photo by unsplash user @ro_la

COMP527
COMPUTATIONAL IMAGING

Lecture #03 – Noise and Color

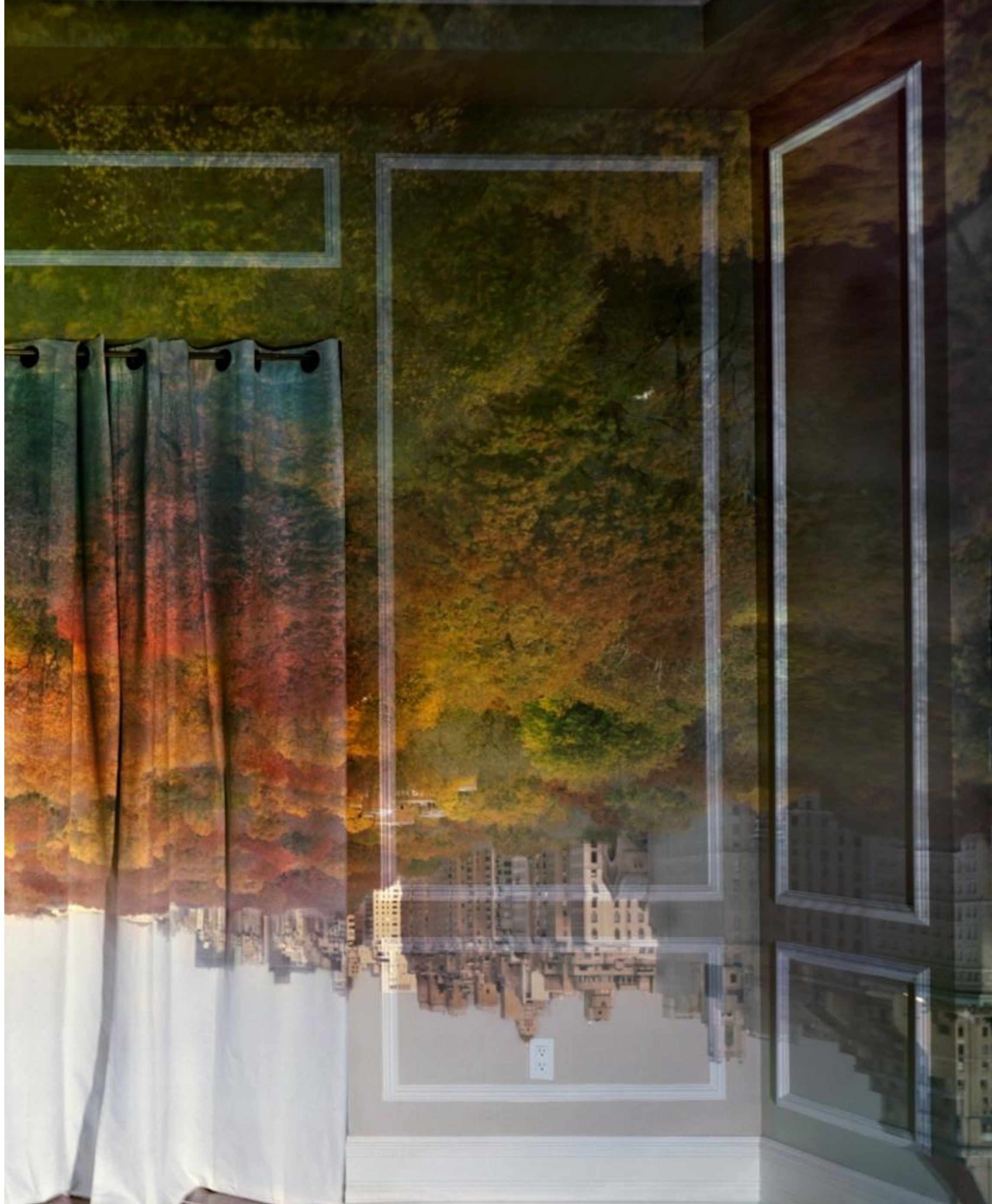
KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2023

Previously on COMP527

- Pinhole camera
- Basics of geometric optics and lenses
- Field of view
- Magnification and perspective
- Zooming
- Orthographic camera and telecentric lenses

View of Central Park Looking North—Fall by Abelardo Morell, 2008



Today's Lecture

- Digital photography
- Standard camera pipeline
- Noise
- Color

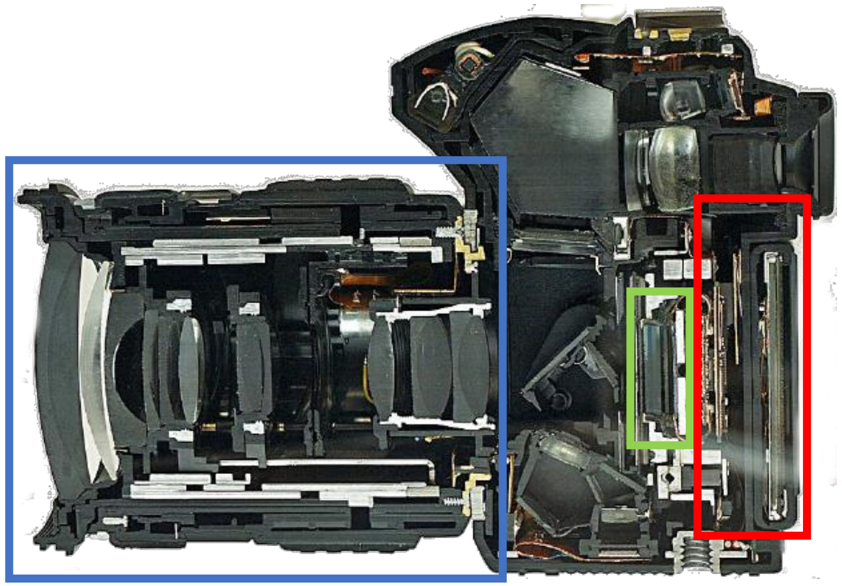
Disclaimer: The material and slides for this lecture were borrowed from
—Ioannis Gkioulekas' 15-463/15-663/15-862 "Computational Photography" class
—Steve Marschner's CS6640 "Computational Photography" class

