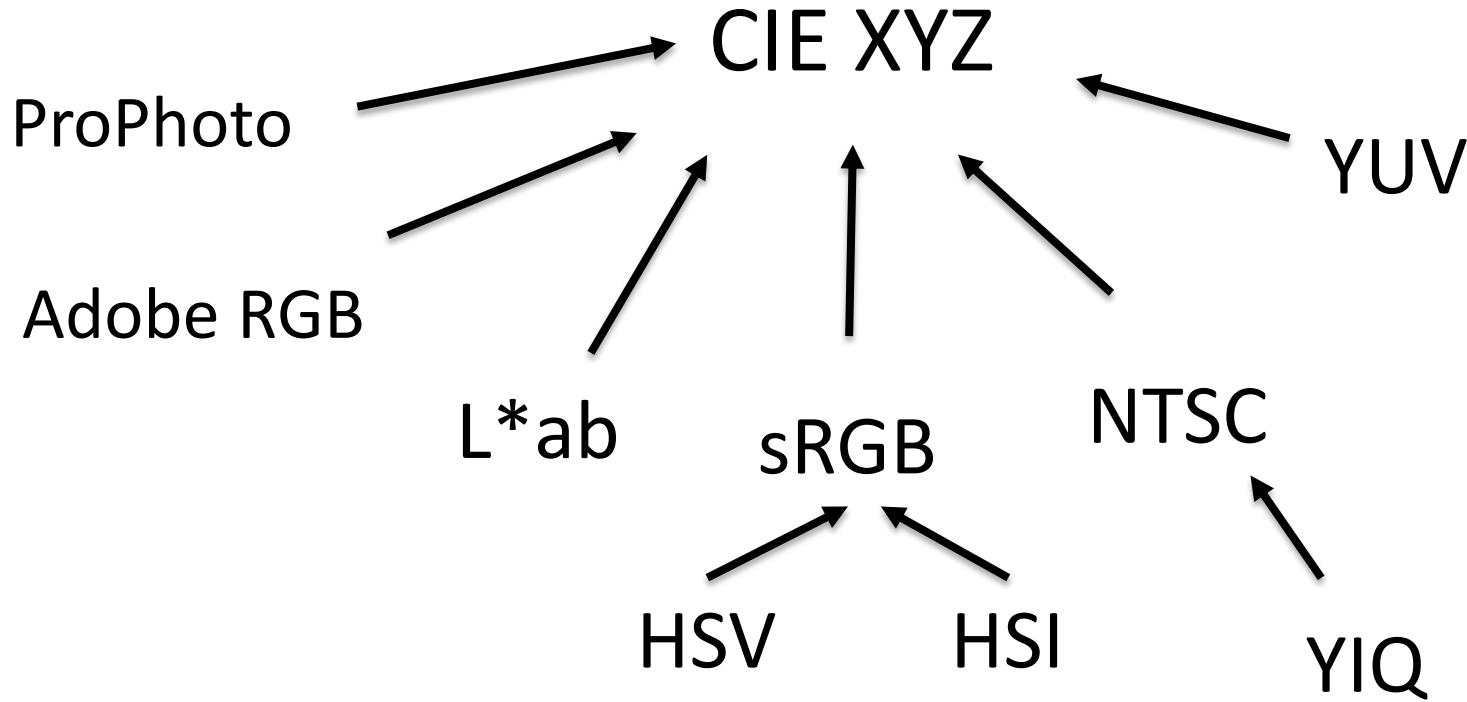
Linear-sRGB back to XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

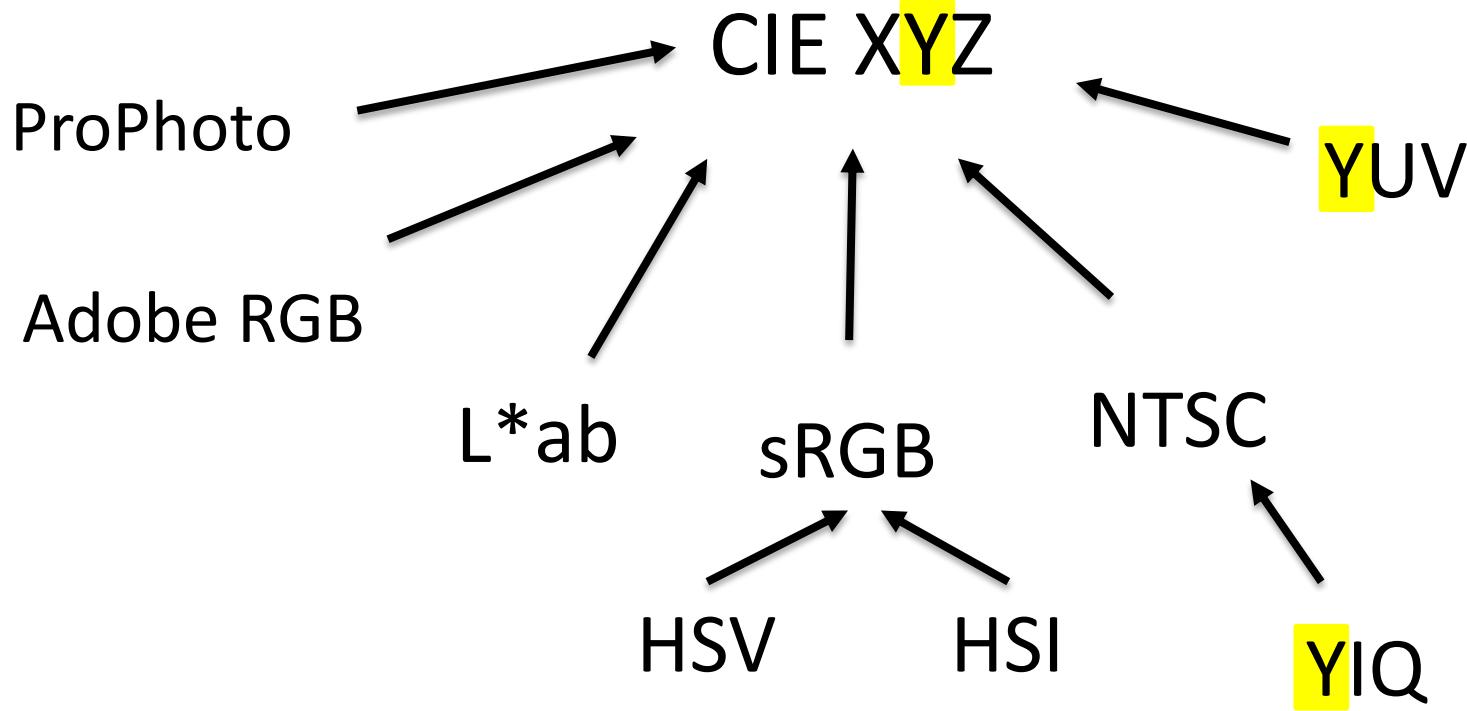
Linear-NTSC back to XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.6071 & 0.1736 & 0.1995 \\ 0.2990 & 0.5870 & 0.1140 \\ 0.0000 & 0.0661 & 1.1115 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

# CIE XYZ: The <sup>grand</sup>mother of color spaces

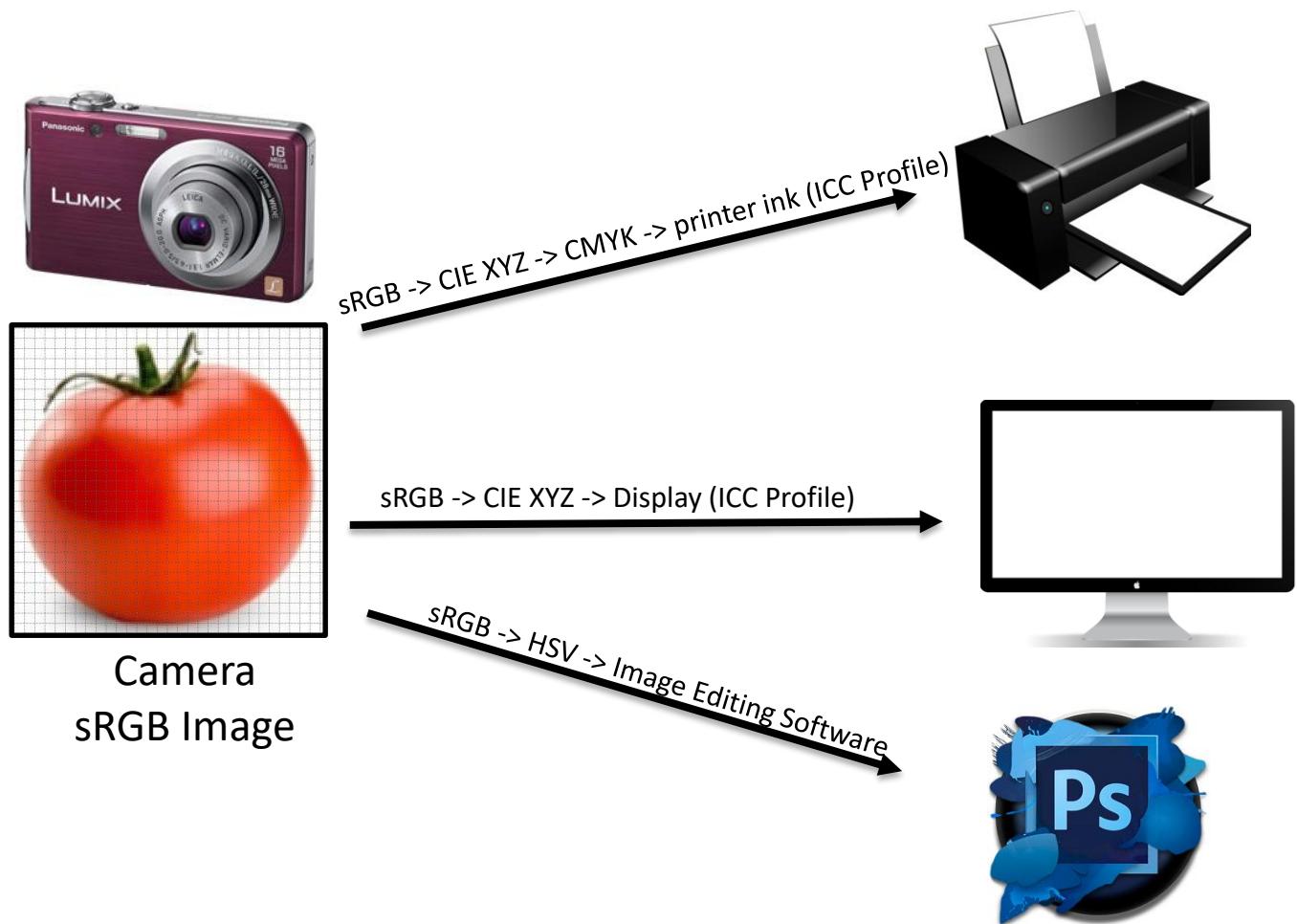


# CIE XYZ: The <sup>grand</sup>mother of color spaces



The Y is the same for these color spaces.

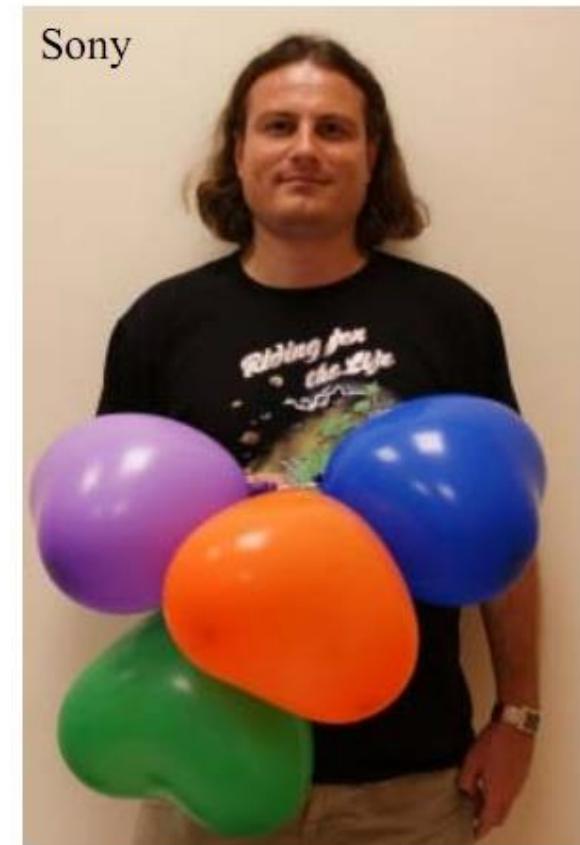
# Standard color spaces are great



# Benefits of sRGB

- Like CIE XYZ, sRGB is a device independent color space (often called an output color space)
- If you have two pixels with the same sRGB values, they will have the same CIE XYZ value, which means “in theory” they will appear as the same perceived value
- Does this happen in practice?

# See for yourself



sRGB values that are the same will be perceived as the same on the screen. However, from “real scene SPD” to final sRGB, cameras are clearly doing different things.

# Congratulations!



# Crash course on color is over!

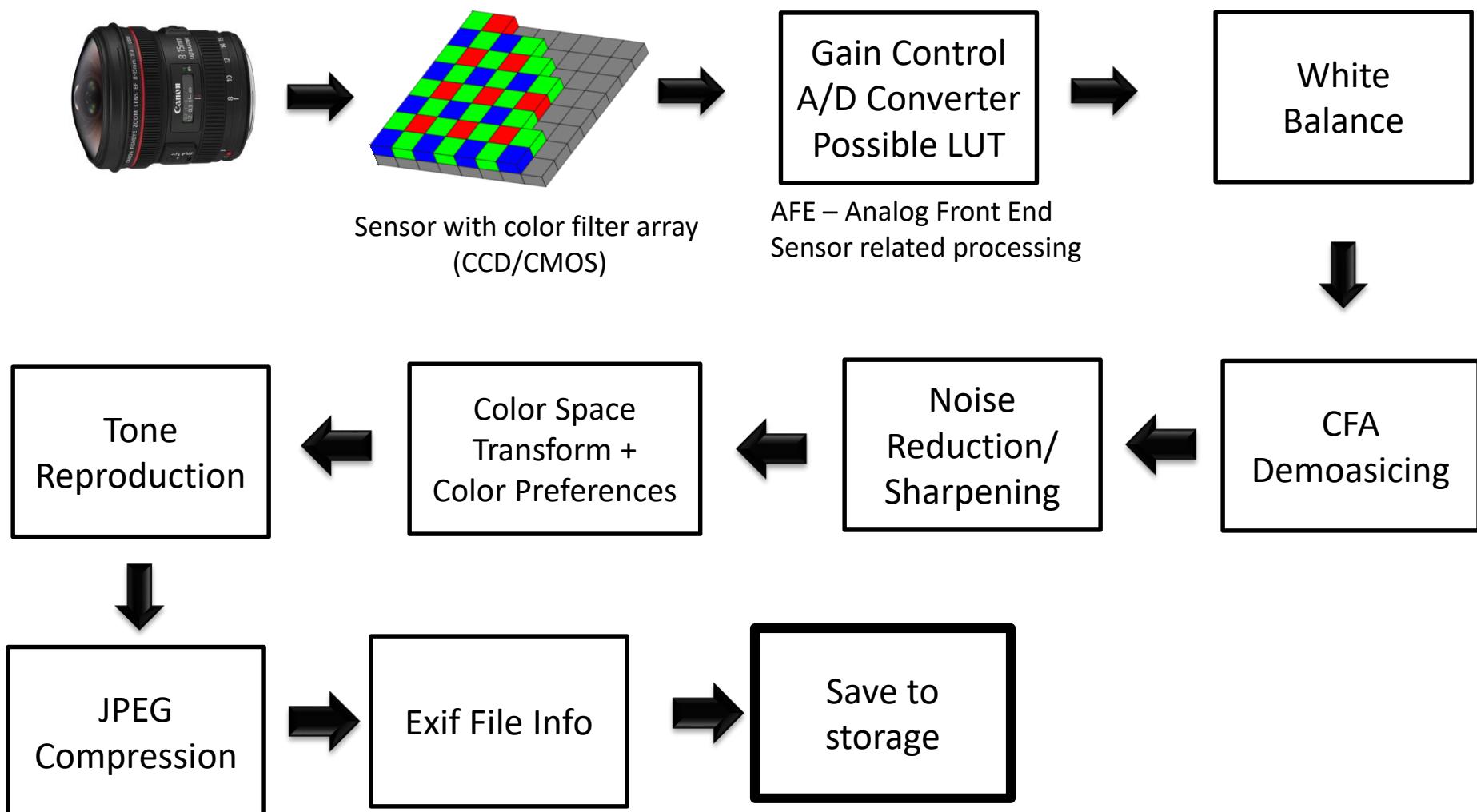
- A lot of information to absorb
- Understanding colorimetry is required to understand imaging devices
- CIE XYZ and CIE illuminants will make many appearances in color imaging/processing discussions

# Tutorial schedule

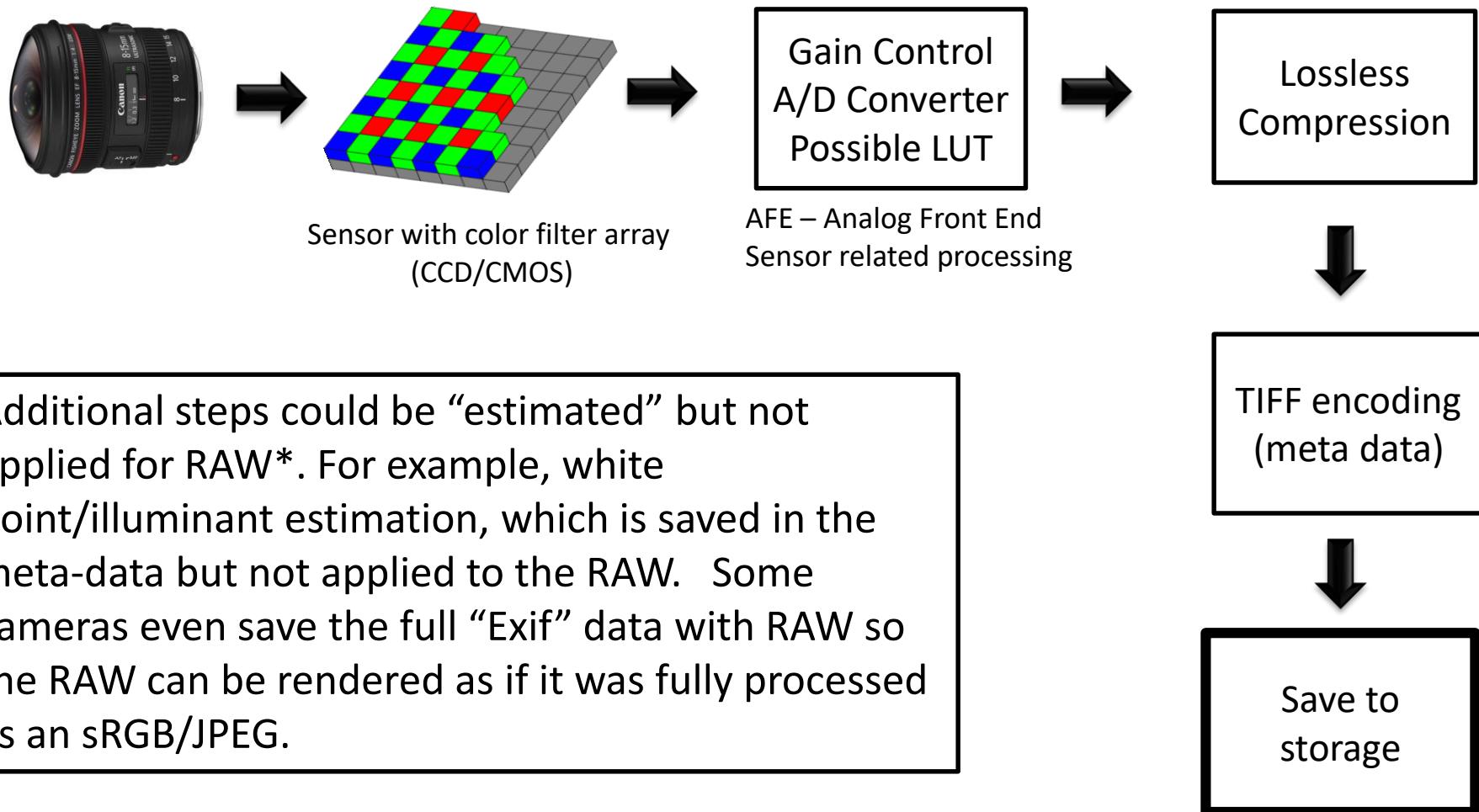
- Part 1 (General Part)
  - Motivation
  - ~~Review of color/color spaces~~
  - Overview of camera imaging pipeline
- Part 2 (Specific Part)
  - Modeling the in-camera color pipeline
  - Photo-refinishing
- Part 3 (Wrap Up)
  - The good, the bad, and the ugly of commodity cameras and computer vision research
  - Concluding Remarks

# **Overview of the Camera Imaging Pipeline**

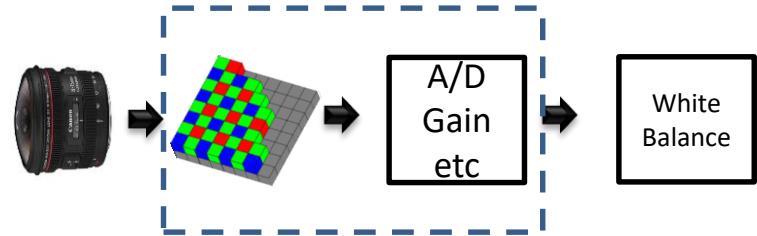
# Pipeline for sRGB (JPEG)



# Pipeline for RAW

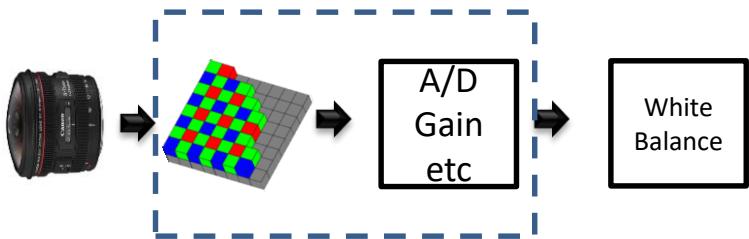


# Imaging sensor

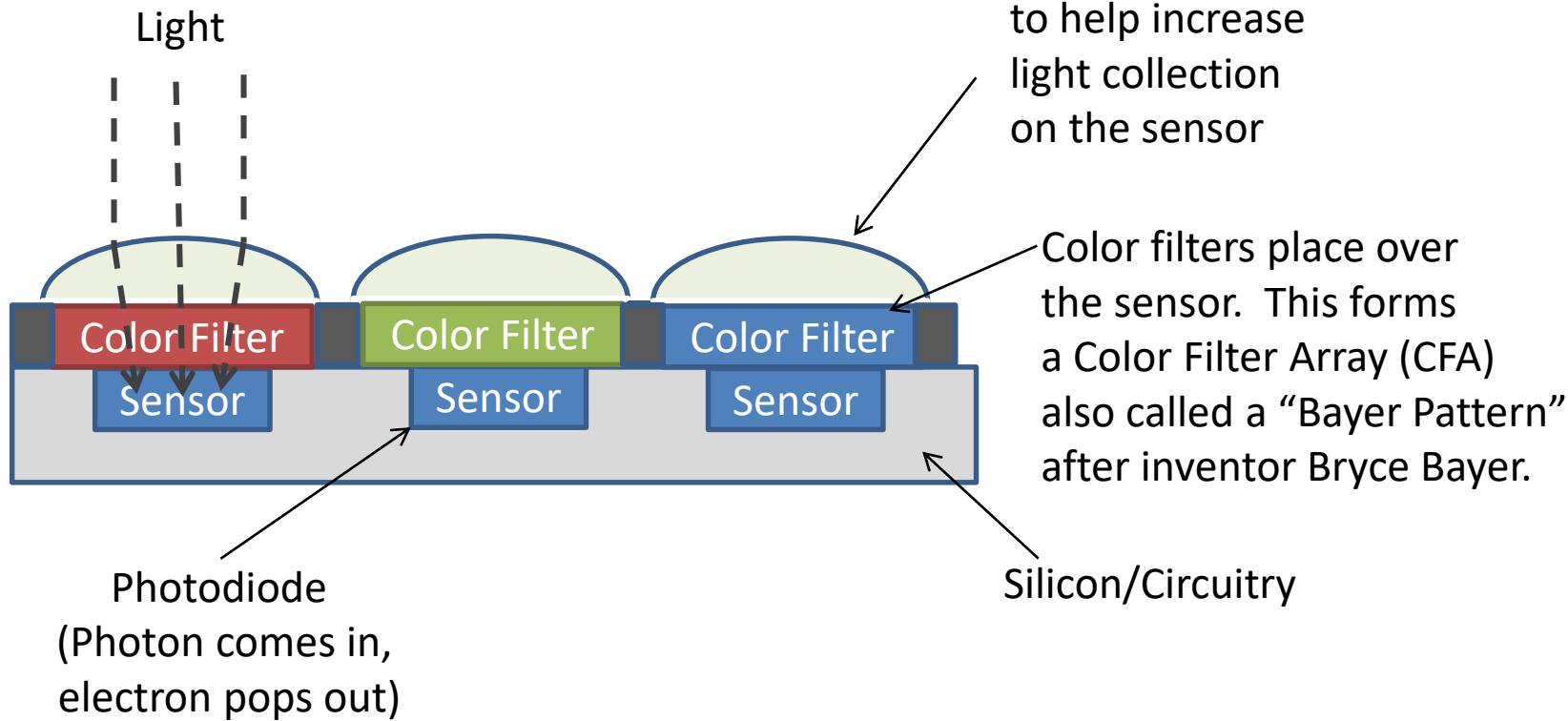


- We will discuss the basics of the imaging sensor at a high level
  - Could have a whole tutorial just on sensors
- The assumption is after this step, the digital values coming off the sensor are linear with respect to the amount of falling on the pixel over a given amount of time
- Will discuss some common steps to make this happen

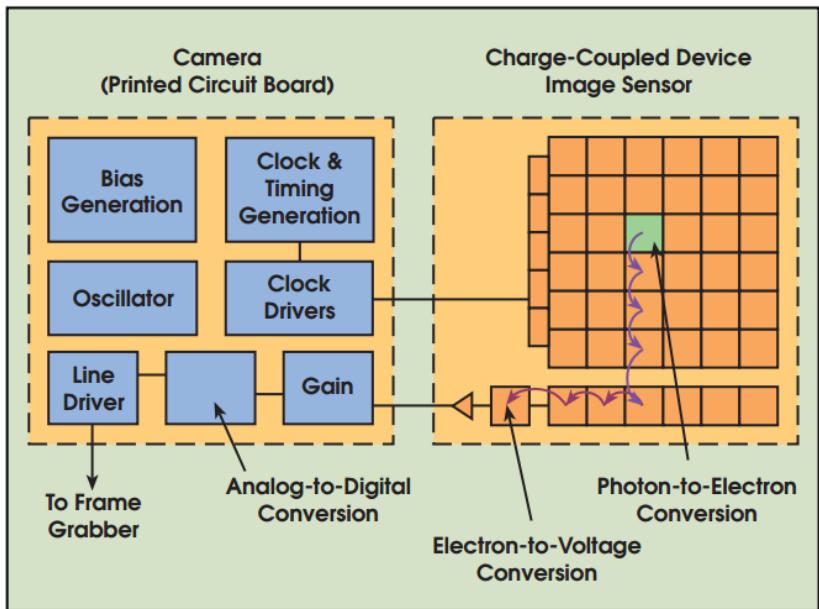
# Imaging sensor



Basic imaging sensor design

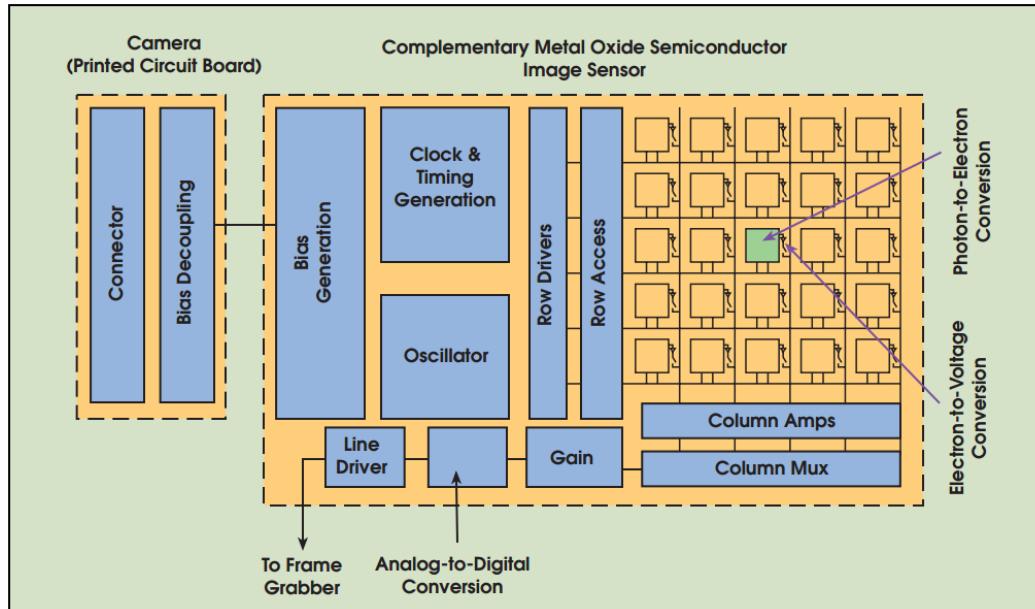


# CCD & CMOS



## CCD

CCD (**c**harge **c**oupled **d**evice) has a different readout technology to convert charge to voltage and buffer it for output. The plus side is there is more space on the pixel for the photo sensor.

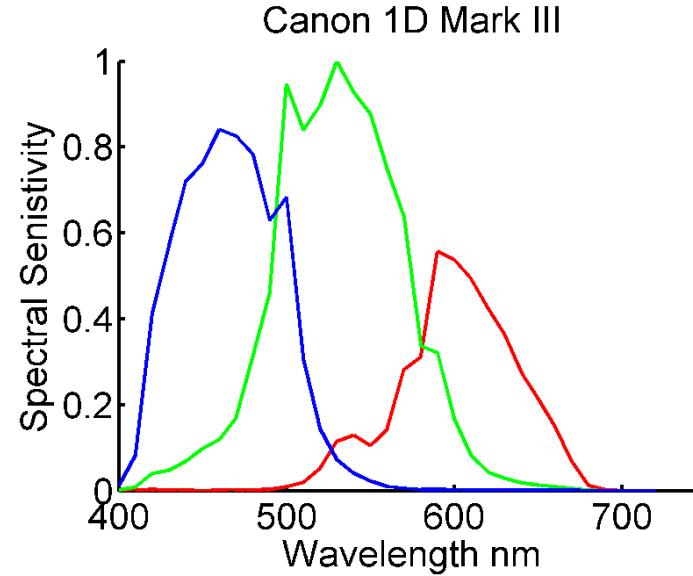
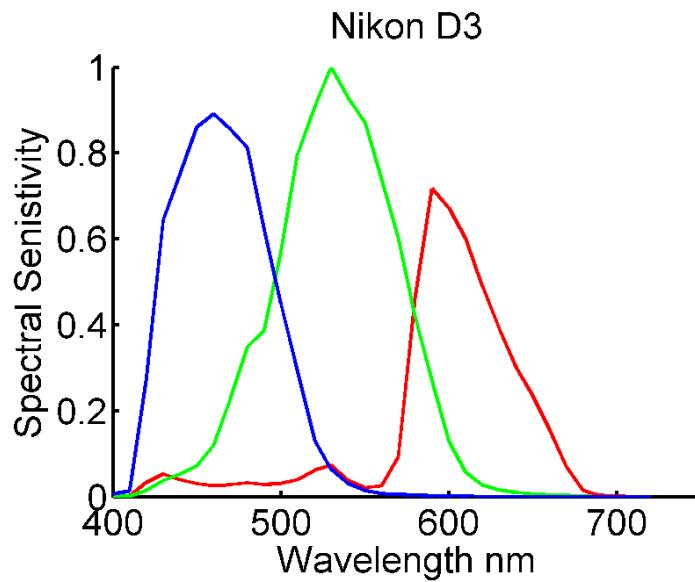


## CMOS

CMOS (**c**omplementary **m**etal **o**xide **s**emiconductor) converts charge to voltage at the pixel site. This allows faster readouts, but less space for the photo sensor per pixel.

# Camera RGB sensitivity

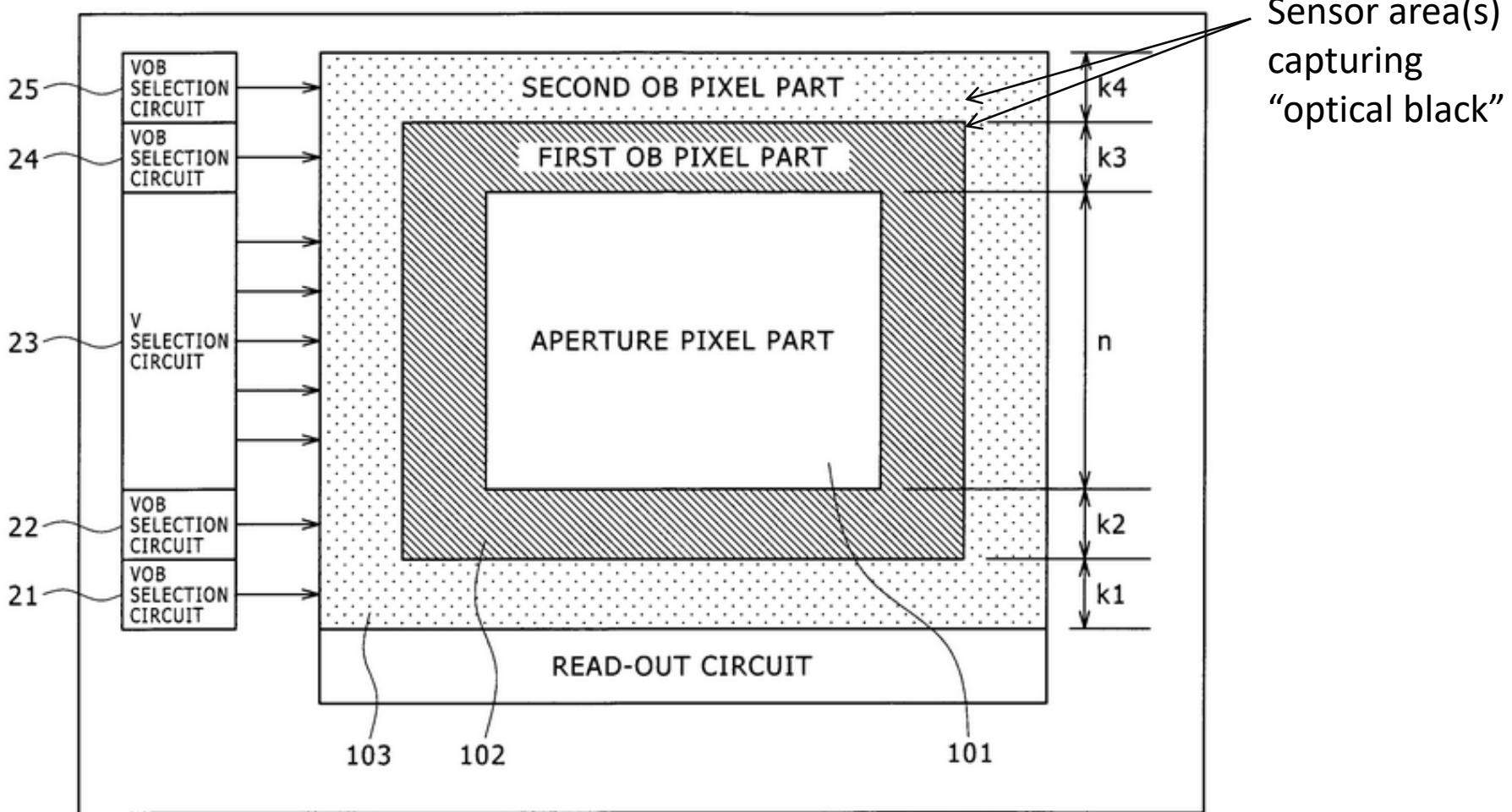
- The color filter array (CFA) on the camera filters the light into three primaries



# Black light subtraction

- Sensor values for pixels with “no light” should be zero
- But, often, this is not the case for various reasons
  - Cross talk on the sensor, etc. .
  - This can also change with sensor temperature
- This can be corrected by capturing a set of pixels that do not see light
- Place a dark-shield around sensor
- Subtract the level from the “black” pixels

# Optical black (OB)



Black light capturing areas (likely exaggerated) from Sony US Patent US8227734B2 (Filed 2008).

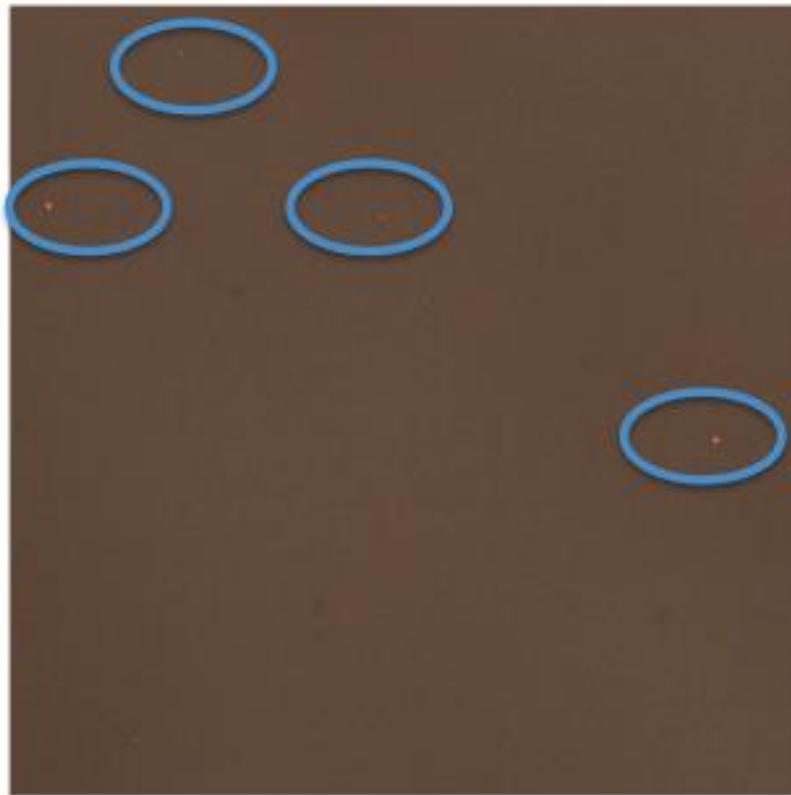
# Signal amplification (gain)

- Imaging sensor signal is amplified
- Amplification to assist A/D
  - Need to get the voltage to the range required to the desired digital output
- This gain could also be used to accommodate camera ISO settings
  - Unclear if all cameras do this here or use a simple post-processing gain to the RAW for ISO settings
  - DSLR cameras RAW is modified by the ISO setting

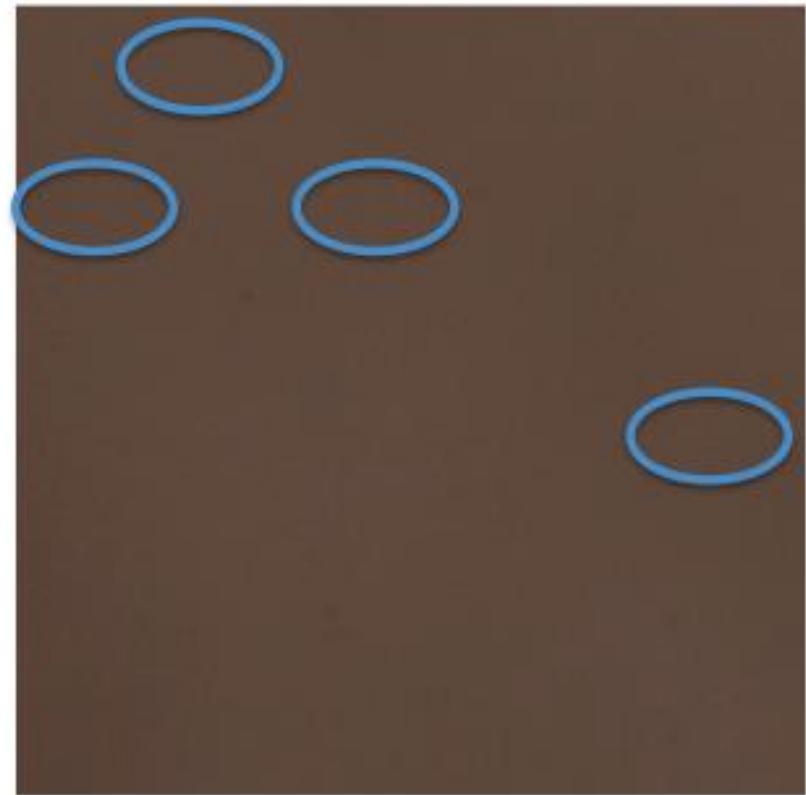
# Defective pixel mask

- CCD/CMOS have pixels that are defective
- Dead pixel masks are pre-calibrated at the factory
  - Using “dark current” calibration
  - Take an image with no light
  - Record locations reporting values to make “mask”
- Bad pixels in the mask are interpolated
- This process seems to happen before RAW is saved
  - If you see dead pixels in RAW, these are generally new dead pixels that have appeared after leaving the factory

# Example



Identifying “dead pixels”



After interpolation

# Nonlinear response correction

- Some image sensor (generally CMOS) often have a non-linear response to different amounts of irradiance
- A non-linear adjustment or look up table (LUT) interpolation can be used to correct this

# Other possible distortion correction

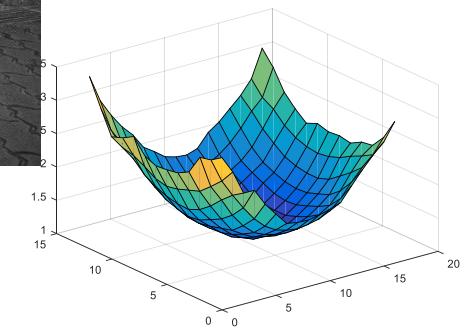
- Sensor readout could have spatial distortion for various reasons, e.g. sensor cross-talk
- For point-and-shoot/mobile cameras with fixed lens, vignetting correction for lens distortion could be applied
- Such corrections can be applied using a LUT or polynomial function (in the case of vignetting)

# Ex: Flat field correction

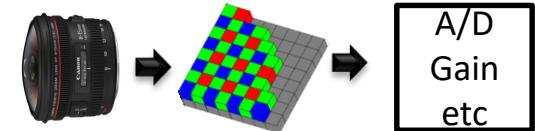


Black level  
subtraction and  
linearization<sup>1</sup>

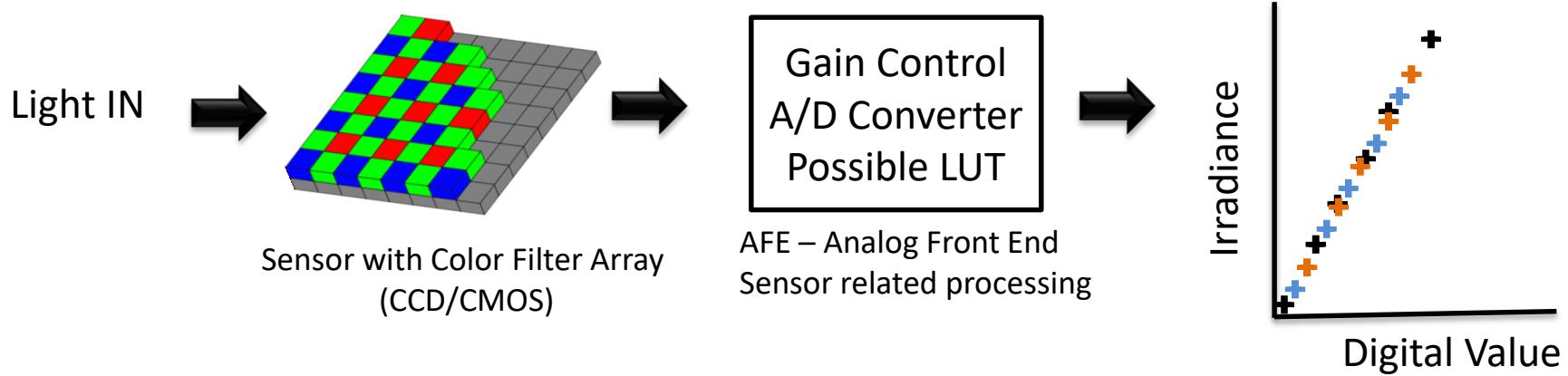
# Ex: Flat field correction (non-uniform gain)



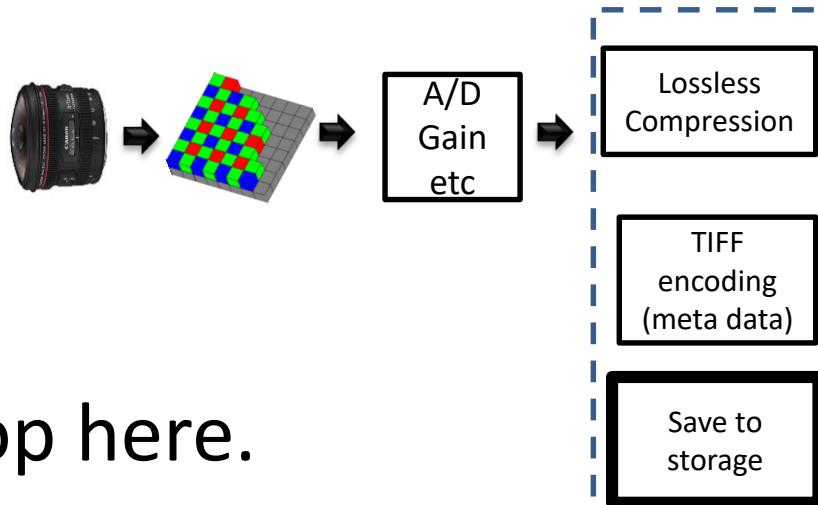
# At this stage



- We now have a reading from the sensor that is linear with respect to light coming in
- Defective pixels have been interpolated
- Potential distortion has been reduced



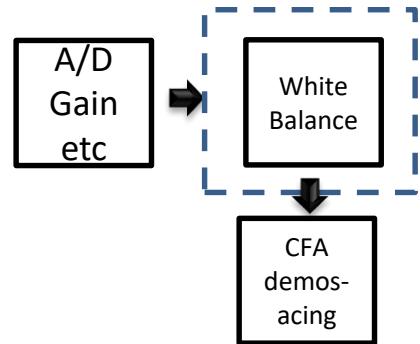
# RAW – stop here



- If saving in RAW, we can stop here.
- Convert to TIFF + metadata, save to media
- RAW generally represents gained + linearized sensor response before CFA demosaicing and white-balance correction
- We like to think of RAW as “unprocessed” sensor response

***Important: RAW image color space will be camera-specific***

# White balance



- White balance is intended to mimic chromatic adaptation of the eye
- Users can manually set the white balance
  - Camera specific white-balance matrices are used selected illuminant (see next slides)
  - This is often stored in the Exif metadata
- Otherwise auto white balance (AWB) is performed

# WB manual settings

WB SETTINGS	COLOR TEMPERATURE	LIGHT SOURCES
	10000 - 15000 K	Clear Blue Sky
	6500 - 8000 K	Cloudy Sky / Shade
	6000 - 7000 K	Noon Sunlight
	5500 - 6500 K	Average Daylight
	5000 - 5500 K	Electronic Flash
	4000 - 5000 K	Fluorescent Light
	3000 - 4000 K	Early AM / Late PM
	2500 - 3000 K	Domestic Lightning
	1000 - 2000 K	Candle Flame

Typical mapping of WB icons to related color temperature.  
White-balance matrix is often stored in the exif file.

# Examples of manual WB matrices

Sunny

$$\begin{bmatrix} 2.0273 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.3906 \end{bmatrix}$$

## Nikon D7000

Incandescent

$$\begin{bmatrix} 1.3047 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 2.2148 \end{bmatrix}$$

Shade

$$\begin{bmatrix} 2.4922 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.1367 \end{bmatrix}$$

Daylight

$$\begin{bmatrix} 2.0938 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.5020 \end{bmatrix}$$

## Canon 1D

Tungsten

Shade

$$\begin{bmatrix} 2.4628 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.2275 \end{bmatrix}$$

Daylight

$$\begin{bmatrix} 2.6836 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.5586 \end{bmatrix}$$

## Sony A57K

Tungsten

$$\begin{bmatrix} 1.6523 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 2.7422 \end{bmatrix}$$

Shade

$$\begin{bmatrix} 3.1953 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.2891 \end{bmatrix}$$

# Auto white balance (AWB)

- If manual white balance is not used, then an AWB algorithm is performed
- This is not entirely the same as chromatic adaptation, because it doesn't have a target illuminant, instead AWB (as the name implies) attempts to make what is assumed to be white map to “pure white”
- Next slides introduce two well known methods: “Gray World” and “White Patch”

# AWB: Gray world algorithm

- This method assumes that average reflectance of a scene is achromatic (i.e. gray)
  - Gray is just the white point not at its brightest, so it serves as an estimate of the illuminant
  - This means that image average should have equal energy, i.e.  $R=G=B$
- Based on this assumption, the algorithm adjusts the input average to be gray as follows:

First, estimate the average response:

$$R_{avg} = \frac{1}{Nr} \sum R_{sensor}(r) \quad G_{avg} = \frac{1}{Ng} \sum G_{sensor}(g) \quad B_{avg} = \frac{1}{Nb} \sum B_{sensor}(b)$$

$r$  = red pixels values,  $g$ =green pixels values,  $b$  =blue pixels values

$Nr$  = # of red pixels,  $Ng$  = # of green pixels,  $Nb$  = # blue pixels

Note: # of pixel per channel may be different if white balance is applied to the RAW image before demosaicing. Some pipelines may also transform into another colorspace, e.g. LMS, to perform the white-balance procedure.

# AWB: Gray world algorithm

- Based on averages, white balance can be expressed as a matrix as:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} G_{avg}/R_{avg} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & G_{avg}/B_{avg} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

White-balanced sensor RGB    Sensor RGB

Matrix scales each channel by its average and then normalizes to the green channel average.

Note: some (perhaps most) pipelines may also transform into another colorspace, e.g. LMS, to perform the white-balance procedure.

# AWB: White patch algorithm

- This method assumes that highlights represent specular reflections of the illuminant
  - This means that maximum R, G, B values are a good estimate of the white point
- Based on this assumption, the algorithm works as follows:

$$R_{max} = \max(R_{sensor}(r)) \quad G_{max} = \max(G_{sensor}(g)) \quad B_{max} = \max(B_{sensor}(b))$$

r = red pixels values, g=green pixels values, b =blue pixels values

# AWB: White patch algorithm

- Based on RGB max, white balance can be expressed as a matrix as:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} G_{max}/R_{max} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & G_{max}/B_{max} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

White-balanced sensor RGB    Sensor RGB

Matrix scales each channel by its maximum value and then normalizes to the green channel's maximum.

# AWB example



Input



Gray World

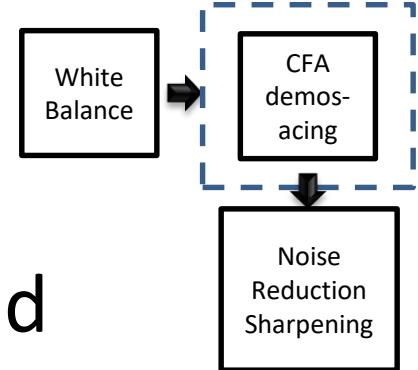


White Patch

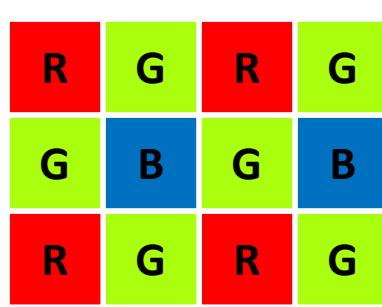
# Better AWB methods

- Gray world and white patch are *very basic* algorithms
  - These both tend to fail when the image is dominated by large regions of a single color (e.g. a sky image)
- **There are many improved versions**
- Most improvements focus on how to perform these white point estimation more robustly
- **Note:** AWB matrix values are often stored in an images Exif file

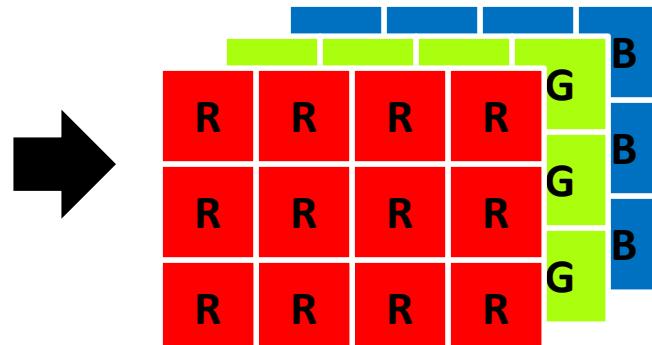
# CFA demosaicing



- Color filter array/Bayer pattern placed over pixel sensors
- We want an RGB value at each pixel, so we need to perform interpolation



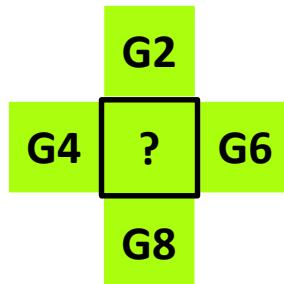
Sensor RGB layout



Desired output with RGB per pixel.

# Simple interpolation

B1	G2	B3
G4	R5	G6
B7	G8	B9

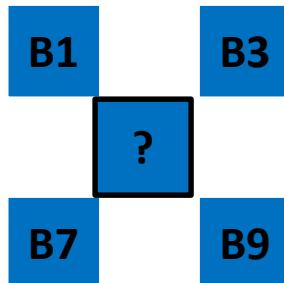


R5

G5 ?

B5 ?

Simply interpolate  
based on neighbor  
values.



$$G5 = \frac{G2 + G4 + G6 + G8}{4}$$

$$B5 = \frac{B1 + B3 + B7 + B9}{4}$$

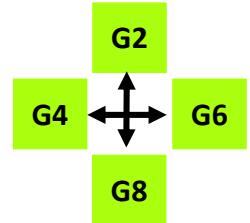
# Simple “edge aware” interpolation



If ( $|G2 - G8| \&\& |(G4 - G8)|$  both < Thres)

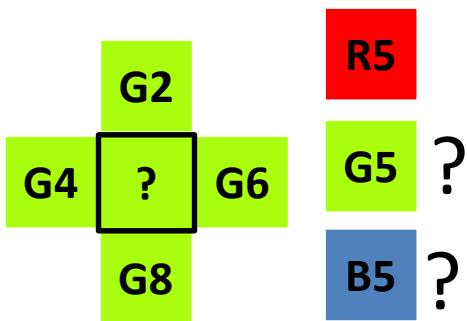
$$G5 = \frac{G2 + G4 + G6 + G8}{4}$$

Case 1



All about the same.

elseif ( $|G2 - G8| > Thres$ )

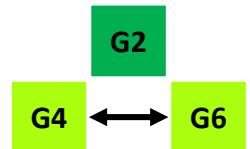


$$G5 = \frac{G4 + G6}{2}$$

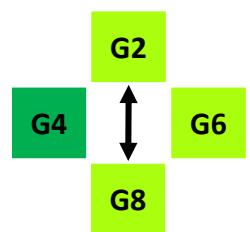
else

$$G5 = \frac{G2 + G8}{2}$$

Case 2



G2 and G8 differ

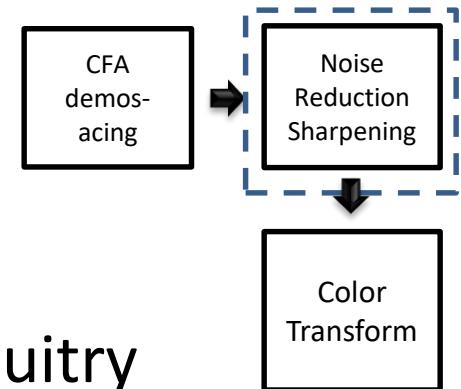


G4 and G6 differ  
127

# Demosaicing

- These examples are simple algorithms
- Cameras almost certainly use more complex and proprietary algorithms
- Demosaicing can be combined with additional processing
  - Highlight clipping
  - Sharpening
  - Noise reduction

# Noise Reduction (NR)



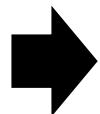
- All sensors inherently have noise
- Some of this is mitigated in the sensor circuitry (discussed previously)
- Some cameras apply NR after A/D conversion
- A couple simple methods are presented here
- For high-end cameras, it is likely that cameras apply slightly different strategies depending on ISO settings, e.g. high ISO will result in more noise, so a more aggressive NR could be used
- Examples given are more likely on lower-end point-and-shoot cameras

# NR – Rank order statistics

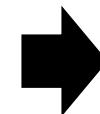
- Sliding window **median** filter
- Sort all pixels in a 3x3 (or larger) widow about center pixel by value
- Select median (i.e. pixel #5 in rank)

10	11	10
9	5	10
10	8	9

input



[5, 8, 9, 9, 10, 10, 10, 10, 11]  
rank #5



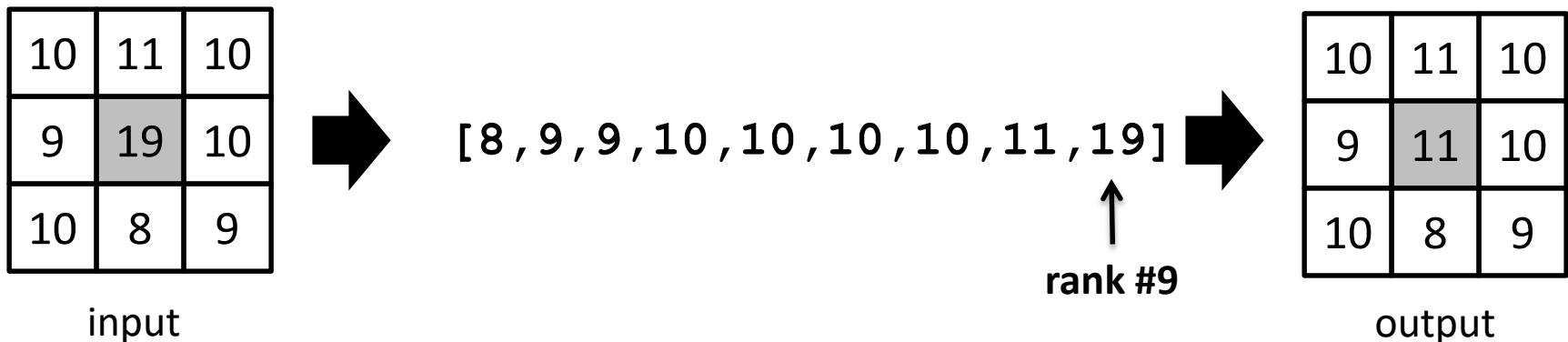
10	11	10
9	10	10
10	8	9

output

Median filter is nice because it preserves edges.

# NR – Rank order statistics

- Sliding window **despeckle** filter
- Sort all pixels in a 3x3 (or larger) widow about center pixel by value
- If center pixel maps to extreme (rank 1 or 9) and is significantly different than closest neighbor, take neighboring ranked pixel value

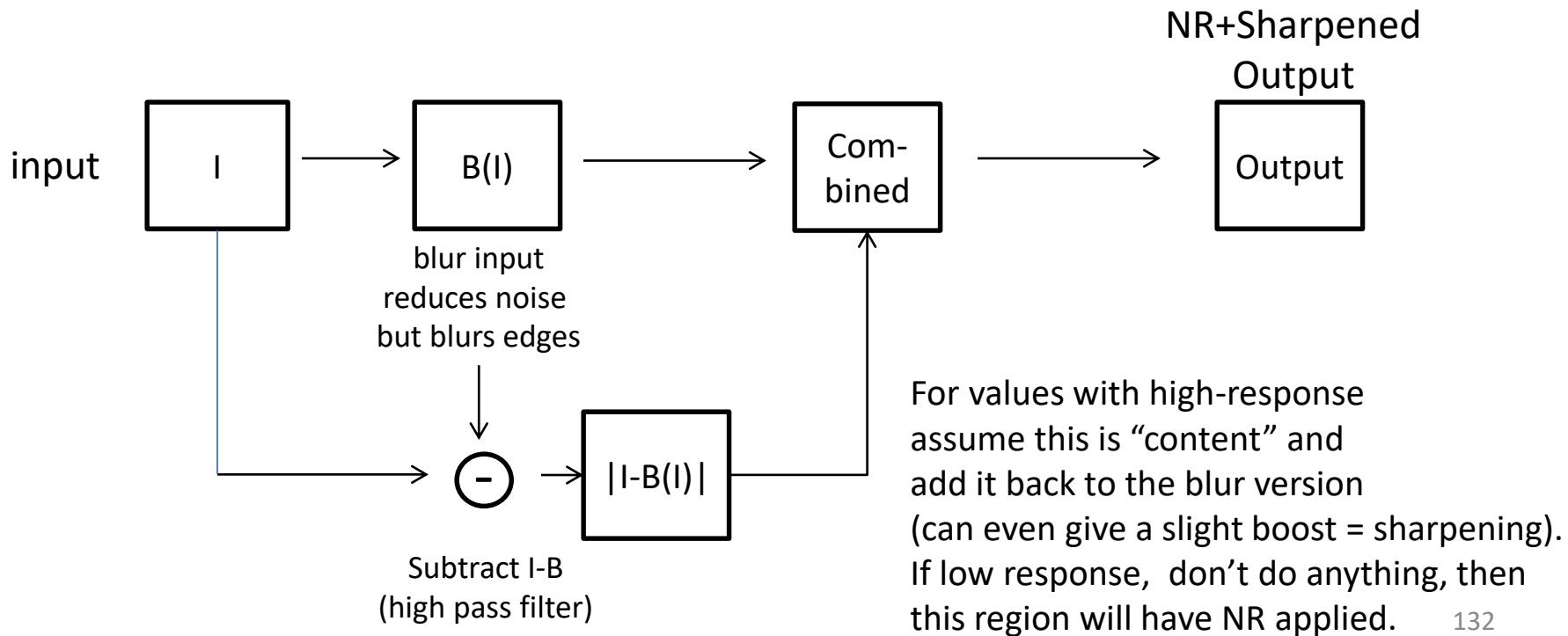


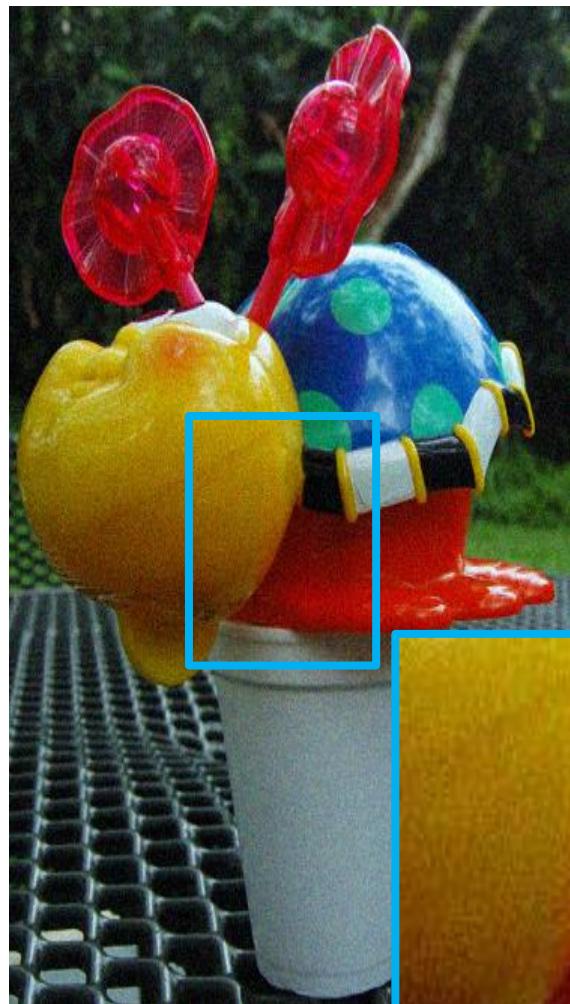
Center pixel mapped to an extreme.  
 $| \text{Rank 9 (19)} - \text{Rank 8 (11)} | > \text{threshold}$   
Replace with Rank 8

# NR + Sharpening

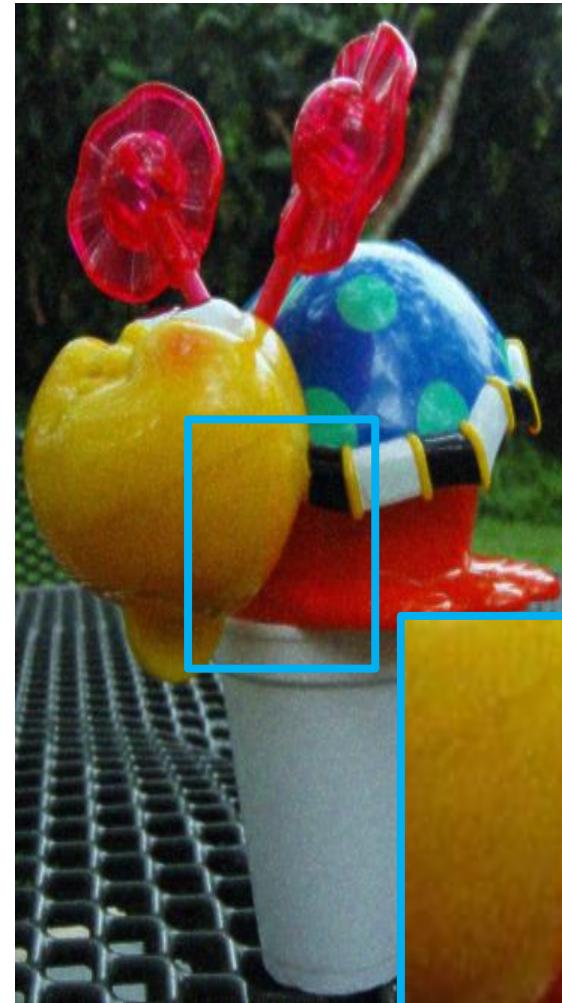
- Another strategy is to apply a blur and add the detail back in “content regions”, or even boost content to perform sharpening

Sketch of the procedure here:



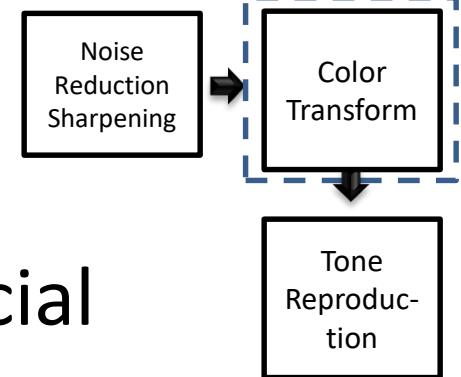


Input



NR

# Color space manipulation

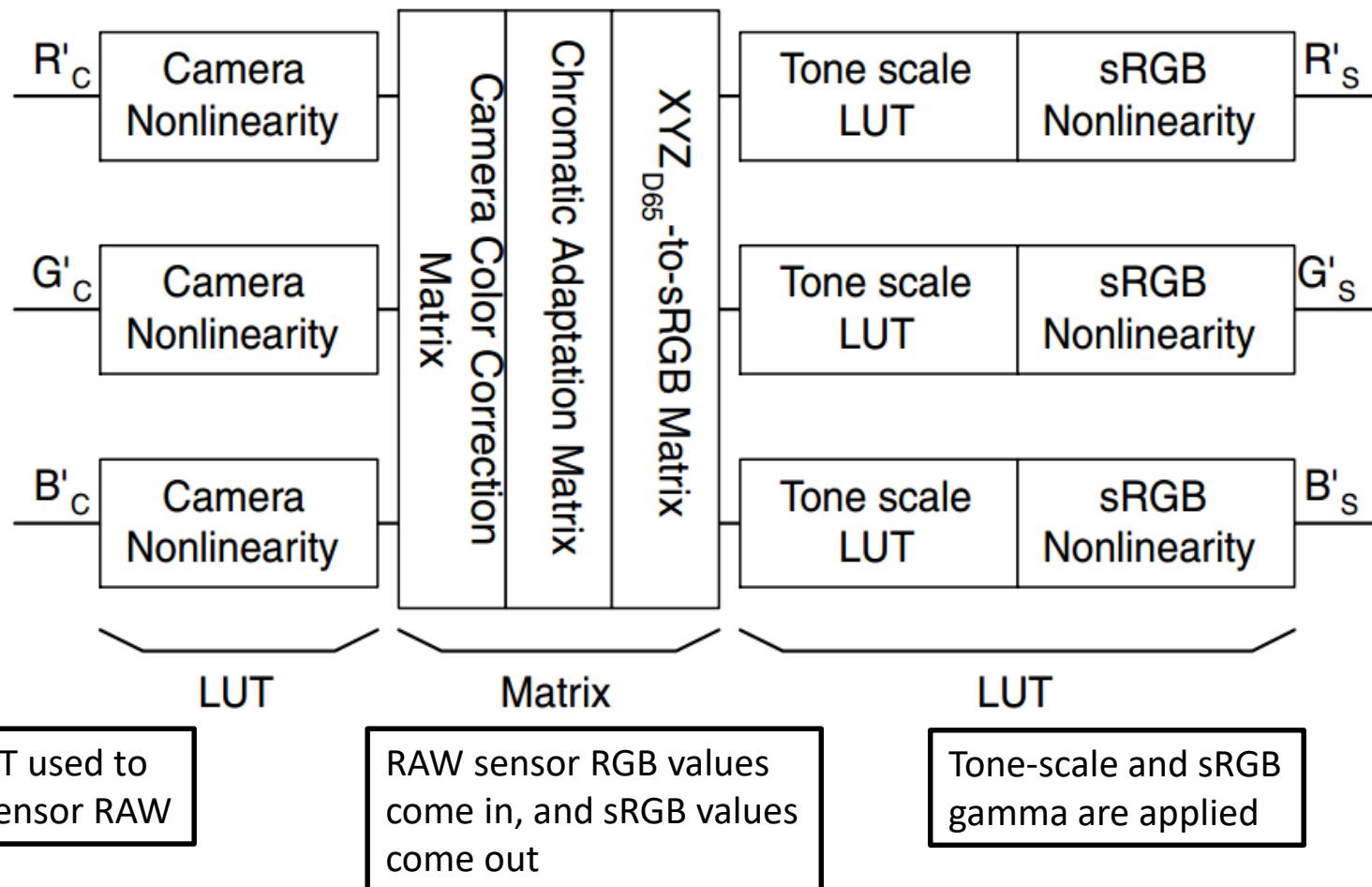


- We are now at one of the most crucial steps in the pipeline
- Up until this point, the “RGB” values have been related to the camera’s RGB sensitivity, i.e. sensor RGB color space
- We now need to map these values to the output color space (sRGB)

# Color space transform

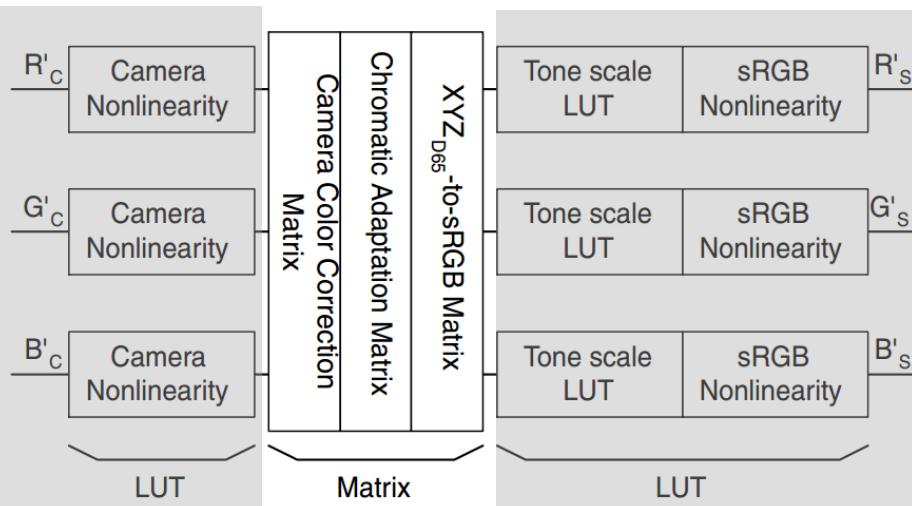
- Color Correction Matrix (CCM)
  - Transforms sensor native RGB values into some canonical colorspace (e.g. CIE XYZ) that will eventual be transform to the final sRGB colorspace
- It is important that the white-balance has been performed correctly

# Color transform conceptually

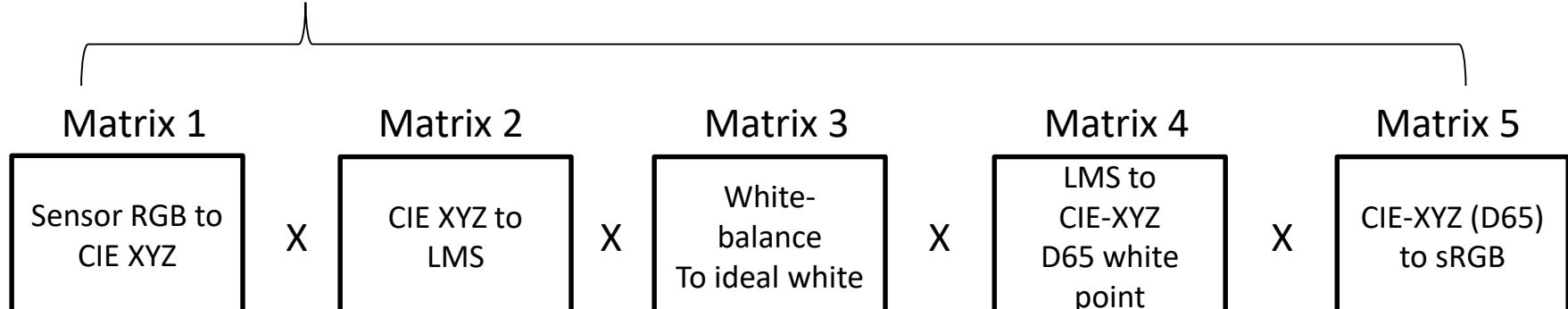


Top figure from: Parulski & Spaulding, Chapter 12 “Color image processing for digital camera”,  
Digital Color Imaging Handbook, 2003

# Matrix decompositions



Focusing on the matrix part only. Conceptually we could breakdown the color transform in to a series of transforms. It may desirable to perform AWB in a different colorspace, e.g. LMS, then to sRGB (using CIE XYZ as an intermediate colorspace to try things together)



First, convert sensor RGB to CIE (1931) XYZ

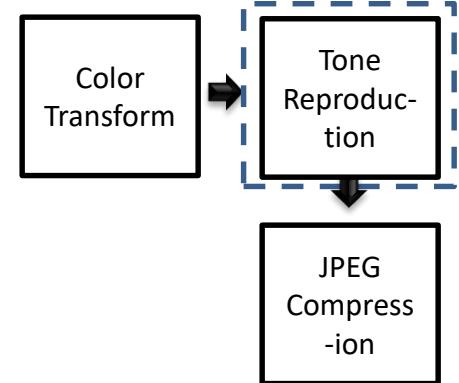
CIE XYZ to LMS

LMS to CIE-XYZ (w/ optional white point)

CIE-XYZ D65 to linearized sRGB

This is from Parulski & Spaulding, however, it is more common to perform white-balance first – more on this later.

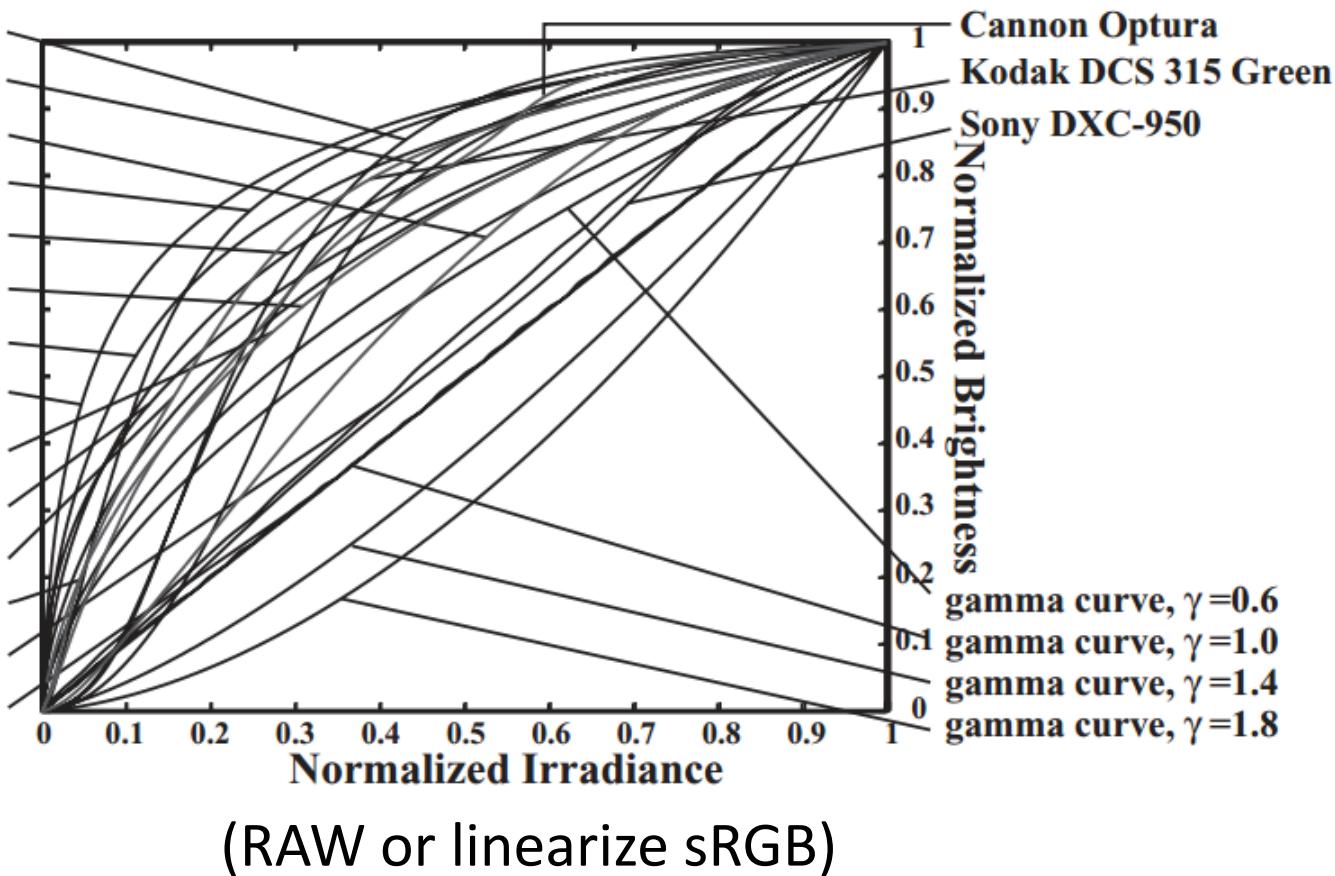
# Tone-mapping



- Non-linear mapping of RGB tones
- Applied to achieve some preferred tone-reproduction
  - This is not sRGB gamma
  - This is to make the images look nice
- To some degree this mimics the nonlinearity in film (known as the Hurter-Driffield Curves)
- Each camera has its own unique tone-mapping (possibly multiple ones)

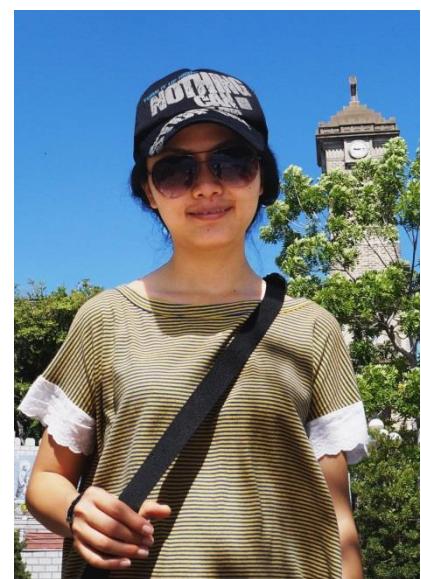
# Examples

Kodak Ektachrome-100plus Green  
Kodak Ektachrome-64 Green  
Agfachrome CTPrecisa100 Green  
Agfachrome RSX2 050 Blue  
Agfacolor Futura 100 Green  
Agfacolor HDC 100 plus Green  
Agfacolor Ultra 050 plus Green  
Agfapan APX 025  
Agfa Scala 200x  
Fuji F400 Green  
Fuji F125 Green  
Kodak Max Zoom 800 Green  
Kodak KAI0372 CCD  
Kodak KAF2001 CCD

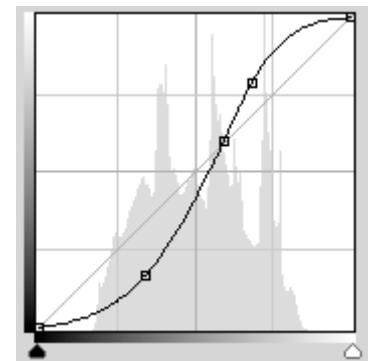
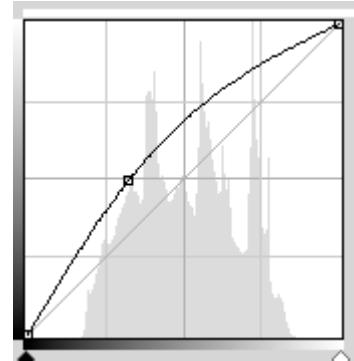
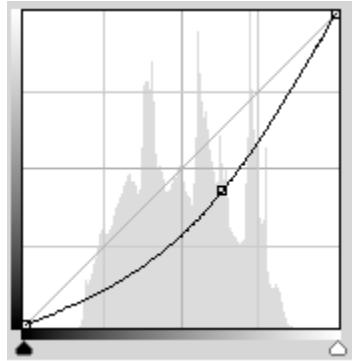


Here are several examples of tone-mappings for a white range of cameras.  
The x-axis is the input “RAW” (or linearized-sRGB), the y-axis is the output  
After tone mapping.

# Tone-mapped examples



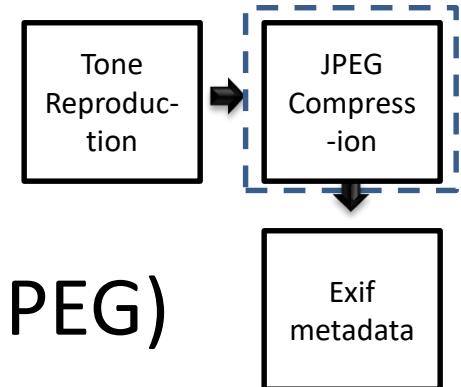
Input



# Note about tone-mapping

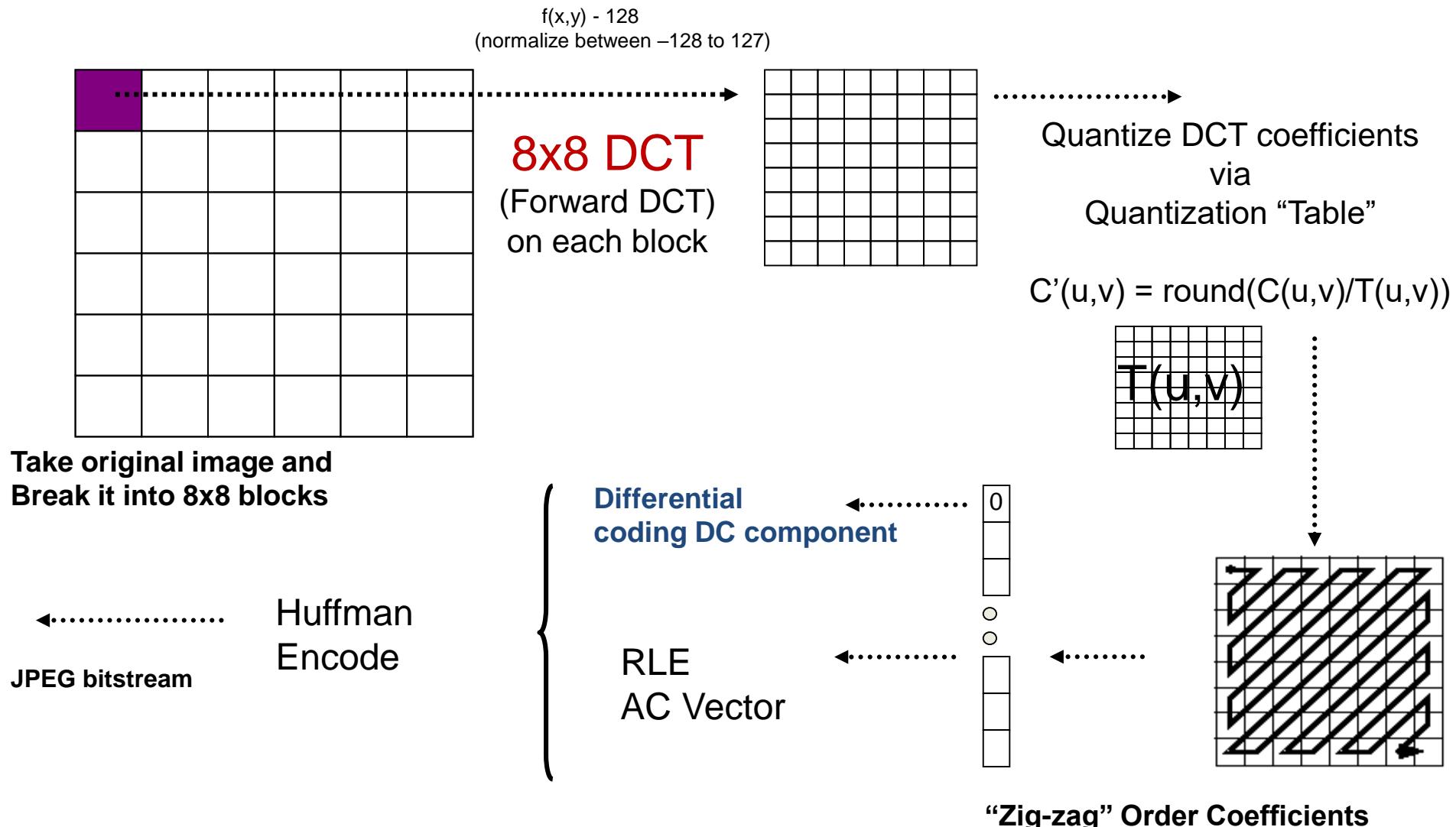
- It is worth noting, that up until this stage, our color values (either RAW or linear sRGB) are related to incoming light in a linear fashion
- After this step, that relationship is broken
- Unlike the sRGB gamma (which is known), the tone-mapping is proprietary and can only be found by a calibration procedure

# JPEG compression



- Joint Photographic Experts Group (JPEG)
- Lossy compression strategy based on the 2D Discrete Cosine Transformation (DCT)
- The by far the most widely adopted standard for image storage

# JPEG approach



JPEG applies almost every compression trick known.

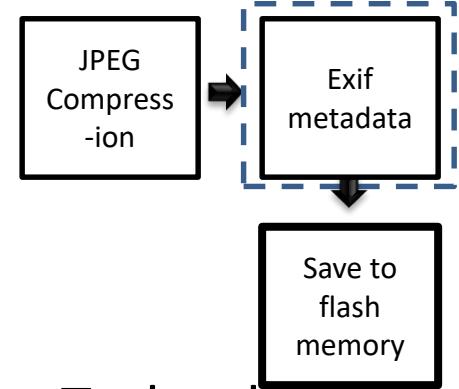
1) Transform coding, 2) psychovisual (loss), 3) Run-length-encoding (RLE), 4) Difference coding, and Huffman.

# JPEG quality

- The amount of quantization applied on the DCT coefficients amounts to a “quality” factor
  - More quantization = better compression (smaller file size)
  - More quantization = lower quality
- Cameras generally allow a range that you can select

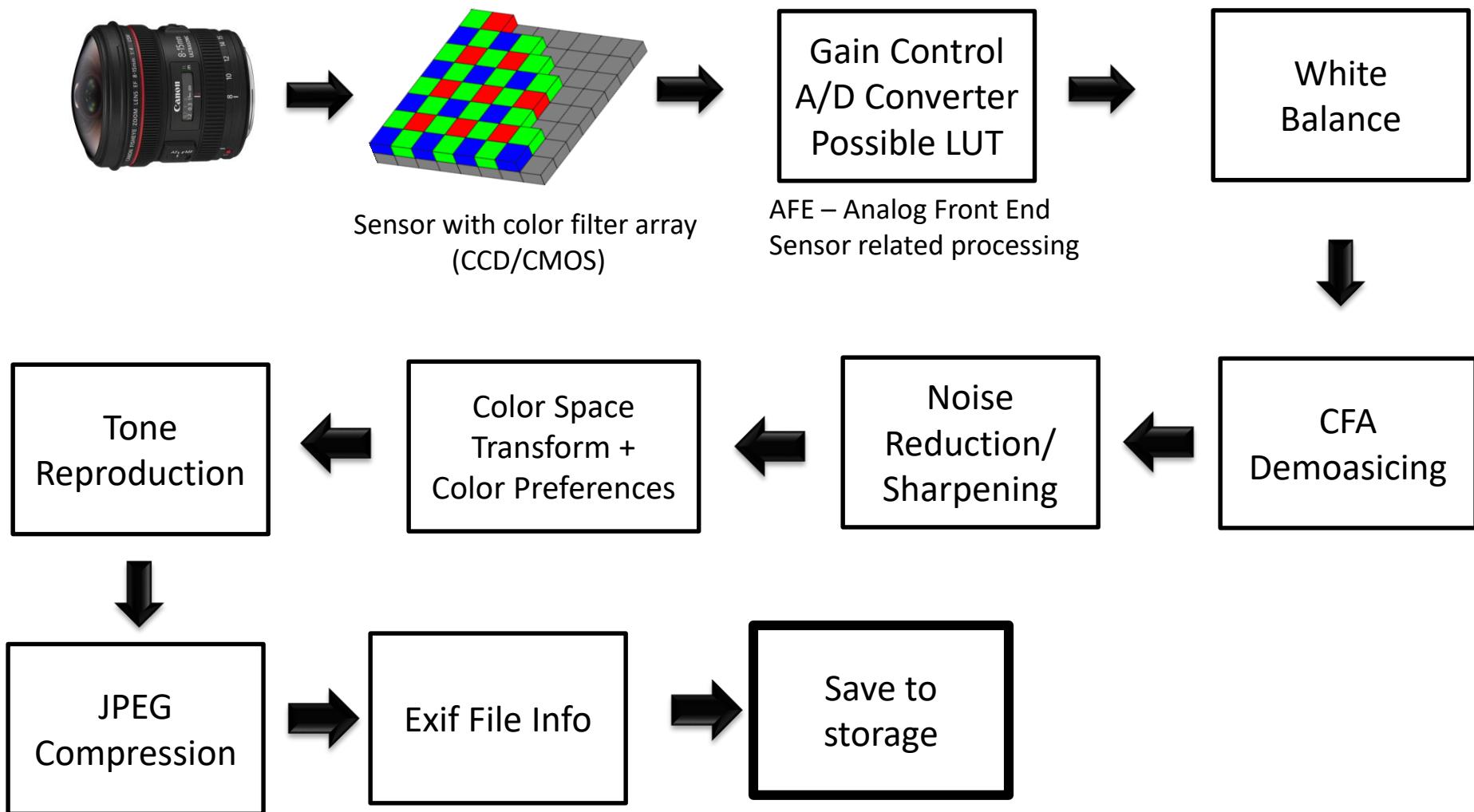


# Exif metadata



- Exchangeable image file format (Exif)
- Created by the Japan Electronics and Information Technology Industries Association (JEITA)
- Associates meta data with images
  - Date/time
  - Camera settings (basic)
    - Image size, aperture, shutter speed, focal length, ISO speed, metering mode (how exposure was estimated)
  - Additional info (from in some Exif files)
    - White-balance settings, even matrix coefficients of white-balnace
    - Picture style (e.g. landscape, vivid, standard, portrait)
    - Output color space (e.g. sRGB, Adobe RGB, RAW)
    - GPS info
    - More . . .

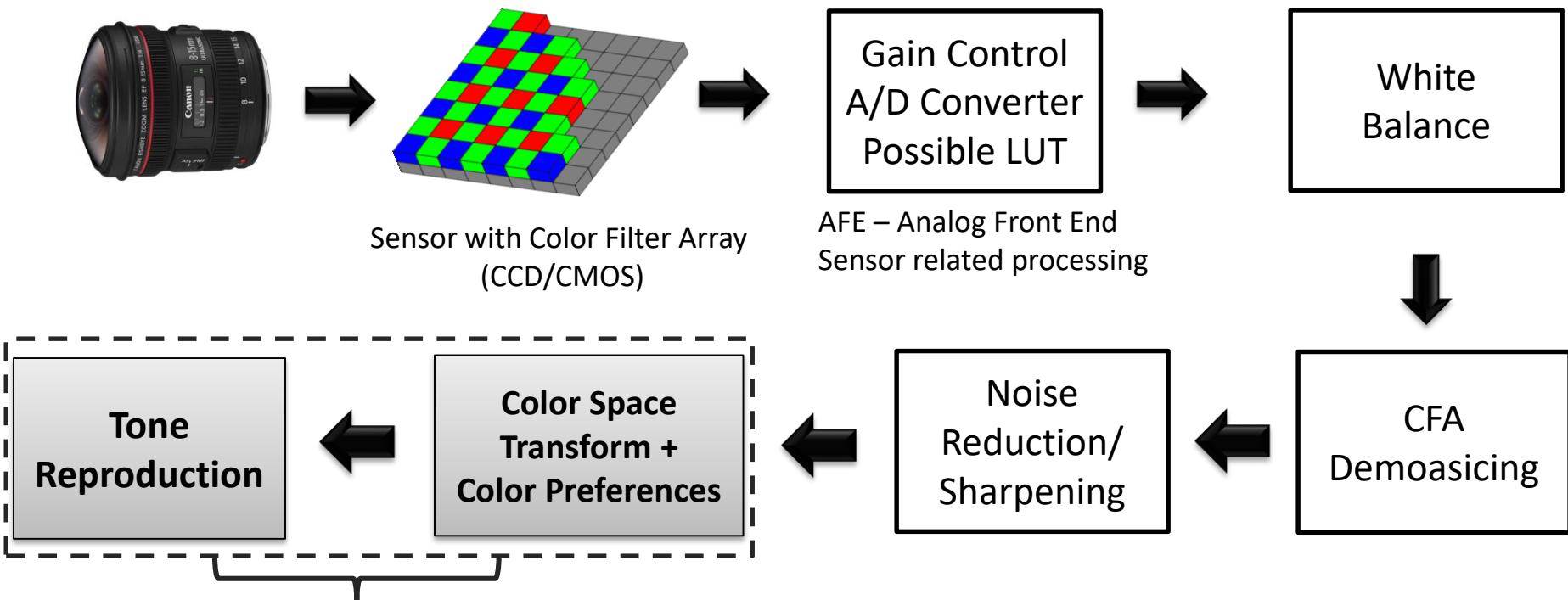
# Saved to storage. We are done!



# Pipeline comments

- Again, important to stress that the exact steps mentioned in these notes only serve as a guide of common steps that take place
- For different camera makes/models, these could be performed in different order (e.g. white-balance after demosaicing) and in different ways (e.g. combining sharpening with demosaicing)

# Color manipulation



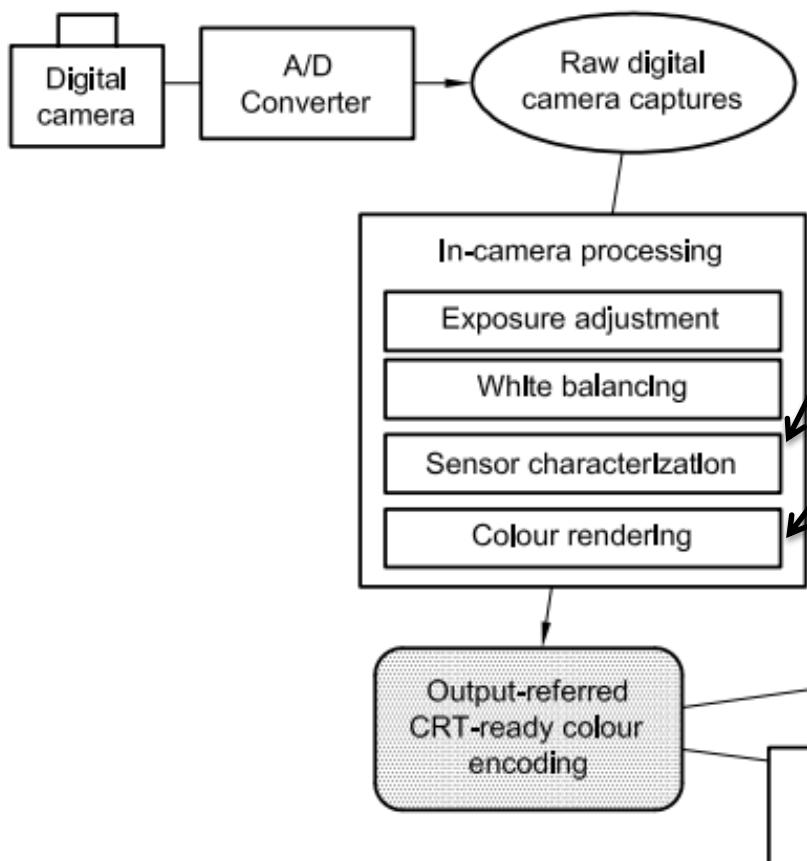
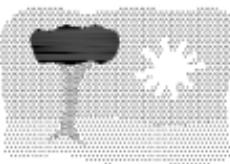
These two steps combined is often referred to as “color rendering” or “color processing/interpolation”. What actually happens here is very much specific to individual cameras, *and even settings on the camera*.

# ICC and color profiles

- International Color Consortium (ICC)
  - In charge of developing several ISO standards for color management
- Promote the use of ICC profiles
- ICC profiles are intended for device manufacturers to describe how their respective color spaces (e.g. sensor RGB) map to canonical color spaces called Profile Connection Spaces (PCS)
- PCS are similar to linking all devices to CIE XYZ, but are more flexible allowing for additional spaces to be defined (beyond CIE XYZ)

# From the ICC – ISO 22028

Original scene



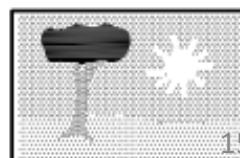
**Photography and graphic technology —  
Extended colour encodings for digital  
image storage, manipulation and  
interchange —**

In this case, “sensor characterization” is related to the Color Adaptation Matrix described earlier.

We can see tone-mapping is not explicitly denoted, instead it is grouped with “color render”.



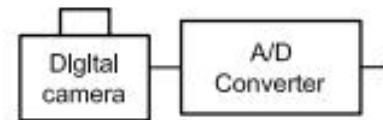
Softcopy image



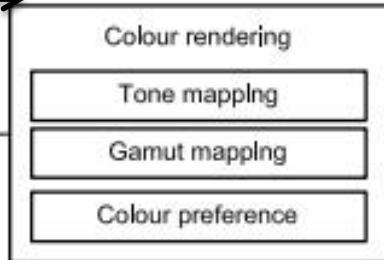
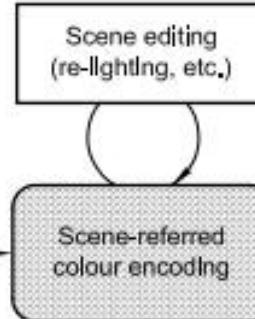
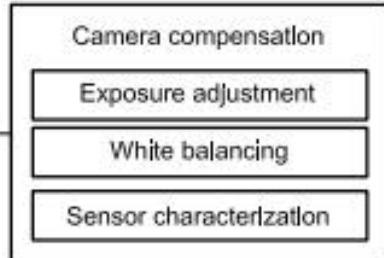
Hardcopy image

# From the ICC ISO 22028

Original scene



Raw digital camera captures



We can see a more detailed breakdown of color rendering.

Picture editing (add text, etc.)