Today's Lecture

- Digital photography
- Standard camera pipeline
- Noise
- Color

The modern photography pipeline

post-capture processing



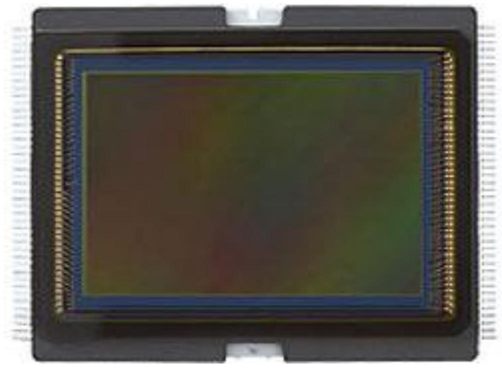
optics and optical controls → sensor, analog front-end, and color filter array → in-camera image processing pipeline

Imaging sensor primer

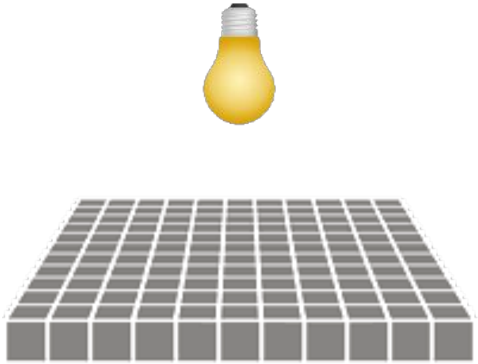
What does an imaging sensor do?

When the camera shutter opens…

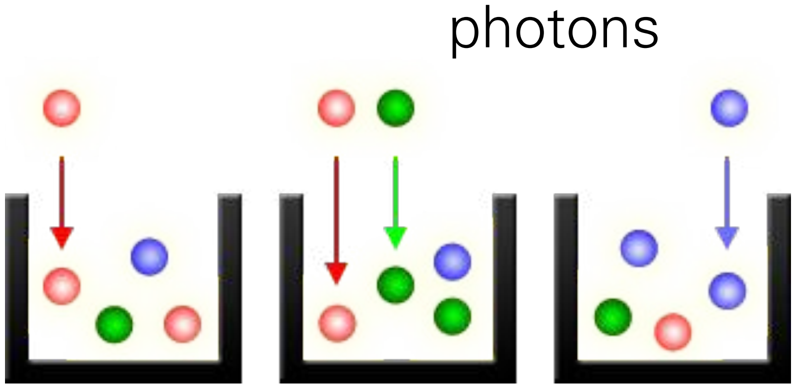
… exposure begins…



Canon 6D sensor
(20.20 MP, full-frame)