Output-referred colour encoding

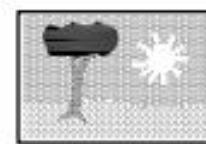
Display colour transform

Softcopy Image



Printer colour transform

Hardcopy Image



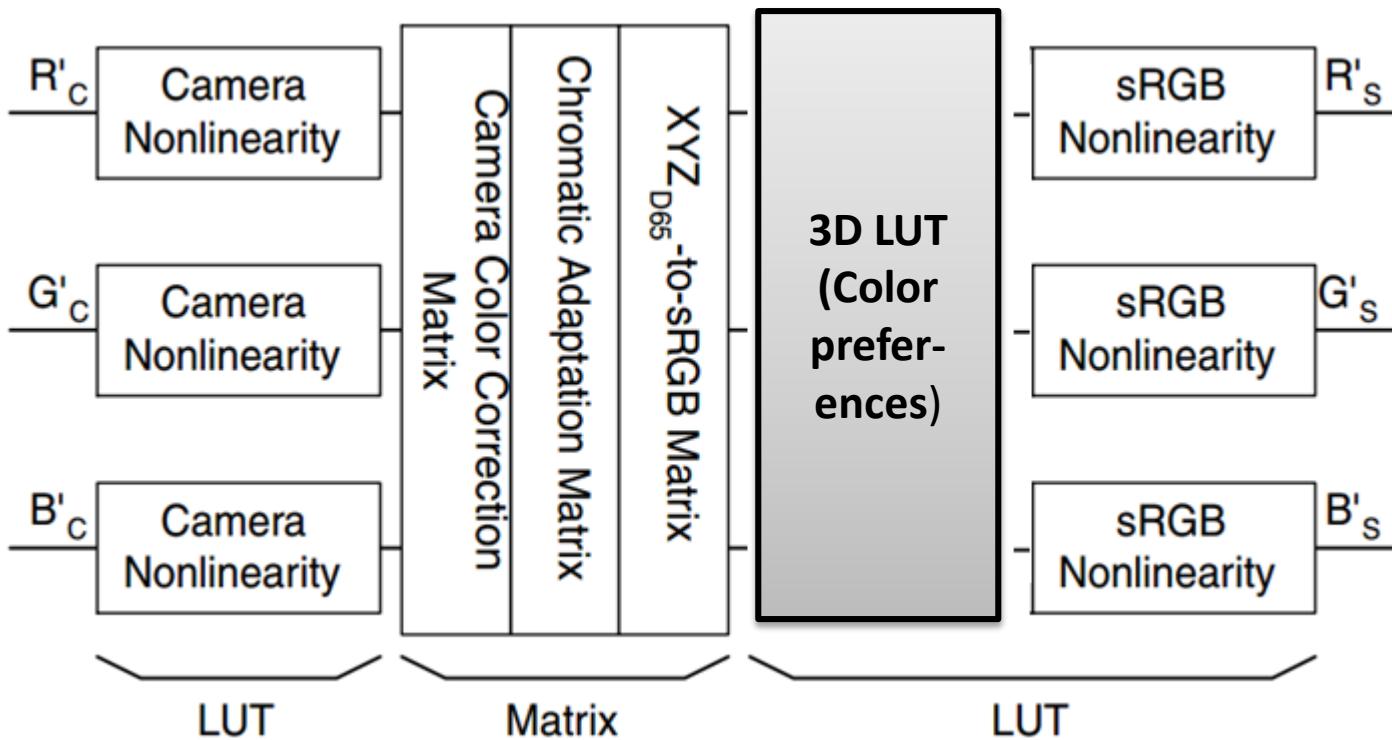
This describes a basic digital camera pipeline in more detail.

RGB values linked to the device are considered “scene referred”.

After the color transform to sRGB they are denoted as “output referred” color encodings.

# Tone mapping/color rendering

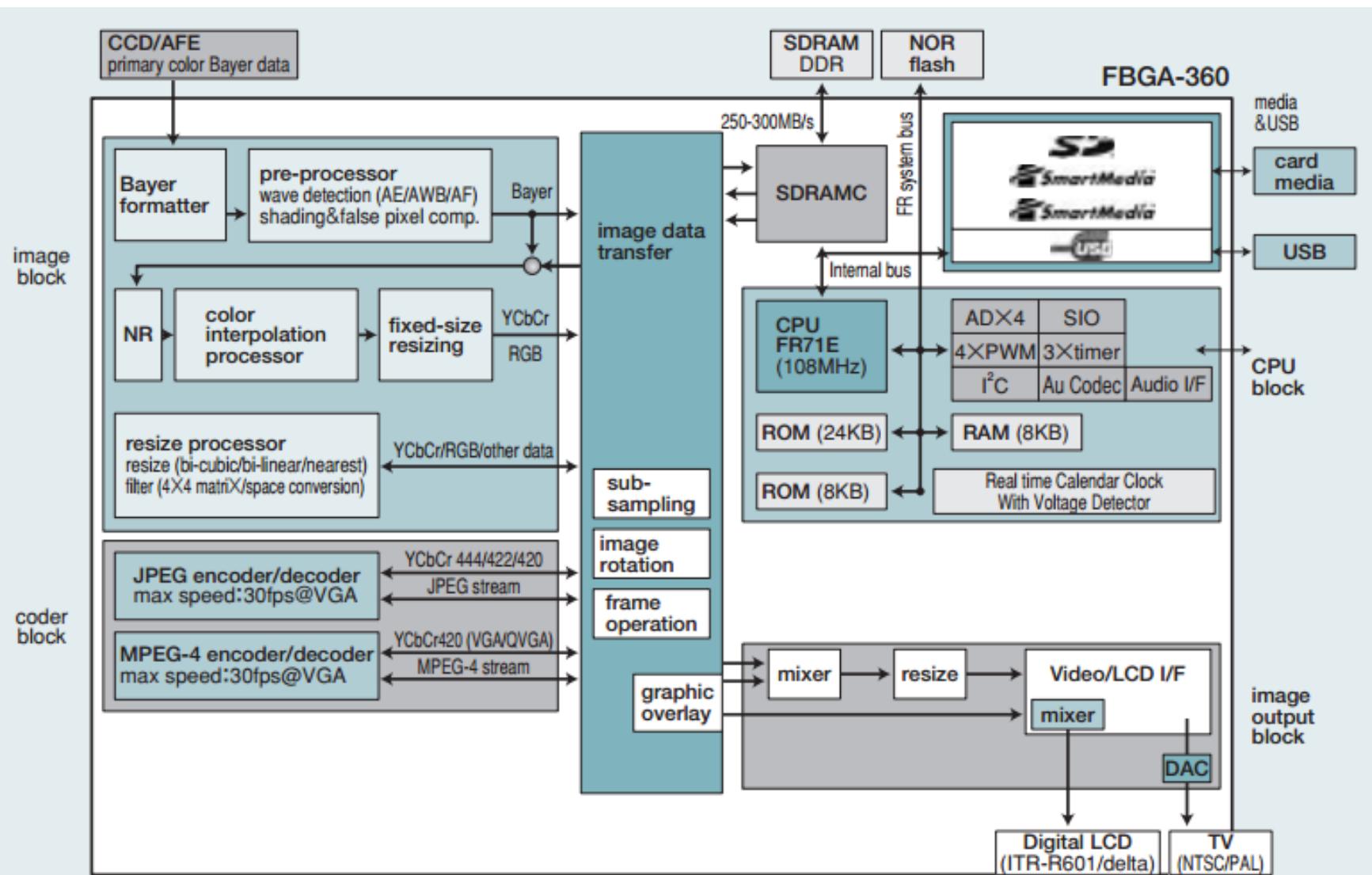
- From the ICC ISO standards, we see this part is more complex than just matrix + tone-map
- In fact, it can often involve *3D LUTs*



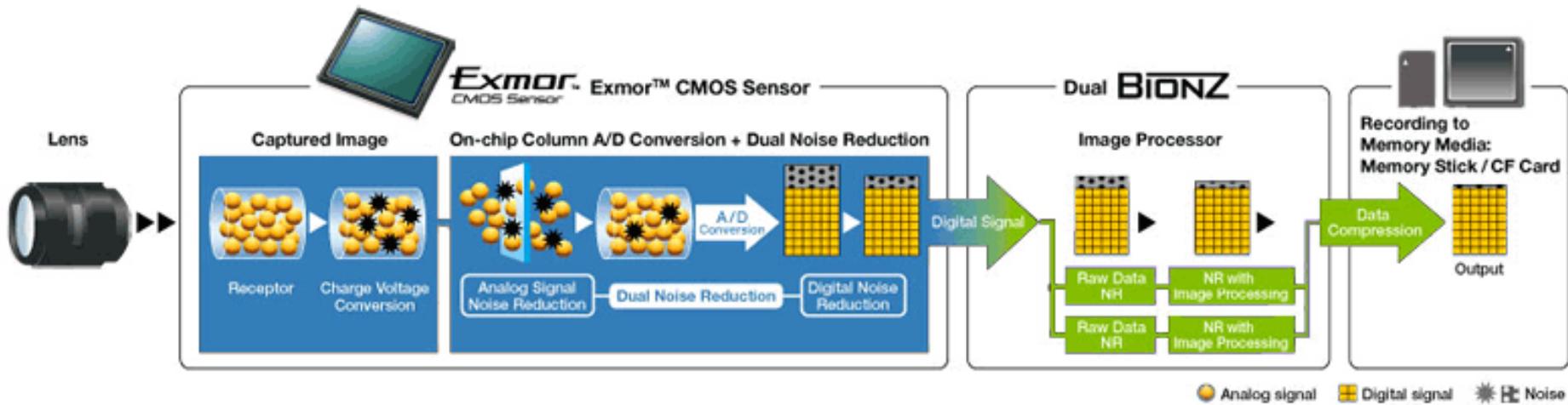
# Camera image processors

- Steps applied after the sensor output are generally performed by an “image processor” on the camera
- Different cameras use different processing boards/software
- High-end cameras and associated processors
  - Nikon – Expeed
  - Fuji – Real photo engine
  - Canon – DIGIC
  - Sony – BIONZ
  - ...
- Mobile-devices
  - Qualcomm - Snapdragon
  - Nvidia - Tegra

# Example: Expeed block diagram



# Example: Sony Exmor + BIONZ



Not a lot of detail . . .

# Tutorial schedule

- Part 1 (General Part)
  - Motivation
  - Review of color/color spaces
  - Overview of camera imaging pipeline
- Part 2\* (Specific Part)
  - Modeling the in-camera color pipeline
  - Photo-refinishing
- Part 3 (Wrap Up)
  - The good, the bad, and the ugly of commodity cameras and computer vision research
  - Concluding Remarks

# Part 2: Modeling the onboard camera color processing pipeline

# Part 2 Acknowledgements



Prof. Seon Joo Kim  
(Yonsei)



Dr. Hai-ting Lin  
(NUS/U. Delaware)



Prof. Sabine Süsstrunk  
(EPFL)



Dr. Steven Lin  
(MSR-Asia)

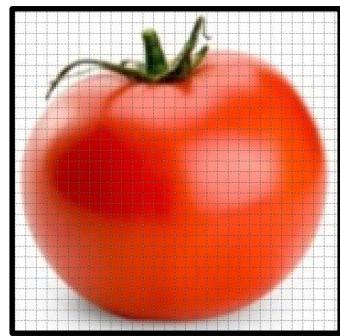


Dr. Zheng Lu  
(NUS/City U Hong Kong)

# Standard color spaces are great



“The real scene”



Camera  
sRGB Image



sRGB → CIE XYZ → CMYK → printer ink (ICC Profile)



sRGB → CIE XYZ → Display (ICC Profile)

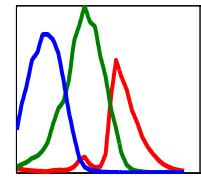


sRGB → HSV → Image Editing Software



# Ideal (simple) camera pipeline

Sensor Image (RAW)



Camera-Specific  
R/G/B Sensitivity  
Functions

White  
balance

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

Color Space  
Transform (CST)

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

$$\begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$



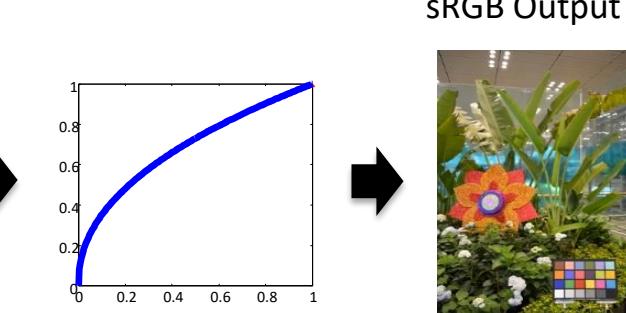
CIE XYZ  
Linear-sRGB

White-balanced  
camera-specific  
raw-RGB

STEP 1

CST maps from camera  
color space to CIE XYZ.  
Then we go to linear-sRGB.

STEP 2



Apply sRGB  
encoding gamma.

STEP 3



# The world isn't ideal

Samsung Galaxy S6 edge



HTC One M9



LG G4



Canon



Nikon

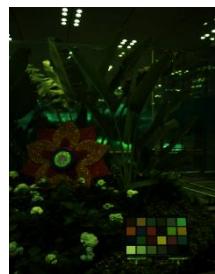


Sony



# Tone-curve is camera-specific

Sensor Image (RAW)



White  
balance

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

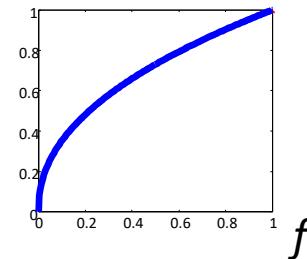
Color Space  
Transform (CST)

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

$$\begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$



CIE XYZ  
Linear-sRGB



*f*

# Tone-curve is camera-specific

Sensor Image (RAW)



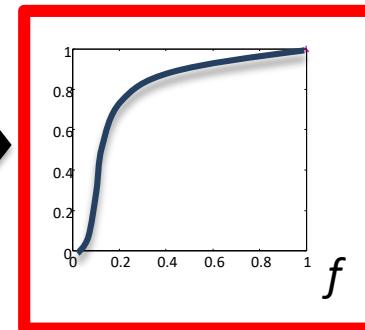
White  
balance

Color Space  
Transform (CST)



$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix} \rightarrow \begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$

CIE XYZ  
Linear-sRGB



We have known for a long time that  
***nobody*** uses the sRGB encoding gamma

Large body of research to  
estimate these camera-specific  
tone-curves  $f$ . All assume this is  
fixed per camera.

Mann and Picard, SPIE'95  
Debevec and Malik, SIG'97  
Mitsunaga and Nayar, CVPR'99  
Farid, TIP'01  
Grossberg and Nayar, TPAMI'03  
Grossberg and Nayar, TPAMI'04  
Lin et al, CVPR'04

...  
Manders et al, ICIP'04  
Pal et al, CVPR'04  
Lin et al, ICCV'05  
Kim and Pollefeys, TPAMI'08  
Chakrabarti et al, BMVC'09

# Manufacturers give us a hint

## > Standard



Glowing prints with crisp finishes.  
It is the basic color of EOS DIGITAL.

## > Portrait



For transparent, healthy skin for women and children

## > Landscape



Crisp and impressive reproduction of blue skies and green trees in deep, vivid color

## > Neutral



Subjects are recorded in rich detail, giving the greatest latitude for image processing

## > Faithful



Accurate recording of the subject's color, close to the actual image seen with the naked eye

## > Monochrome



Filter work and sepia tone with the freedom of digital monochrome

From Canon's user manual

# Nikon D7000 - RAW

SHOOTING MENU	
JPEG compression	
NEF (RAW) recording	
White balance	AUTO
Set Picture Control	ESD
Manage Picture Control	--
Color space	SRGB
Active D-Lighting	OFF
Long exp. NR	OFF



# NEUTRAL



# STANDARD



# PORTRAIT



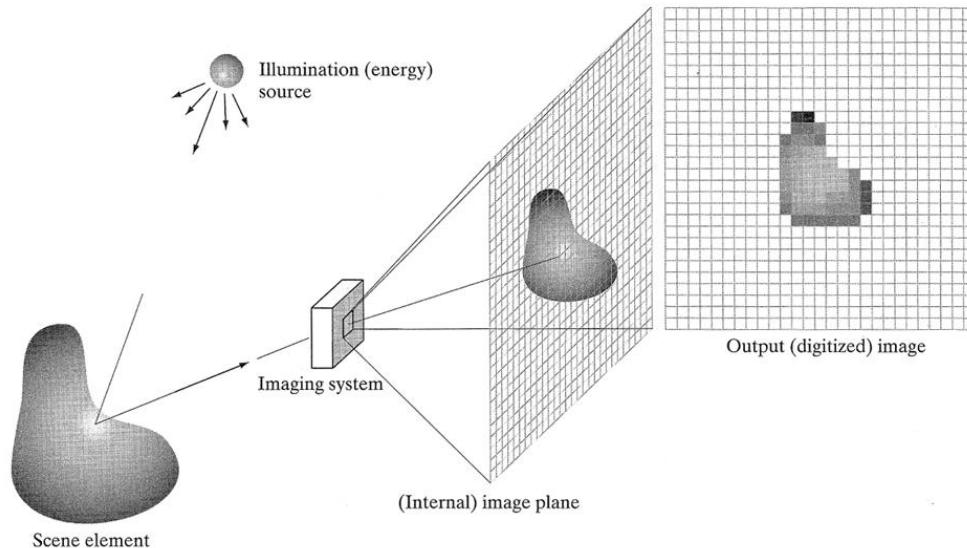
# LANDSCAPE



# VIVID



# Camera = light-measuring device



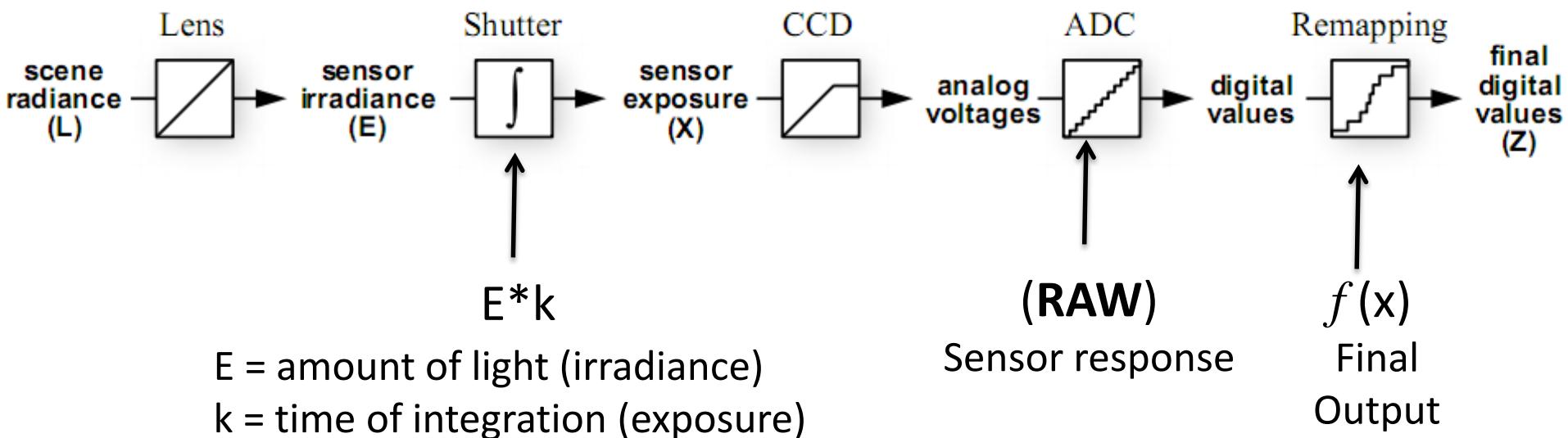
**“All models are wrong, but some are useful;  
how wrong can they be and still be useful.”**



George Box  
Professor Emeritus of Statistics  
U. Wisconsin

# Early work

## “Radiometric Calibration”



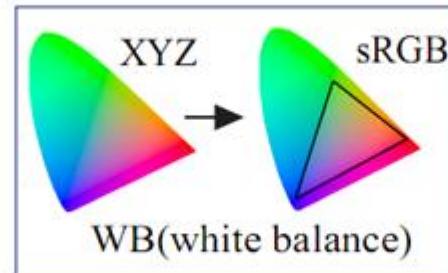
Unknown  $f$  (**tone-map**) is the camera’s non-linear transform of **RAW** to “intensity”.

# Accepted model

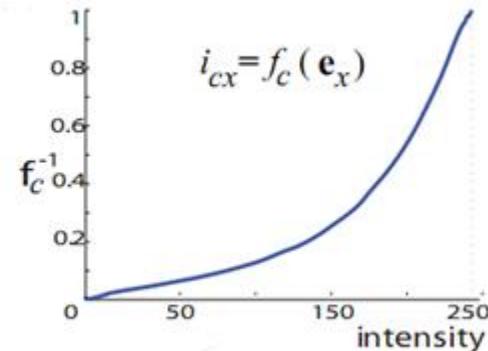
(1) Irradiance  $\mathbf{E}_x$  (RAW)



(2) Color Transform



(3) Radiometric Response



$$\begin{bmatrix} E_{rx} \\ E_{gx} \\ E_{bx} \end{bmatrix}$$

(RAW)

$$\begin{bmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{bmatrix} = \mathbf{T} \begin{bmatrix} E_{rx} \\ E_{gx} \\ E_{bx} \end{bmatrix}$$

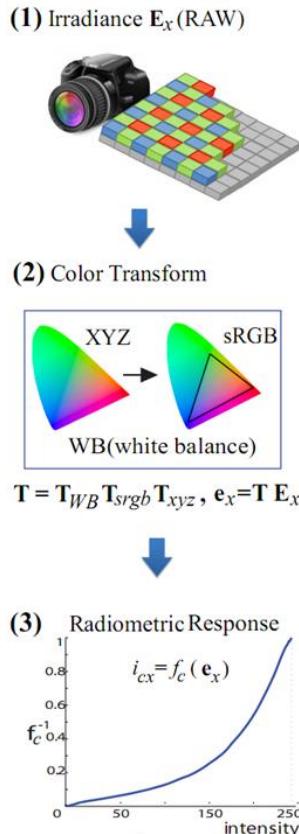
$\mathbf{T}$  is a 3x3 matrix

(small  $e$  are white-balanced  
RAW)

$$\begin{bmatrix} i_{rx} \\ i_{gx} \\ i_{bx} \end{bmatrix} = \begin{bmatrix} f_r(e_{rx}) \\ f_g(e_{gx}) \\ f_b(e_{bx}) \end{bmatrix}$$

$i$  is the sRGB  
output and  $f$  is  
a non-linear  
function

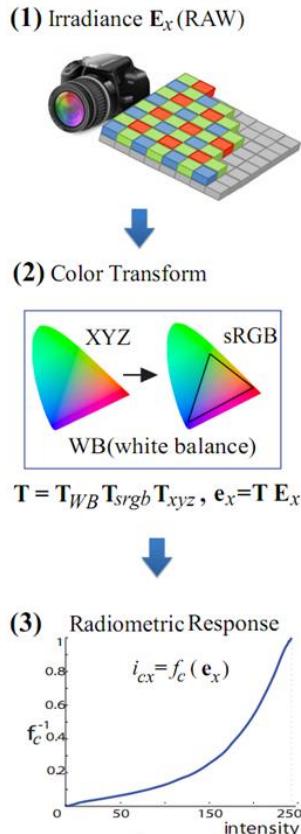
# Prior work



Fixed property of the camera

- Mann and Picard, SPIE'95  
Debevec and Malik, SIG'97  
Mitsunaga and Nayar, CVPR'99  
Farid, TIP'01  
Grossberg and Nayar, TPAMI'03  
Grossberg and Nayar, TPAMI'04  
Lin et al, CVPR'04  
...  
Manders et al, ICIP'04  
Pal et al, CVPR'04  
Lin et al, ICCV'05  
Kim and Pollefeys, TPAMI'08  
Chakrabarti et al, BMVC'09

# Prior work



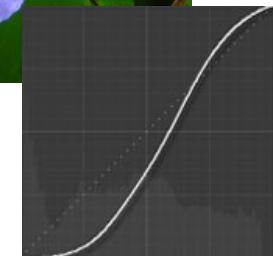
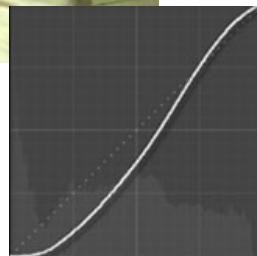
Fixed property of the camera

- Mann and Picard, SPIE'95  
Debevec and Malik, SIG'97  
Mitsunaga and Nayar, CVPR'99  
Farid, TIP'01  
Grossberg and Nayar, TPAMI'03  
Grossberg and Nayar, TPAMI'04  
Lin et al, CVPR'04  
...  
Manders et al, ICIP'04  
Pal et al, CVPR'04  
Lin et al, ICCV'05  
Kim and Pollefeys, TPAMI'08  
Chakrabarti et al, BMVC'09

## Chakrabarti et al conclusions:

- 😊 RAW is meaningful . . . .
- 😢 But, requires a 24 parameter model that is **scene-dependent** to accurately go back from sRGB to RAW.

# Scene dependent. . .



Tone curve,  $f$ , is computed based on scene content.  
**This makes it almost impossible to pre-compute.**

# Accepted model

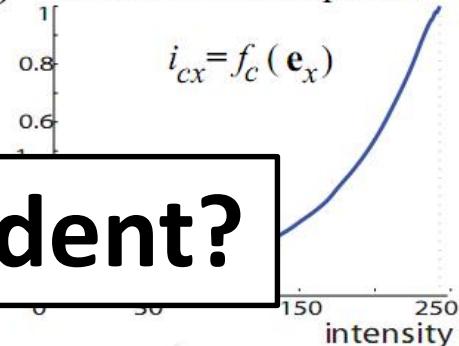
(1) Irradiance  $E_x$  (RAW)



(2) Color Transform



(3) Radiometric Response



Is processing scene dependent?

$$\mathbf{T} = \mathbf{T}_{WB} \mathbf{T}_{srgb} \mathbf{T}_{xyz}, \mathbf{e}_x = \mathbf{T} \mathbf{E}_x$$

Or is this model not good enough?

$$\begin{bmatrix} i_{gx} \\ i_{bx} \end{bmatrix} = \begin{bmatrix} f_g(e_{gx}) \\ f(e_{bx}) \end{bmatrix} \quad \leftarrow \quad \begin{bmatrix} e_{gx} \\ e_{bx} \end{bmatrix} = \mathbf{T} \begin{bmatrix} E_{gx} \\ E_{bx} \end{bmatrix}$$

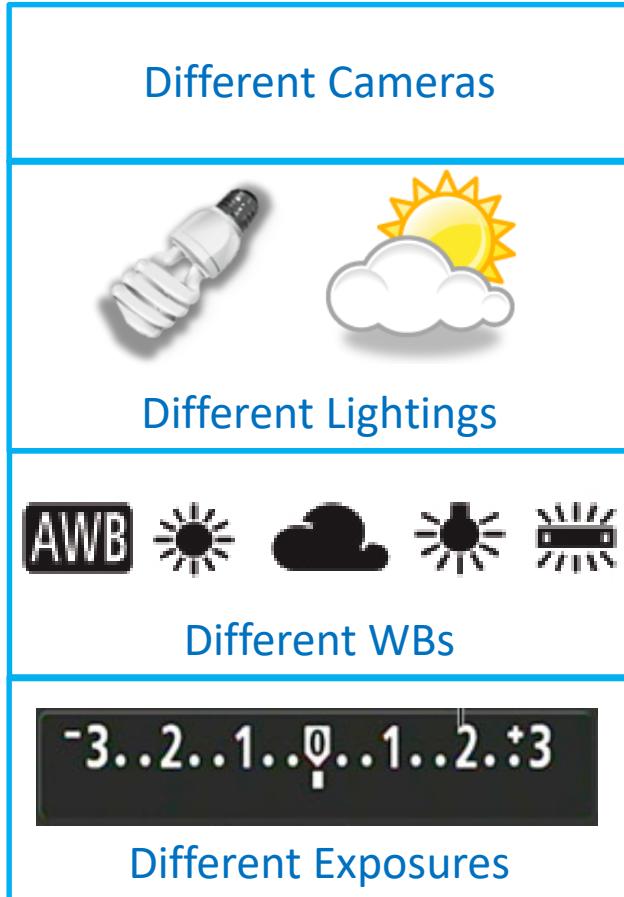
# Our experiment: data collection

- More than **10,000 images** from **33 cameras** from DSLRs to point-and-shoots
  - Images of color charts under indoor / outdoor (cloudy)
  - Images are taken at all possible shutter speeds, at multiple aperture, and white balance settings. JPEG / RAW both captured if possible.
- \* Special shooting features such as lighting optimizer are turned off



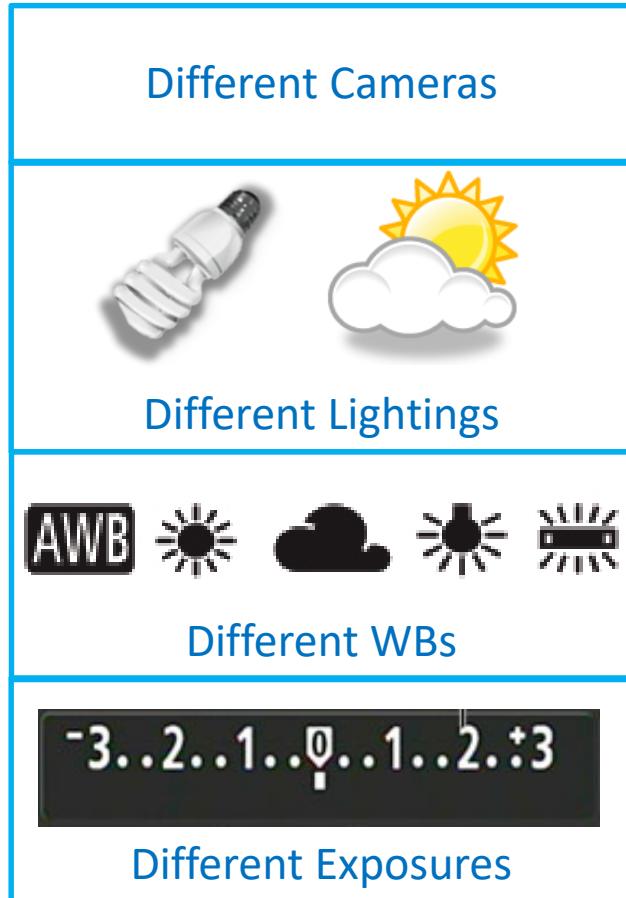
# Data collection

More than **10,000 images** from **33 cameras**



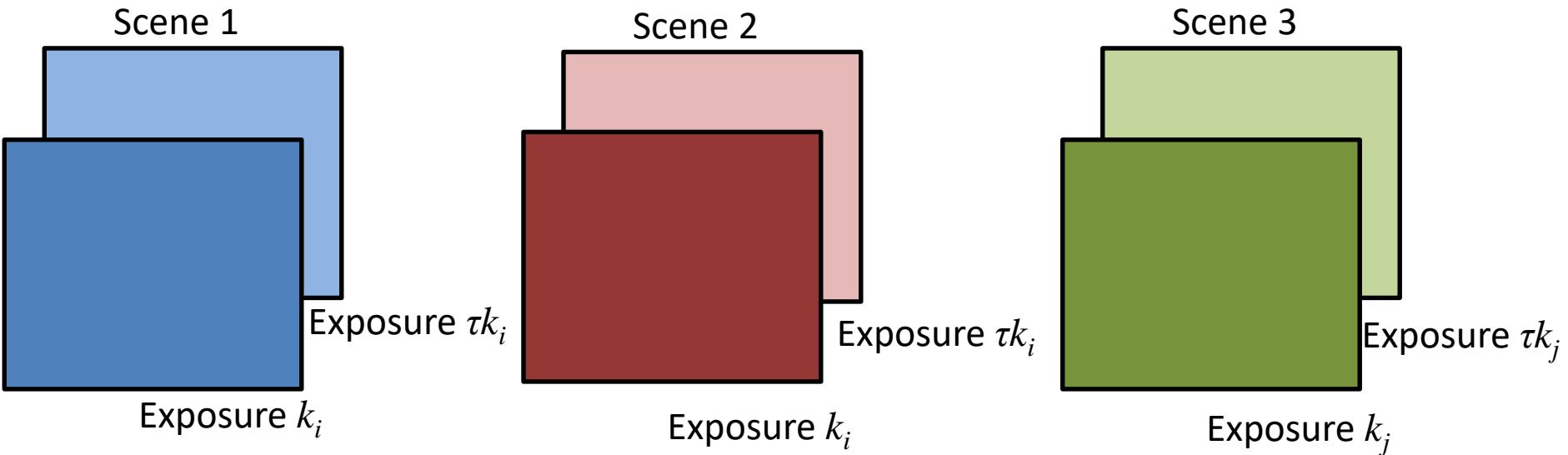
# Data collection

More than **10,000 images** from **33 cameras**



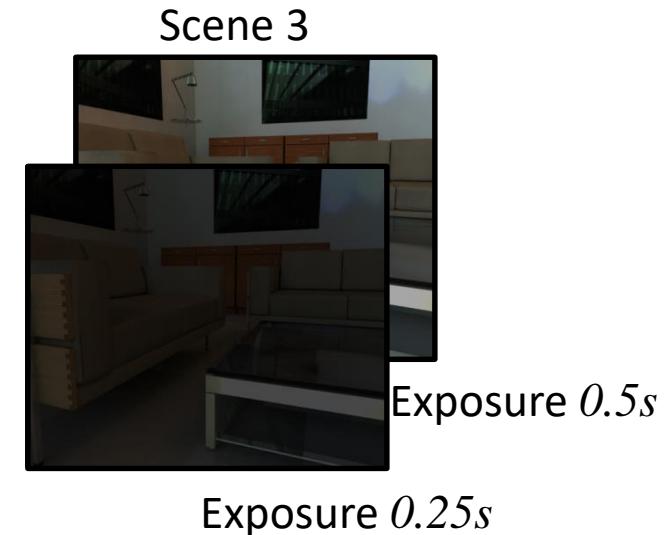
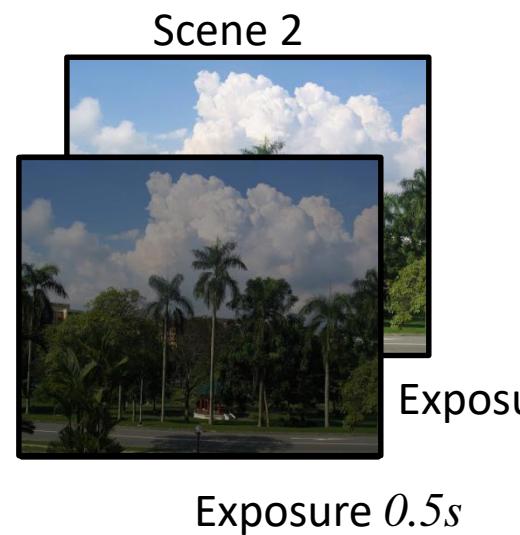
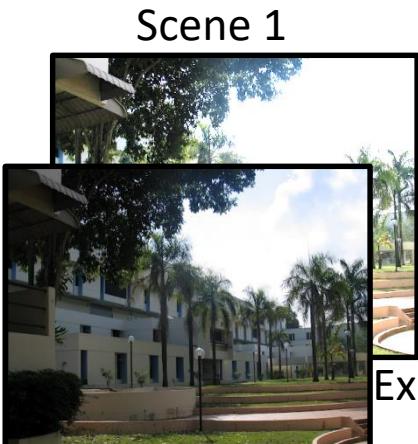
# Checked if $f$ is fixed or scene dependent

- How?
- Plot the **brightness transfer function (BTF)**
  - Plot points from image pairs of different scenes
  - Each pair has the same ratio,  $\tau$ , of exposure change



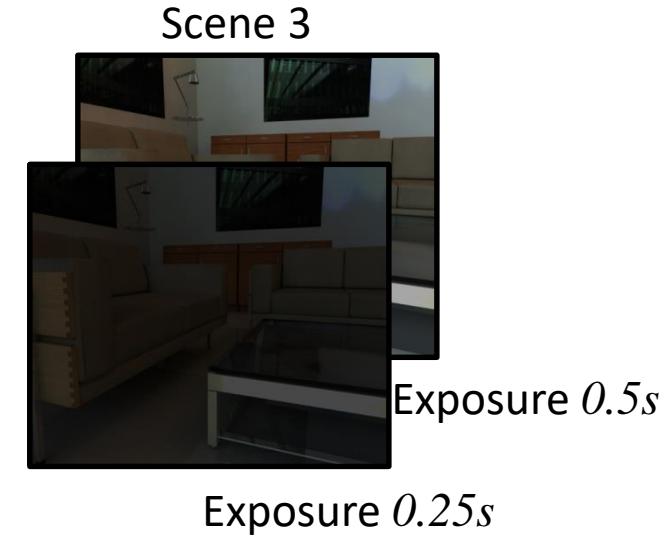
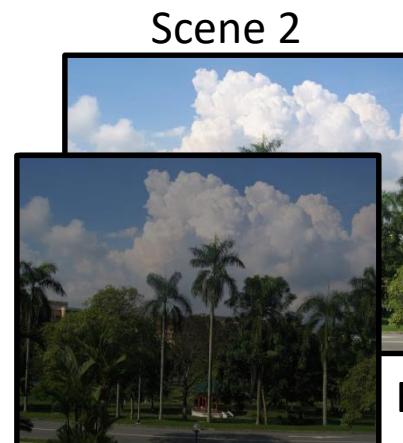
# Checked if $f$ is fixed or scene dependent

- How?
- Plot the **brightness transfer function (BTF)**
  - Plot points from image pairs of different scenes
  - Each pair has the same ratio,  $\tau$ , of exposure change

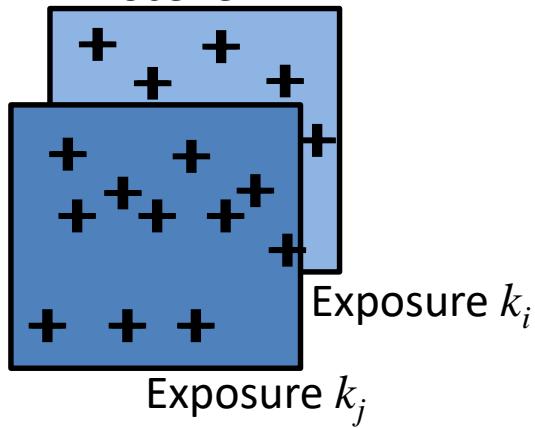


# Checked if $f$ is fixed or scene dependent

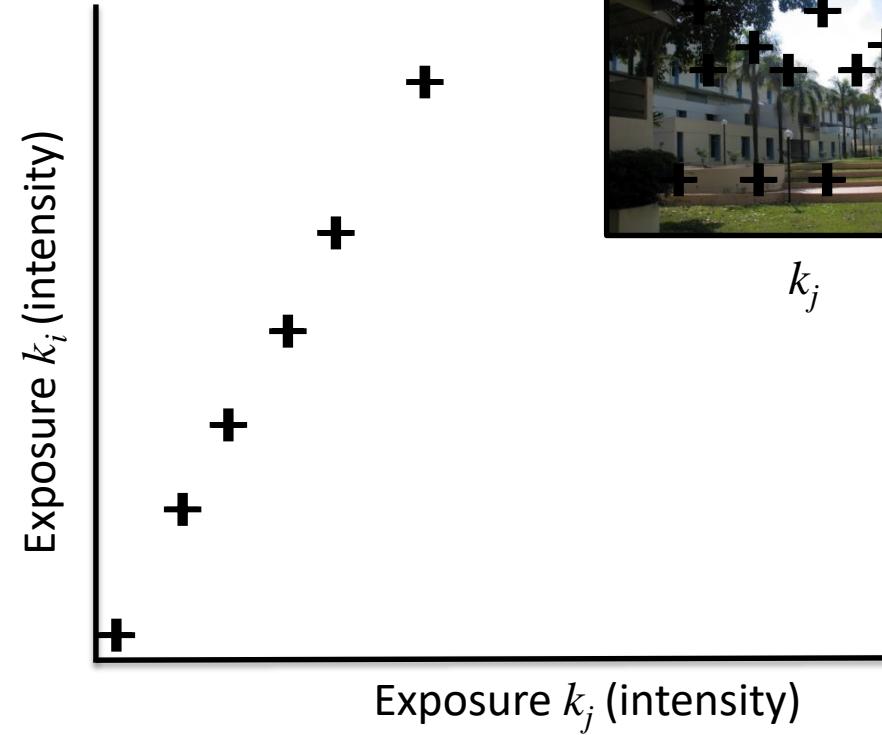
- How?
- Plot the **brightness transfer function (BTF)**
  - Plot points from image pairs of different scenes
  - Each pair has the same ratio,  $\tau$ , of exposure change



Scene 1

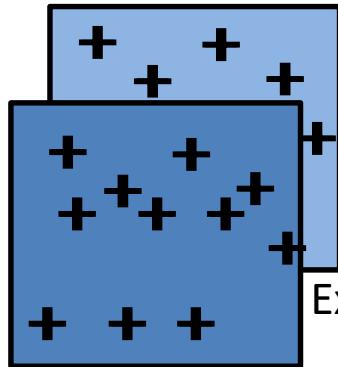


BTF

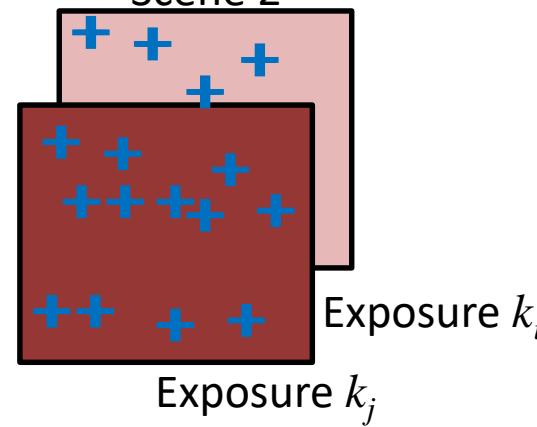


Linear function looks like this. . .

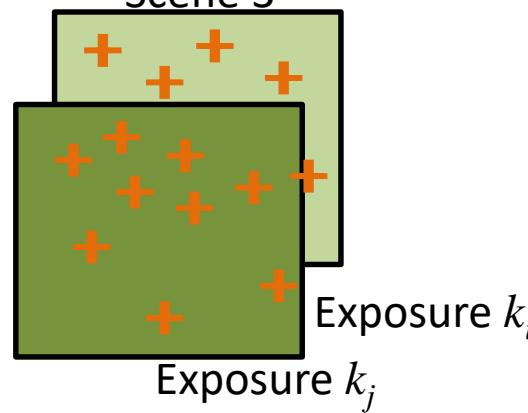
Scene 1



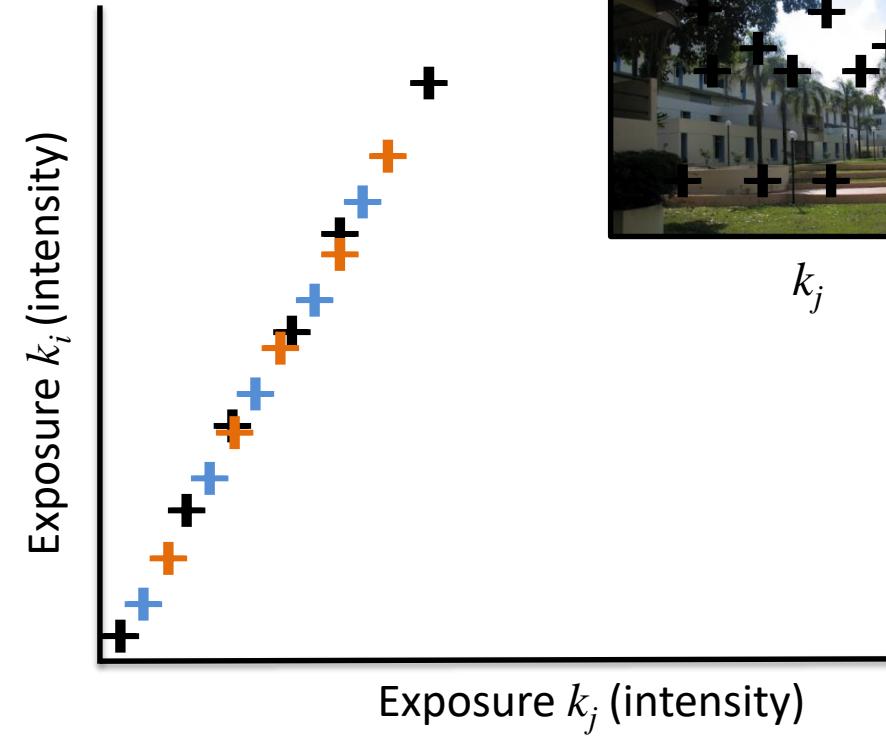
Scene 2



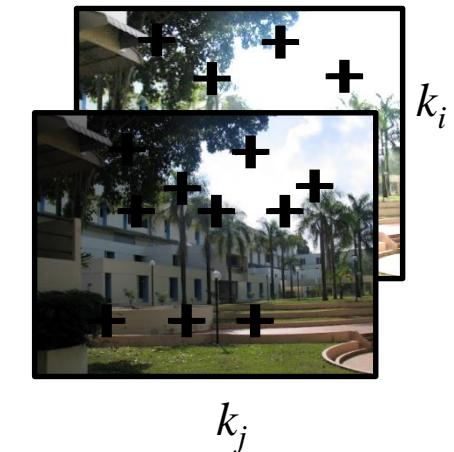
Scene 3



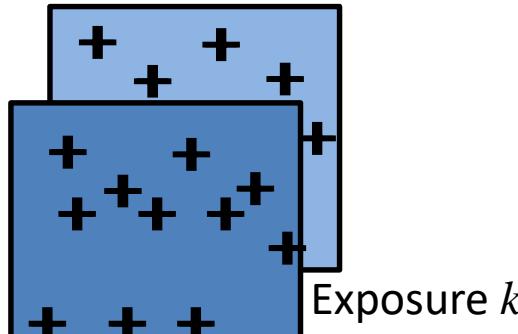
# BTF



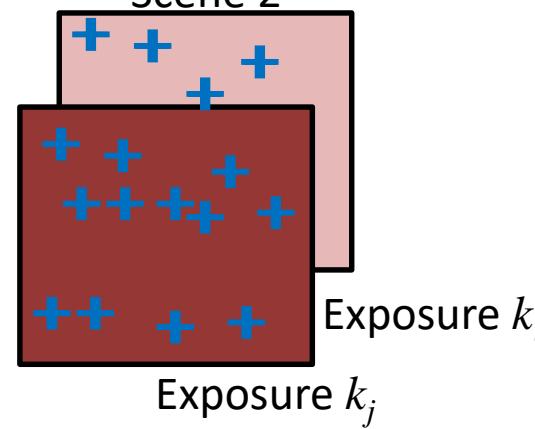
Linear function looks like this...



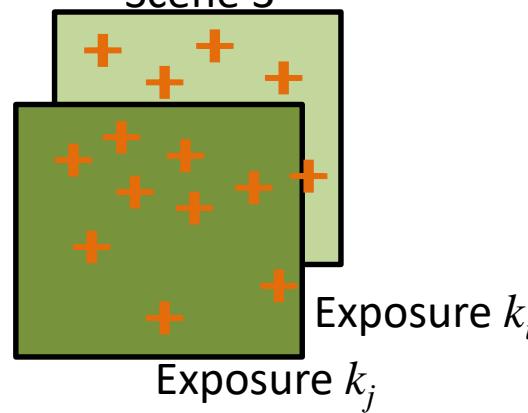
Scene 1



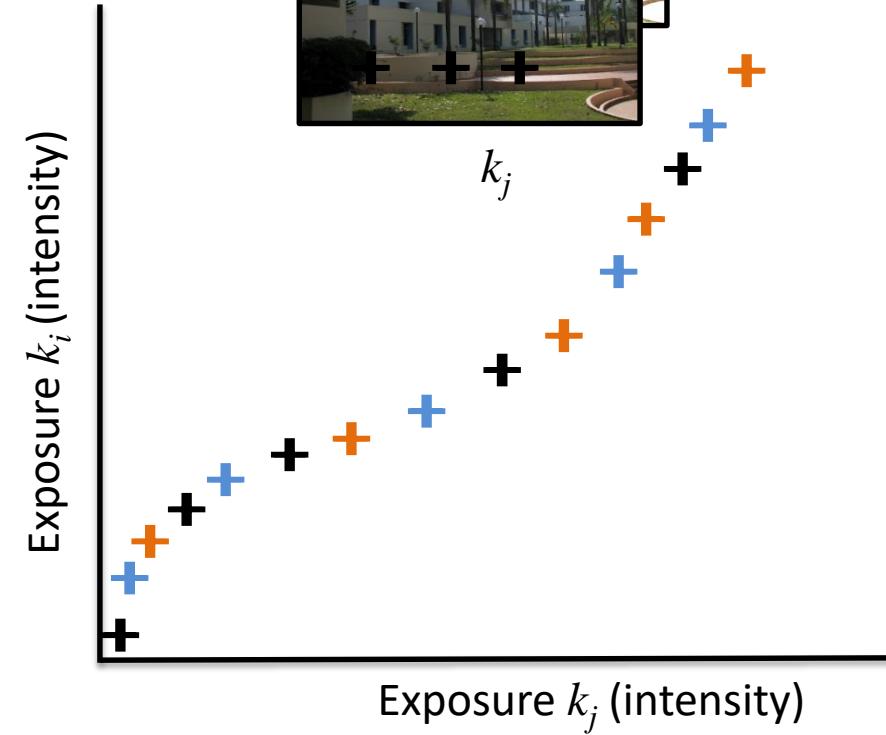
Scene 2



Scene 3

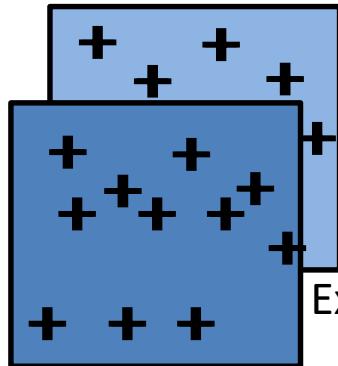


BTF

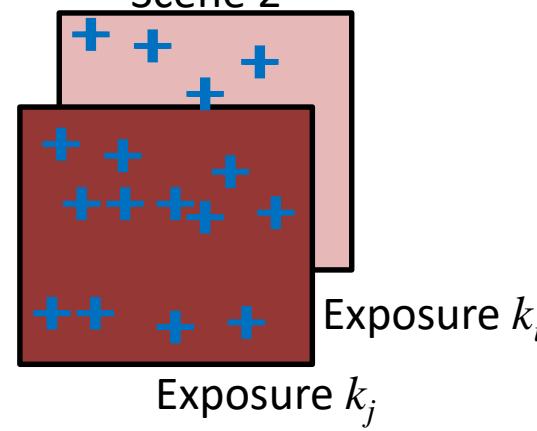


Non-linear BTF looks like this . .

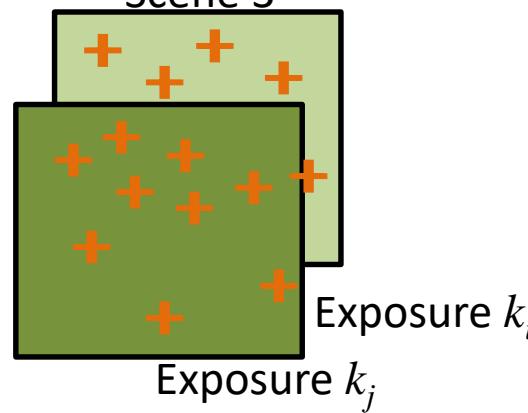
Scene 1



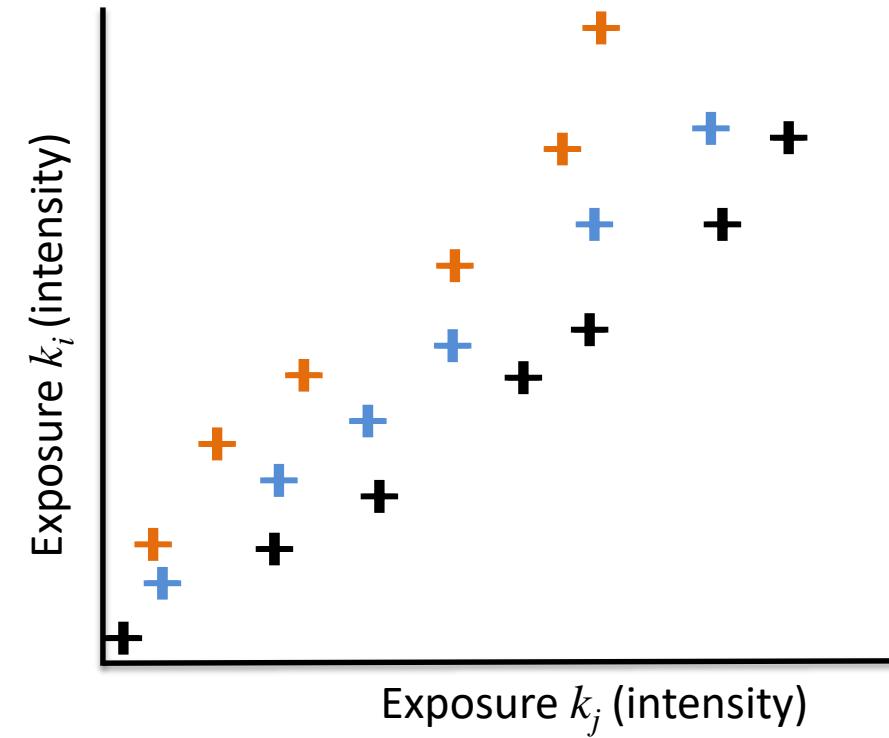
Scene 2



Scene 3



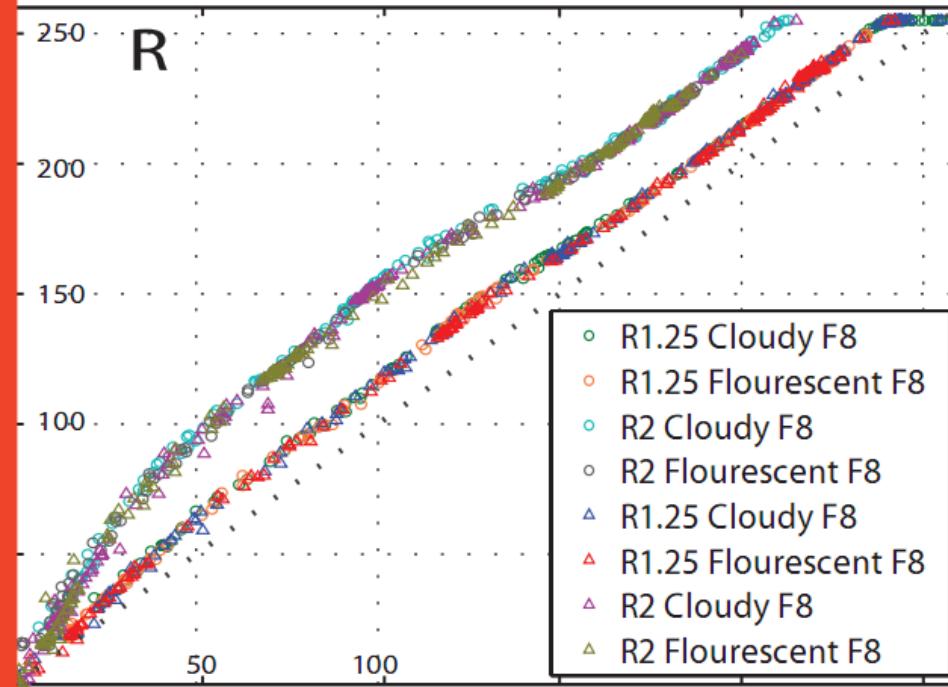
# BTF



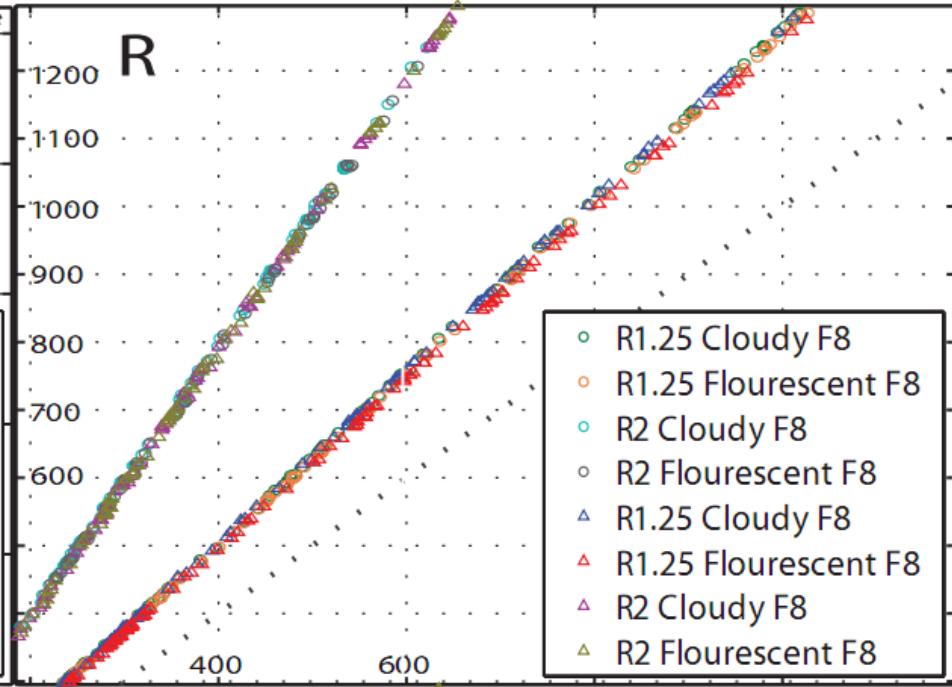
**Scene dependent** non-linear BTF  
looks like this . .

# For the most part . . . it was ok

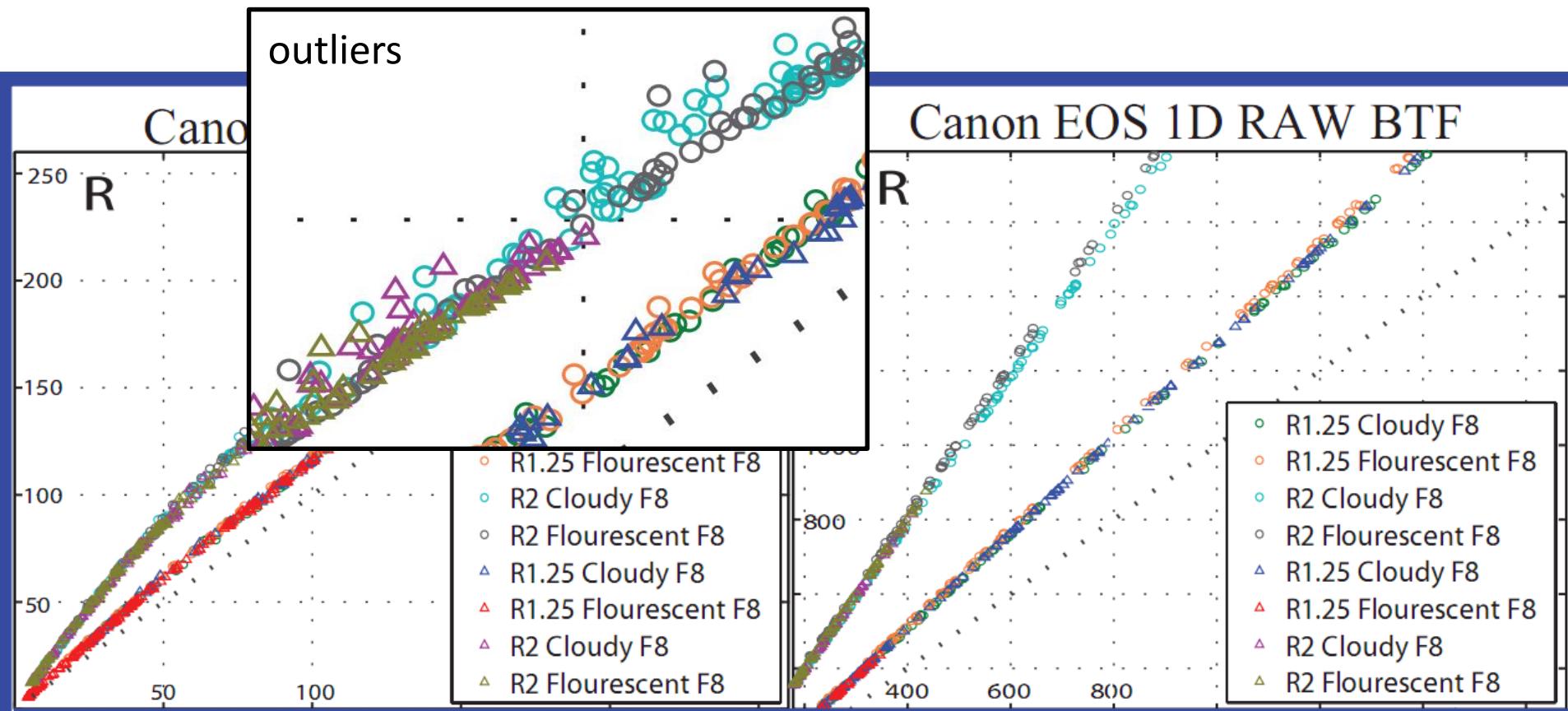
Nikon D50 sRGB BTF



Nikon D50 RAW BTF

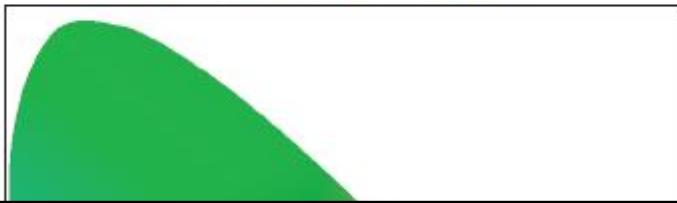


# For the most part . . . it was ok



# Where are the outliers?

Canon EOS 1D



Nikon D50

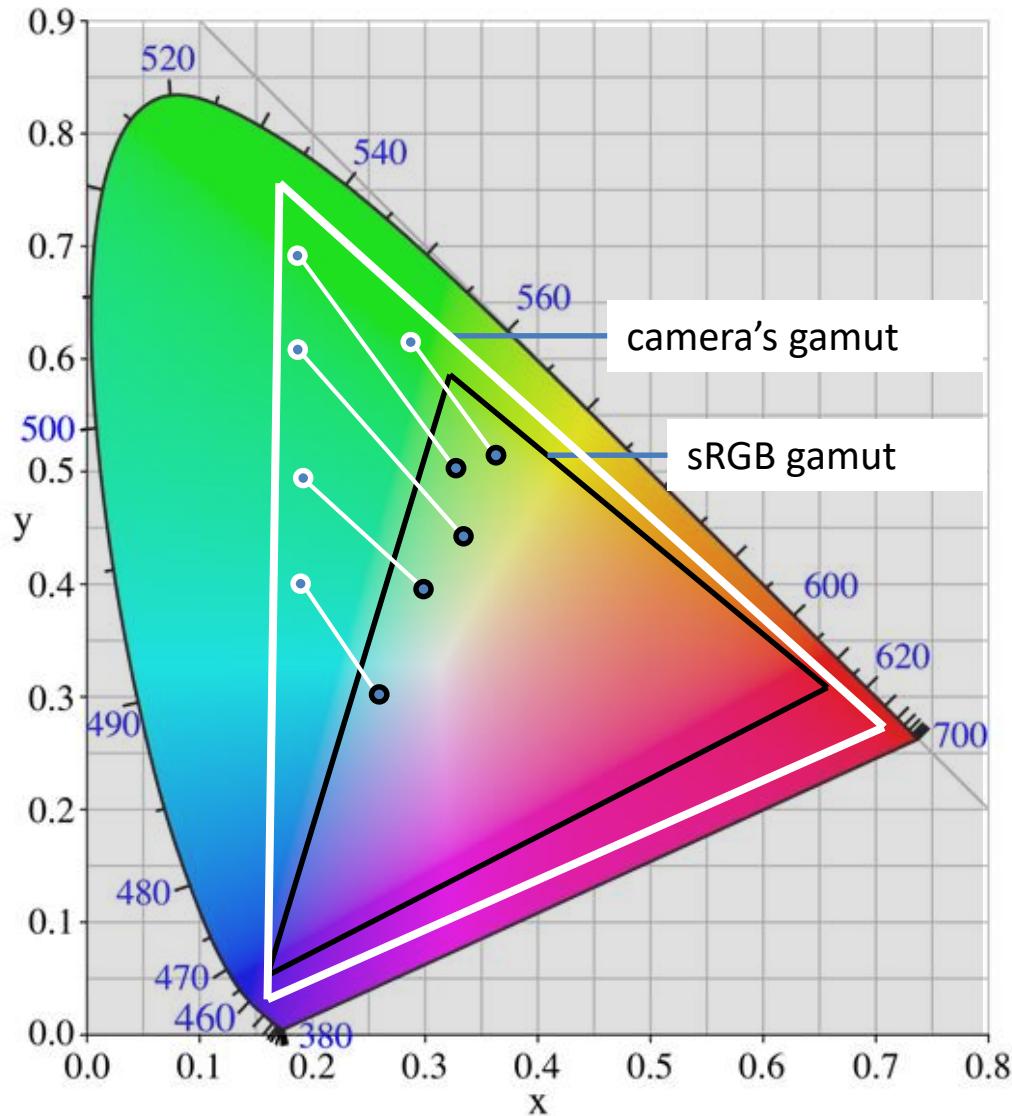


Outliers were not scene  
dependent.

Outliers were color dependent.



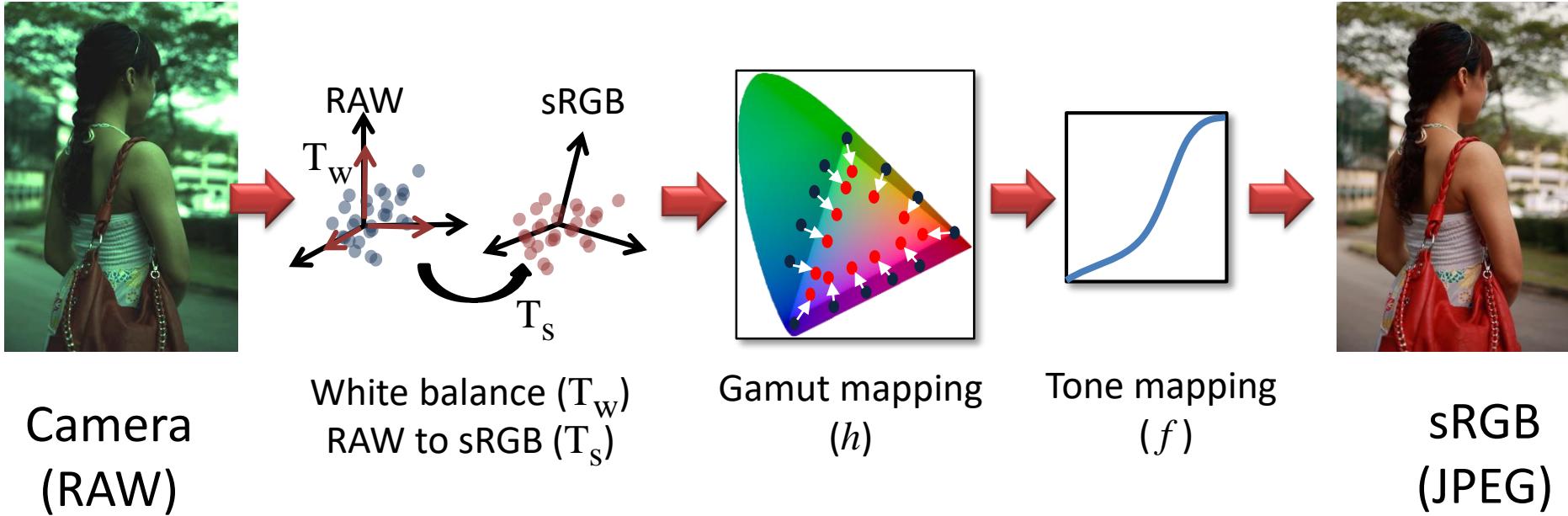
# Gamut mapping



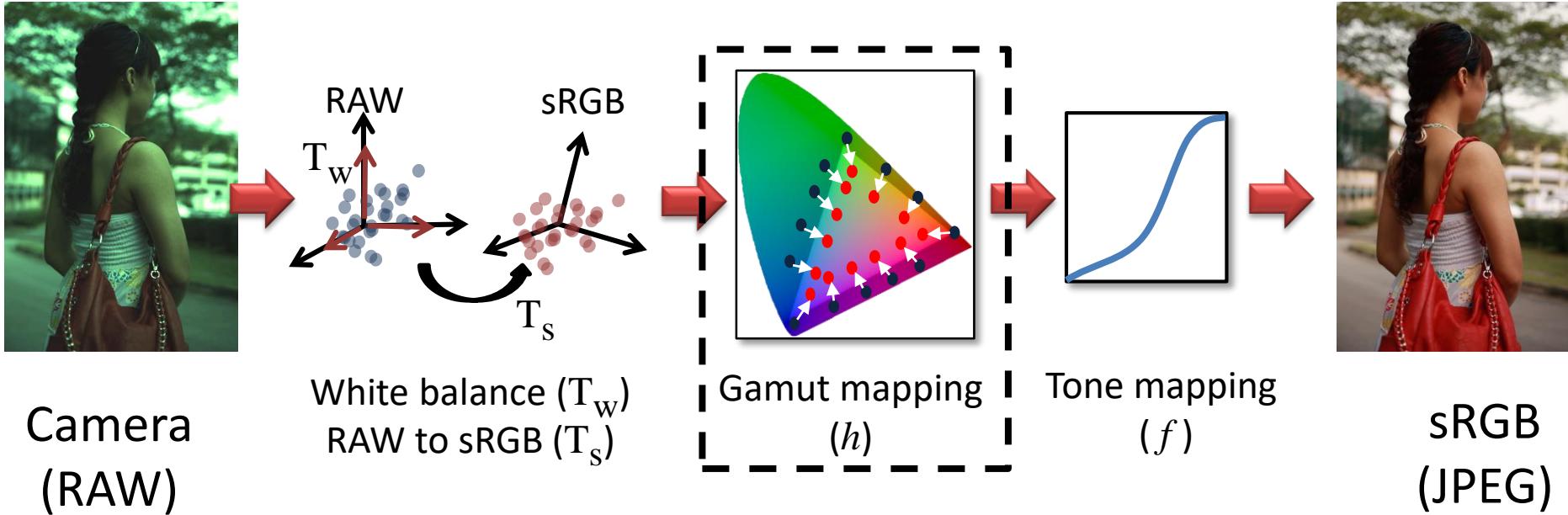
Gamut mapping is necessary because the gamut of the camera's color space is different from the gamut of sRGB.

Gamut mapping is a natural mechanism to support scene modes, such as *vivid* mode, *portrait* mode, *landscape* mode, etc.

# Proposed a new model



# Proposed a new model



Introduce  $h$ , a 3D function that takes in input RGB and maps it to a new RGB value

$$\begin{bmatrix} i_{rx} \\ i_{gx} \\ i_{bx} \end{bmatrix} = f(h \left( \mathbf{T} \begin{bmatrix} E_{rx} \\ E_{gx} \\ E_{bx} \end{bmatrix} \right))$$

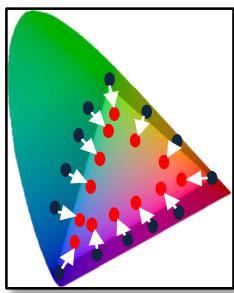
$$h: \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

# sRGB Image to RAW

$$\begin{bmatrix} f^{-1}(i_{rx}) \\ f^{-1}(i_{gx}) \\ f^{-1}(i_{bx}) \end{bmatrix} = h \begin{pmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{pmatrix} = h \left( \mathbf{T} \begin{bmatrix} E_{rx} \\ E_{gx} \\ E_{bx} \end{bmatrix} \right) \rightarrow \begin{bmatrix} E_{rx} \\ E_{gx} \\ E_{bx} \end{bmatrix} = \mathbf{T}^{-1} \cdot h^{-1} \left( \begin{bmatrix} f_r^{-1}(i_{rx}) \\ f_g^{-1}(i_{gx}) \\ f_b^{-1}(i_{bx}) \end{bmatrix} \right)$$

Based on several sRGB-RAW pairs,

- $f^1$  &  $\mathbf{T}^{-1}$  are computed using less saturated points
- $h^{-1}$  is computed with scatter point interpolation via *radial basis func.*

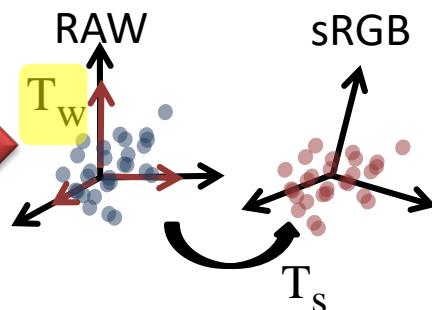
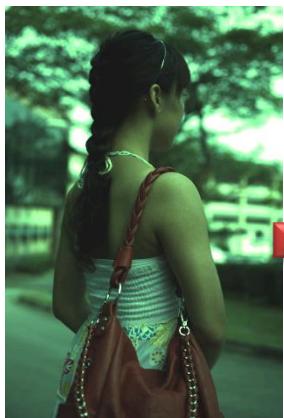
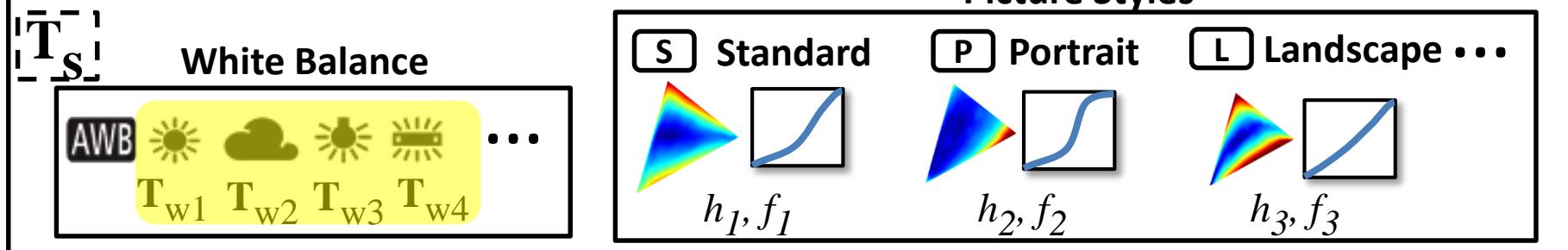


Gamut Mapping  
( $h$ )

Gamut mapping modeled using  
radial basis function

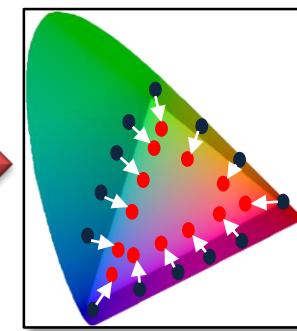
$$h^{-1}(\mathbf{X}) = p(\mathbf{X}) + \sum_{i=1}^N \lambda_i \phi(\|\mathbf{X} - \mathbf{X}_i\|)$$

# Canon EOS1Ds Mark III



Camera  
(RAW)

White balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Tone mapping  
( $f$ )

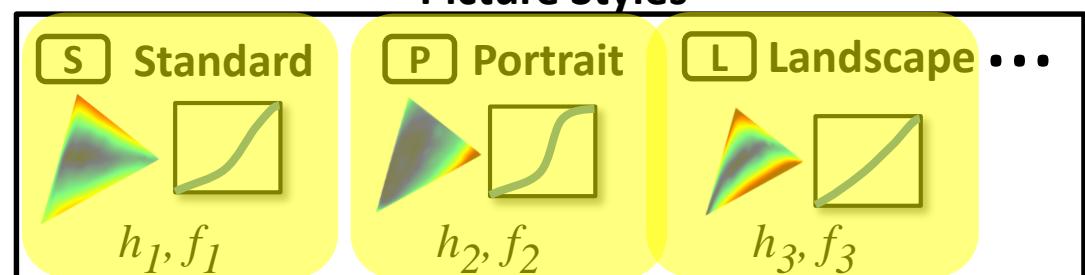
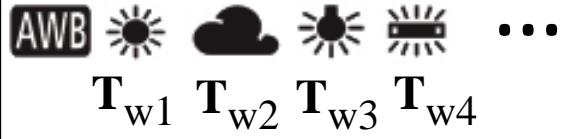


sRGB  
(JPEG)

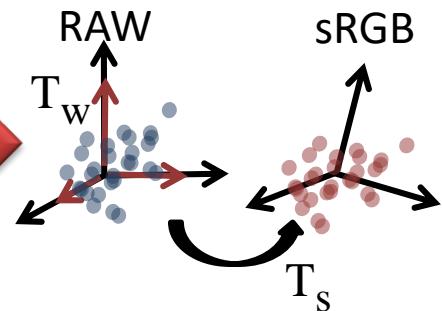
# Canon EOS1Ds Mark III

$T_w$

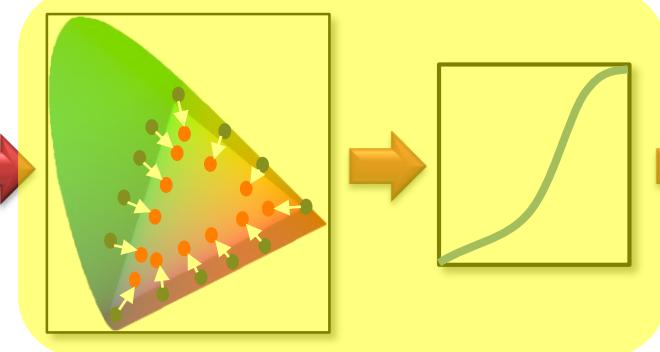
White Balance



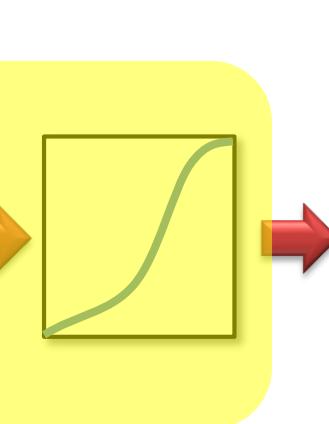
Camera  
(RAW)



White balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut mapping  
( $h$ )

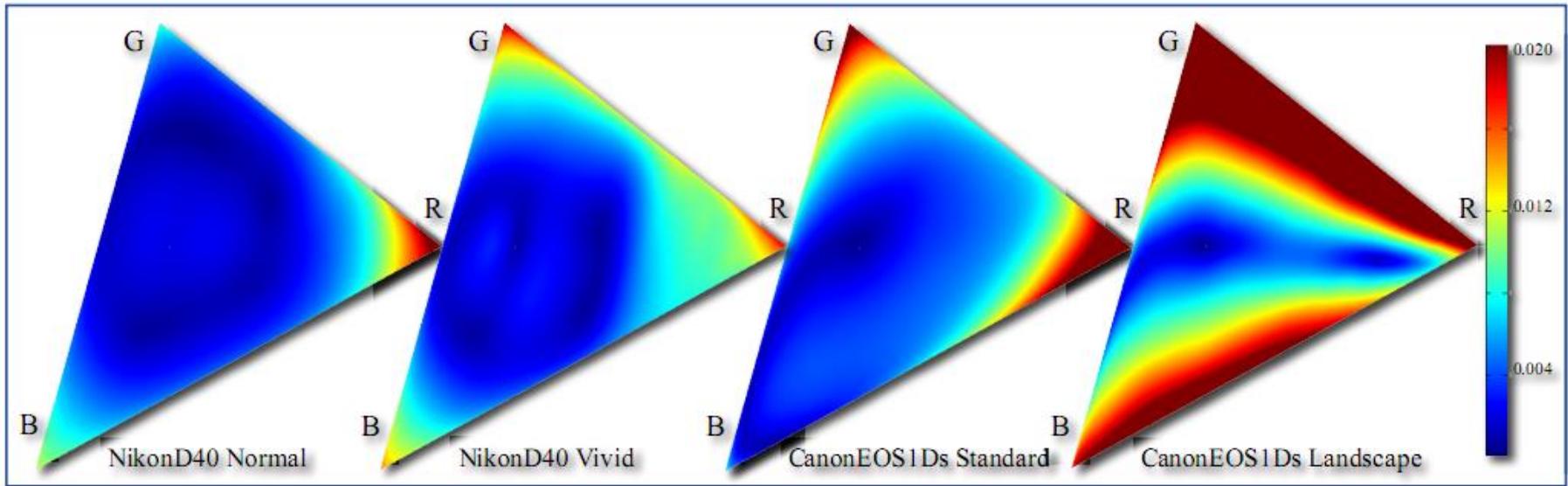


Tone mapping  
( $f$ )

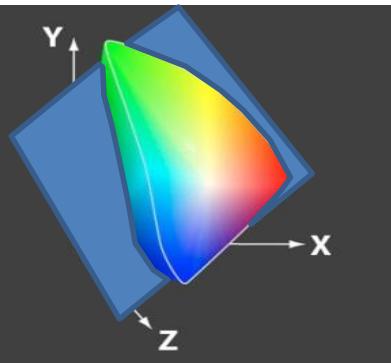


sRGB  
(JPEG)

# Gamut mapping



Mapping is represented as a **displacement map** of the camera's original RGB value to its sRGB location.



The plots above are displacement map of the function  $h$ , projected into a 2D plane. This is intended to help visualize how much deformation to the color is taking place. E.g. we can see in "Landscape style", the green and blue colors are more manipulated for the Canon EOS 1Ds.

# Experiments : Mapping sRGB back to RAW



input sRGB image



ground truth RAW

**Canon EOS1D**

# Experiments : Mapping sRGB back to RAW



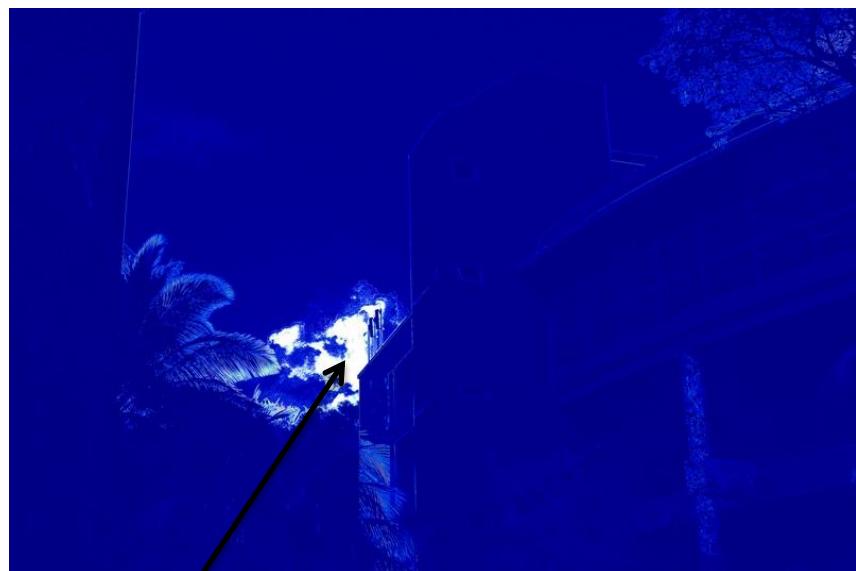
input sRGB image



estimated RAW

**Canon EOS1D**

# Experiments : Mapping sRGB back to RAW



**Our model ( $f, T, h$ )**

We cannot handle  
fully saturated points.



**old model ( $f, T$ )**

**Canon EOS1D**

# Experiments : Mapping sRGB back to RAW



input sRGB image



ground truth RAW

**Canon EOS550D**

# Experiments : Mapping sRGB back to RAW



input sRGB image



estimated RAW

**Canon EOS550D**

# Experiments : Mapping sRGB back to RAW



new model ( $f, T, h$ )

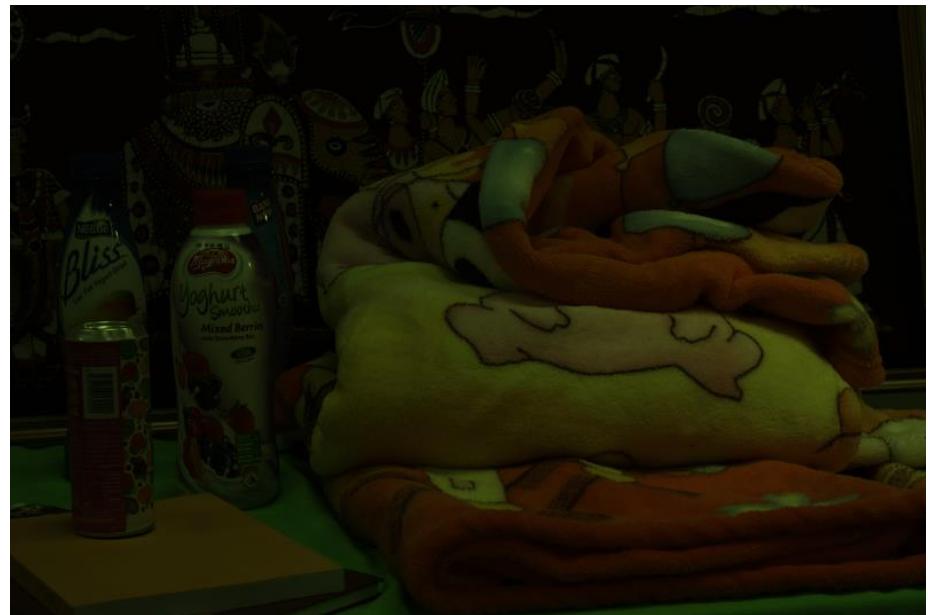
old model ( $f, T$ )

**Canon EOS1D**

# Experiments : Mapping sRGB back to RAW



input sRGB image



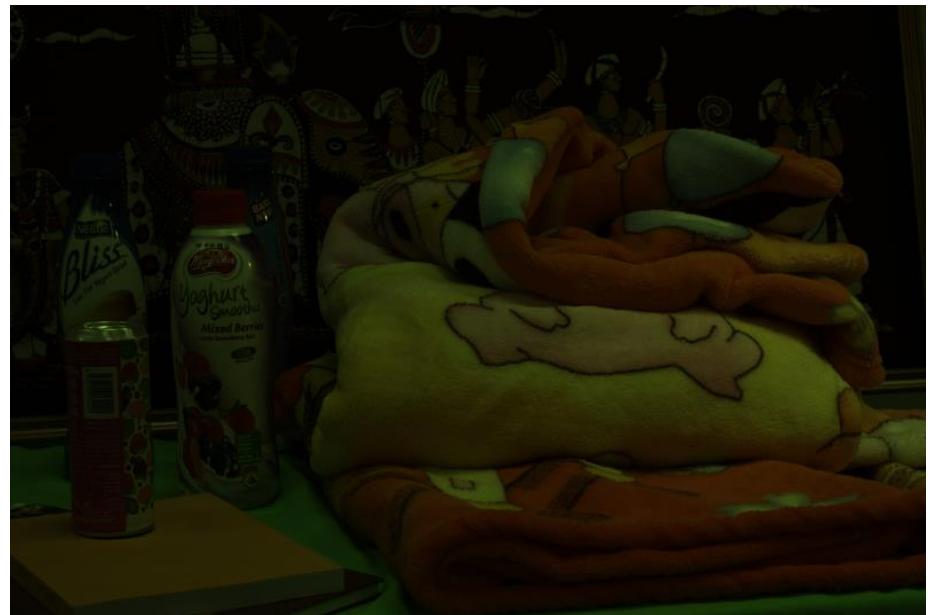
ground truth RAW

**Sony A200**

# Experiments : Mapping sRGB back to RAW



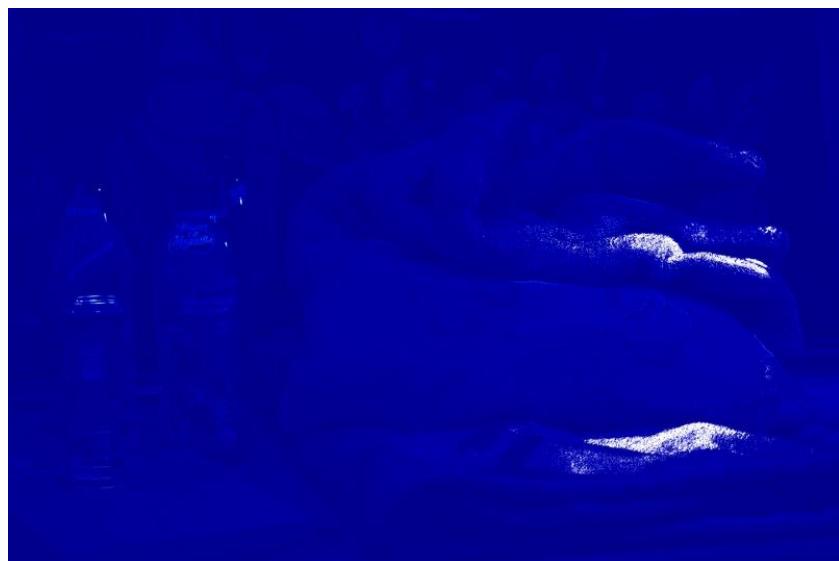
input sRGB image



estimated RAW

Sony A200

# Experiments : Mapping sRGB back to RAW



**Our model ( $f, T, h$ )**



**old model ( $f, T$ )**

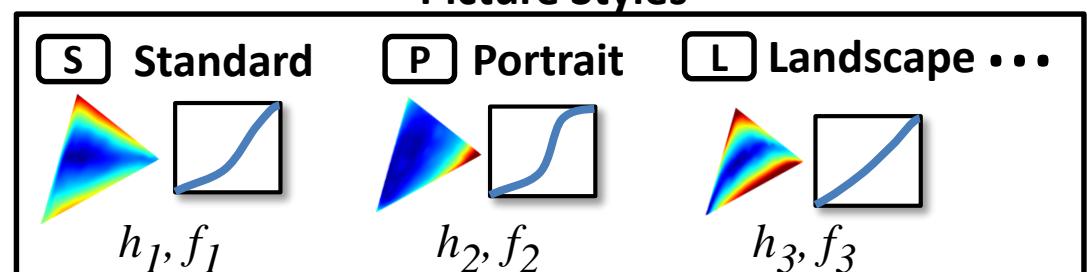
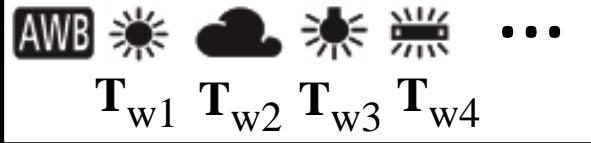
**Sony A200**

# Application: *Photo Refinishing*

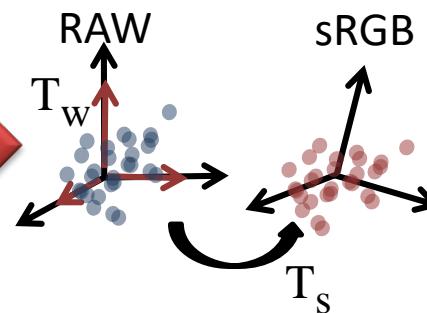
# Canon EOS1Ds Mark III

$T_S$

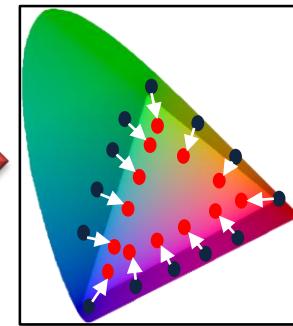
White Balance



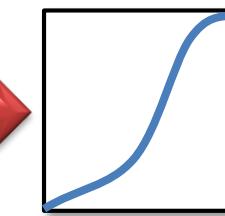
Camera  
(RAW)



White Balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut Mapping  
( $h$ )



Tone Mapping  
( $f$ )

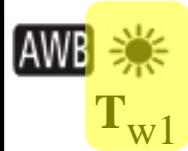


sRGB  
(JPEG)

# Canon EOS1Ds Mark III

$T_S$

## White Balance



AWB



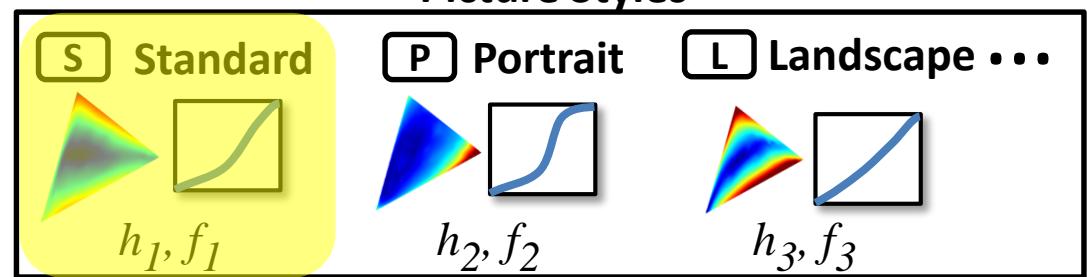
...

$T_{w1}$

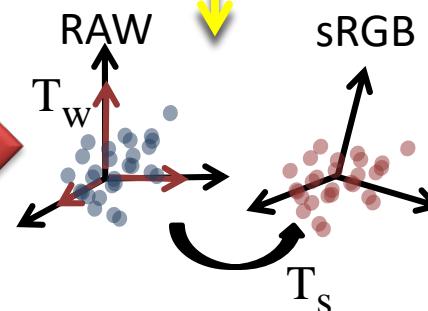
$T_{w2}$

$T_{w3}$

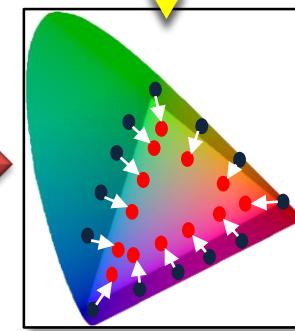
$T_{w4}$



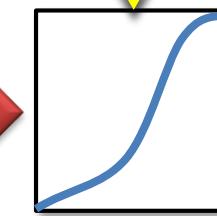
Camera  
(RAW)



White Balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut Mapping  
( $h$ )

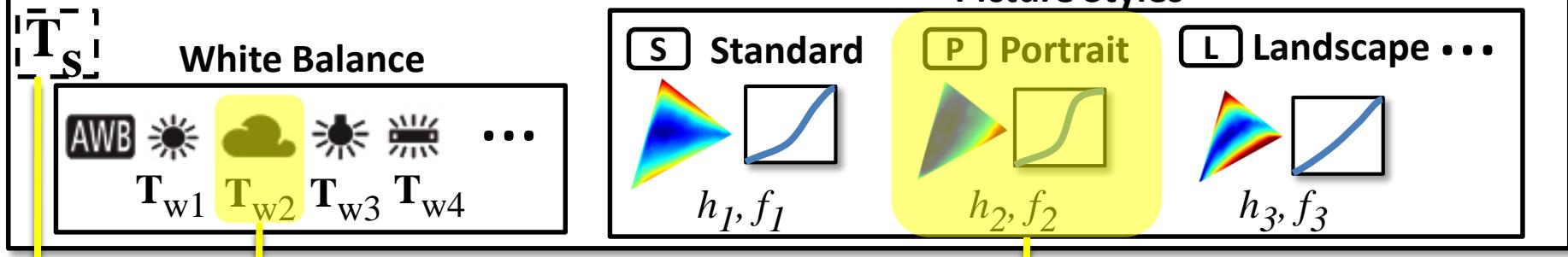


Tone Mapping  
( $f$ )

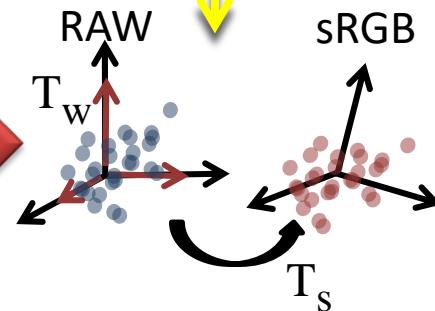


sRGB  
(JPEG)

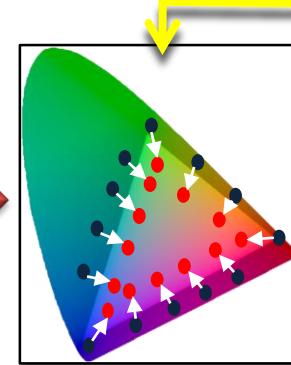
# Canon EOS1Ds Mark III



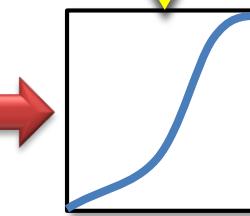
Camera  
(RAW)



White Balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut Mapping  
( $h$ )

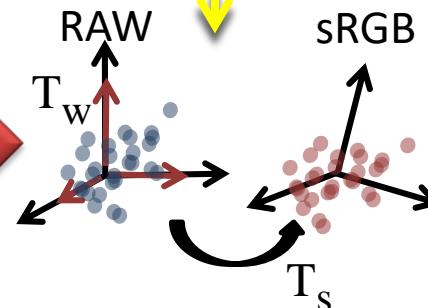
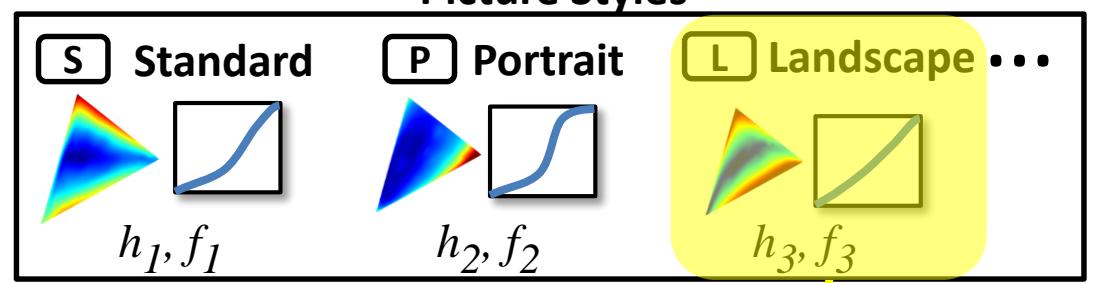
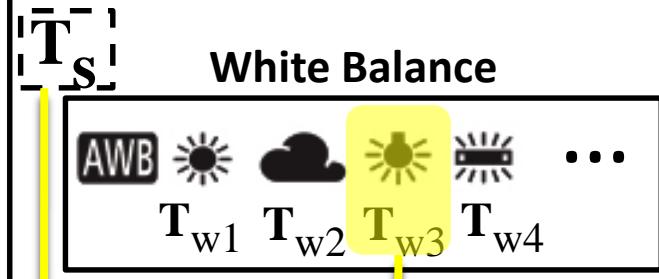


Tone Mapping  
( $f$ )



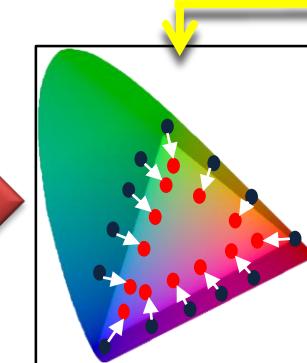
sRGB  
(JPEG)

# Canon EOS1Ds Mark III

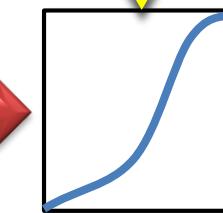


Camera  
(RAW)

White Balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut Mapping  
( $h$ )



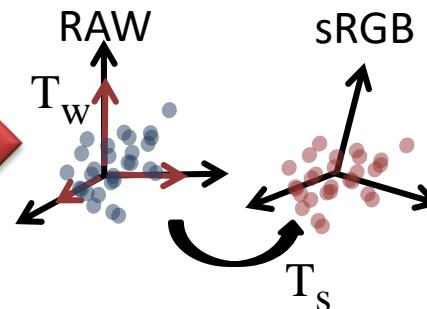
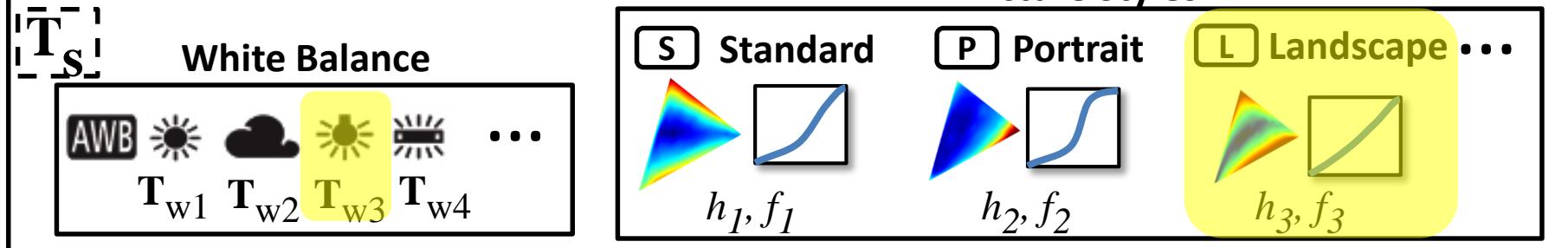
Tone Mapping  
( $f$ )



sRGB  
(JPEG)

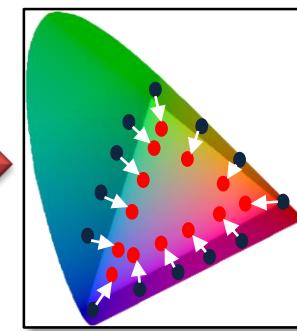
What if you took a photo  
with the wrong settings?