array of photon buckets



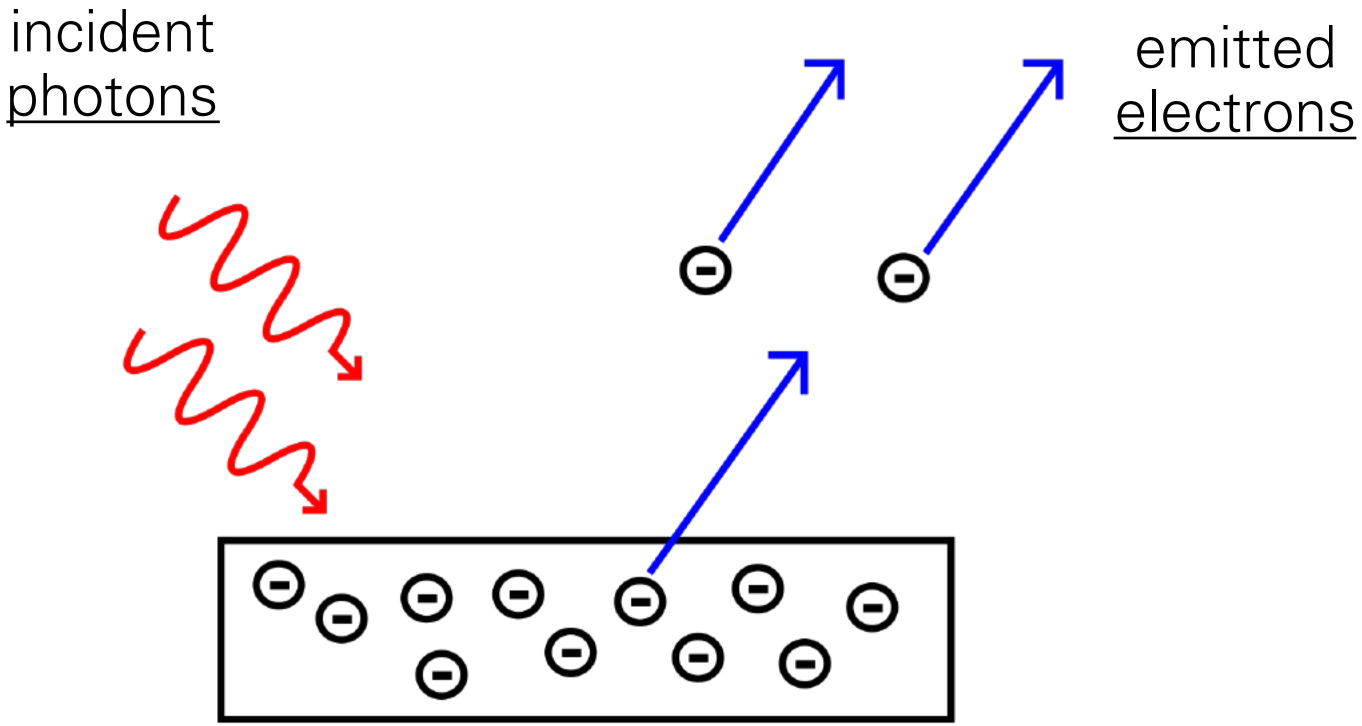
close-up view of photon buckets

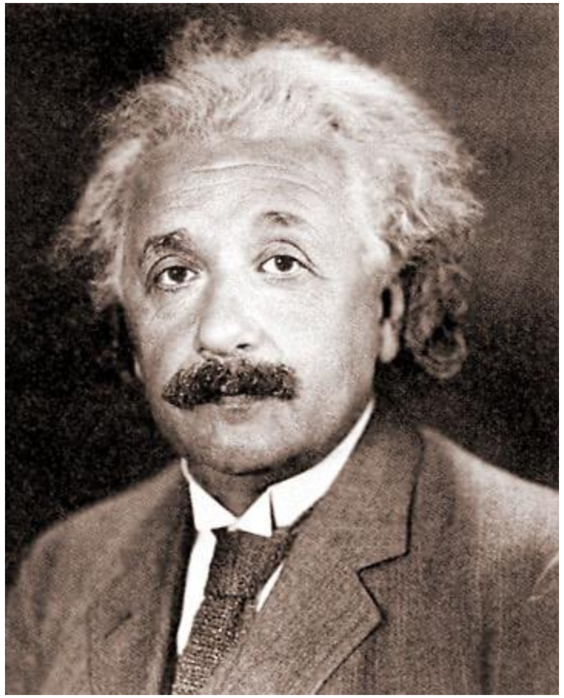
… photon buckets begin to store photons...

… until the camera shutter closes. Then, they convert stored photons to intensity values.