# Canon EOS1Ds Mark III

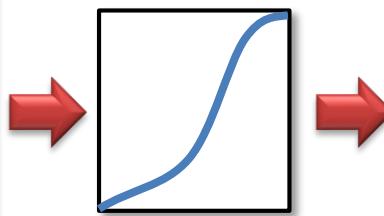


Camera  
(RAW)

White Balance ( $T_w$ )  
RAW to sRGB ( $T_s$ )



Gamut Mapping  
( $h$ )



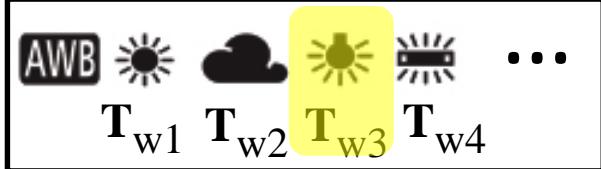
Tone Mapping  
( $f$ )



# Canon EOS1Ds Mark III

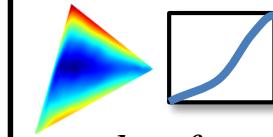
$T_S^{-1}$

## White Balance

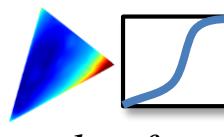


## Picture Styles

**S** Standard



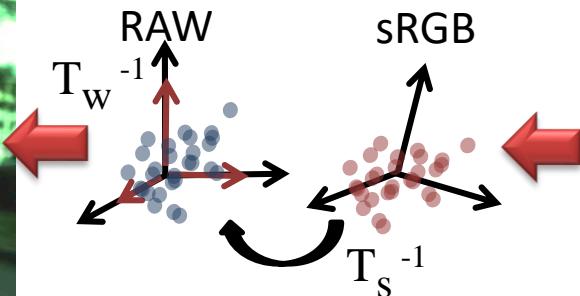
**P** Portrait



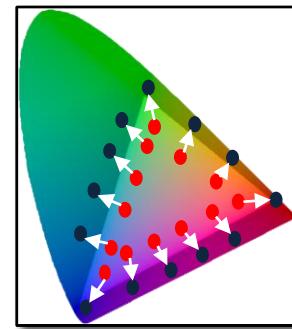
**L** Landscape ...



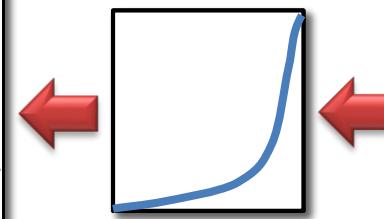
Camera  
(RAW)



White Balance ( $T_w^{-1}$ )  
RAW to sRGB ( $T_s^{-1}$ )



Gamut Mapping  
( $h^{-1}$ )

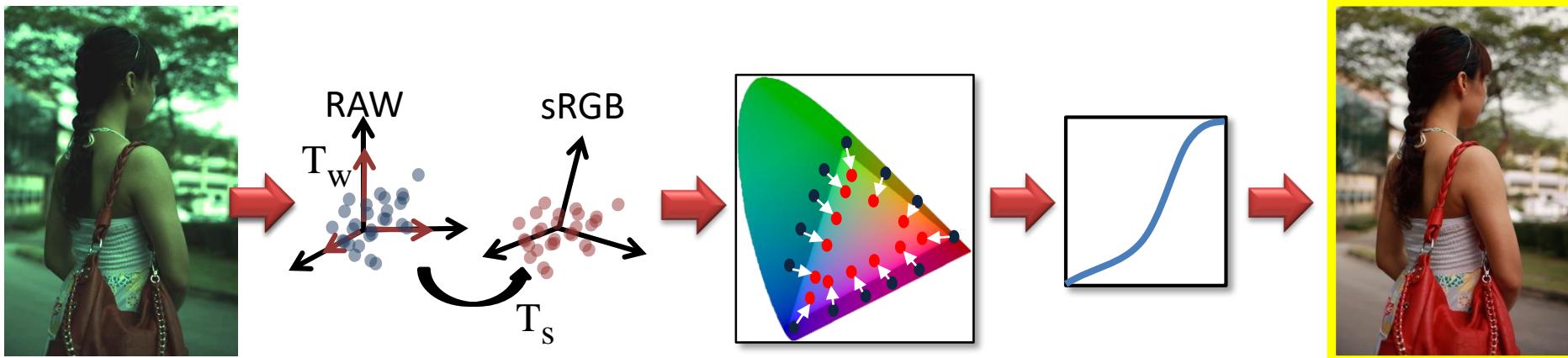
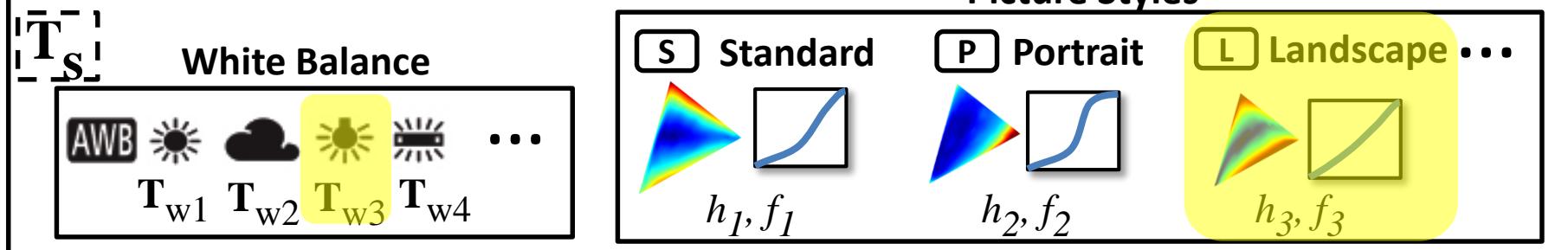


Tone Mapping  
( $f^{-1}$ )

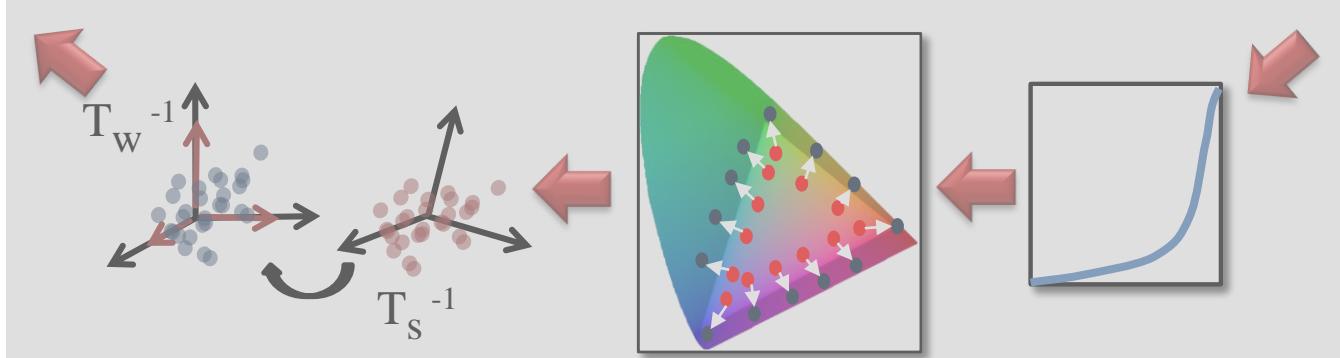


sRGB  
(JPEG)

# Canon EOS1Ds Mark III



Camera  
(RAW)



White Balance ( $T_w^{-1}$ )  
RAW to sRGB ( $T_s^{-1}$ )

Gamut Mapping  
( $h^{-1}$ )

Tone Mapping  
( $f^{-1}$ )

sRGB  
(JPEG)

# Result - Canon EOS 1Ds Mark III



Input: cloudy WB + landscape style

# Result - Canon EOS 1Ds Mark III



Ground truth: fluorescent WB + standard style

# Result - Canon EOS 1Ds Mark III



Photoshop result

# Result - Canon EOS 1Ds Mark III



Refinished result

# Result - Canon EOS 1Ds Mark III



Ground truth: fluorescent WB + standard style

# Result - Canon EOS 1Ds Mark III



Input



Ground truth



Photoshop



Our refined result

# Result - Canon EOS 1Ds Mark III



Input: tungsten WB + standard style

# Result - Canon EOS 1Ds Mark III



Ground truth: daylight WB + standard style

# Result - Canon EOS 1Ds Mark III



Photoshop result

# Result - Canon EOS 1Ds Mark III



Our refinished result

# Result - Canon EOS 1Ds Mark III



Ground truth: daylight WB + standard style

# Result - Canon EOS 1Ds Mark III



Input



Ground truth



Photoshop



Our refinished result

# Result – Nikon D200



Input: tungsten WB + standard style

# Result – Nikon D200



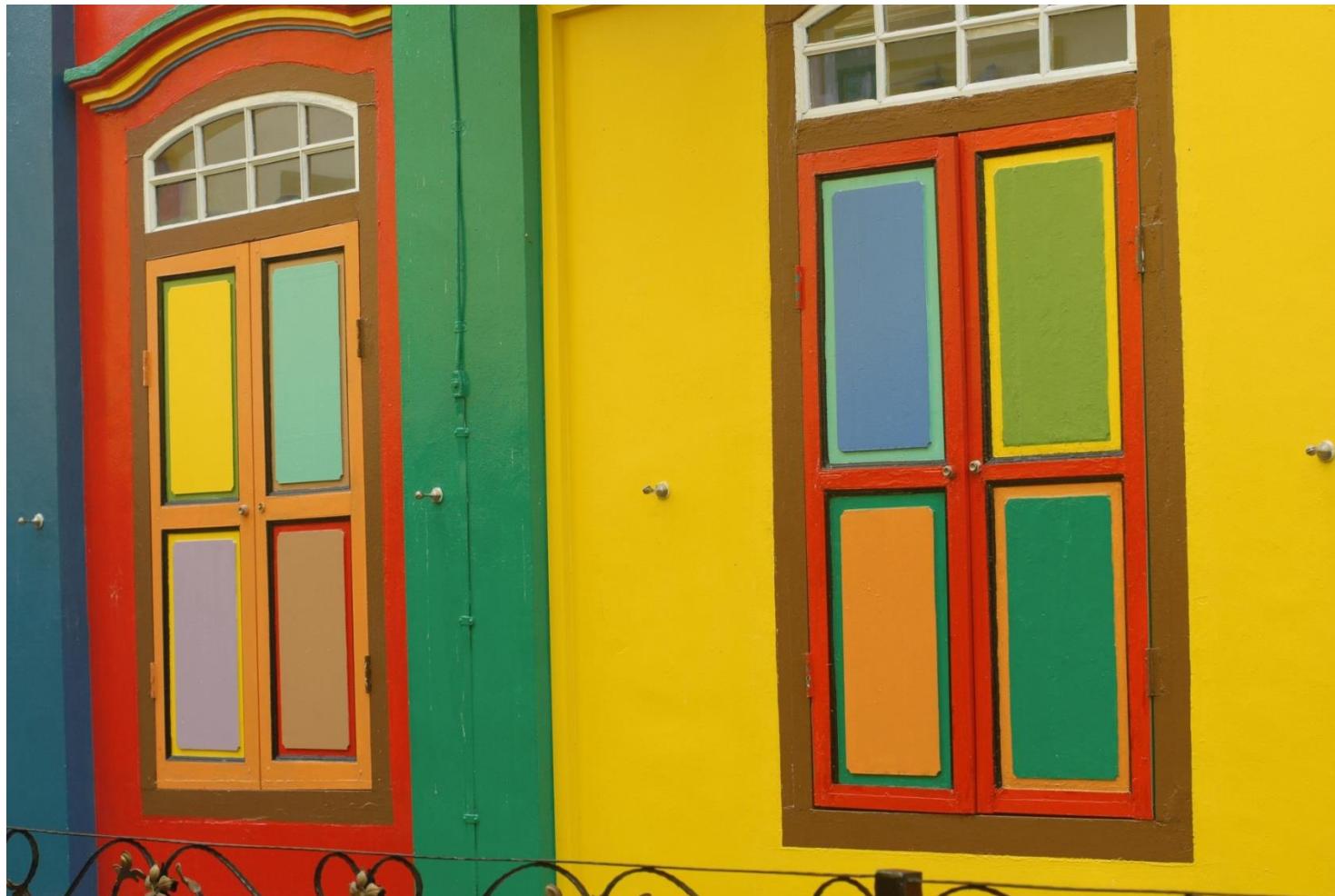
Ground truth: daylight WB + standard style

# Result – Nikon D200



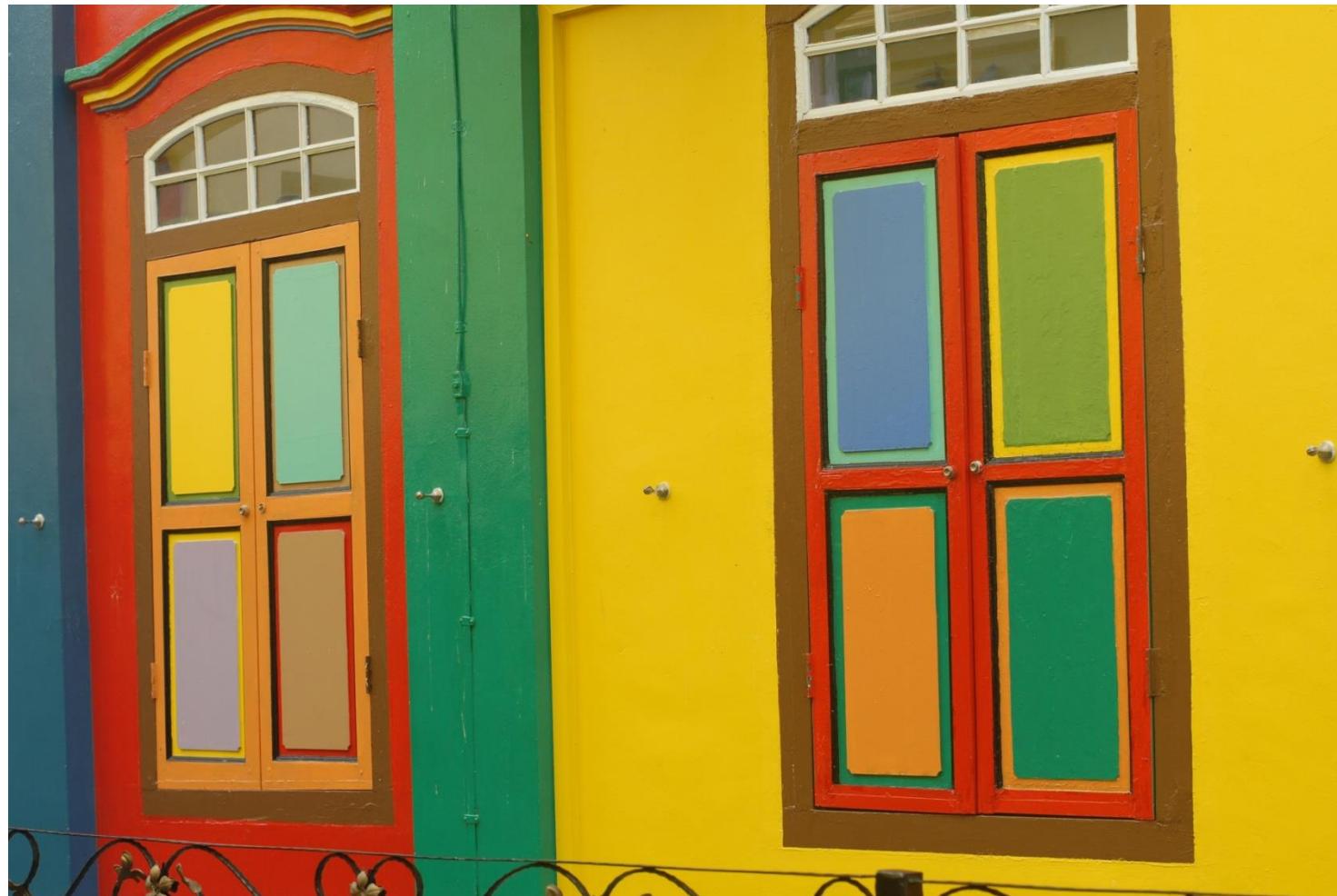
Photoshop result

# Result – Nikon D200



Refinished result

# Result – Nikon D200



Ground truth: daylight WB + standard style

# Result – Nikon D200



Input



Ground truth



Photoshop



Photo refinish

# Result - Sony α200



Input: tungsten WB + standard style

# Result - Sony α200



Ground truth: daylight WB + standard style

# Result - Sony α200



Photoshop result

# Result - Sony α200



Our refinished result

# Result - Sony α200



Ground truth: daylight WB + standard style

# Result - Sony α200



Input



Ground truth



Photoshop



Our Refinished Result

# Remember these guys?

Canon



Nikon



Sony



Nikon



Canon  
↓  
Nikon



Sony  
↓  
Nikon



Sony



Canon  
↓  
Sony



Nikon  
↓  
Sony



# Aside: Probabilistic Approach

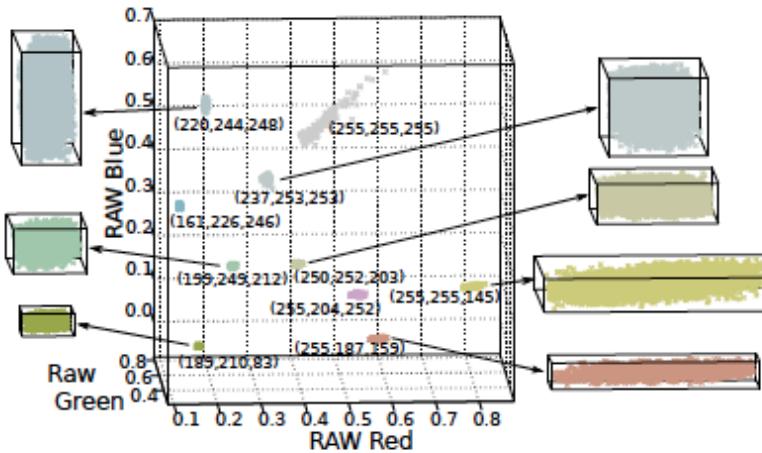


Fig. 1. Clusters of RAW measurements that each map to a single JPEG color value (indicated in parentheses) in a digital SLR camera (Canon EOS 40D). Close-ups of the clusters emphasize the variations in cluster size and orientation. When inverting the tone-mapping process, this structured uncertainty cannot be avoided.

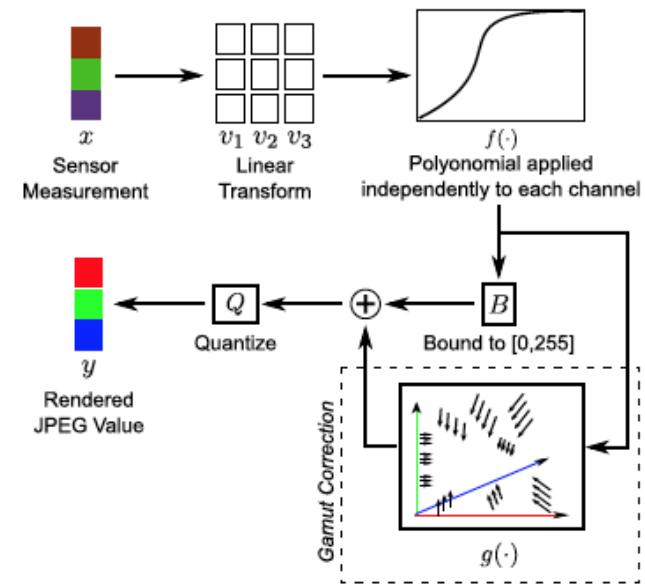


Fig. 3. Rendering Model. We model a camera's processing pipeline using a two step-approach: (1) a  $3 \times 3$  linear transform and independent per-channel polynomial; followed by, (2) a correction to account for deviations in the rendering of saturated and out-of-gamut colors.

The mapping of the function,  $h$ , is not one-to-one. Chakrabarti et al [TPAMI 2014] has a nice paper on a probabilistic approach for this inverse mapping problem.

# Aside: Probabilistic Approach

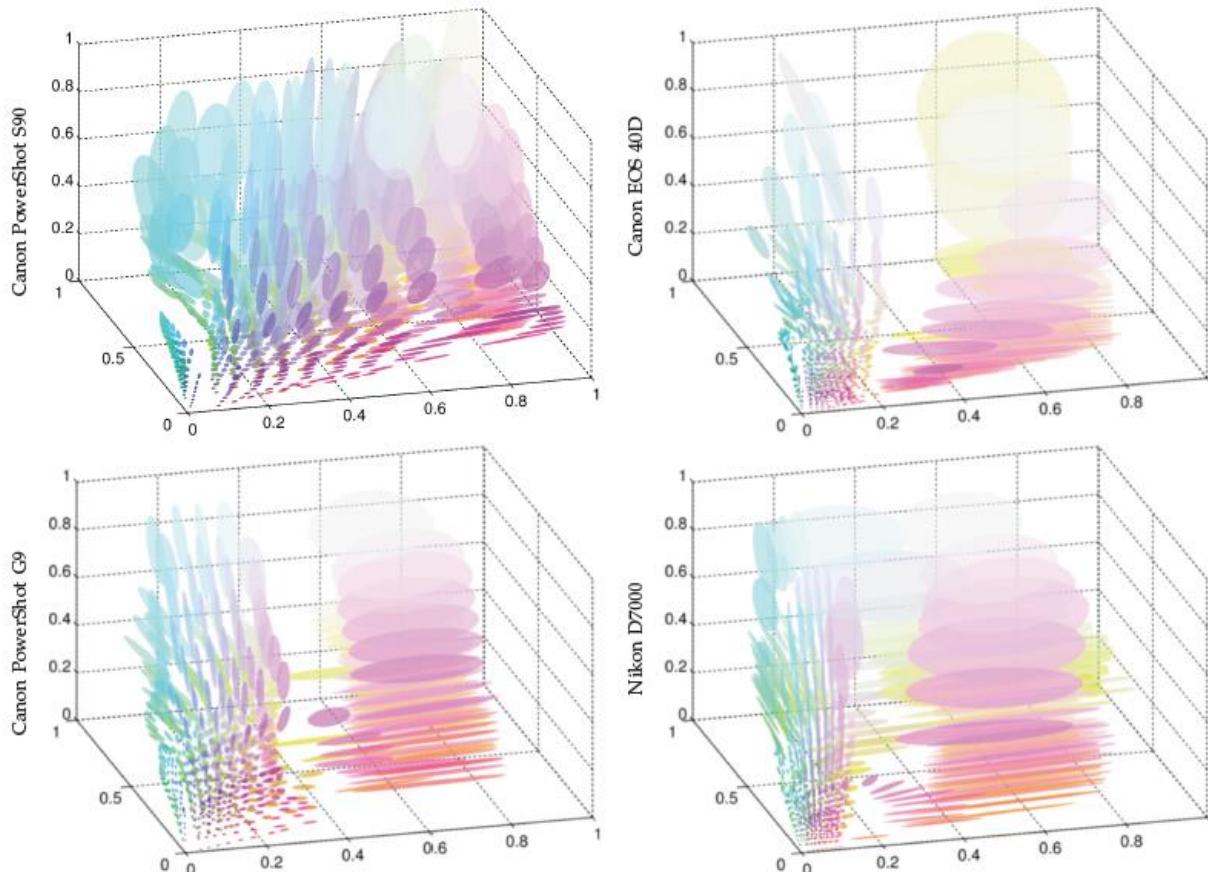


Fig. 5. Probabilistic Inverse. For different cameras, we show ellipsoids in RAW space that denote the mean and covariance of  $p(x|y)$  for different JPEG values  $y$ —indicated by the color of the ellipsoid. These values  $y$  are uniformly sampled in JPEG space, and we note that the corresponding distributions can vary significantly across cameras.

A. Chakrabarti, Y. Xiong, B. Sun, T. Darrell, D. Scharstein, T. Zickler, and K. Saenko,  
"Modeling Radiometric Uncertainty for Vision with Tone-mapped Color Images,"  
*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014

# Tutorial schedule

- Part 1 (General Part)
  - Motivation
  - Review of color/color spaces
  - Overview of camera imaging pipeline
- Part 2 (Specific Part)
  - Modeling the in-camera color pipeline
  - Photo-refinishing
- Part 3 (Wrap Up)
  - The good, the bad, and the ugly of commodity cameras and computer vision research
  - Concluding Remarks

# Part 3: Wrap Up



Image Credit: Timothy Anderson

*The Good, The Bad, and the Ugly* - 1966

Western Movie directed by Sergio Leone

One of the most influential “Western” films and one of the “Great Films of All Times”



THE  
**BAD**

# Color on cameras is not standard

- We have already discussed the “bad” in part 2
- No camera outputs true “sRGB” with respect to the incoming light
- Each camera applies its own color rendering
- Calibration of the camera is required
- While we have good calibration methods, such calibration needs to be done for each camera, and for many different settings (not practical)

# No standard for RAW color space

- RAW is good because it is linear
- However, RAW is a scene-referred color space, specific to the sensor
- This mean RAW RGB values from image on difference cameras of the same scene will be different

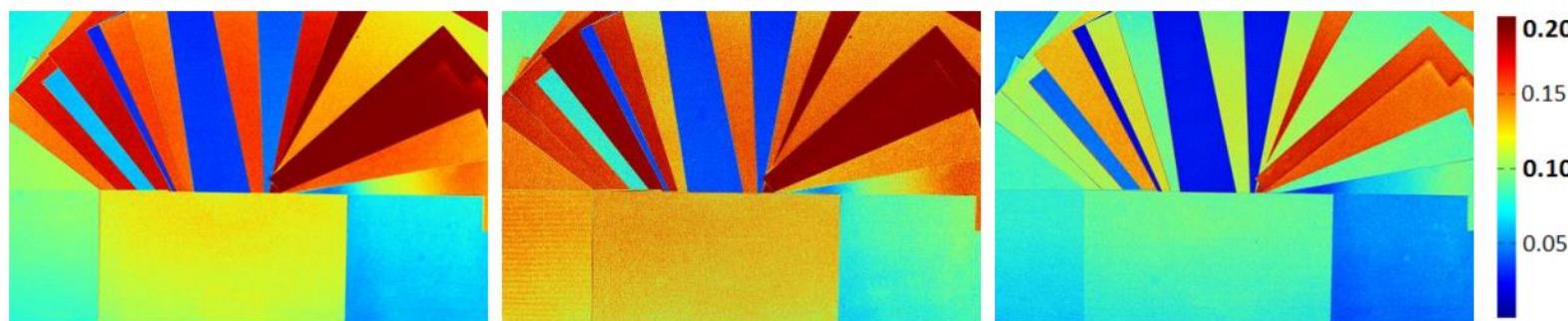
# Example – RAW is not standard



Canon 1D

Nikon D40

Sony  $\alpha$ 57



Canon 1D – Nikon D40

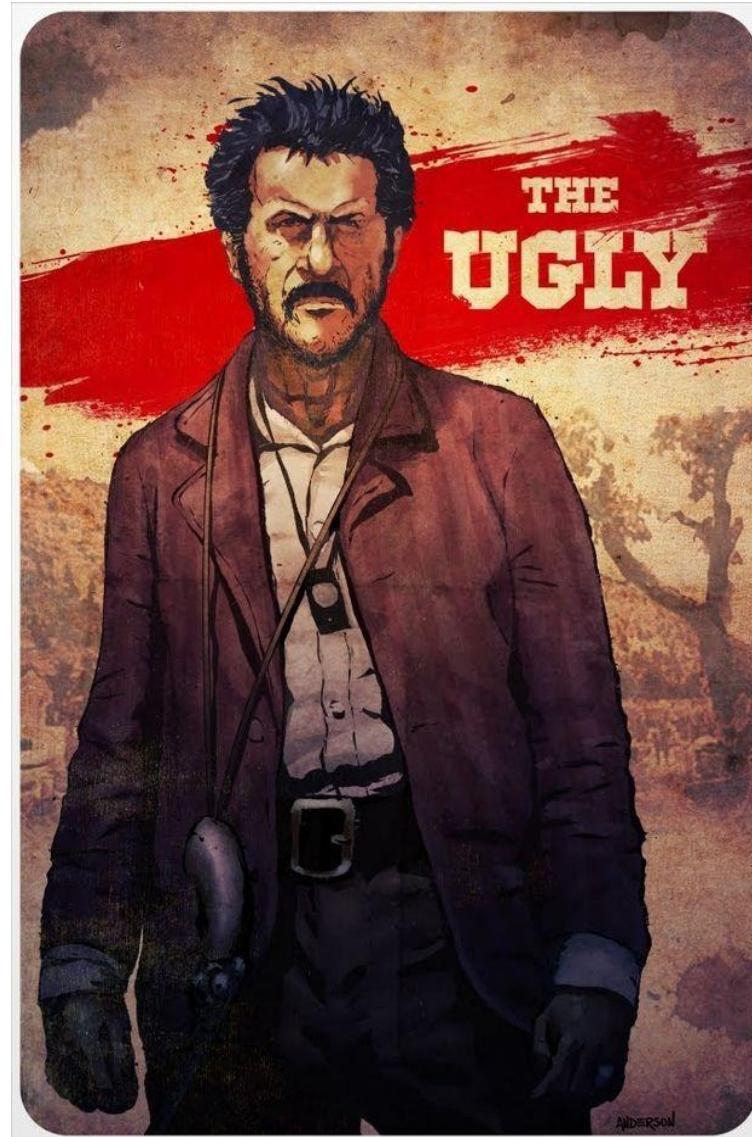
Canon 1D - Sony  $\alpha$ 57

Nikon D40 - Sony  $\alpha$ 57

## Example

Top: RAW images from three cameras, all of the same scene.

Bot: Error plots showing the pixel-wise  $L_2$  difference between camera pairs



# Problems in academic research

1. Lack of understanding of color on cameras and relationship to “real” color spaces
2. Research and results performed “out of context” of the camera processing pipeline
3. Lack of ability to emulate full camera pipelines

# Ugly Example – Color spaces

- Recall our color space transforms
- Camera images are saved in sRGB

Linear-sRGB back to XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Linear-NTSC back to XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.6071 & 0.1736 & 0.1995 \\ 0.2990 & 0.5870 & 0.1140 \\ 0.0000 & 0.0661 & 1.1115 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

# Ugly examples

The conversion of RBG to YCbCr is done by the equation given as in equation (2):

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= B - Y \\ Cr &= R - Y \end{aligned} \quad (2) \quad \left. \right\}$$

The skin segmentation step thus employed exploits the 2D chromatic subspace to reduce the dependence of illumination. A skin color map is derived and used on the chrominance components of the input image to detect pixels that are of skin color. According to the authors the most suitable ranges of Cb and Cr that can

# Ugly examples

scaling by a different factor. This can be achieved easily with the following formula:

$$Y(x, y) = 0.299 R(x, y) + 0.587 G(x, y) + 0.114 B(x, y), \quad (1)$$

$$U(x, y) = 0.492 (B(x, y) - Y(x, y)), \text{ and} \quad (2)$$

$$V(x, y) = 0.877 (R(x, y) - Y(x, y)). \quad (3)$$

In our approach, the color image is first transformed to YUV color space. However, the proposed texture feature is based on the DCT coefficients that are calculated only from the Y (luminance) component. The main reason for this decision is that human visual system is more sensitive to Y than to other chrominance components.

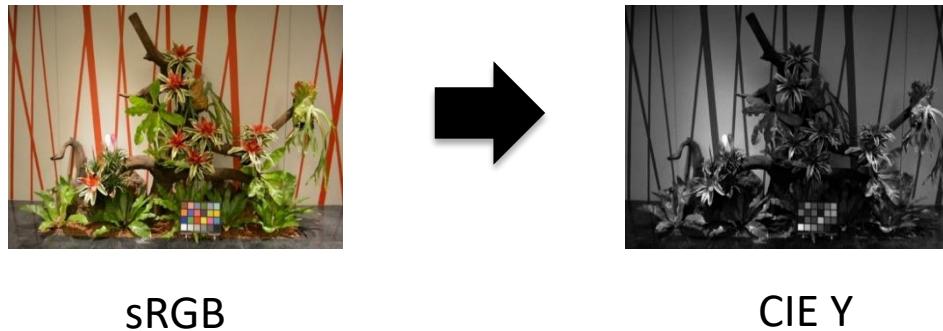
# Ugly examples

## II. ALGORITHM FOR SKIN DETECTION

The Figure 4 illustrates the algorithm steps. This was implemented using the language C and openCv library. First, the image in RGB was converted to HSV color space, because it is more related to human color perception [1]. The skin in channel H is characterized by values between 0 and 50, in the channel S from 0.23 to 0.68 for Asian and Caucasian ethnics [2]. In this work we used images from different Caucasian people, from different places of the world. Figure 1 illustrates one of the original image processed.

# Analyzing ugly

- One of the most common examples in the academic literature – obtaining the luminance (CIE Y) channel from an sRGB image



- Goal is to often *purported* to find the imaged scene's “brightness” (i.e. CIE Y)

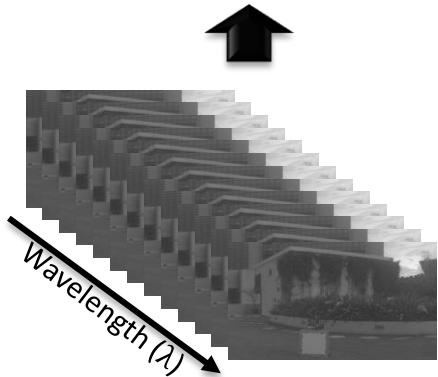
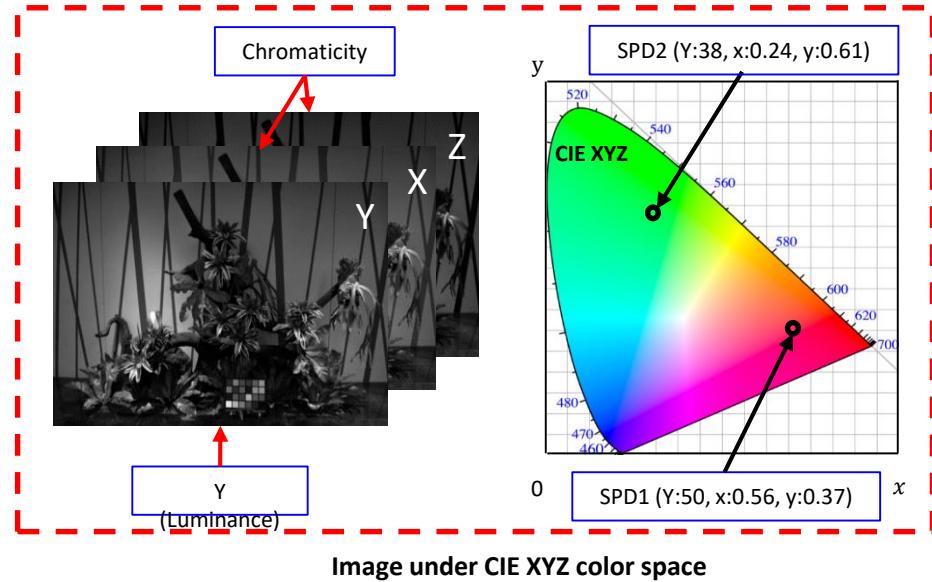
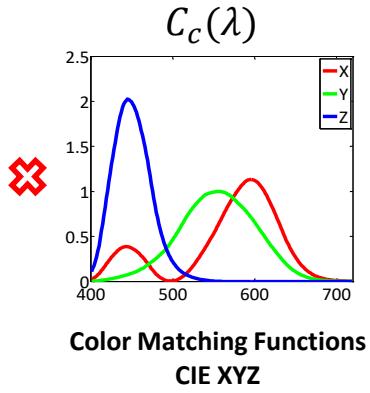
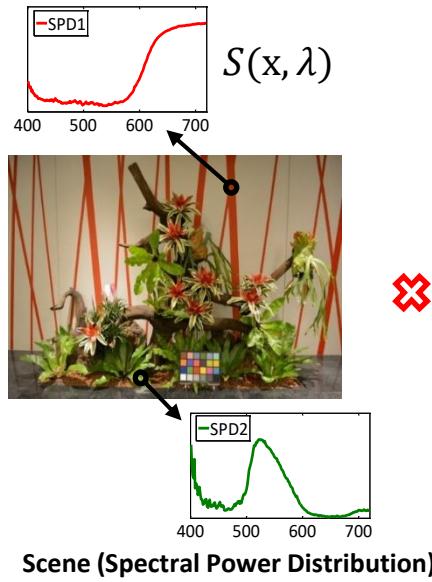
# Analyzing sRGB to luminance ( $Y$ )



- 1) Perform experiments to see just how accurate sRGB to luminance conversion
- 2) Examine the common mistakes made in the academic literature

# Experimental setup

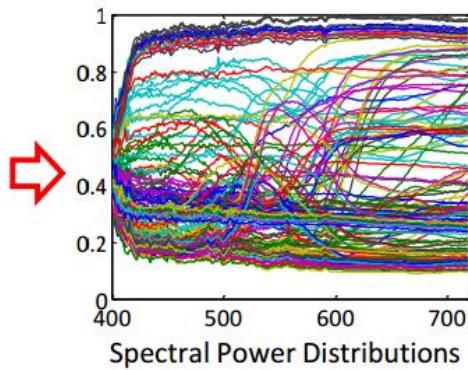
Generate “true CIE XYZ” images (ground truth)



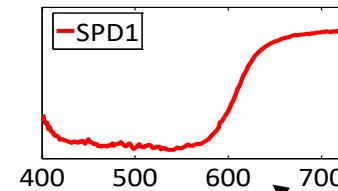
# Test on two types of inputs



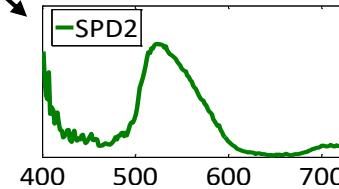
Color Checker



Spectral Power Distributions



$$S(x, \lambda)$$

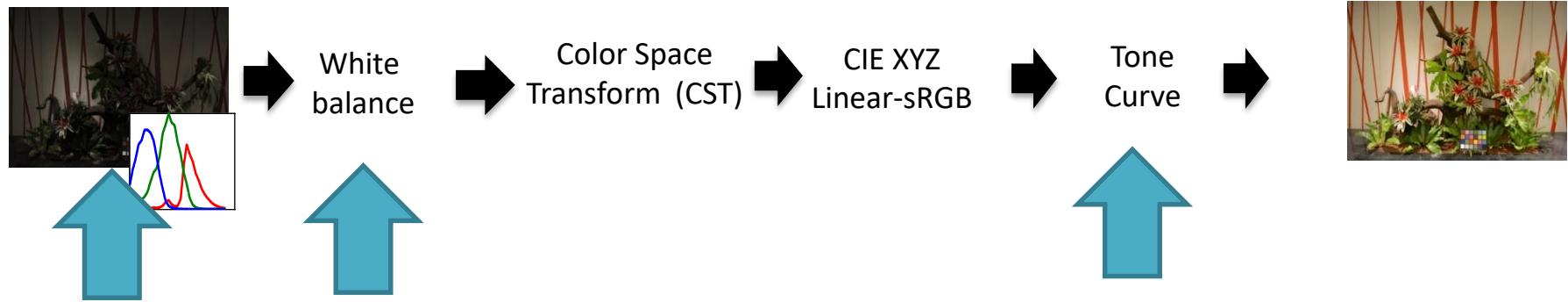


Color rendition chart

Full scenes

# Generate synthetic camera images

Sensor Image (RAW)



Simulate different cameras using sensitivity functions

Generate images with different white balance settings on camera

Generate images with camera-specific tone-curves

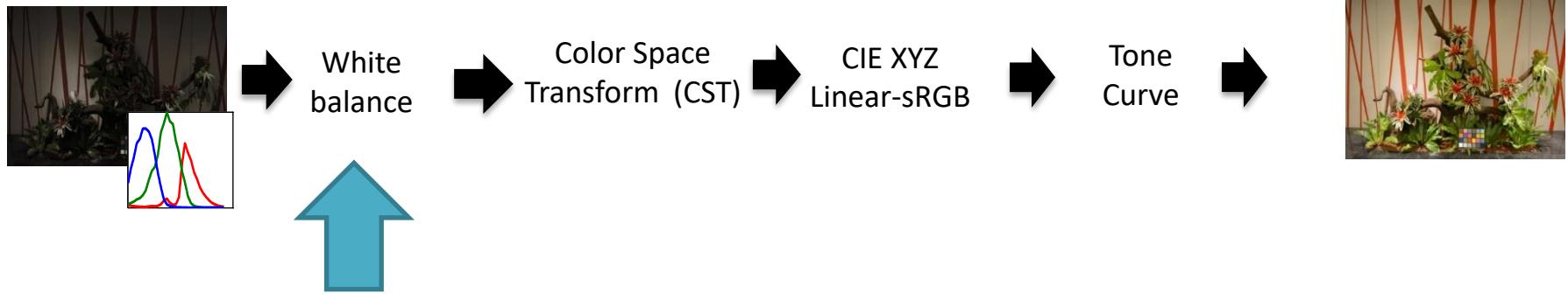
# Three common mistakes in the literature

(when attempting to map a sRGB camera image to CIE – Y)

- Assumption that white balance is correct
- Using the wrong equations (and calling it CIE Y)
- Not applying the correct linearization step

# 1. White balance assumption

Sensor Image (RAW)



There is an implicit assumption  
that the white balance was  
estimated correctly

## 2. Wrong Equations

NTSC is used instead of sRGB (often under the guise of YUV/YIQ)

$$Y = 0.299R + 0.587G + 0.114B$$

Correct eq ( $Y = 0.2126R + 0.7152G + 0.0722B$ )

Average of RGB is used

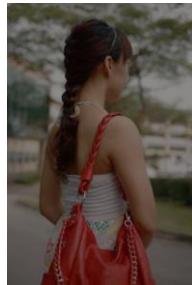
$$Y = \frac{1}{3}(R + G + B)$$

Hue, Saturation, Value (HSV) - Value is taken to equate to scene Luminance

$$Y = \max(R, G, B)$$

# 3. Linearization is incorrect

Sensor Image (RAW)



White-  
balance

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

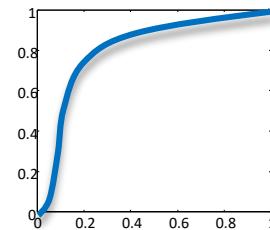
Color Space  
Transform (CST)

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

$$\begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$

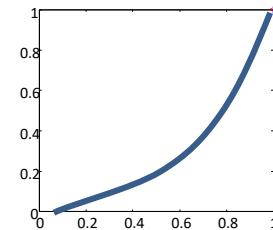


CIE XYZ  
Linear-sRGB



Camera-specific  
tone-curve

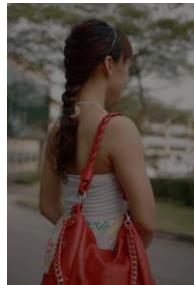
Apply inverse  
equations



Wrong tone-curve  
(sRGB gamma)

# 3. Linearization is not applied

Sensor Image (RAW)



White-  
balance

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

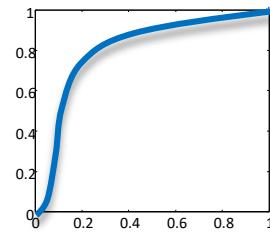
Color Space  
Transform (CST)

$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$



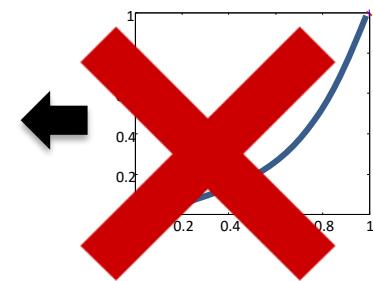
CIE XYZ  
Linear-sRGB

$$\begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$



Camera-specific  
tone-curve

Apply inverse  
equations



Wrong tone-curve  
(sRGB gamma)

### 3. No linearization – “Luma”

Ignoring linearization has a name. It is called Luma instead of Luminance.  
**Variable Y' is used to distinguish it from Luminance Y.**

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

$$Y' = \frac{1}{3}(R' + G' + B')$$

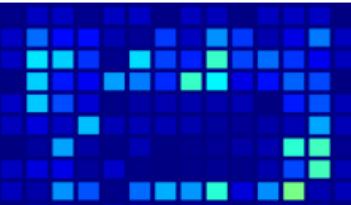
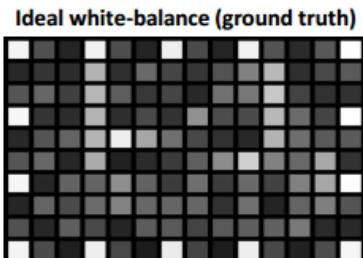
$$Y' = \max(R', G', B')$$

# How ugly is it?

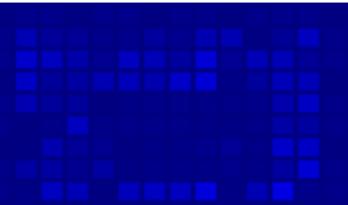
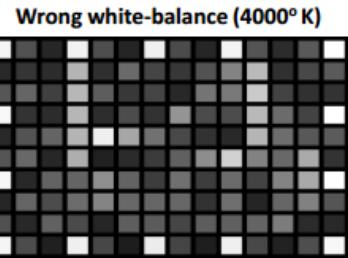
# White balance not correct

CIE – Y

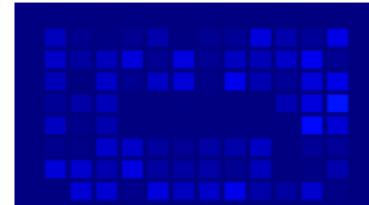
White-balance



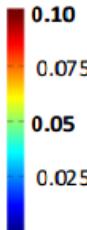
Max: 0.0491, Mean: 0.0115, Std: 0.0120



Max: 0.0097, Mean: 0.0024, Std: 0.0024



Max: 0.0144, Mean: 0.0030, Std: 0.0036

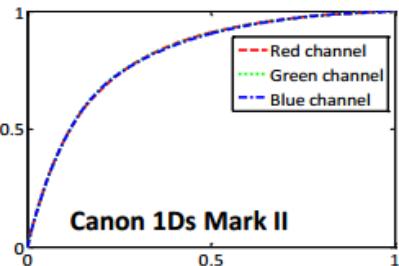


Nikon camera – white balance is wrong, but we have ideal CST, sRGB gamma

Overall error less than 1.5%

# Wrong tone-curve (right equation)

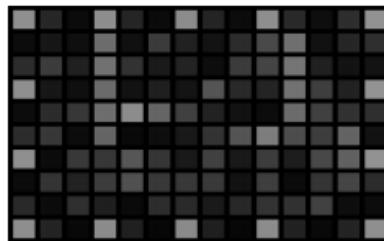
Camera tone-curves



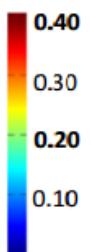
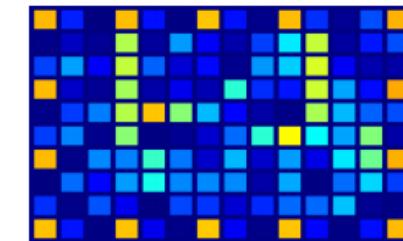
Ground Truth Y



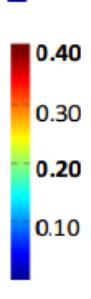
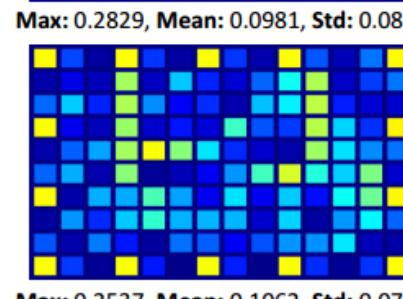
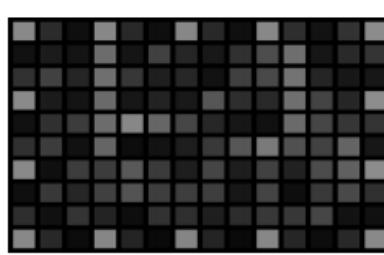
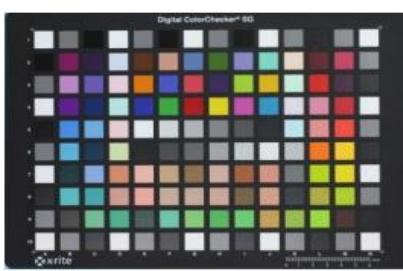
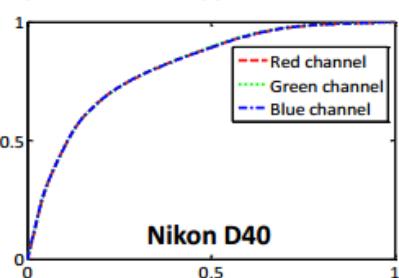
Result with using sRGB gamma



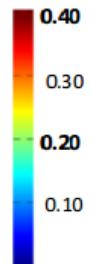
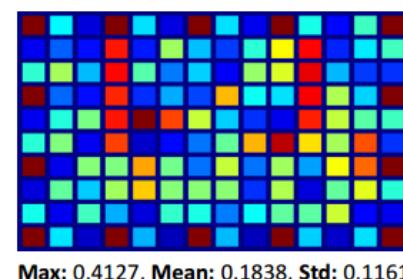
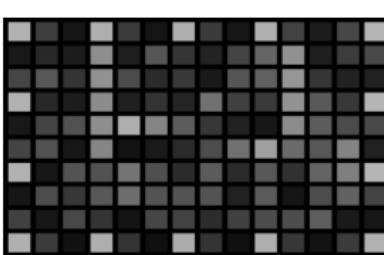
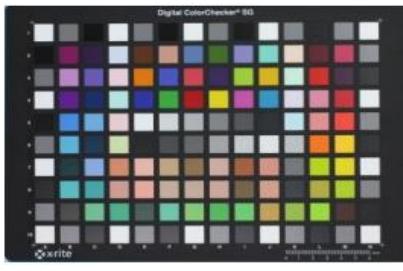
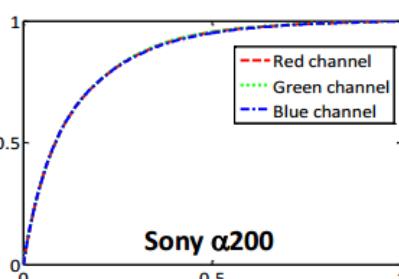
Error Map



Nikon D40



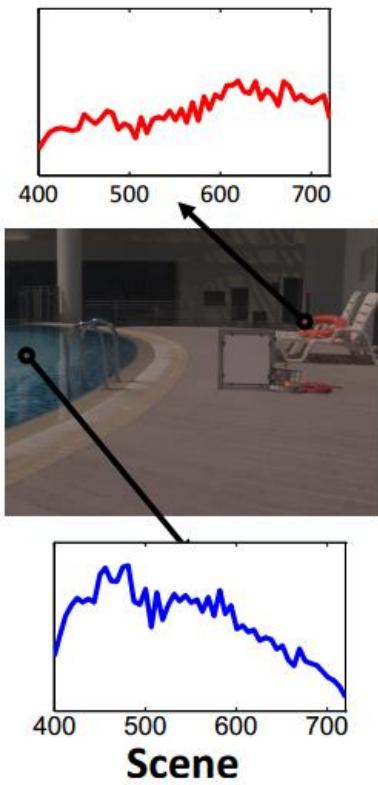
Sony α200



Errors for some colors can be over 25-40%

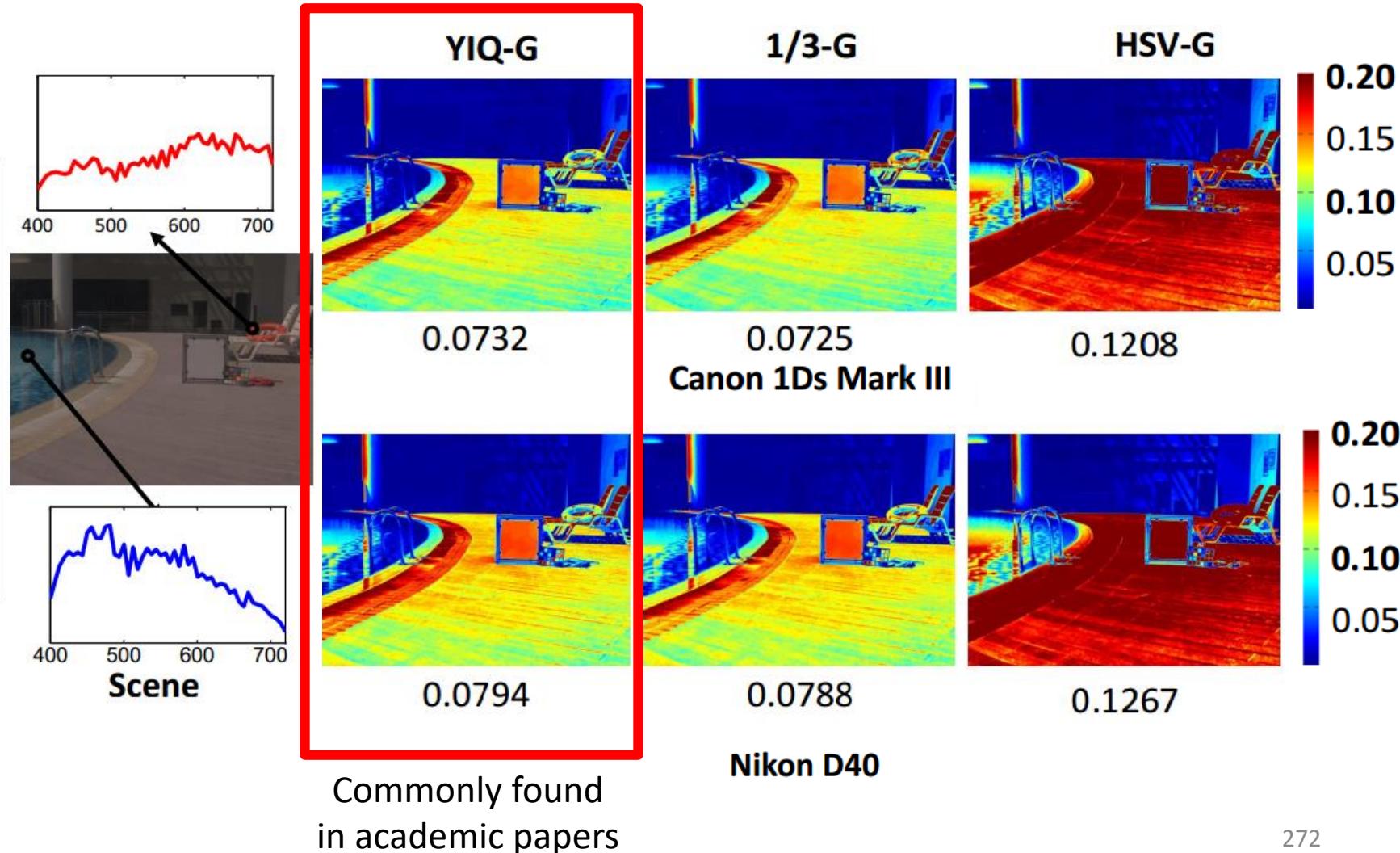
# Wrong equations (right tone-curve)

Luminance Conversion



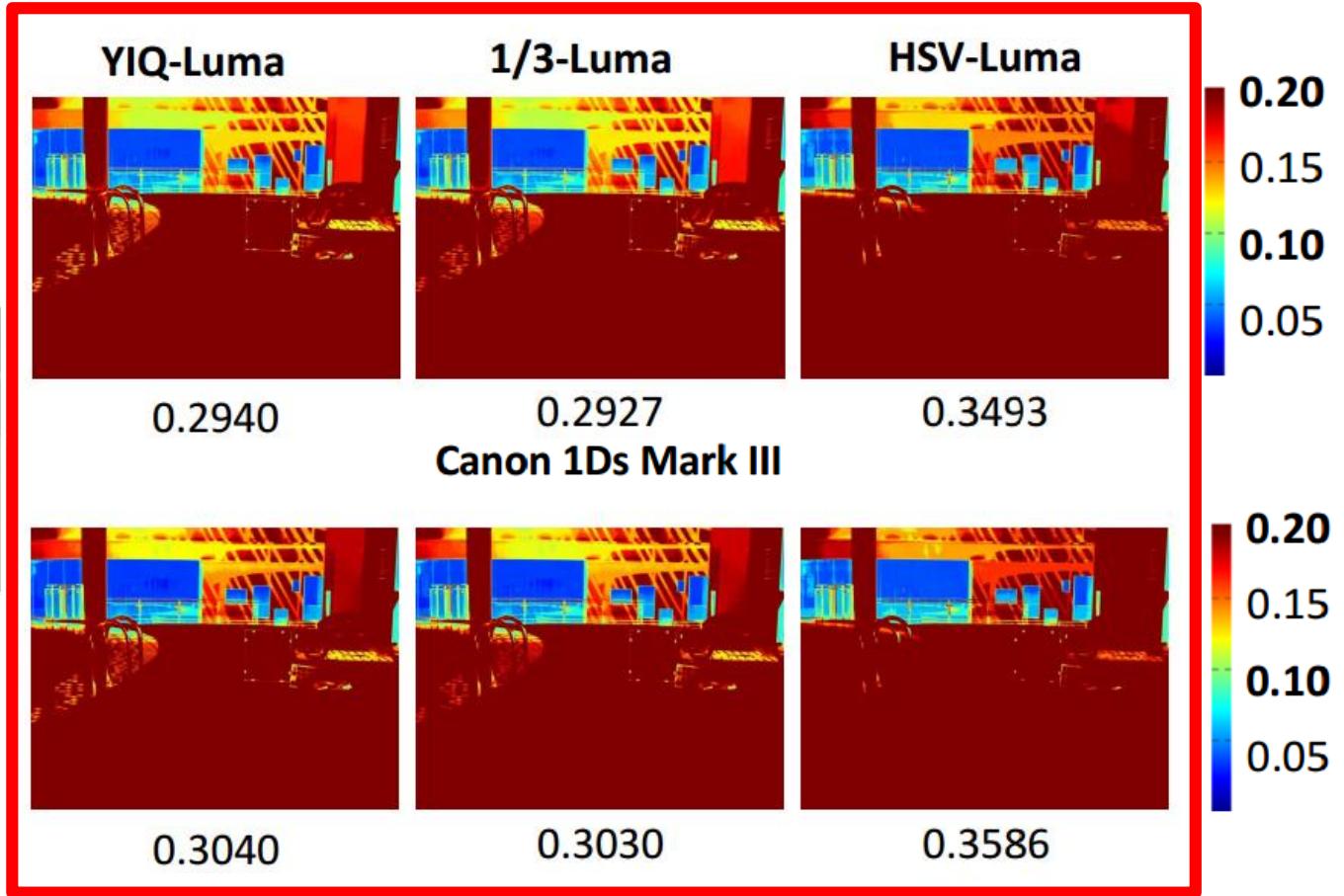
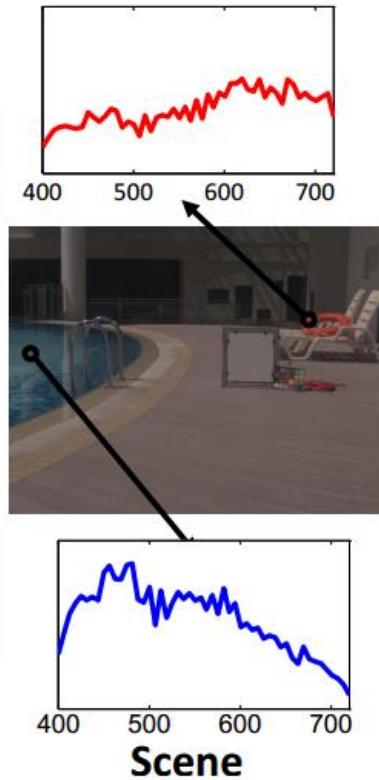
# Wrong equations (wrong tone-curve)

Luminance Conversion



# Wrong equations (no linearization)

Luminance Conversion

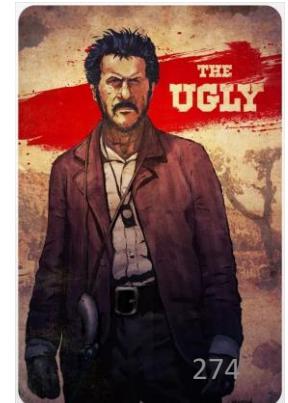


Nikon D40

All very common  
in academic papers<sup>273</sup>

# Ugly analysis

- White balance assumption violation not too serious
- Incorrect tone-curve is significant (25-40% error)
- No tone-curve (Luma) is the worse (over 40% error)
- HSV is the worst of the “wrong” equations, but not worse than using the incorrect tone-curve



# Why bother with CIE-Y?



(a) sRGB image



(b) Y of YIQ



(c) RGB to Gray\*

**SIFT**



(d) SIFT features on (b)

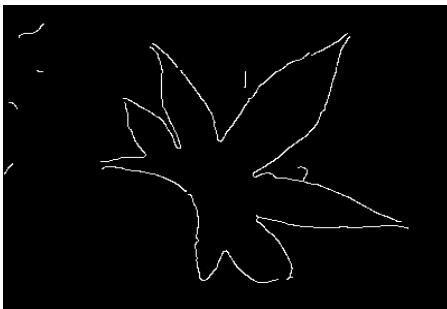


(e) SIFT features on (c)

**Canny**



(f) Canny edges on (b)



(g) Canny edges on (c)

# More examples . .

(a) sRGB image



(b) Y of YIQ



(c) Grayscale proposed in IJCV'14



(d) SIFT features on (b)



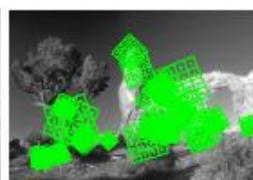
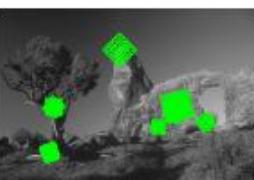
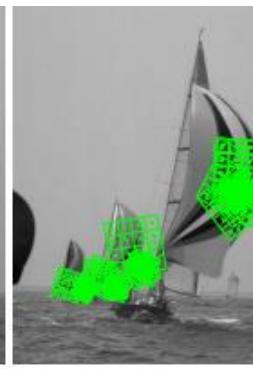
(e) SIFT features on (c)



(f) Canny edges on (b)

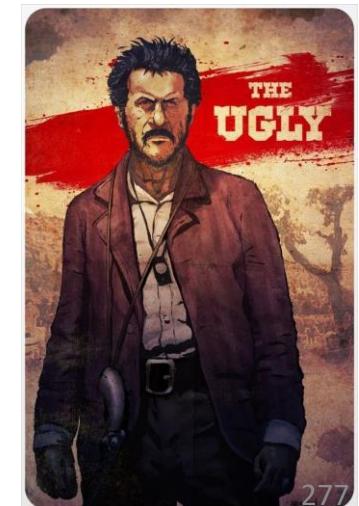


(g) Canny edges on (c)

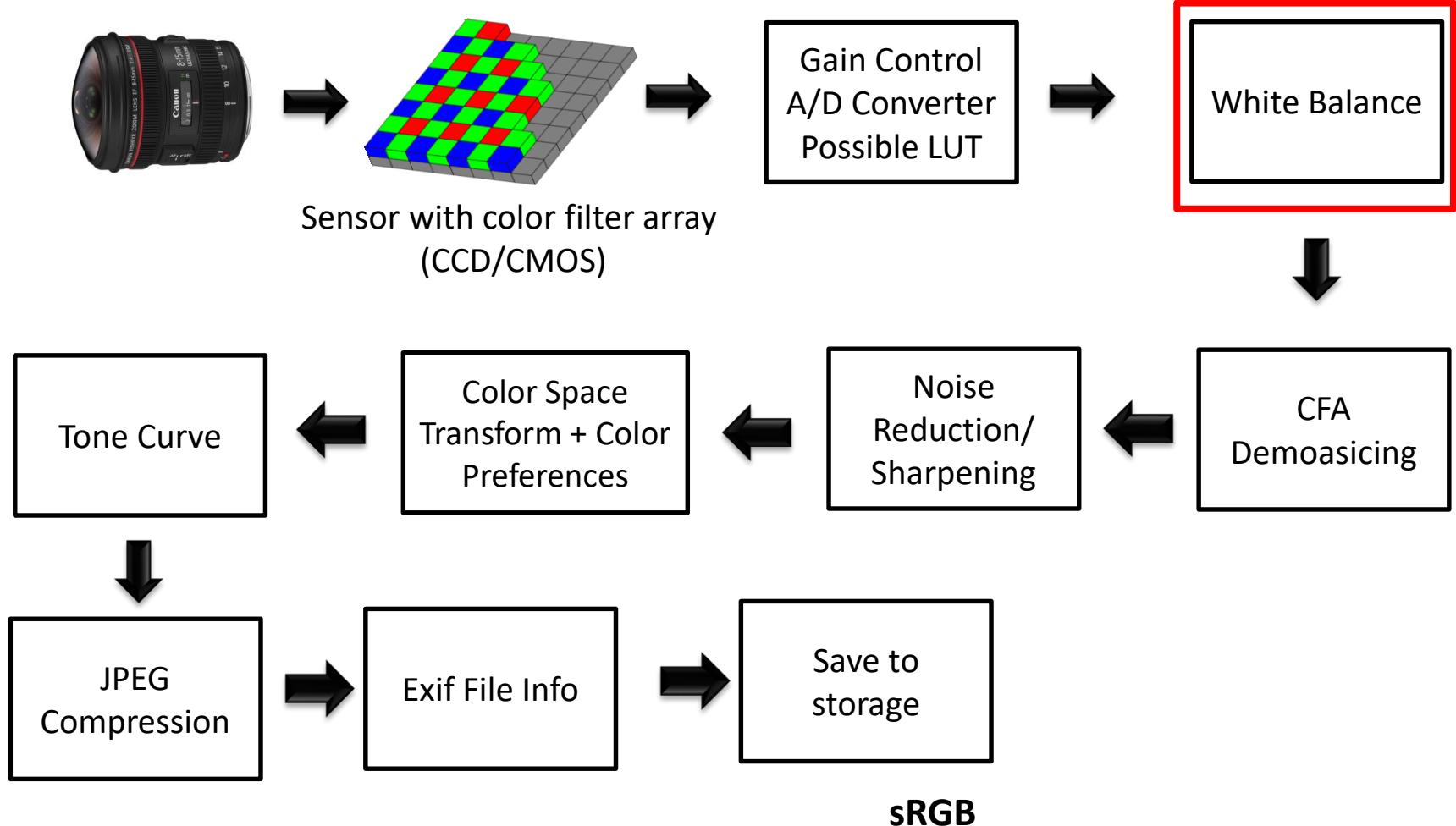


# Another ugly problem . . .

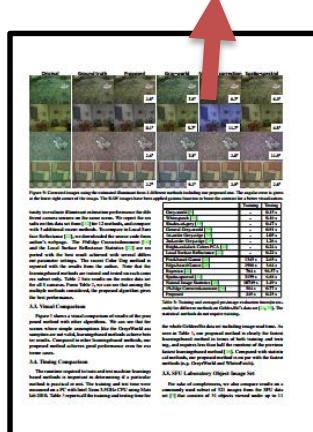
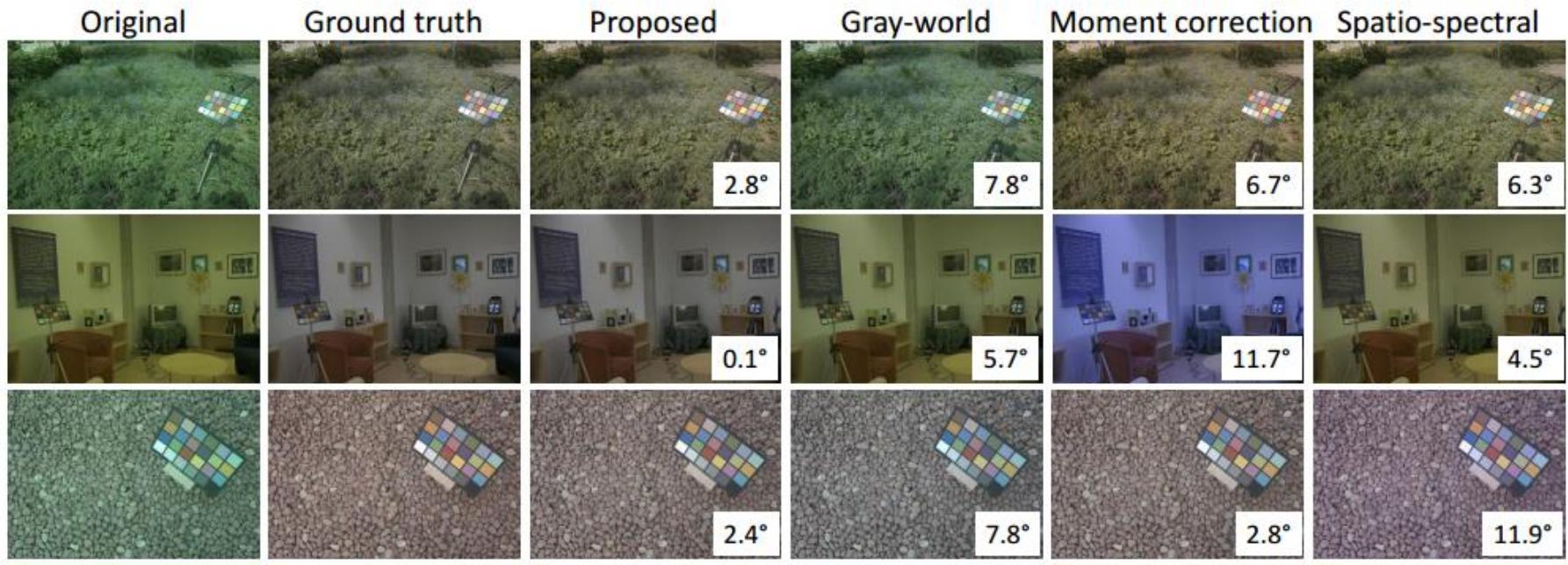
- Applying operations in the “wrong” context . . .
- E.g. applying white balance on sRGB images . . .
- Or, denoising an sRGB image. . .
- Or, deblurring an sRGB image . . .



# Camera processing pipeline



# Classic white balance results out-of-context



“Subjective” white balance results shown  
(images are in camera-raw color space)

These subjective results have **absolutely no** visual meaning.  
The camera-raw is not a standard color space!

# Consider image deblurring

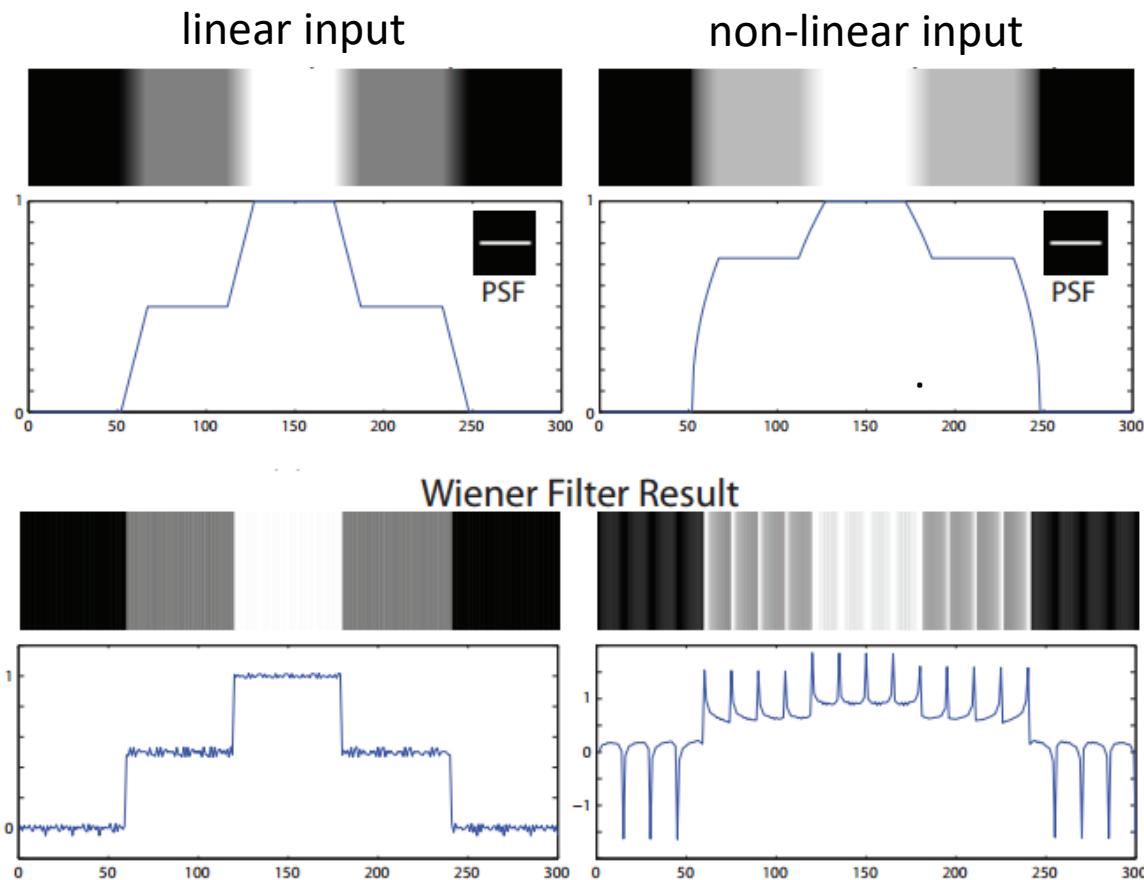
$$\begin{matrix} \text{[Image Block]} & \text{[Image Block]} & = & \text{[Image Block]} & \text{[Image Block]} & \otimes & - \\ B & & & I & & & K \end{matrix}$$

**Assumption is:** linear color space (RAW or linear-sRGB)

$$\begin{matrix} \text{[Image Block]} & \text{[Image Block]} & = & f( \text{[Image Block]} & \text{[Image Block]} & \otimes & - ) \\ B & & & I & & & K \end{matrix}$$

**Reality:** image has been run through the pipeline and some non-linear tone-curve  $f$

# Gamma/tone-curve effect on blur



Tai Y.-W., Chen X., Kim S., Kim S. J., Li F., Yang J., Yu J., Matsushita Y., Brown M. S. (2013) "Nonlinear Camera Response Functions and Image Deblurring: Theoretical Analysis and Practice", *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(10), Oct 2013

# RAW vs sRGB deblurring

Canon 600D



Input with blur



Ground Truth



Deblur on RAW



Deblur on sRGB

# State of affairs (The Ugly)

- Many researchers don't understand camera color
- Attempt to treat sRGB images as true scene measurements (ideal sRGB images)
  - Often use wrong equations, forget linearization, etc. .
  - Results can be erroneous by up to 50%
  - Why even attempt CIE Y?
- Research is often applied without understanding the context of the color processing pipeline





raw image



Web

Images

Shopping

Videos

News

More ▾

Search tools

About 1,020,000,000 results (0.68 seconds)

## Raw image format - Wikipedia, the free encyclopedia

[https://en.wikipedia.org/wiki/Raw\\_image\\_format](https://en.wikipedia.org/wiki/Raw_image_format) ▾ Wikipedia ▾

A camera raw image file contains minimally processed data from the image sensor of either a digital camera, image scanner, or motion picture film scanner.

Rationale - File contents - Processing - Software support

You've visited this page 2 times. Last visit: 10/11/15

### People also ask

What is a RAW image? ▾

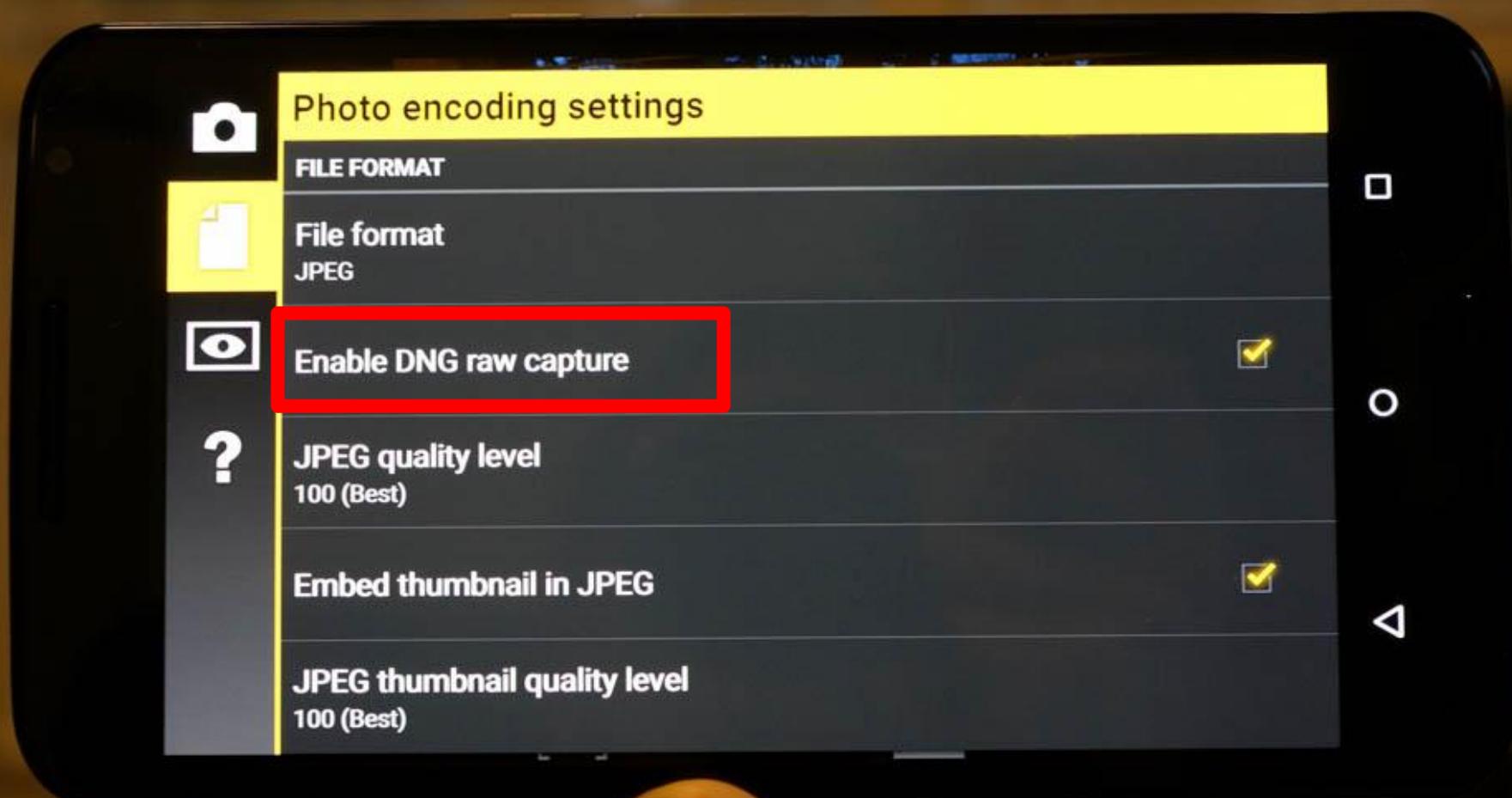
What is a RAW image file? ▾

What is a Camera Raw file? ▾

## 10 Reasons Why You Should Be Shooting RAW

[photographyconcentrate.com/10-reasons-why-you-should-be-shooting-r...](http://photographyconcentrate.com/10-reasons-why-you-should-be-shooting-r...) ▾

You've probably heard over and over that you should be shooting in RAW. But do you know why it's so important? And what it really means for your images?



# Adobe Digital Negative (DNG)

- Public raw-camera image file specification
- Open source SDK for processing DNG to sRGB
- After almost 10 years, this is becoming mainstream

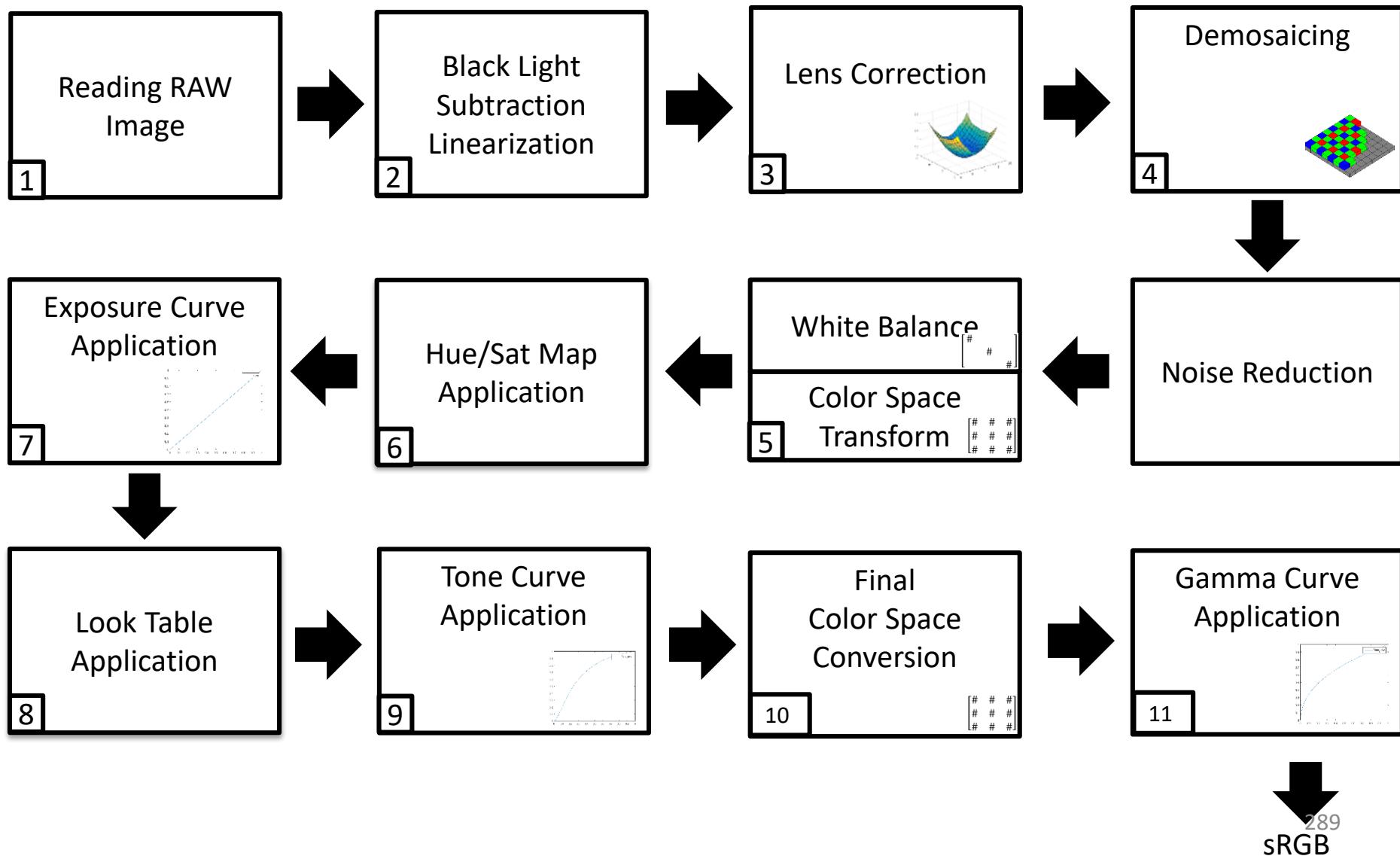


# Android Camera2 API

- Allows access to many of the onboard camera procedures
- Access to camera characteristics (CST, white balance)
- Can capture RAW images in DNG



# DNG SDK “access” to the pipeline

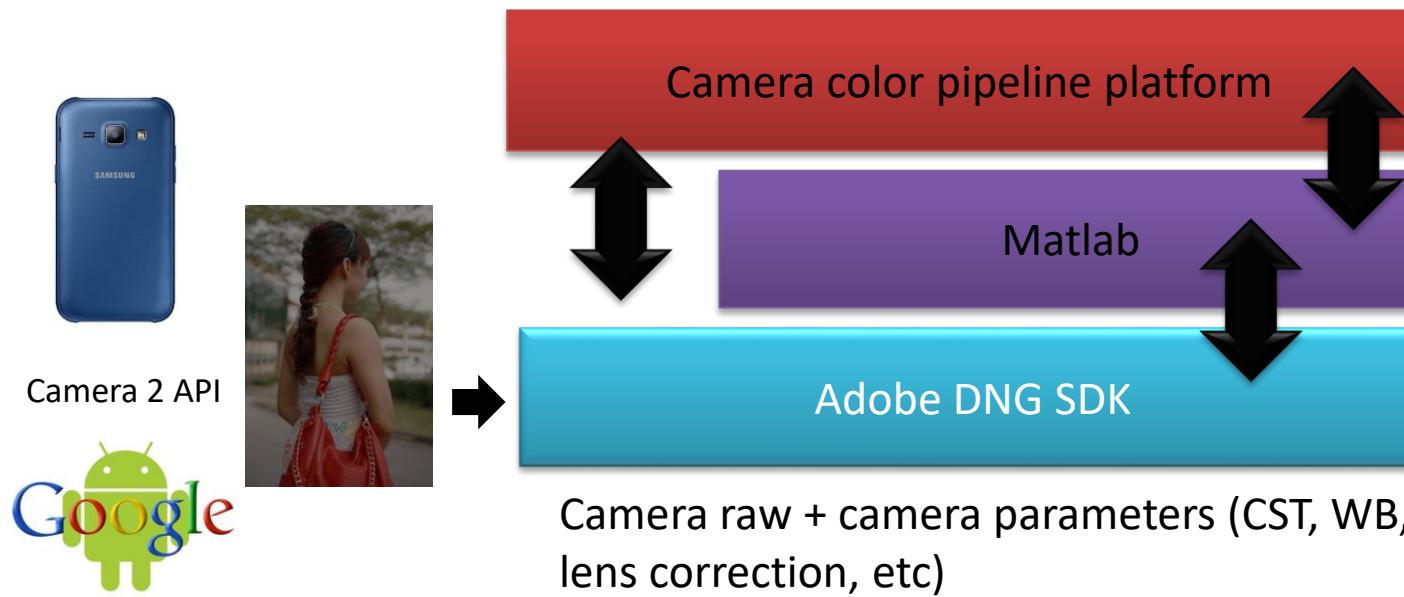


# Matlab platform for color pipeline manipulation

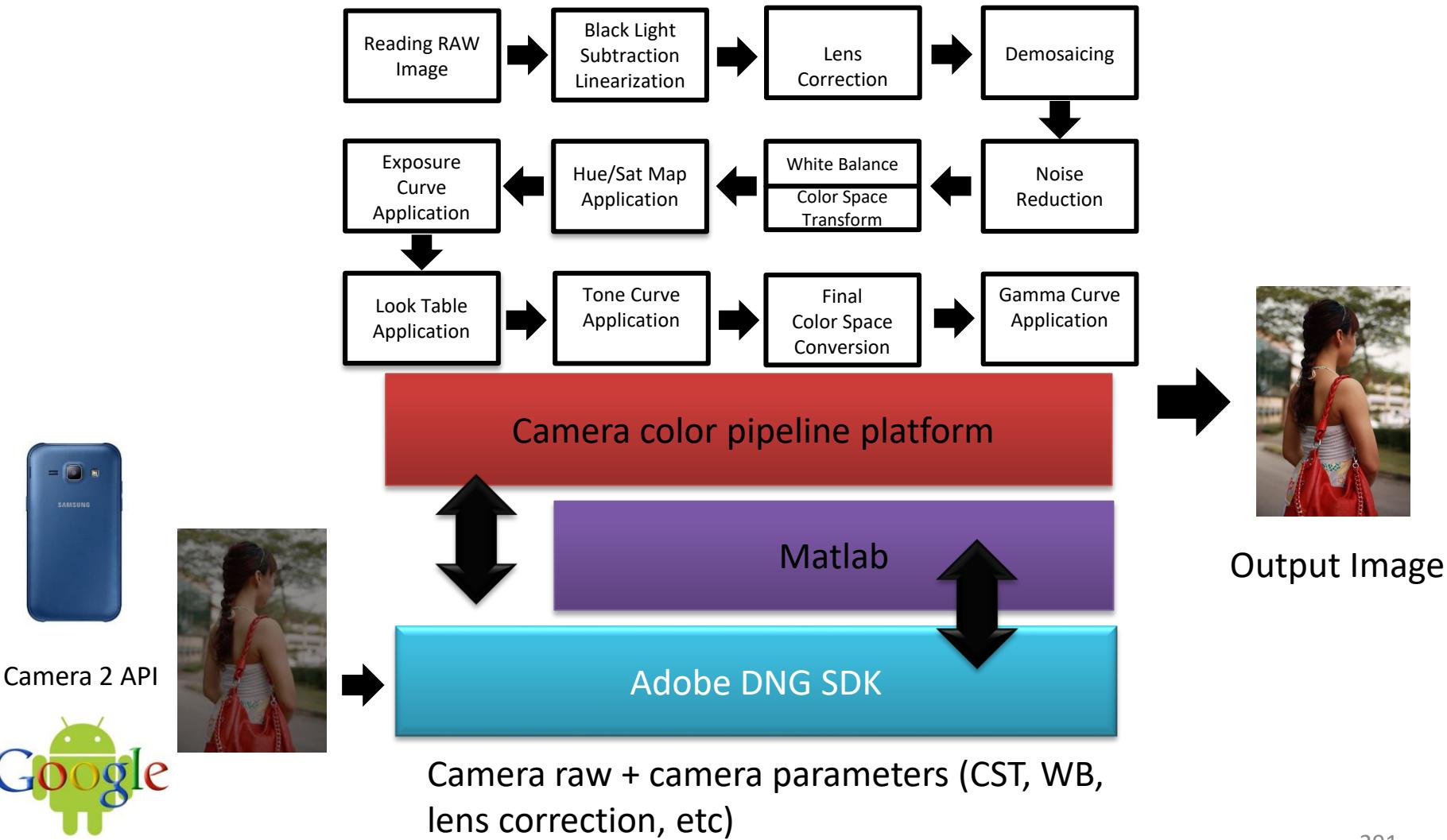
User has access to the image at each step in the color processing pipeline.

Parameters for each step can be modified.

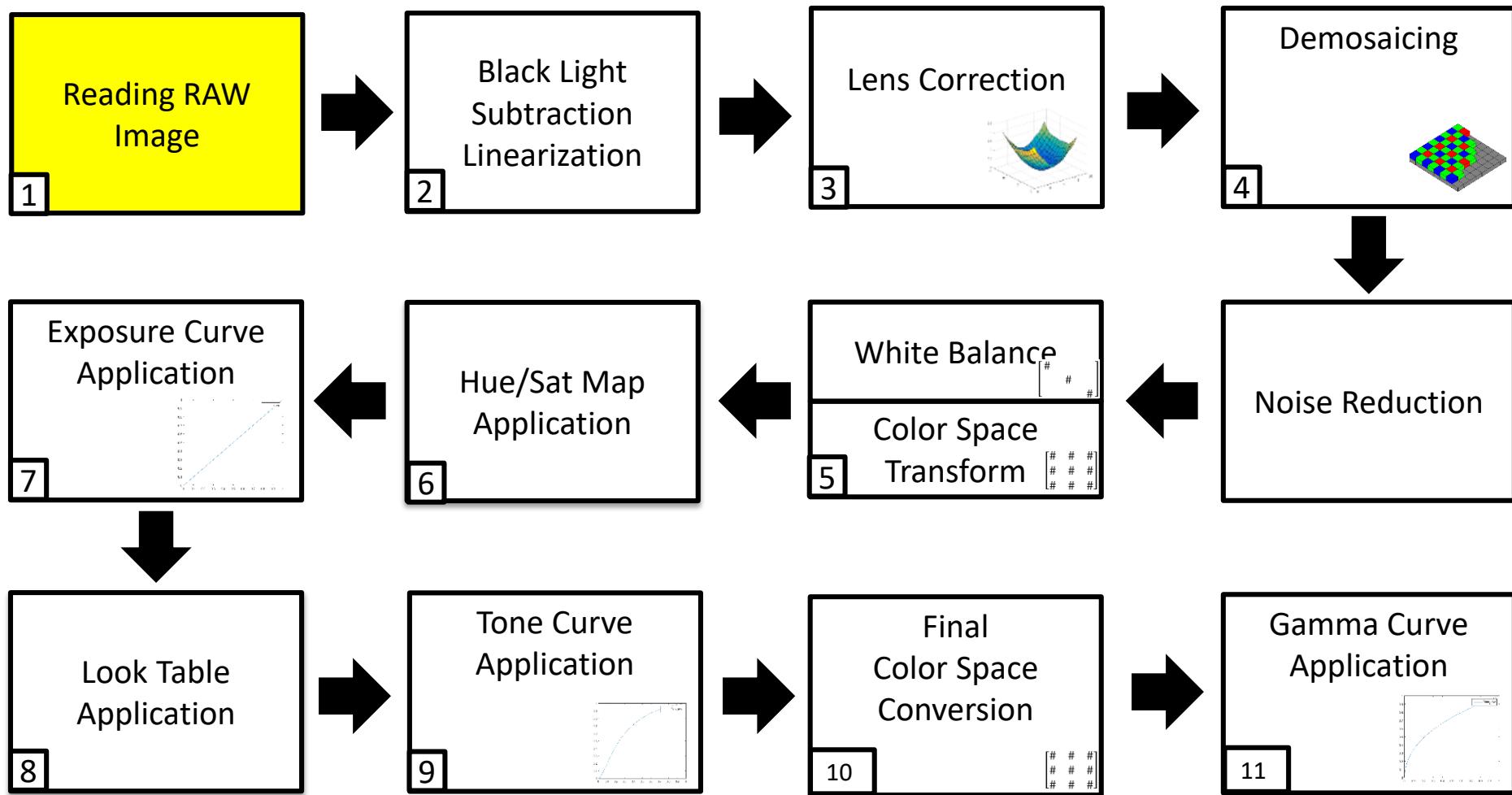
All using Matlab calls.