Nikon D3s

Photoelectric effect

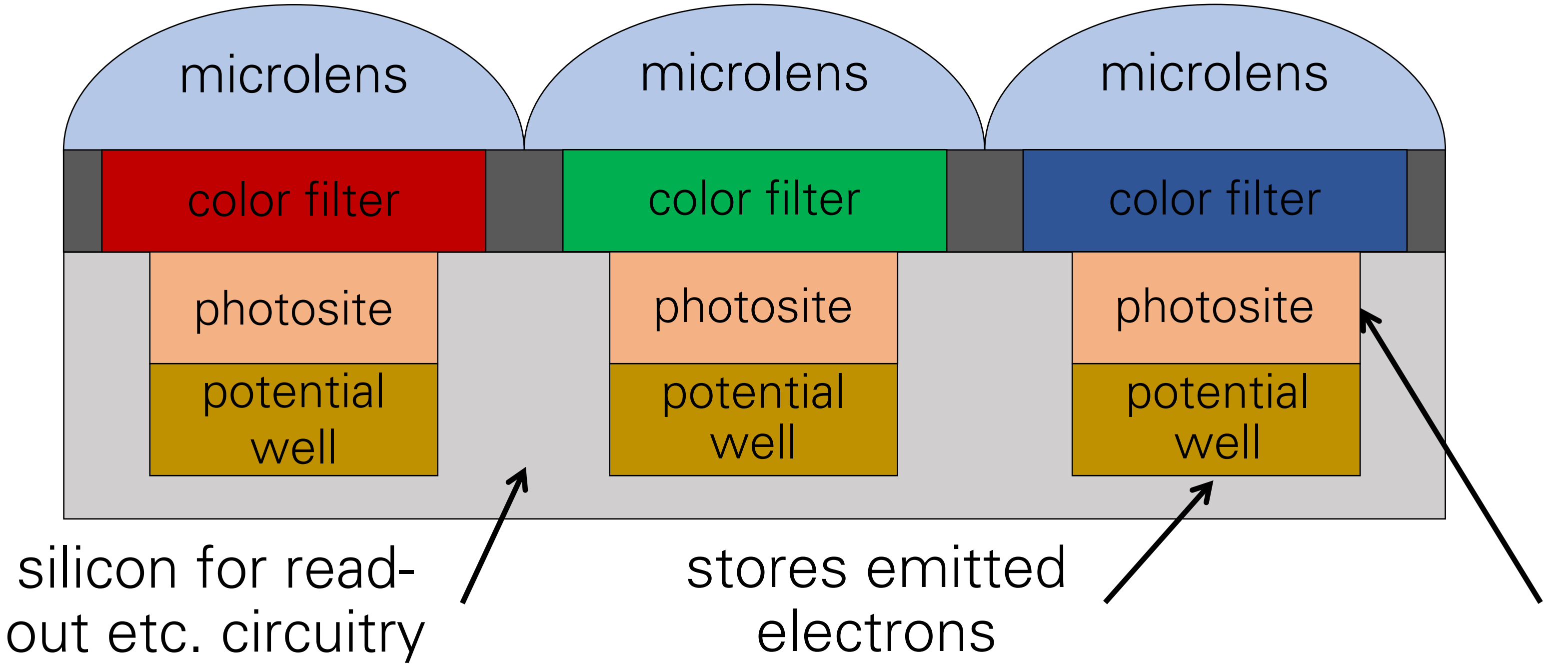


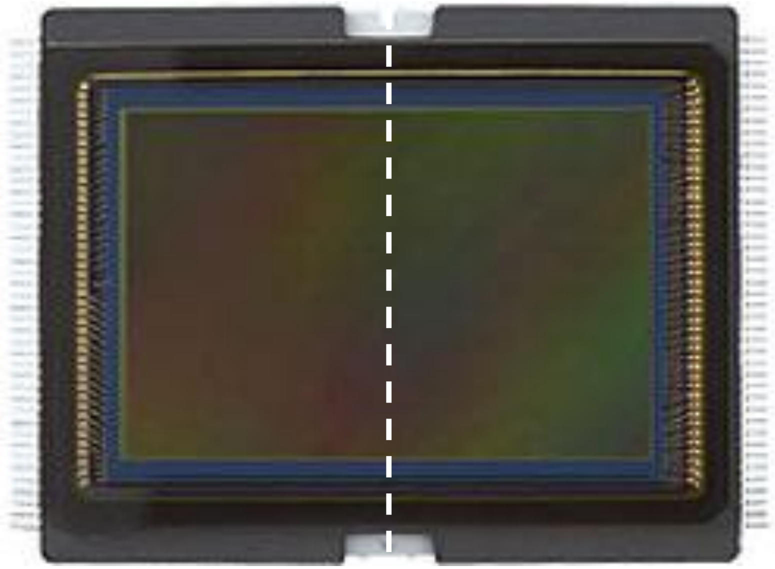


Albert Einstein

Einstein's Nobel Prize in 1921 "for his services to Theoretical Physics, and especially for his discovery of the law of the photoelectric effect"

Basic imaging sensor design





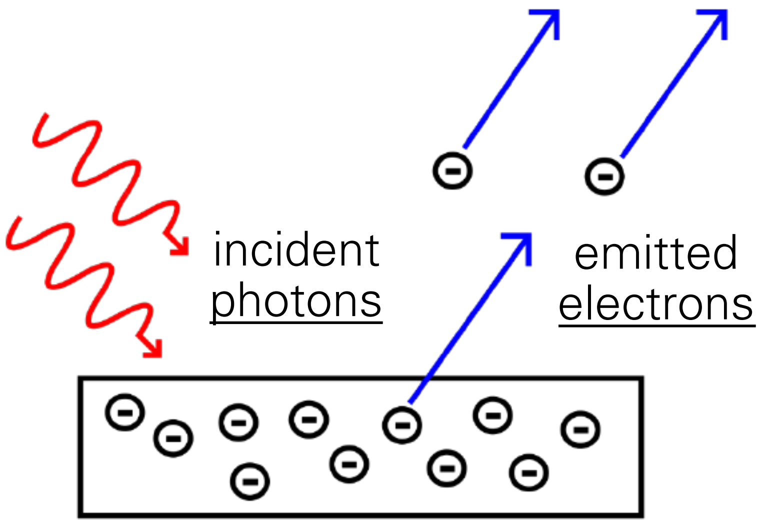
Canon 6D sensor
(20.20 MP, full-frame)

The term "photosite" can be used to refer to both the entire pixel and only the photo-sensitive area.

Photosite quantum efficiency (QE)

How many of the incident photons will the photosite convert into electrons?

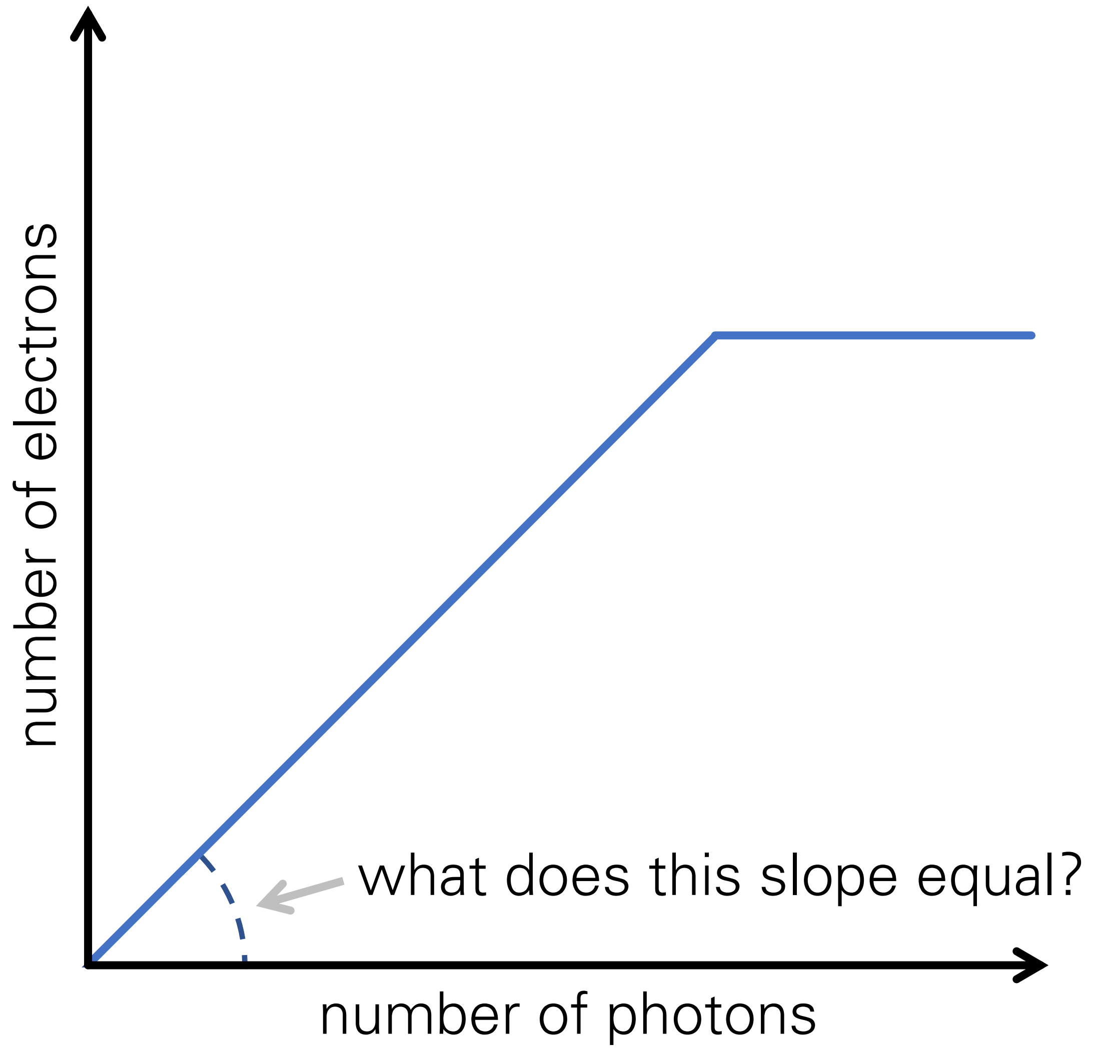
$$QE = \frac{\#\ \text{electrons}}{\#\ \text{photons}}$$