# Matlab platform for color pipeline manipulation



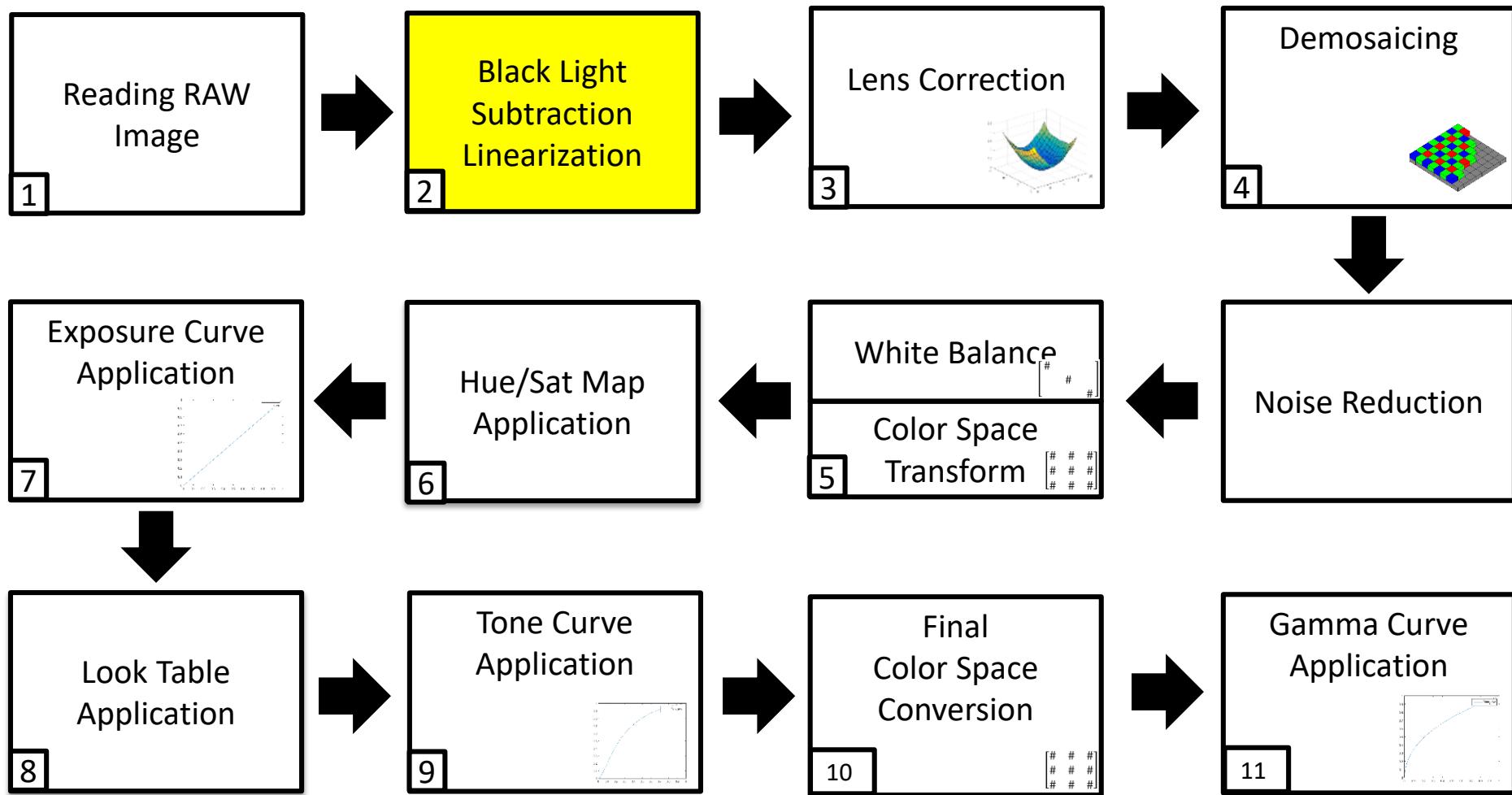
# Matlab camera platform / Adobe DNG



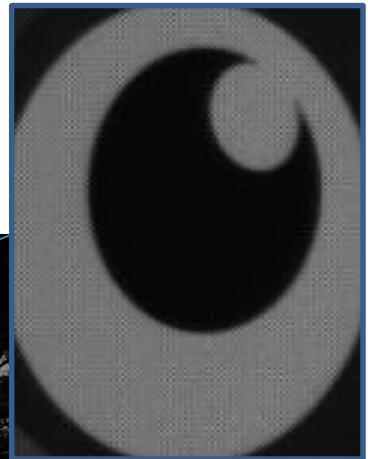
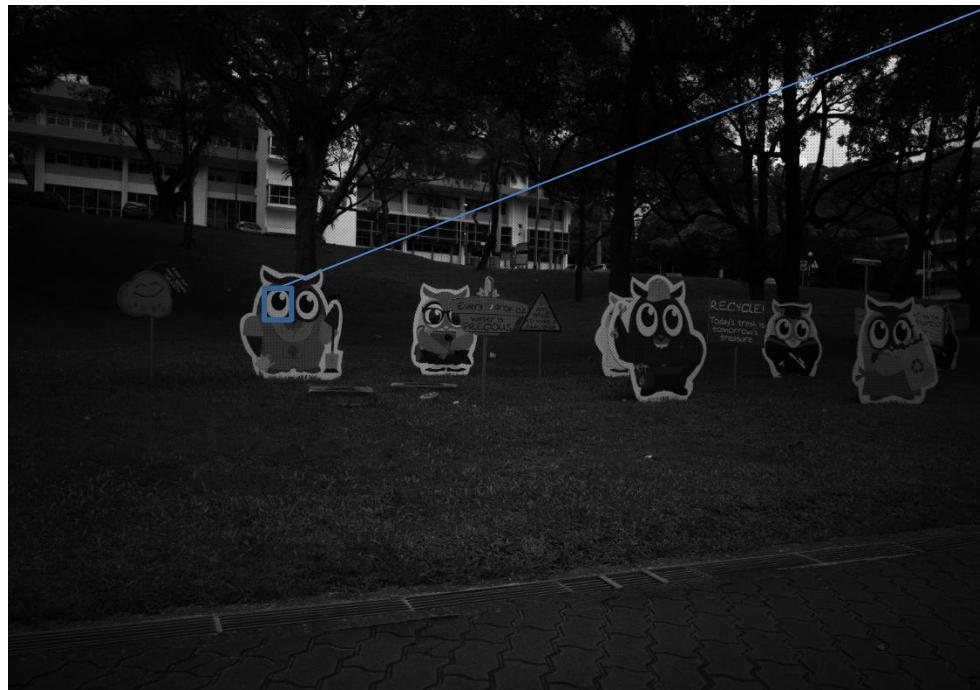
# Reading RAW Image



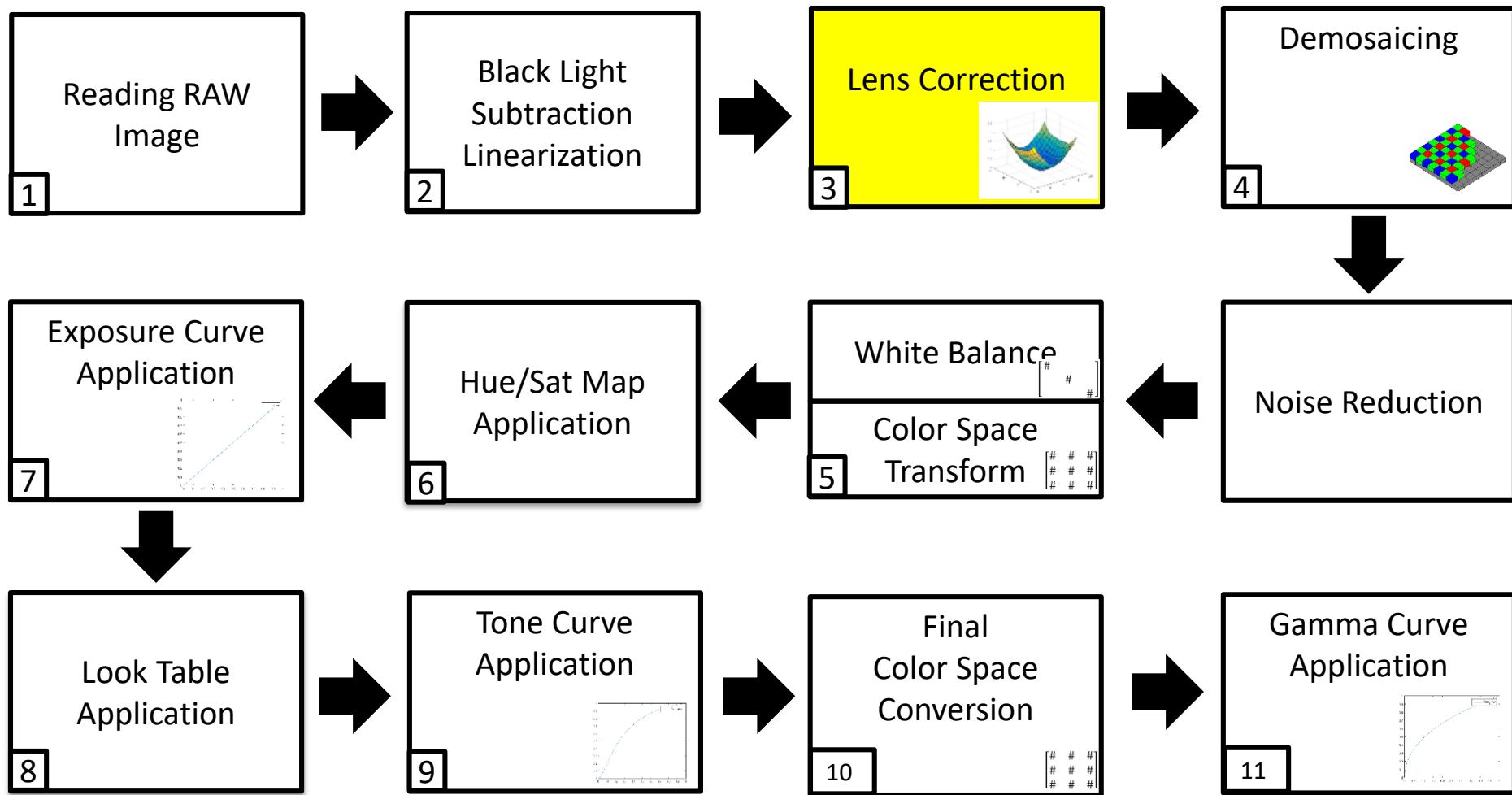
# Matlab camera platform / Adobe DNG



# Black light subtraction + linearization

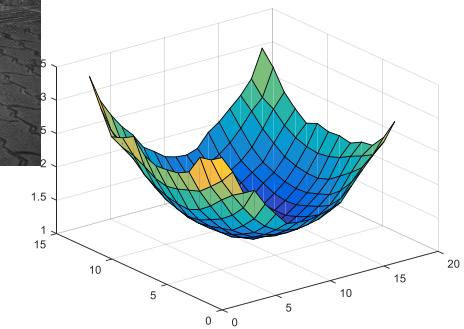


# Matlab camera platform / Adobe DNG

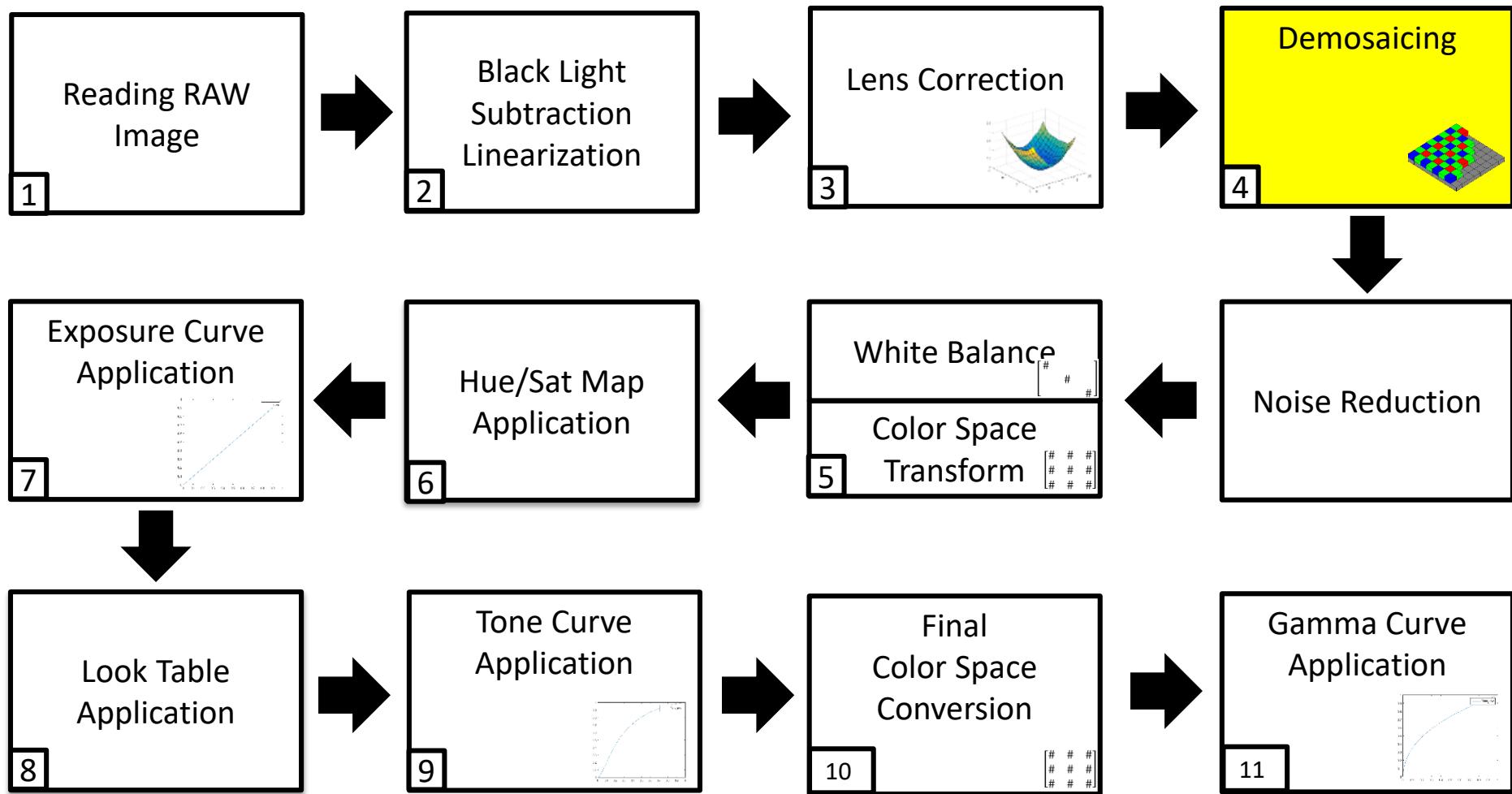


# Lens Correction

## (non-uniform gain)



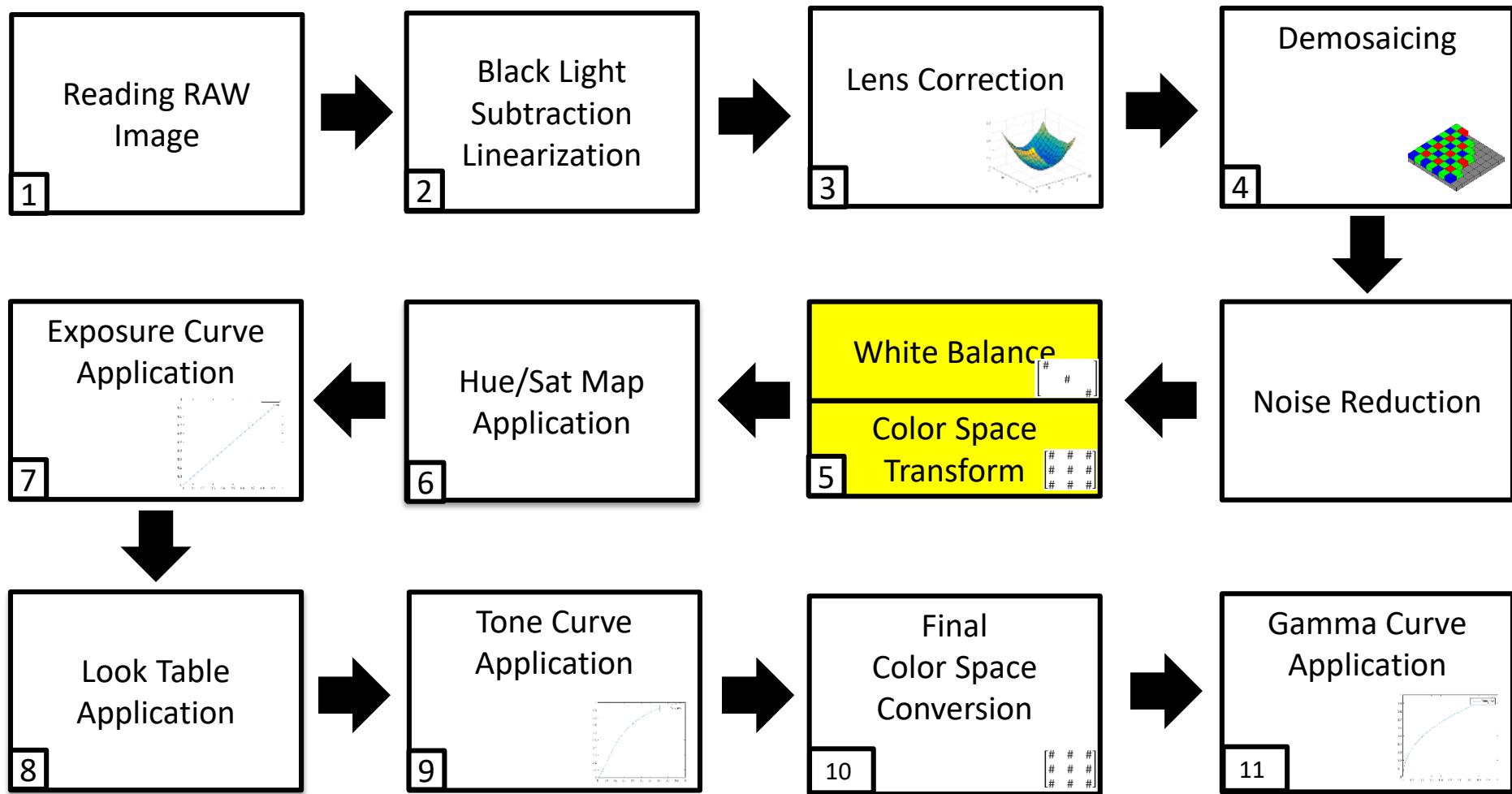
# Matlab camera platform / Adobe DNG



# Demosaic



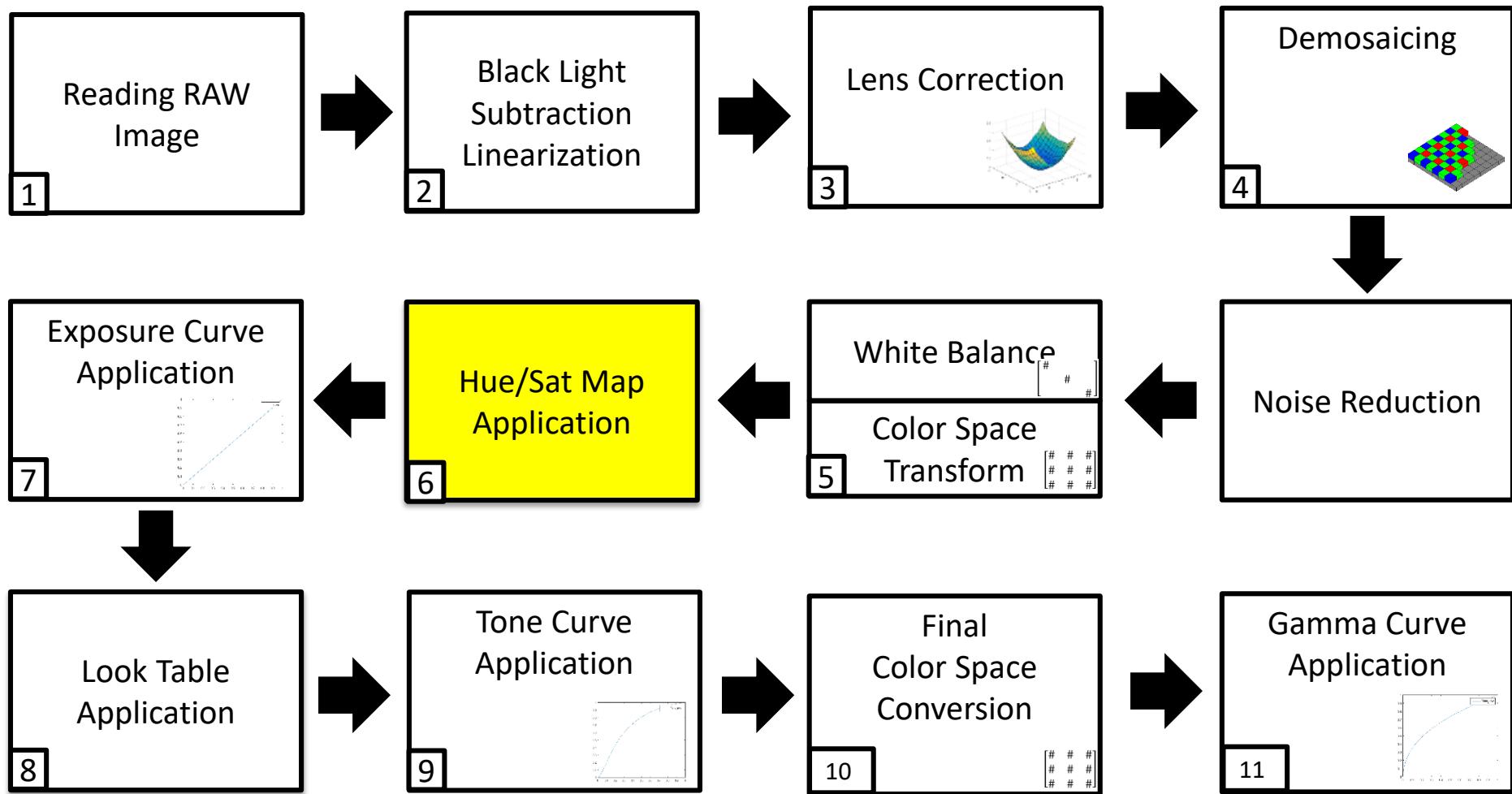
# Matlab camera platform / Adobe DNG



# White balance + color space transform



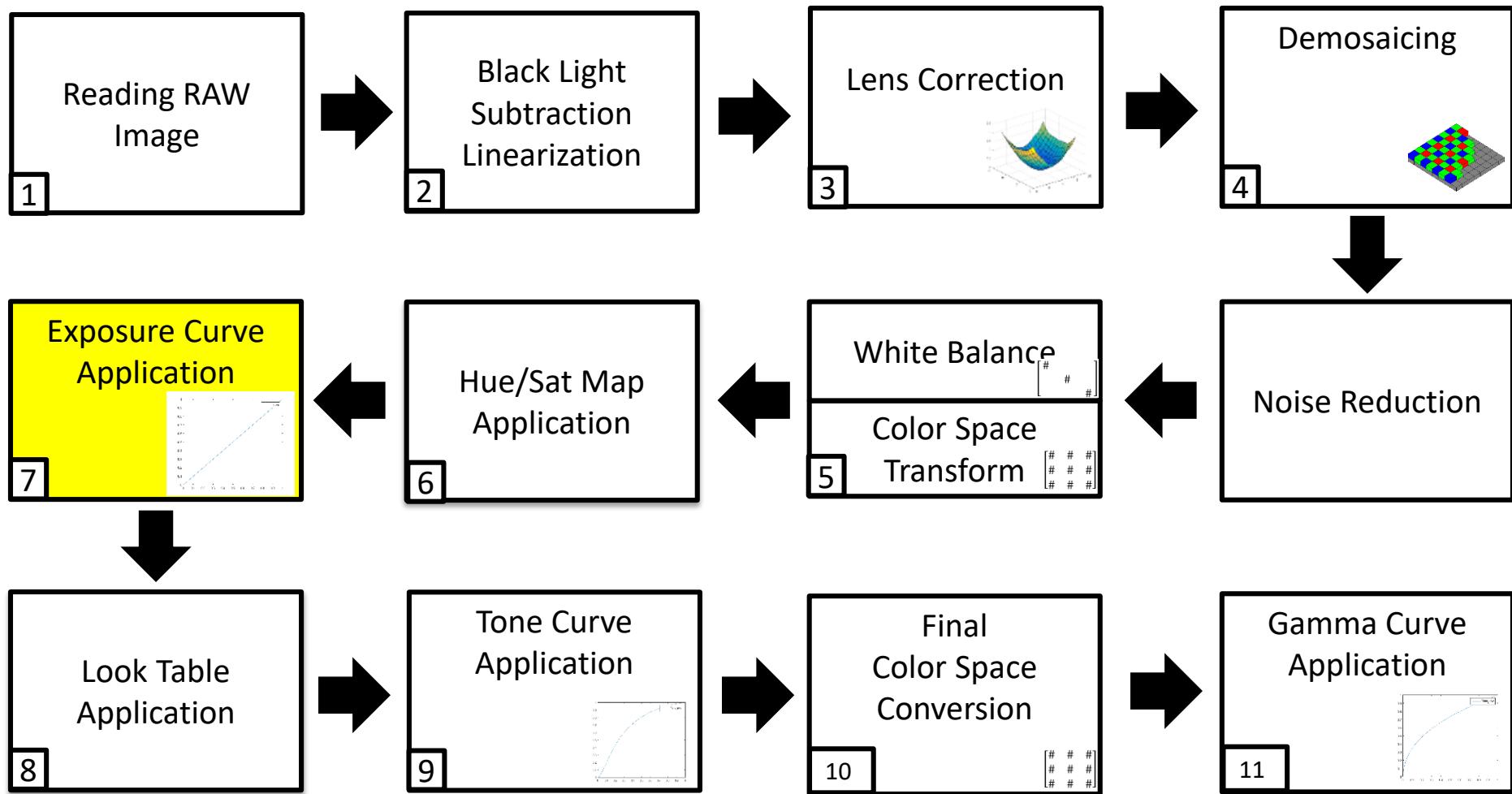
# Matlab camera platform / Adobe DNG



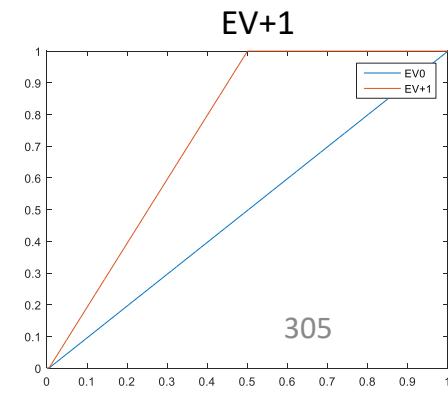
# Hue/Sat map application



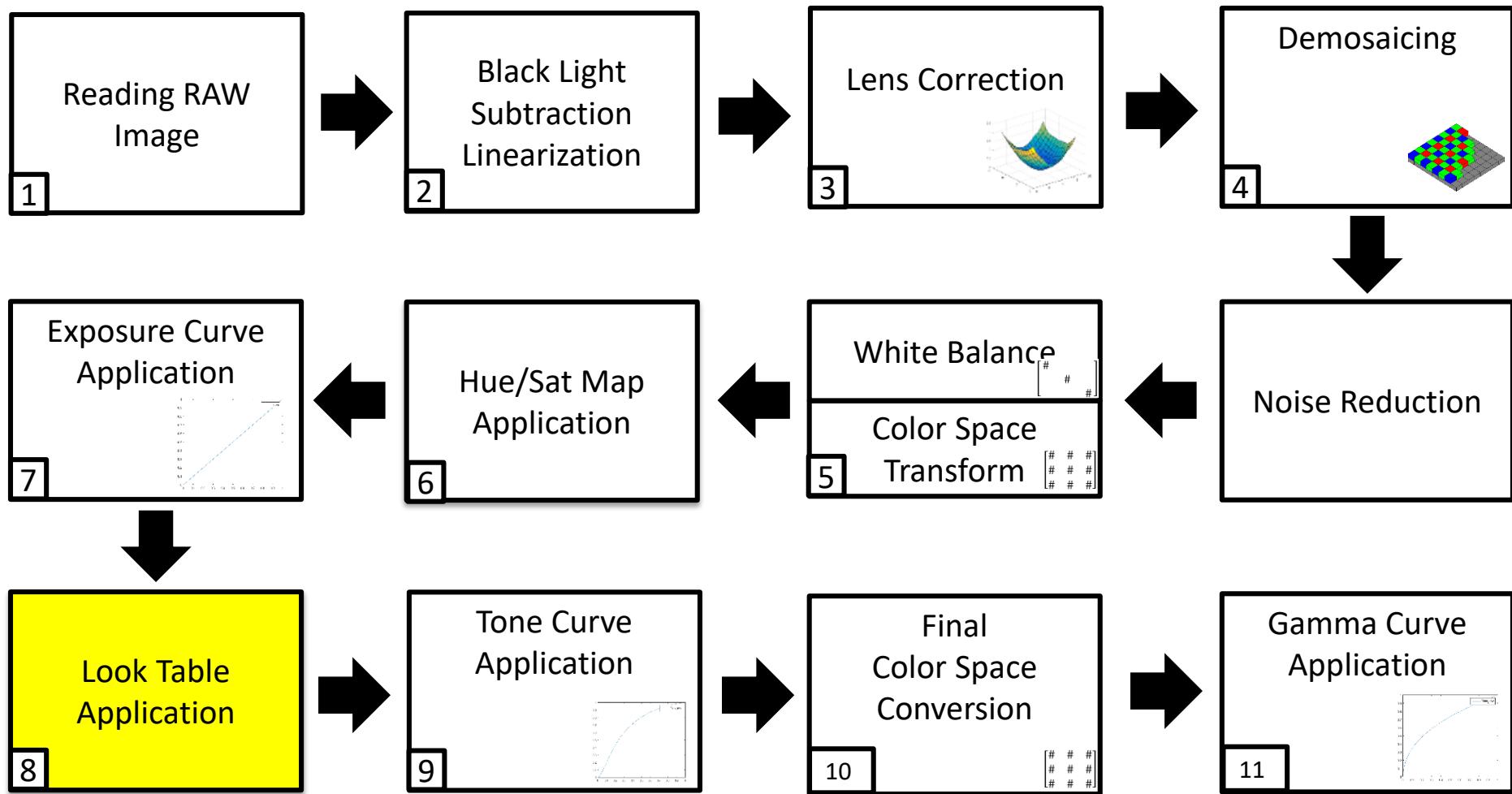
# Matlab camera platform / Adobe DNG



# Exposure curve application



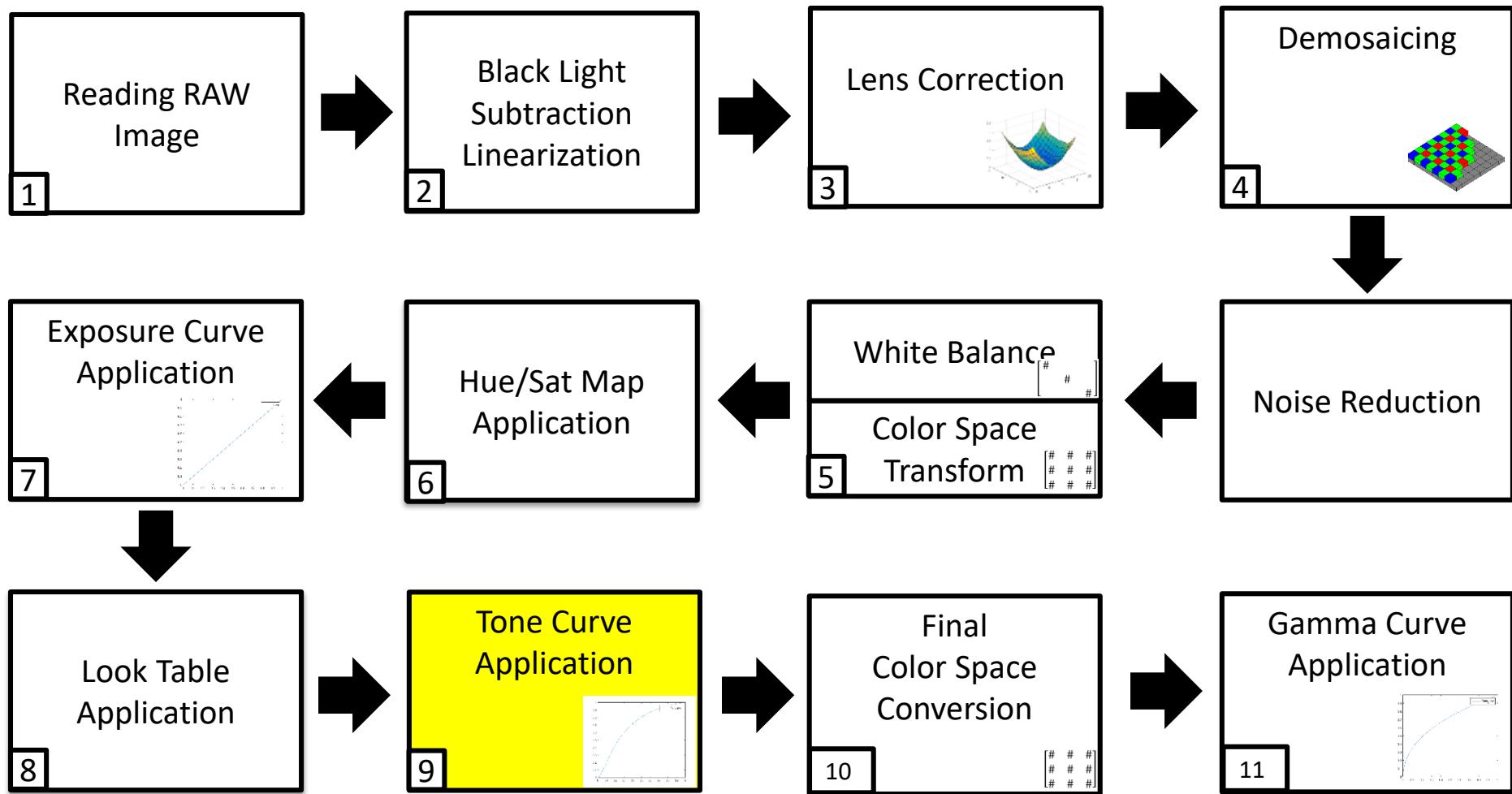
# Matlab camera platform / Adobe DNG



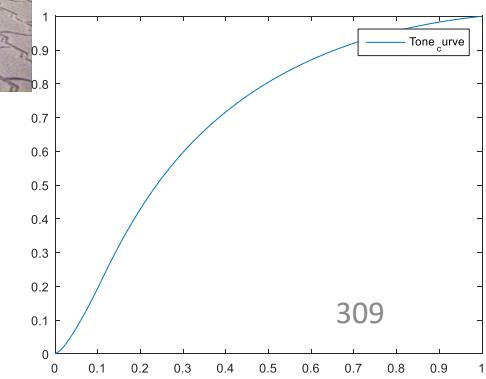
# 3D lookup table application



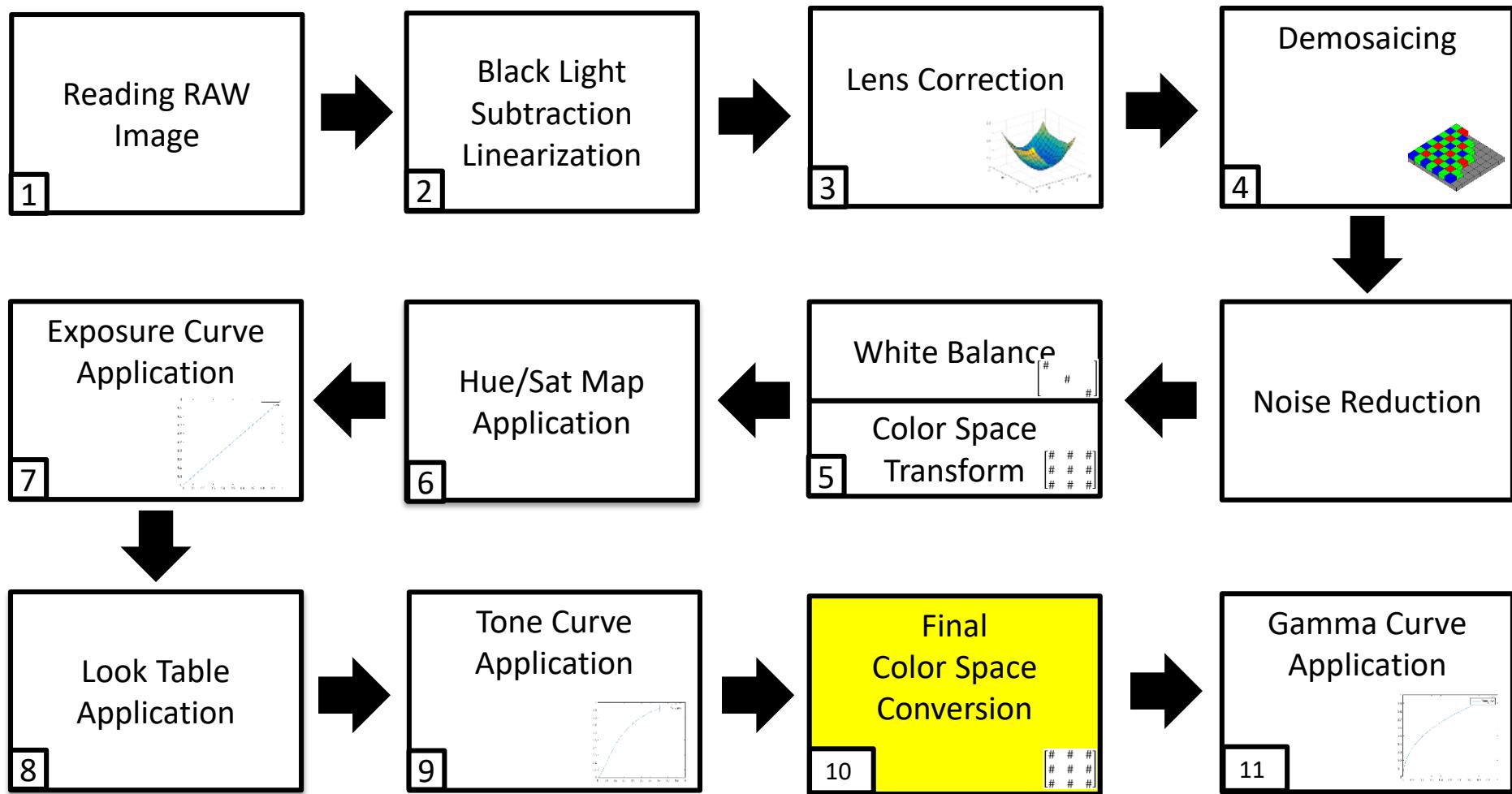
# Matlab camera platform / Adobe DNG



# Tone curve application



# Matlab camera platform / Adobe DNG

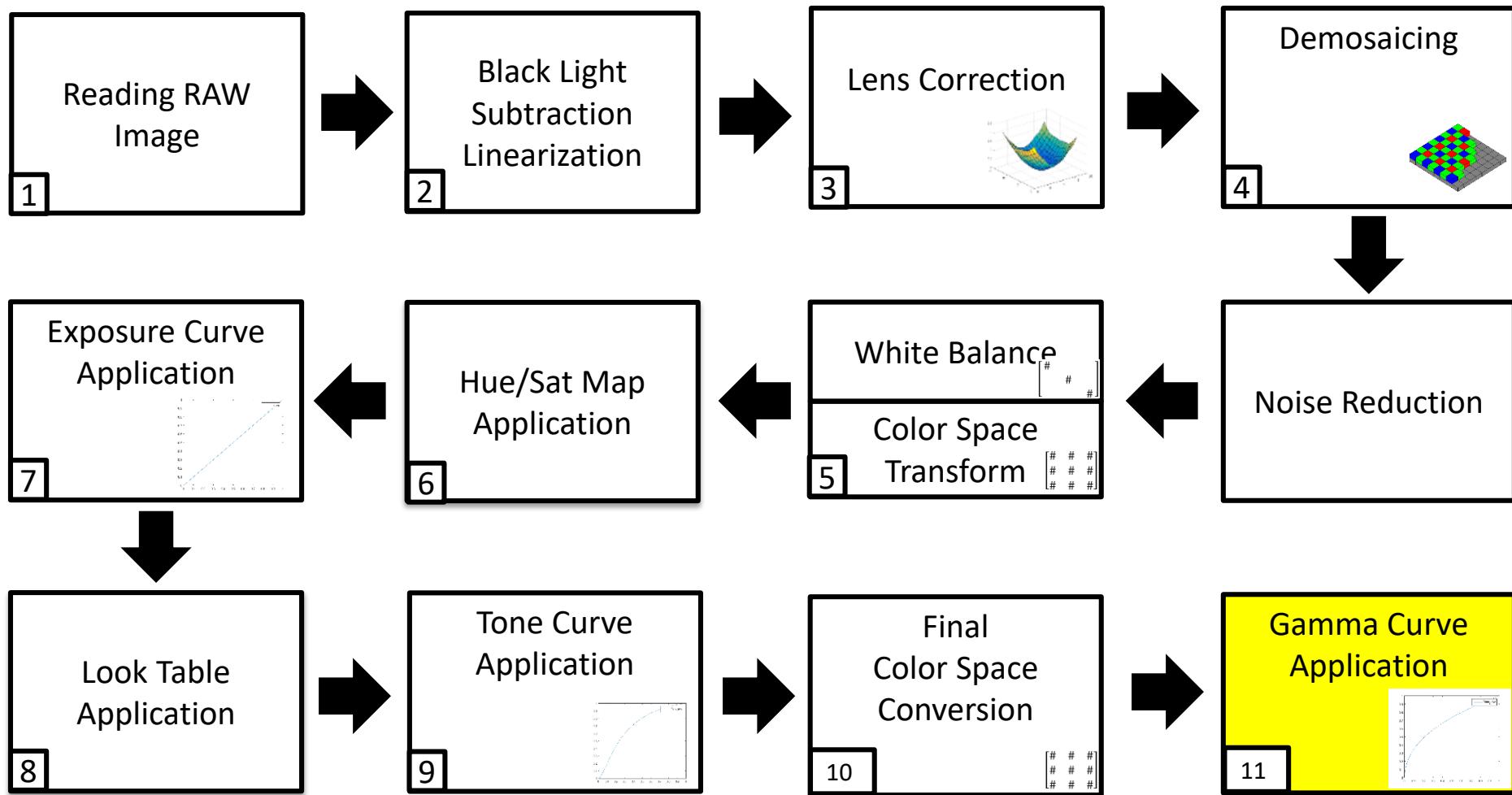


# Final color space conversion

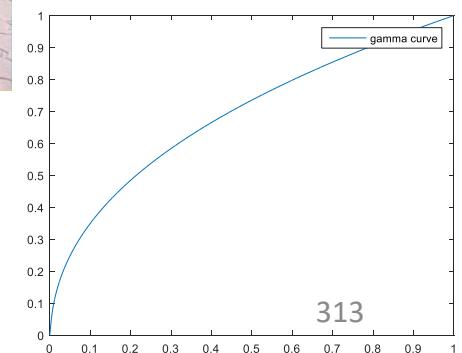
(CIE XYZ -> linear sRGB)



# Matlab camera platform / Adobe DNG



# sRGB gamma curve application



# All together now



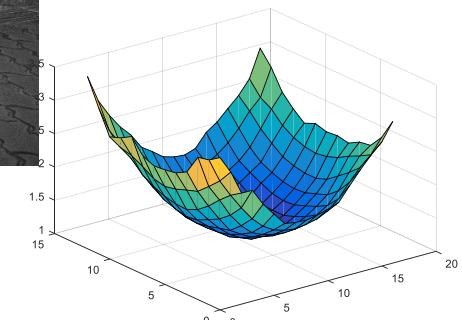
Camera RAW

# All together now



Black level  
subtraction and  
linearization

# All together now



Lens correction  
(non-uniform gain)  
316

# All together now



Demosaicing  
+ Noise Reduction  
<sup>317</sup>

# All together now



White balance  
+ color space transform (CIE XYZ)  
<sup>318</sup>

# All together now

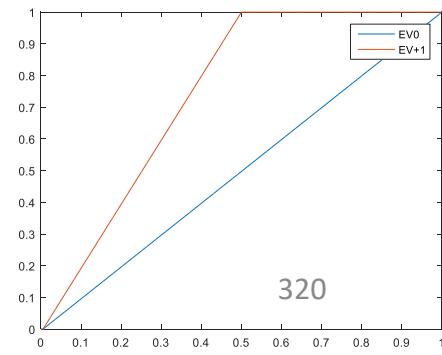


Hue-saturation  
adjustment  
319

# All together now



Exposure  
Compensation

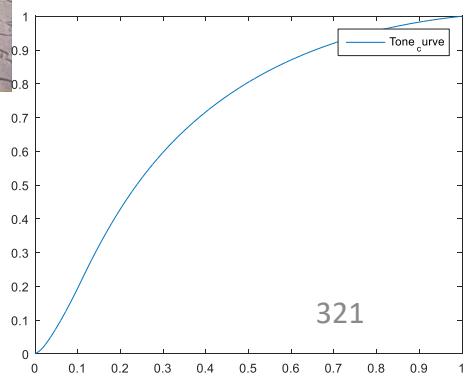


# All together now



Tone-Curve

321



# All together now

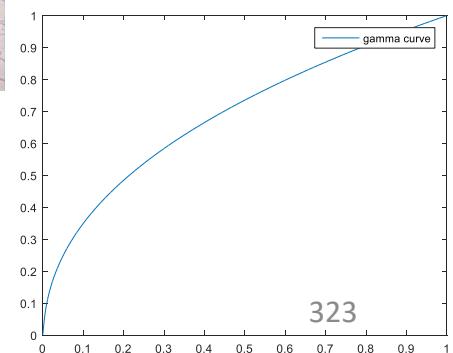


sRGB color space

# All together now



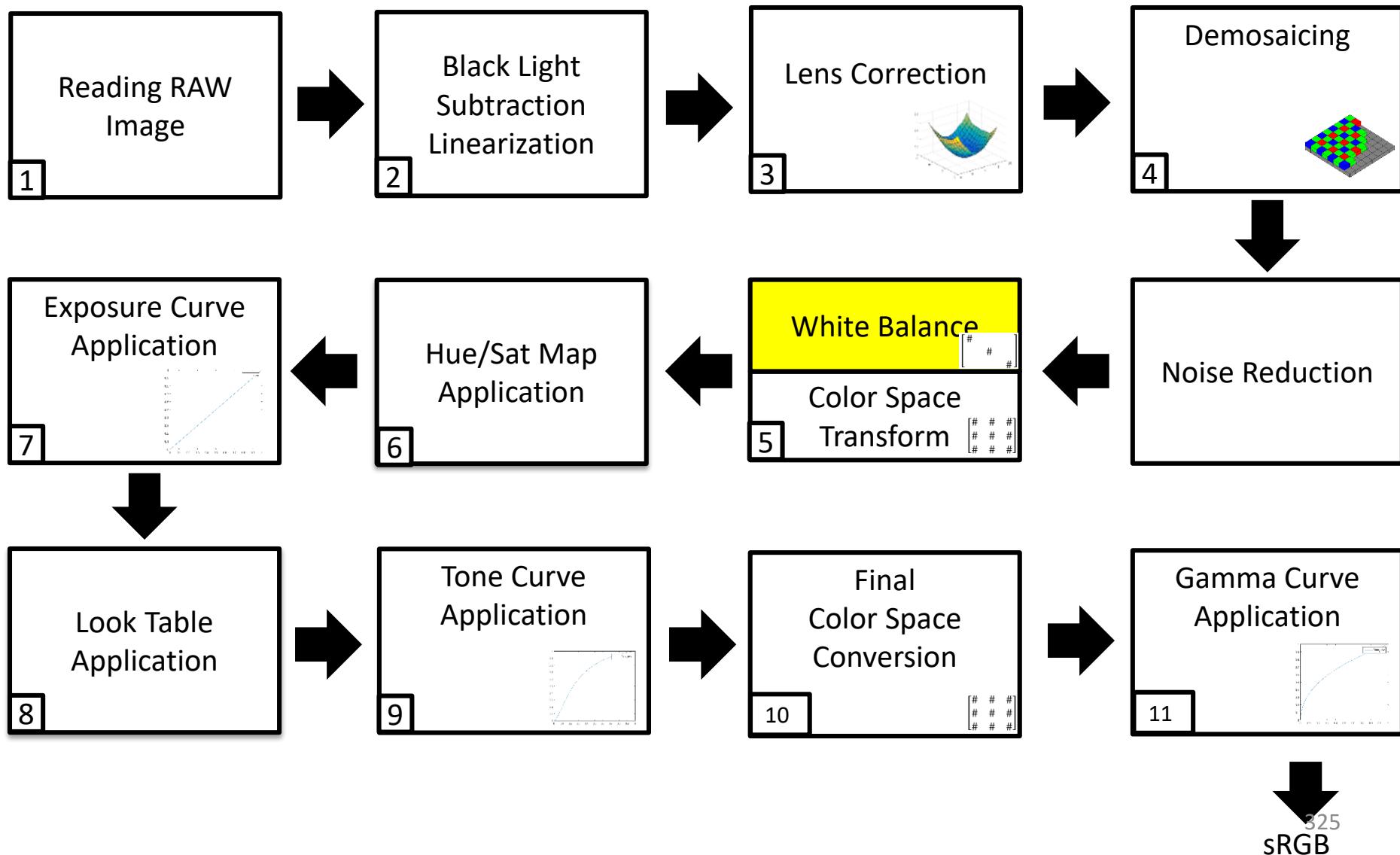
sRGB gamma



# Using the platform - Examples

- White balance “in context”
- Camera colorimetry
- Real sRGB outputs

# “White balance” in context



# White balance example



Input Raw



White balance method 1



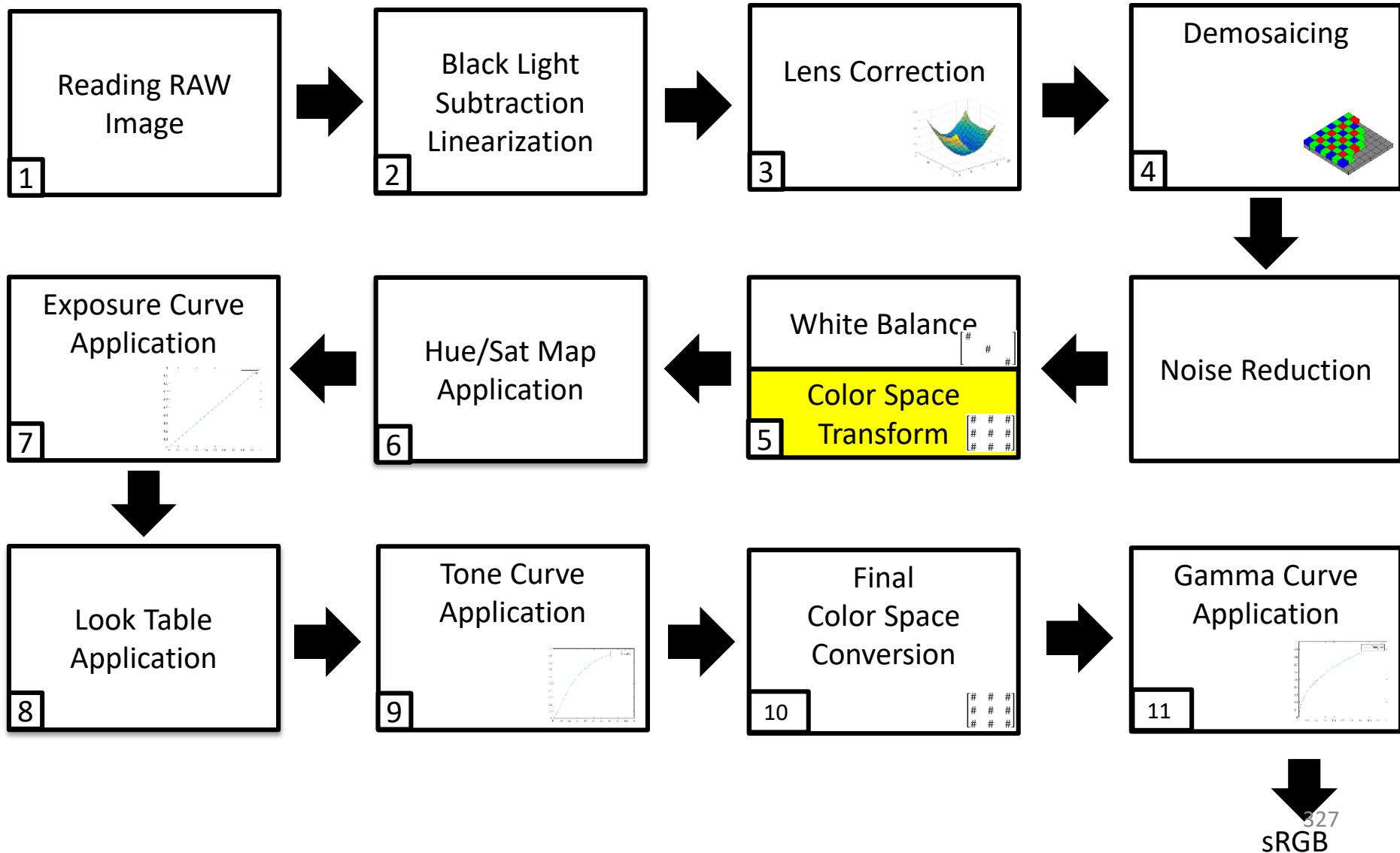
White balance method 2

This is how subjective WB results would have been presented before.



Results through the “full pipeline”

# Colorimetry example



# Example - colorimetry

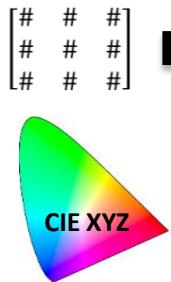
Sensor Image (RAW)



White  
balance

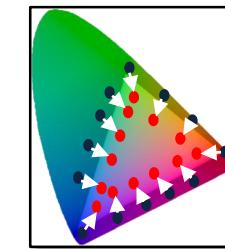
$$\begin{bmatrix} # & # & # \\ # & # & # \\ # & # & # \end{bmatrix}$$

Color Space  
Transform (CST)



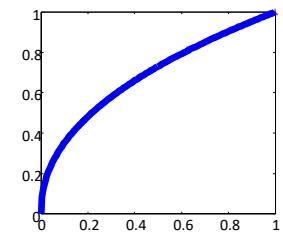
CIE XYZ  
Linear-sRGB

$$\begin{bmatrix} 3.24 & -1.53 & 0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$$



3D RGB Warp

*h*

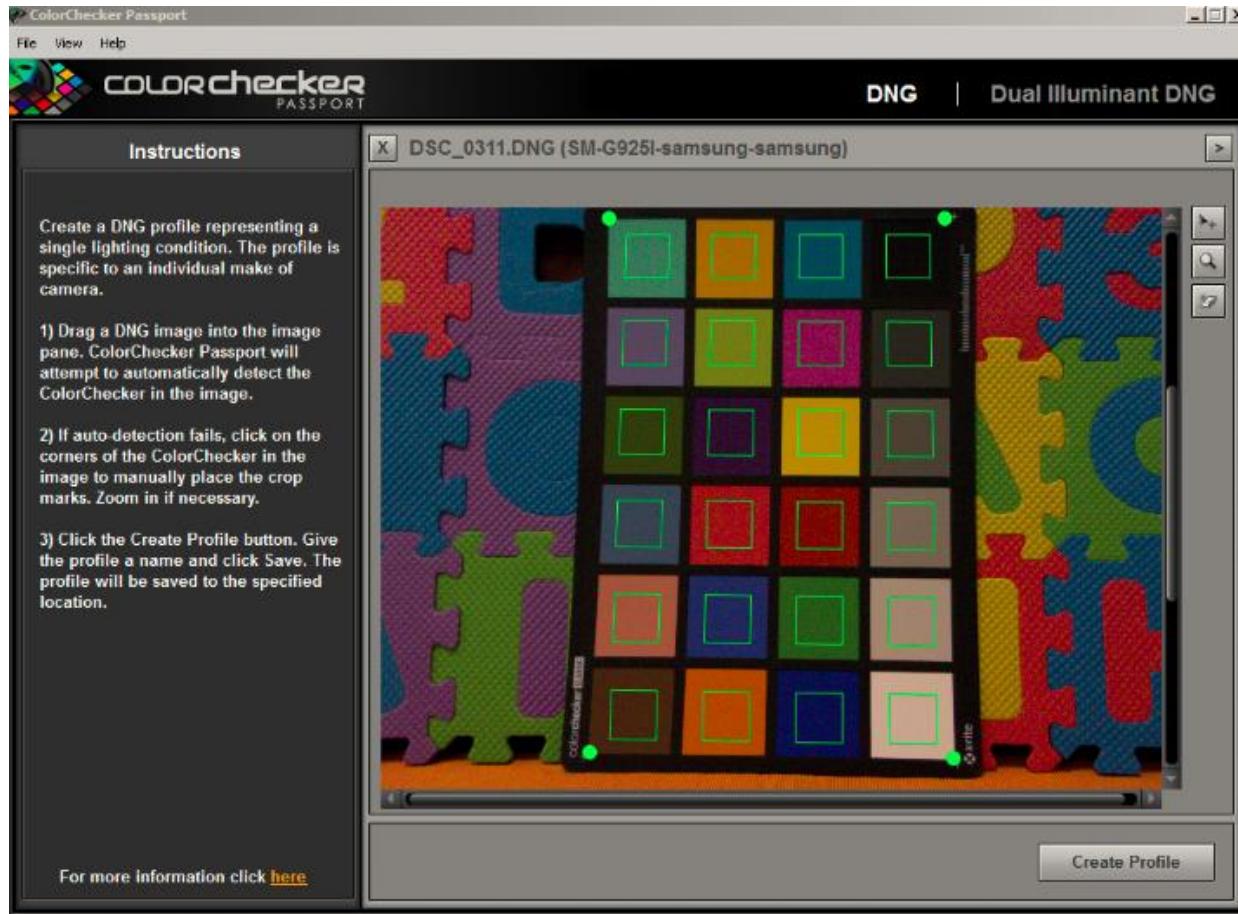


*f*



Can compute our own CST.  
Derive a mapping from the camera raw color space to the CIE XYZ space.  
This is the CST in the pipeline.