- Fundamental optical performance metric of imaging sensors.
- Not the only important optical performance metric!

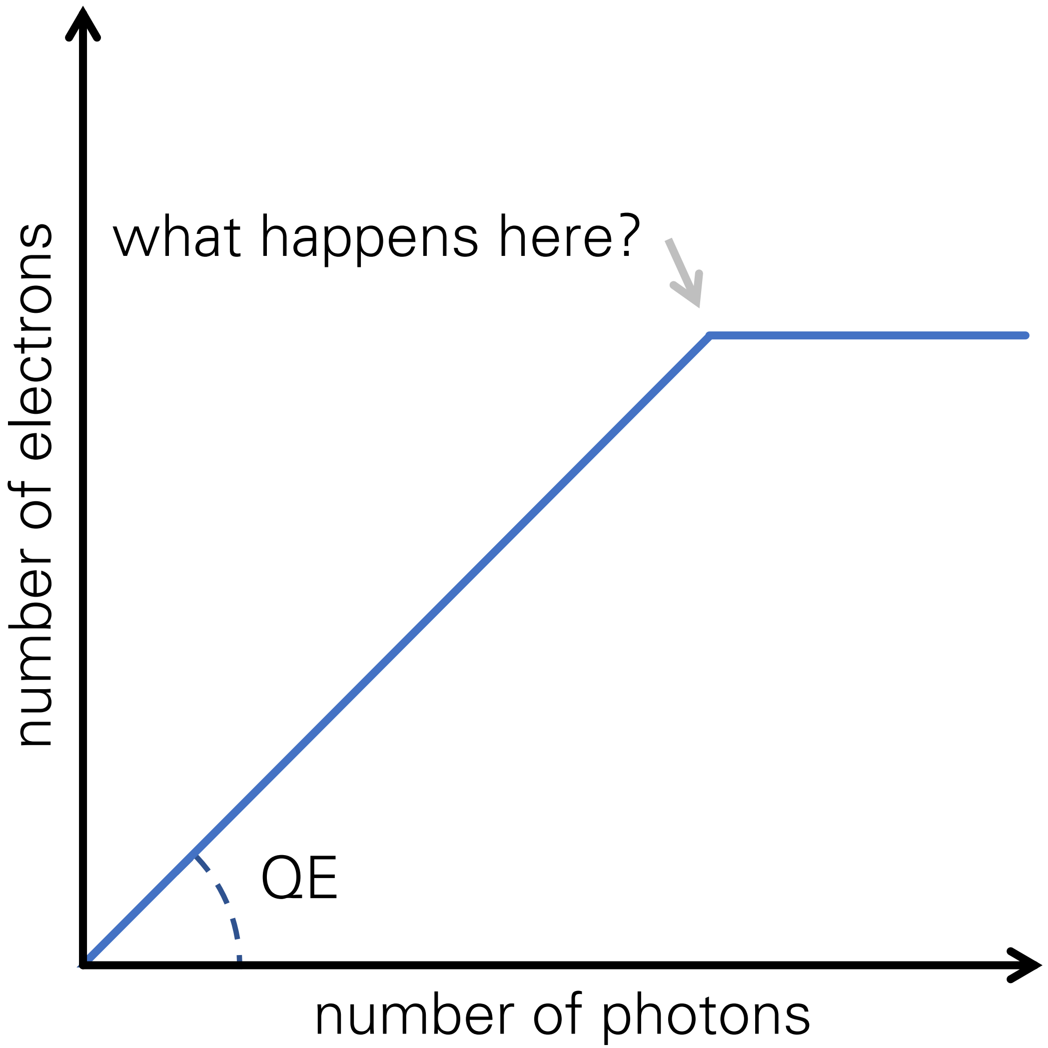
Photosite response

The photosite response is mostly linear



Photosite response

The photosite response is <u>mostly</u> linear



Photosite response

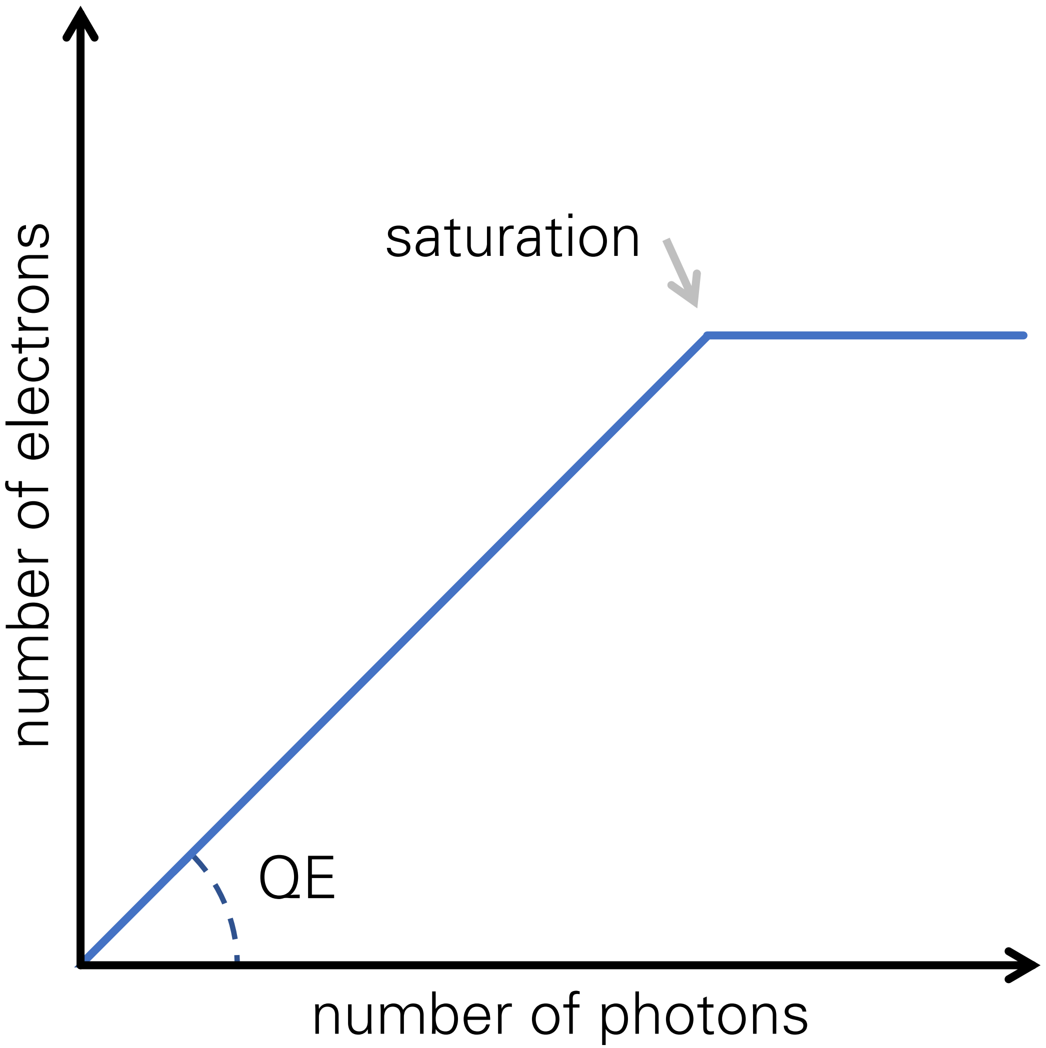
The photosite response is mostly linear, but:

- non-linear when potential well is saturated (over-exposure)
- non-linear near zero (due to noise)

We will see how to deal with these issues in a later lecture (high-dynamic-range imaging).

under-exposure (non-linearity due to sensor noise)

over-exposure (non-linearity due to sensor saturation)

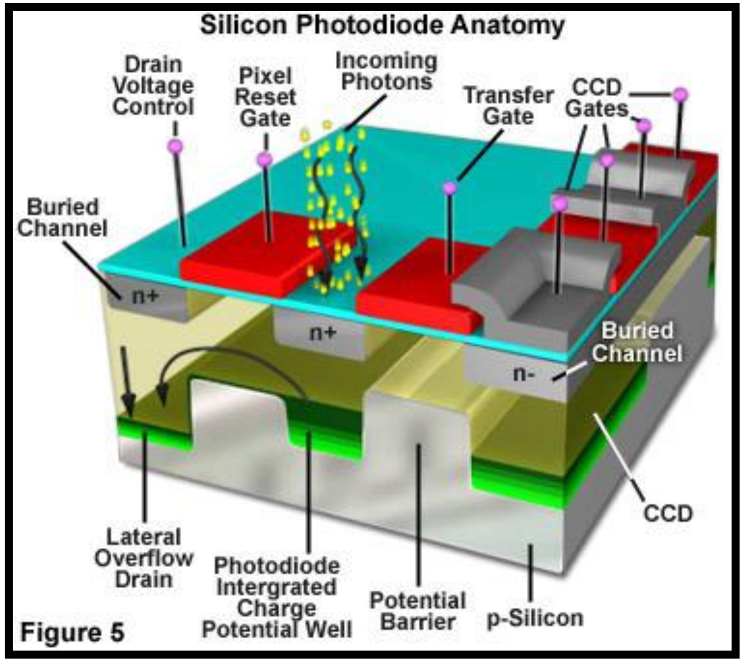


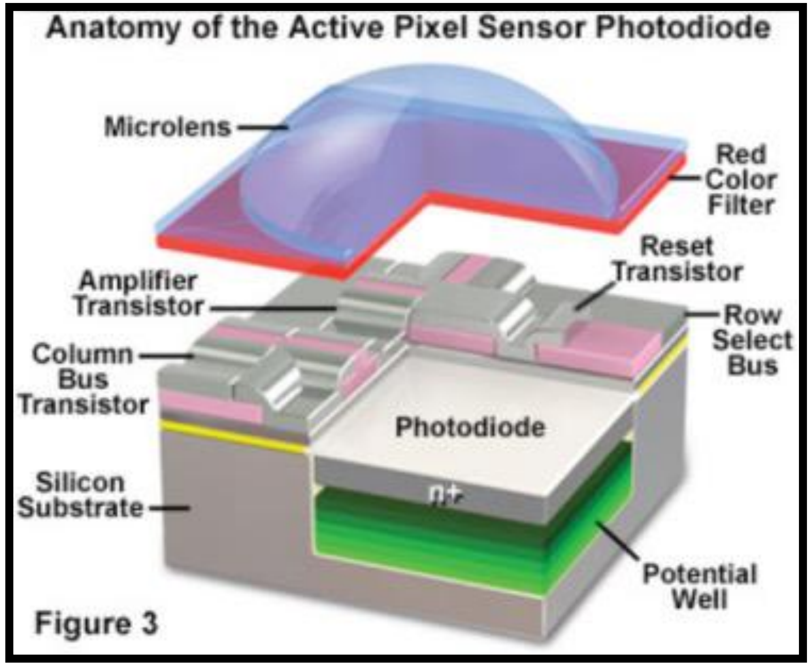
Saturation means that the potential well is full before exposure ends.

Two main types of imaging sensors

Do you know them?

Two main types of imaging sensors





Charged coupled device (CCD):

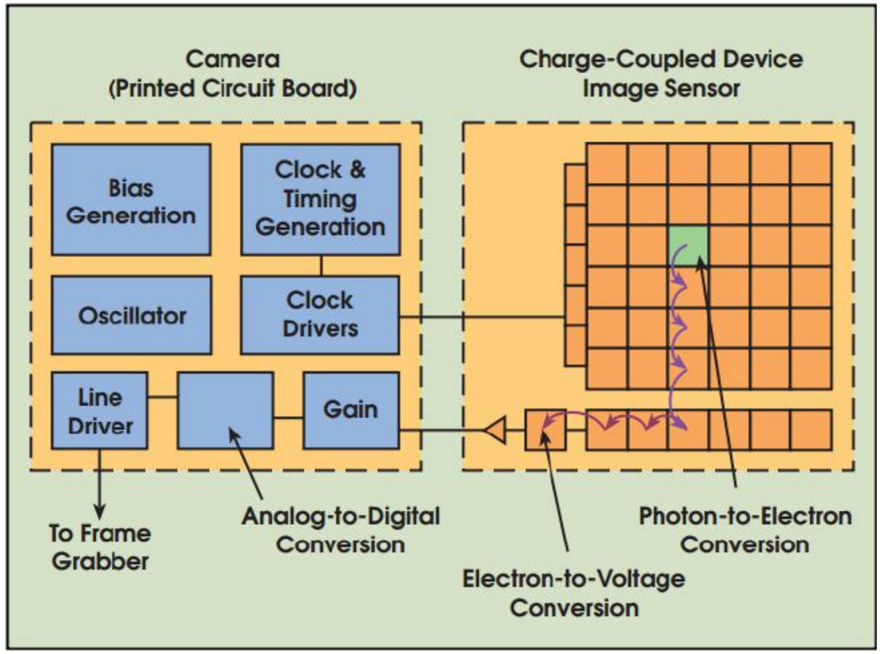
- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

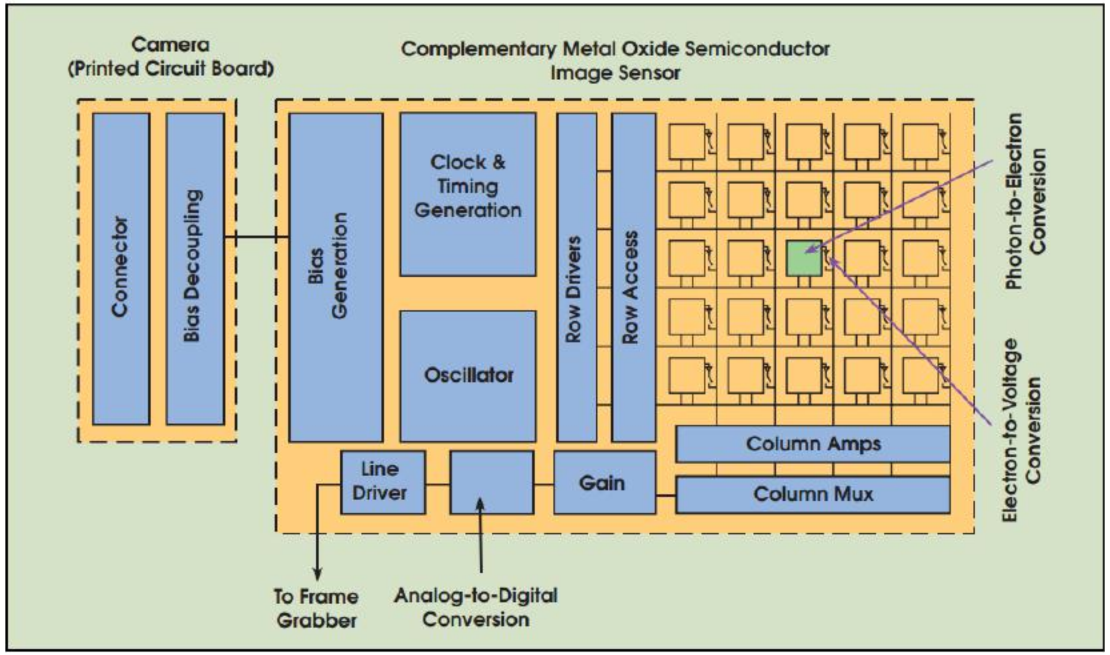
Complementary metal oxide semiconductor (CMOS):

- per-pixel amplifiers convert charges to voltages
- multiplexer reads voltages row-by-row

Can you think of advantages and disadvantages of each type?