# Calibrate using X-Rite software



# Four Android cameras

Samsung S6-Edge



HTC M9



Motorola Nexus 6



LG-G4

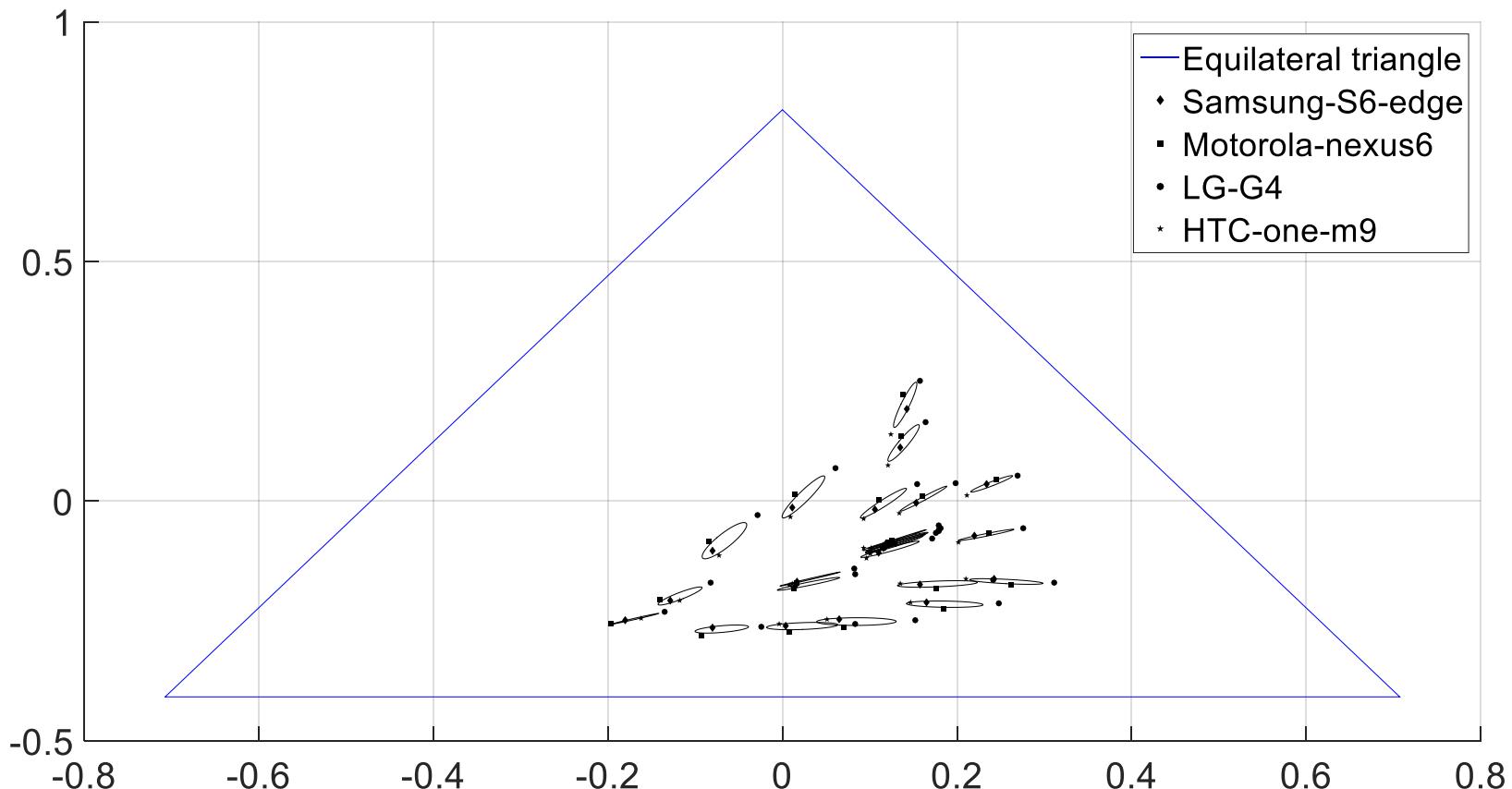


Goal is that the colors for all the cameras are the same.



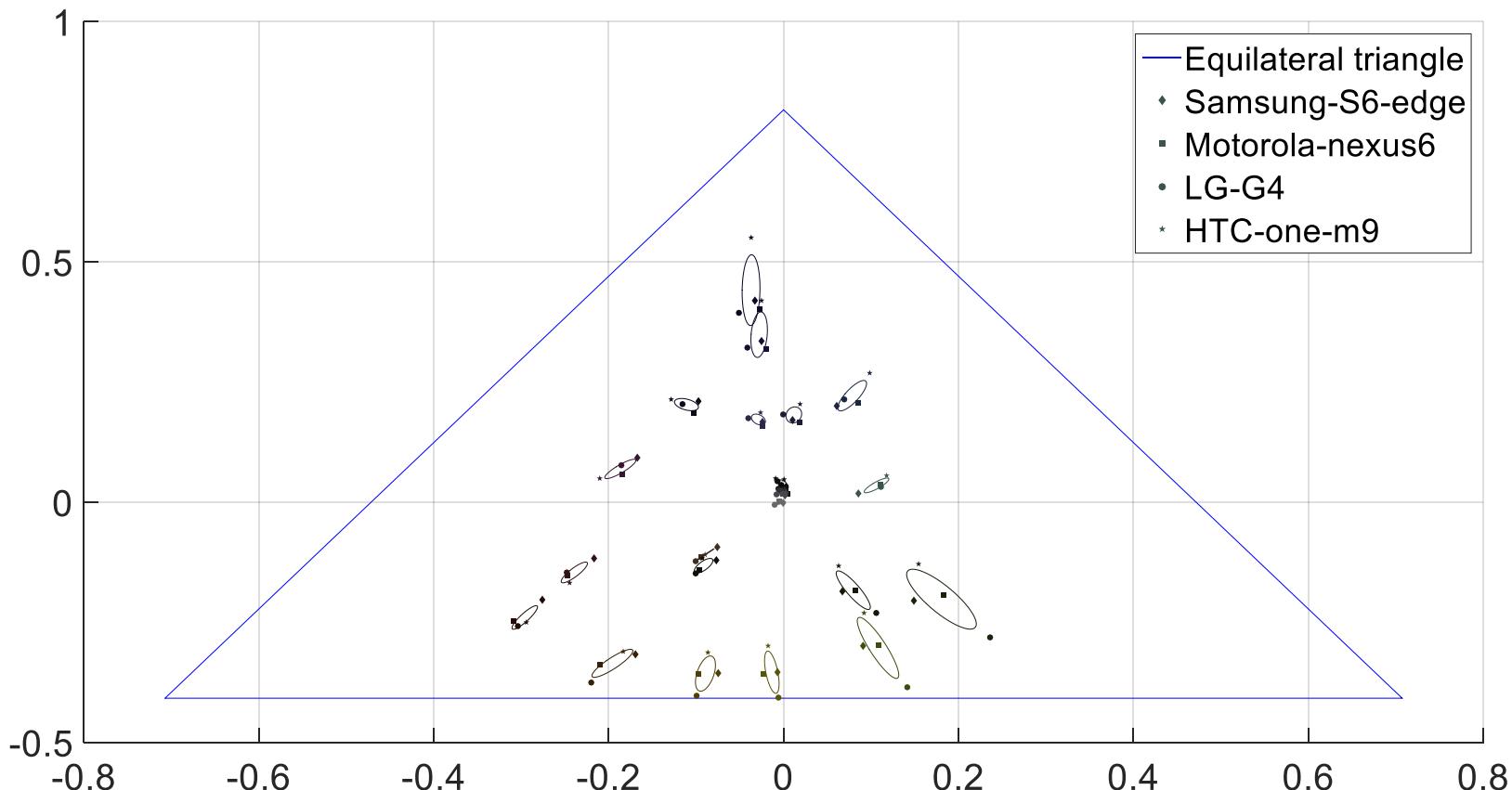
# Color chart patches

RAW— (i.e. without any color transformation)



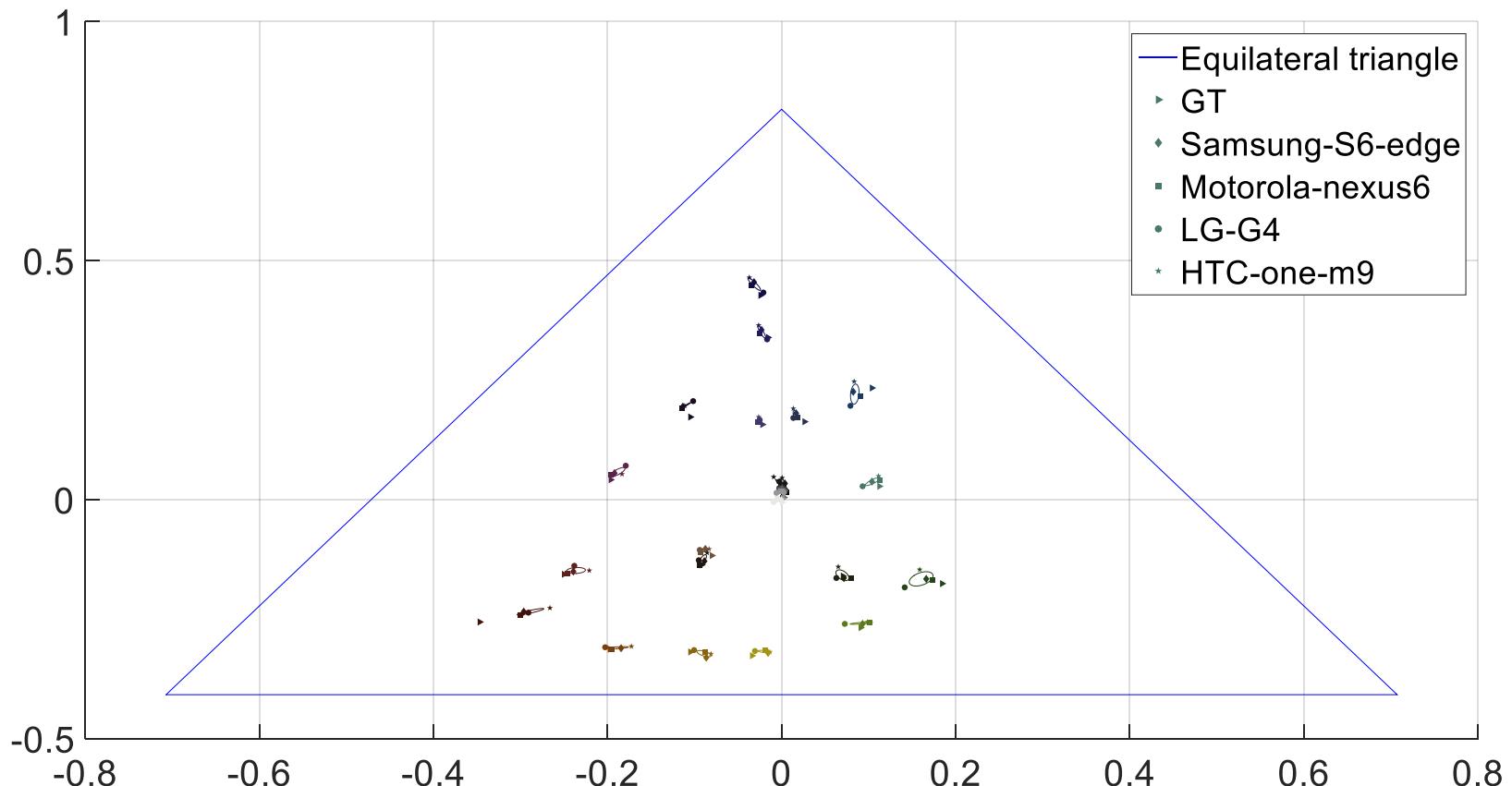
# Color chart patches

RAW + white balanced and CST from the camera



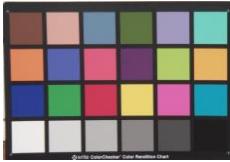
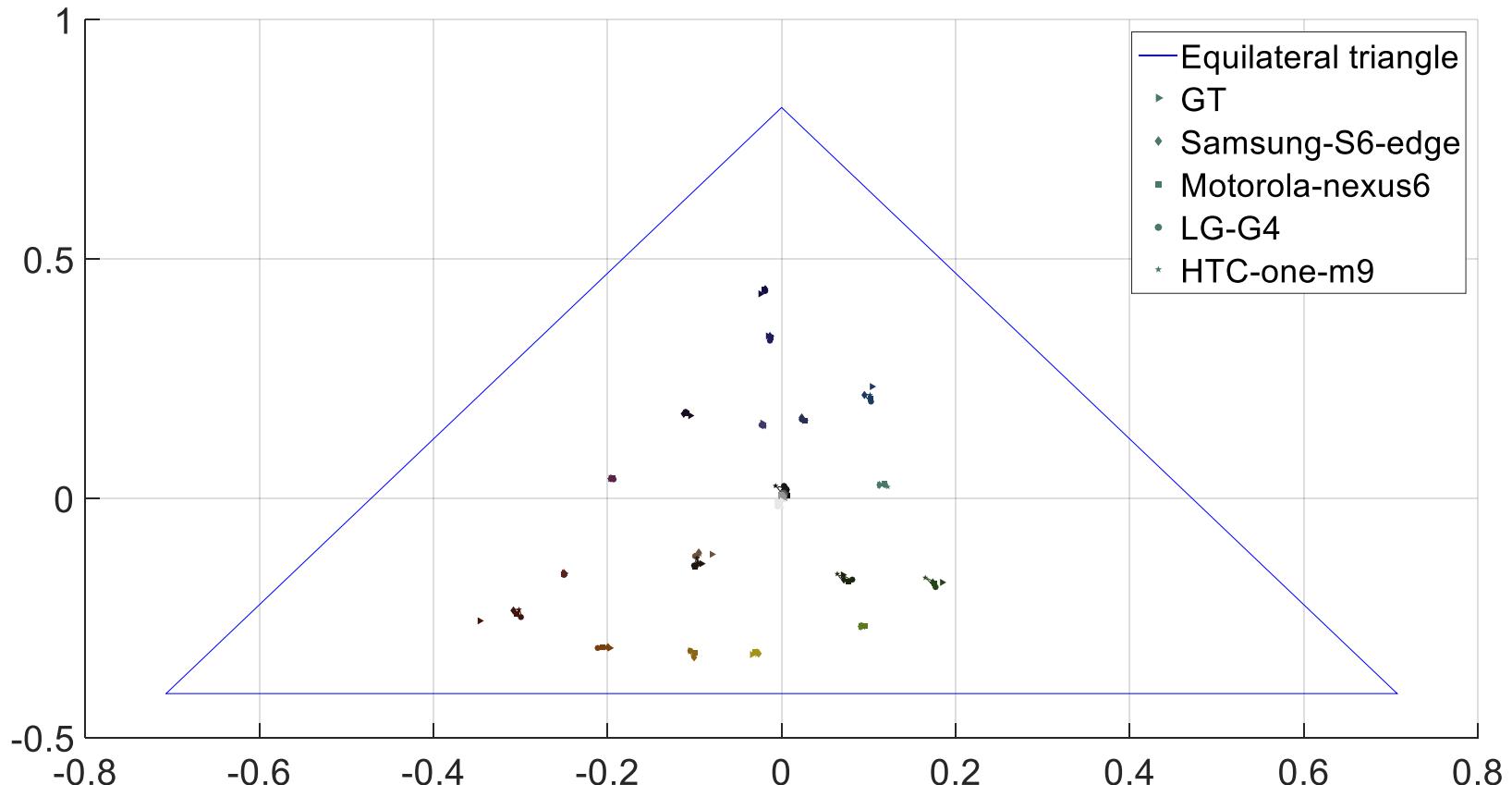
# Color chart patches

## X-Rite profile calibration (CST)



# Color chart patches

## Our CST estimation



Based on a method by Bastani and Funt, SPIE 2014

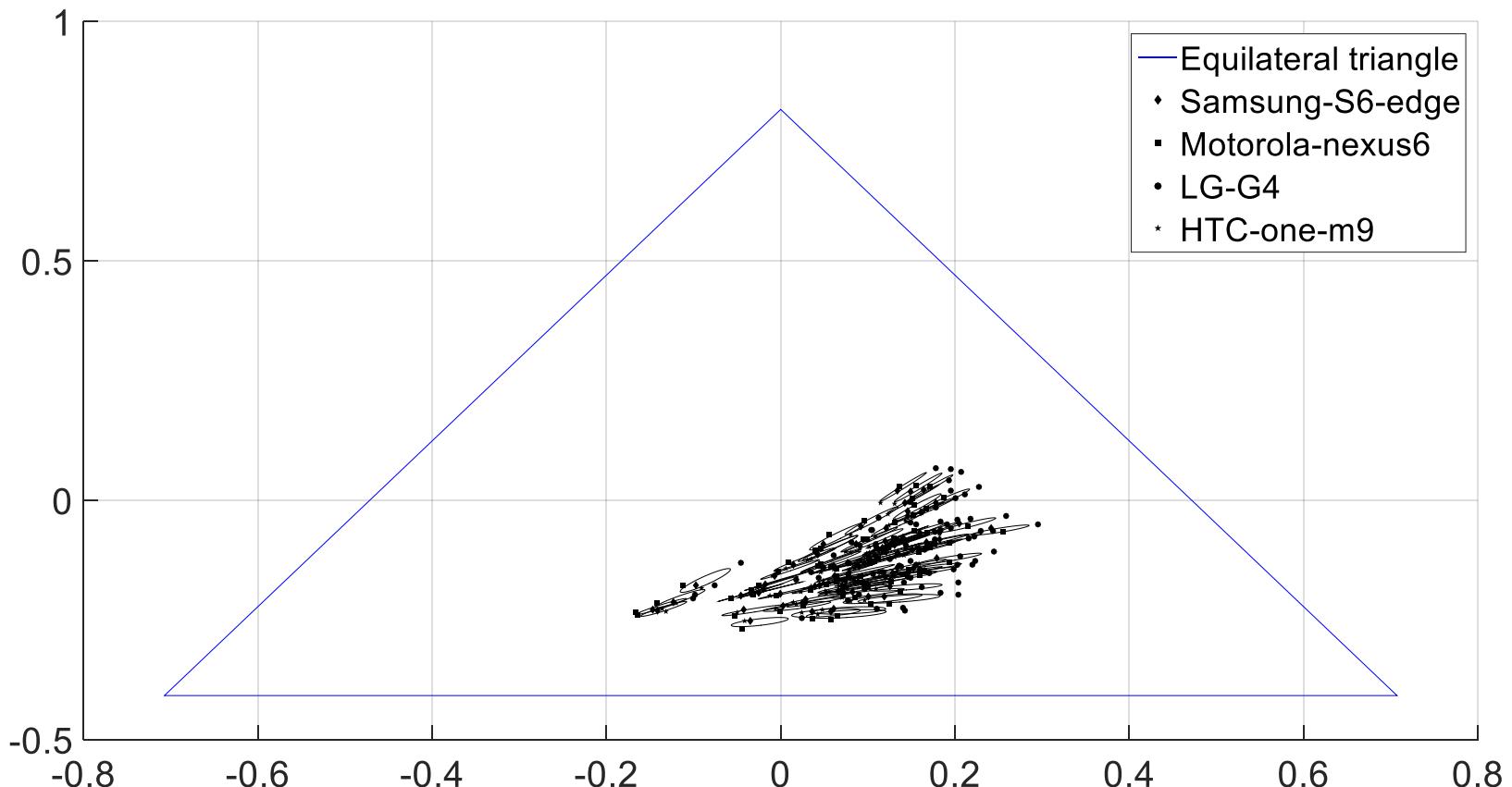
# Applied to other materials



81 Paint chips

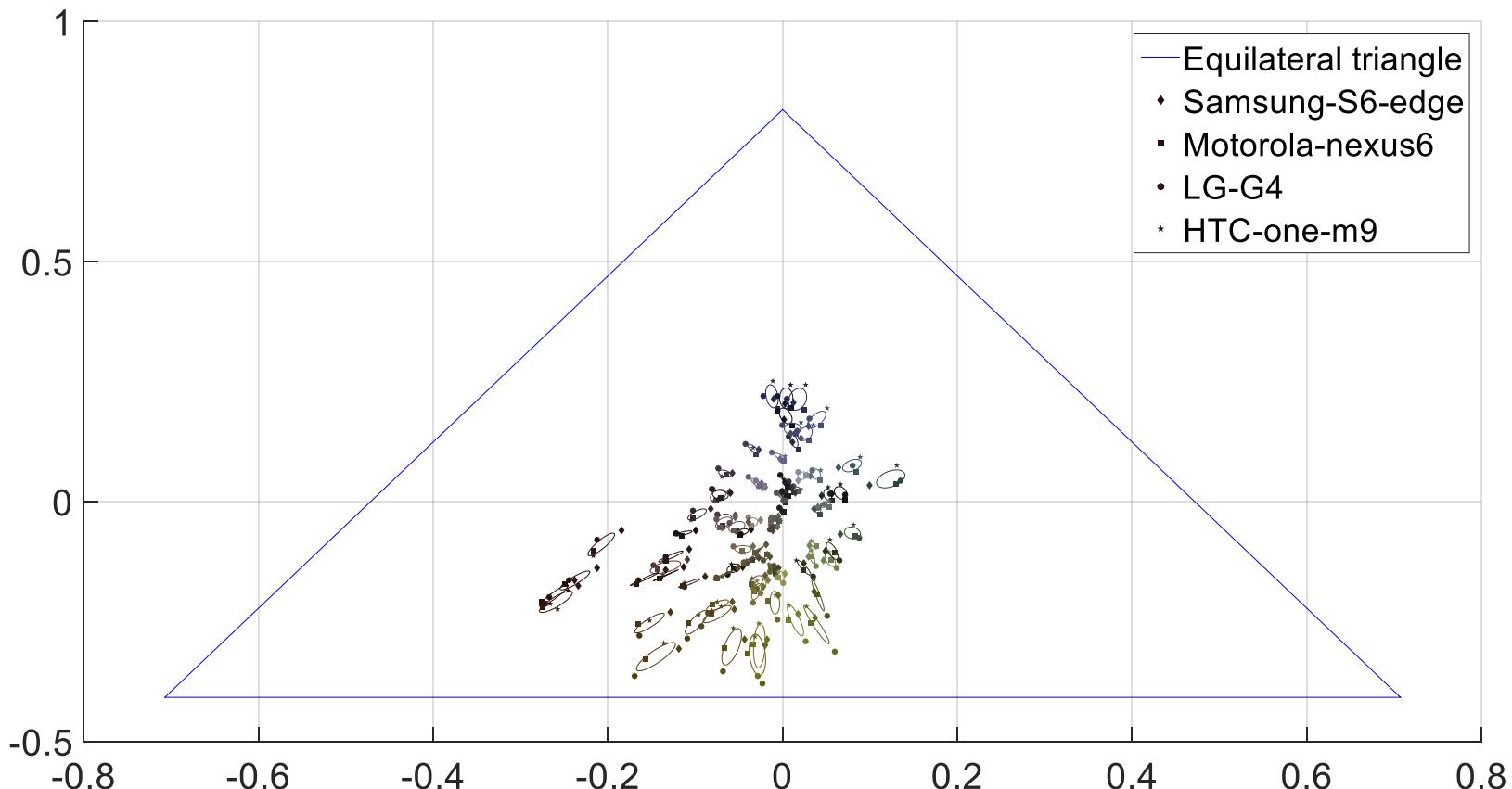
# Additional materials

RAW– (i.e. without any color transformation)



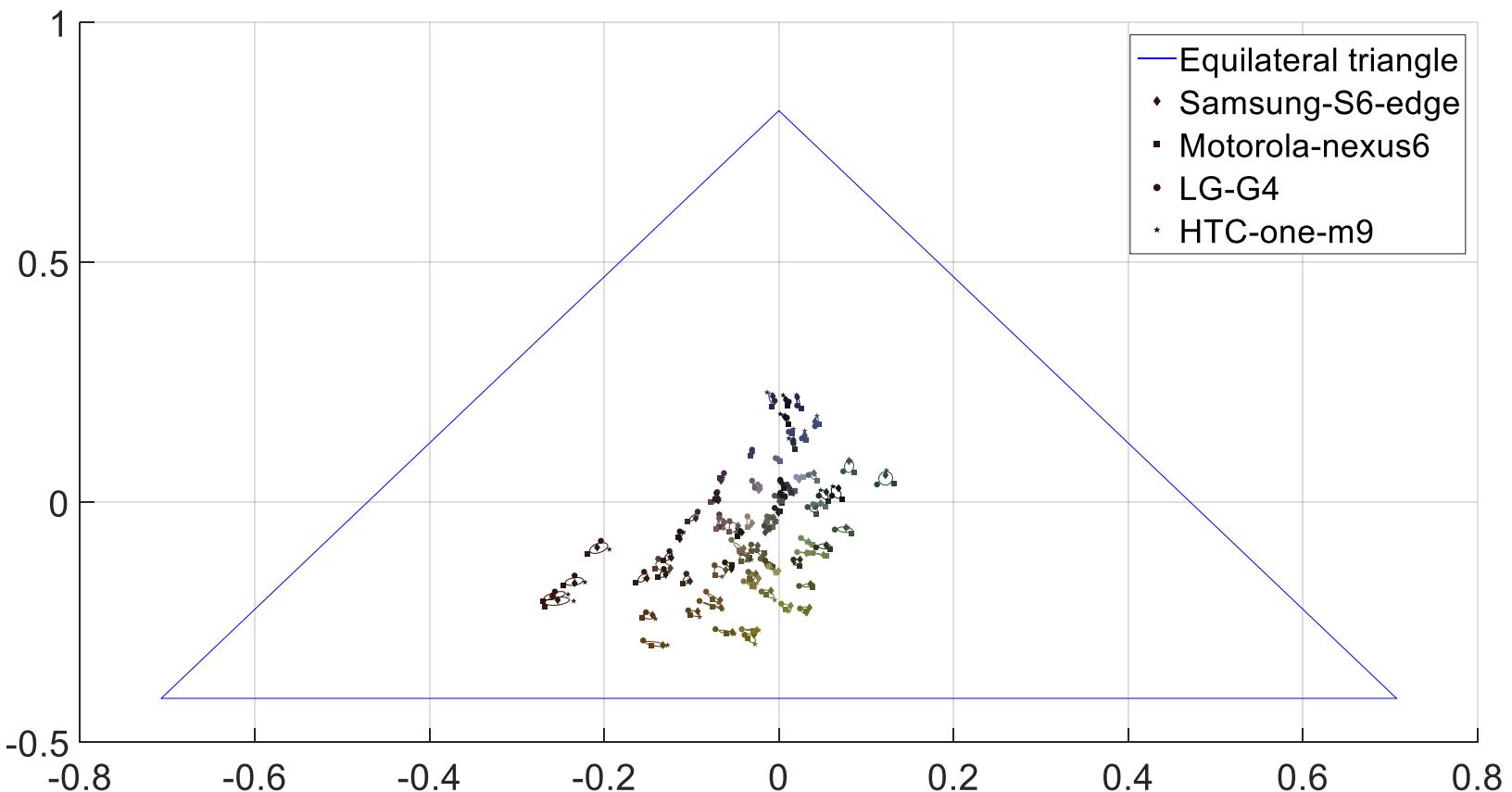
# Additional materials

## RAW (white balanced and CTM of camera)



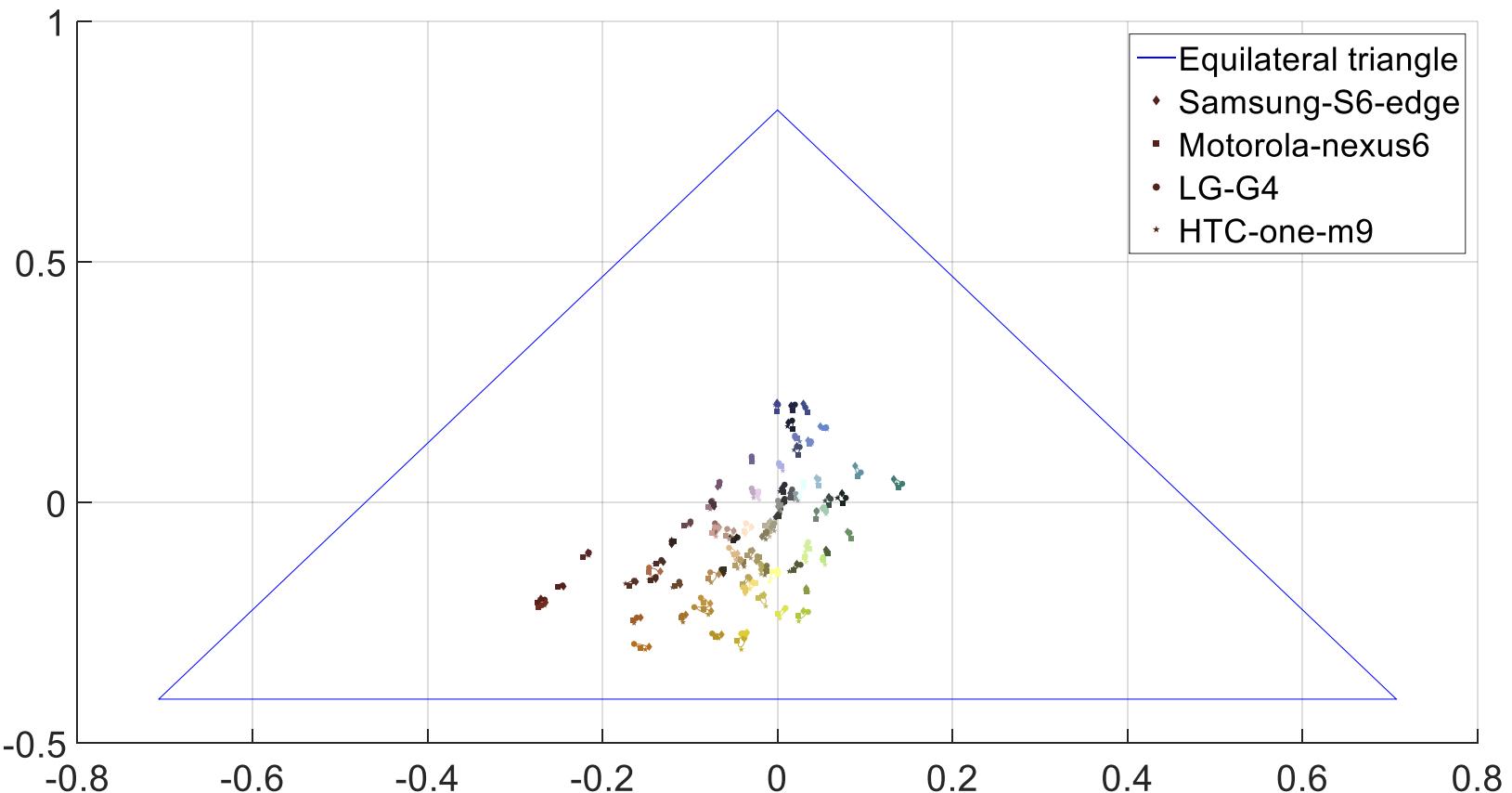
# Additional materials

## X-rite Profile

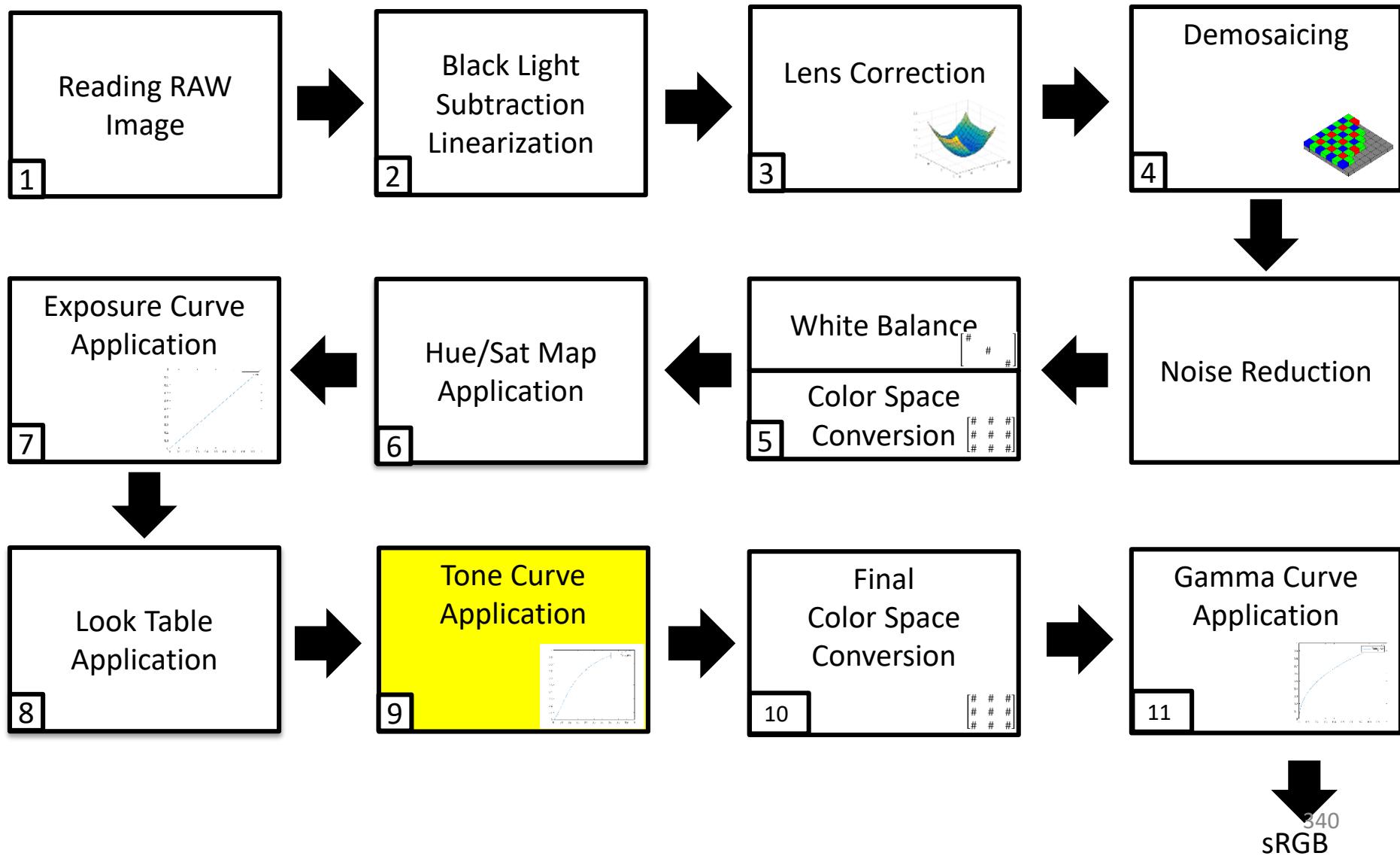


# Additional materials

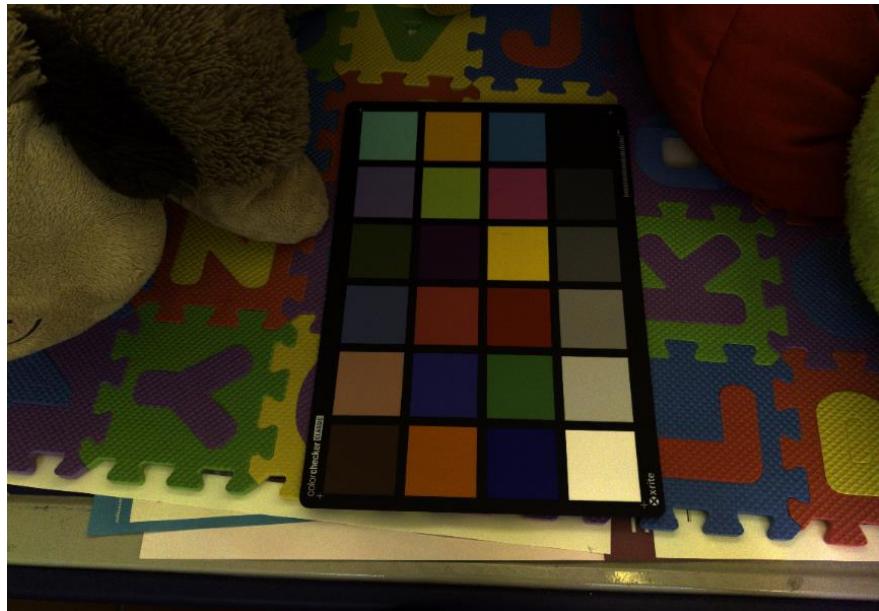
## Our CST estimation



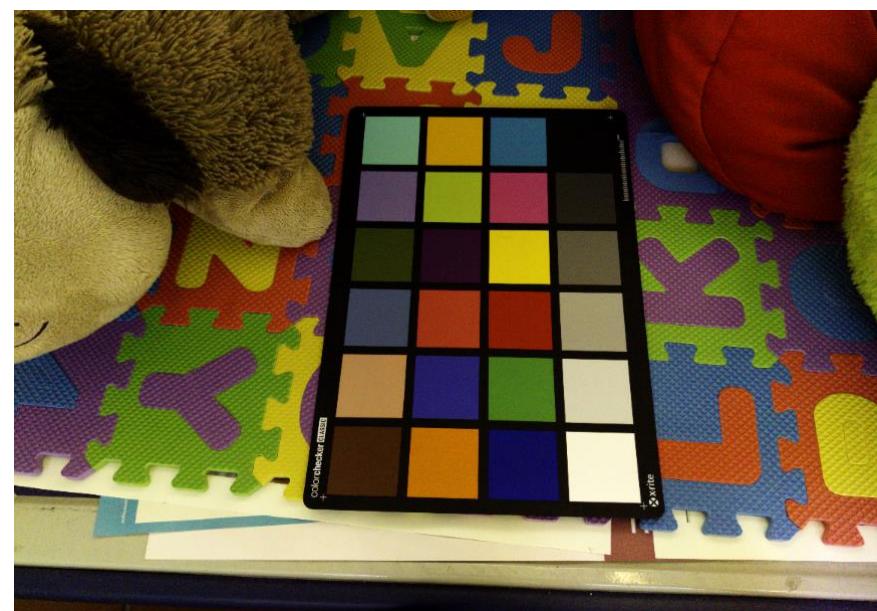
# True sRGB Example



# True sRGB vs. camera RGB



“True sRGB”

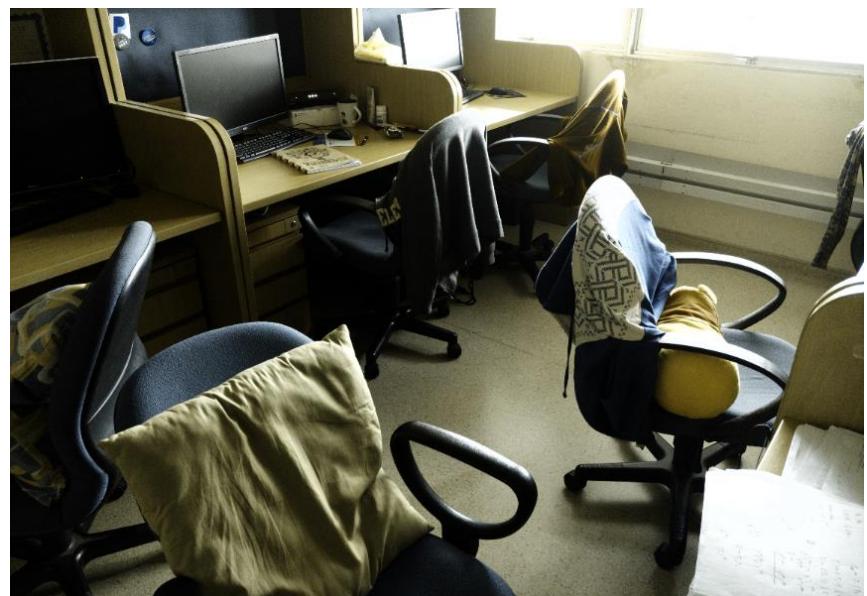


Camera sRGB

# True sRGB vs. camera RGB



“True sRGB”



Camera sRGB

# State of affairs (The Good)

- DNG allows access to processing pipeline parameters
- Android Camera2 API now supports DNG capture
- Developing a Matlab platform for researcher working on low-level vision



# Concluding remarks

- Your camera performs a great deal of processing
- RAW
  - Represents relatively unprocessed sensor response
  - Does not represent CIE XYZ
  - Transform to CIE XYZ is not readily available
  - There is no standard RAW!
- sRGB
  - Is defined with respect to CIE XYZ
  - However, incoming radiant energy (SPDs) to sRGB is not standardized (otherwise all camera images would look the same)

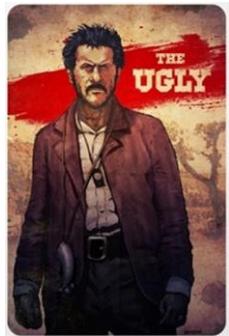
# Concluding remarks

- Our experiments found
  - Mapping from RAW to sRGB changes for different picture styles (portrait, landscape, vivid, standard, etc. . )
  - Picture styles can be modeled with a 3D function + tone-map
  - White-balance (so far) seems independent of picture styles

# Good, Bad, Ugly Summary



- Cameras are black boxes
- sRGB is not standard across cameras
- Makes it hard to develop apps



- Researchers need more education
- Lack of understanding
- Research being performed “out of context”



- Situation is (slowly) changing
- DNG + Android
- Platforms to support research are coming

# Embrace the in-camera processing

- Onboard photofinishing make our photos look great
- Embrace the technology!



# Thank you for attending



Li Yu



Hakki Karaimer



Rang Nguyen



Cheng Dongliang

**Special thanks to my students and colleagues for contributing images, code, and materials to this tutorial.**

# Understanding Color Perception



Hermann Grassmann



Johannes von Kries



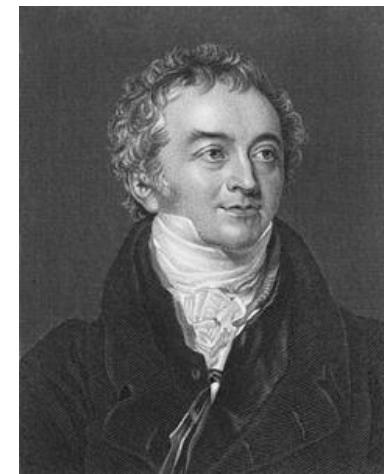
W. David Wright



James Clerk Maxwell



Hermann von Helmholtz



Thomas Young

# Digital Cameras



Bryce E. Bayer

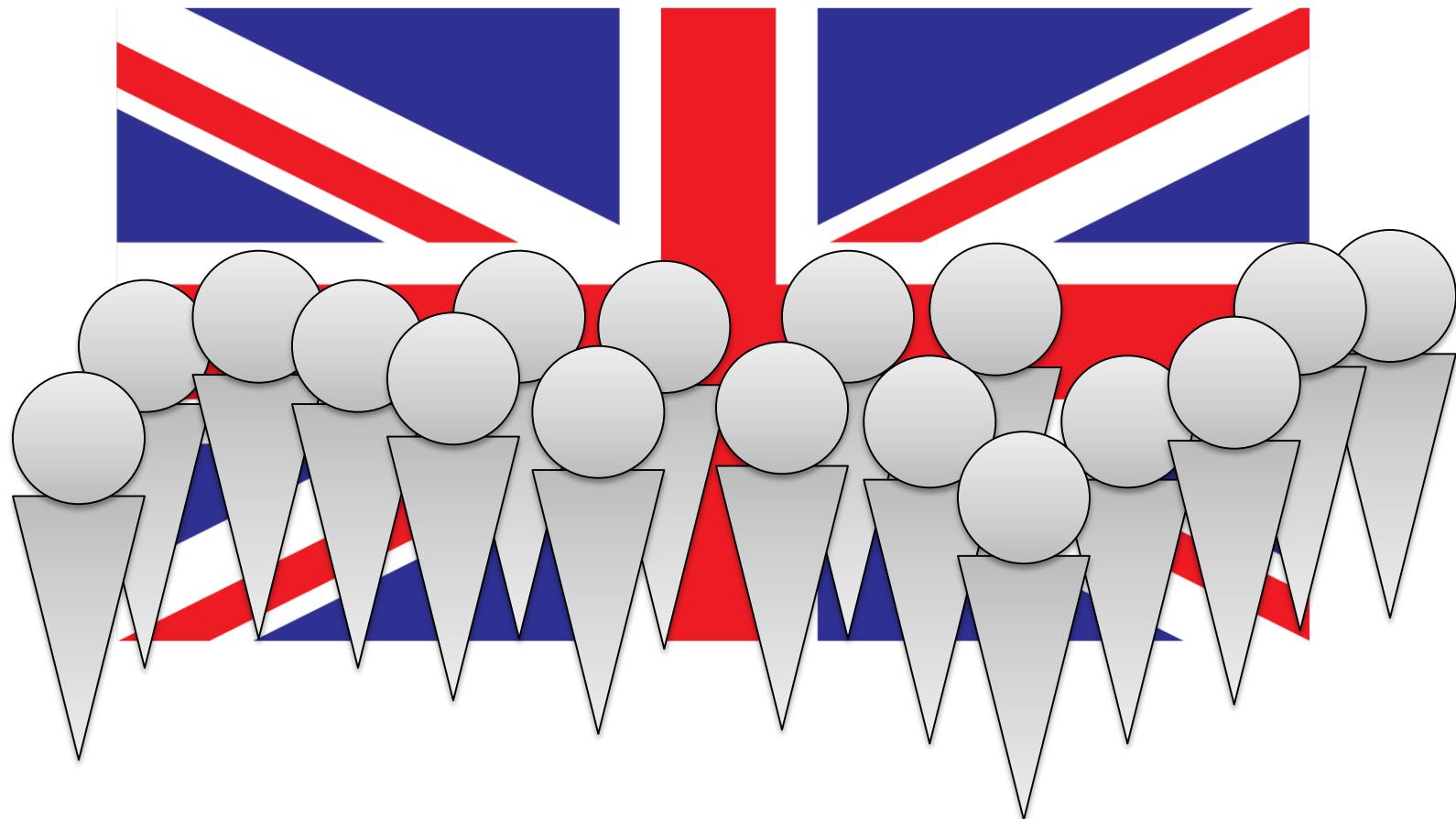


Willard Boyle and George Smith  
(Nobel Prize for inventing the CCD)  
Photo: Reuters



Eric R. Fossum  
(Invented CMOS)

# And of course



“The Standard Observers”

# References/additional materials

## Books

- G. Sharma, Digital Color Imaging Handbook, CRC Press, 2003
- M. Fairchild, Color Appearance Models, Wiley, 2005
- D. Forsyth & J. Ponce, Computer Vision: A modern approach, Prentice Hall, 2011
- R. Lukac, Single-Sensor Imaging: Methods and Applications for Digital Cameras, CRC Press, 2008

## Articles/Conference Papers

- R. Ramantha et al "Color Image Processing Pipeline: a general survey of digital still cameras", IEEE Signal Processing Magazine, Jan 2005
- H. Fairman et al "How the CIE 1931 Color-Matching Functions Were Derived from Wright–Guild Data", Color Research & Application, Feb 1997
- G. Meyer, "Tutorial on Color Science", The Visual Computer, 1986
- S. J. Kim et al "A New In-Camera Imaging Model for Color Computer Vision and its Application", IEEE Transactions on Pattern Analysis and Machine Intelligence, Dec 2012
- E. Garcia, M. Gupta , "Building accurate and smooth ICC profiles by lattice regression", Color and Imaging Conference, 2009
- A. Chakrabarti et al. "Modeling Radiometric Uncertainty for Vision with Tone-mapped Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014
- Cheng D. L., Price B., Cohen S., Brown M. S. (2015) "Beyond White: Ground Truth Colors for Color Constancy Correction", *IEEE International Conference on Computer Vision*, Dec 2015
- Tai Y.-W. , Chen X., Kim S., Kim S. J, Li F., Yang J., Yu J., Matsushita Y., Brown M. S. (2013) "Nonlinear Camera Response Functions and Image Deblurring: Theoretical Analysis and Practice", *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(10), Oct 2013

# References/additional materials

## Websites/Online Resources

**ICC – ISO 22028 (2012) Document**

**Photo Tech Edu Series Jan 2007 (Google)**

Available on youtube

Richard Lyon: Digital Camera Image Processing Pipelines

**<http://www.photo-mark.com>**

Mark Meyer Photography

Professional photography blog with nice tie-ins to color from a  
photographers point of view

**<http://dougkerr.net>**

Doug Kerr postings

A large collection of self-published articles on various aspects of imaging in  
an “accessible language” to most readers

# Acknowledgements

This work was supported in part by generous gifts from:

- Adobe Research
- Google Research (Faculty Award 2015)