Two main types of imaging sensors



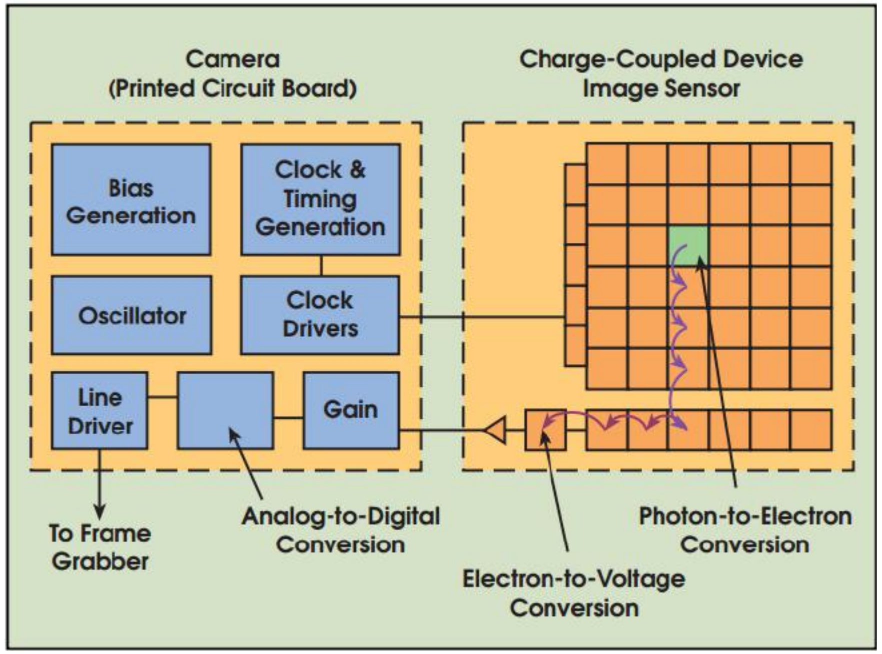


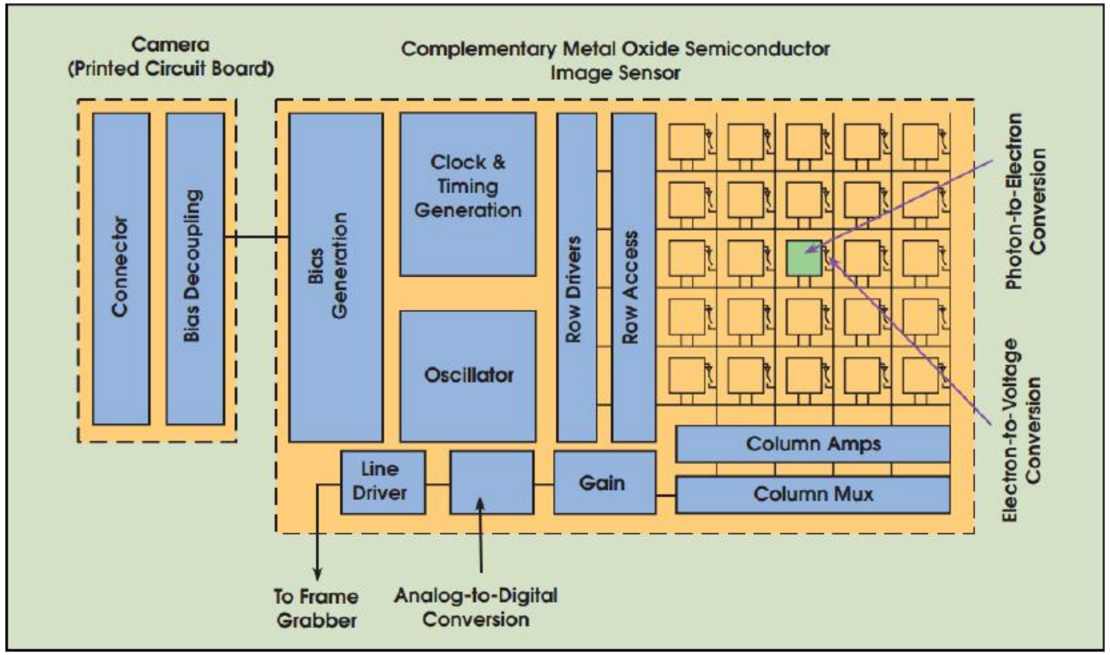
Charged coupled device (CCD):
- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

Complementary metal oxide semiconductor (CMOS):
- per-pixel amplifiers convert charges to voltages
- multiplexer reads voltages row-by-row

Can you think of advantages and disadvantages of each type?

Two main types of imaging sensors





Charged coupled device (CCD):

- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

✓ higher sensitivity

✓ lower noise

Complementary metal oxide semiconductor (CMOS):

- per-pixel amplifiers convert charges to voltages
- multiplexer reads voltages row-by-row

✓ faster read-out

✓ lower cost

Artifacts of the two types of sensors

sensor bloom

smearing artifacts

Which sensor type can have these artifacts?

Artifacts of the two types of sensors

sensor bloom
(CMOS and CCD)

smearing artifacts
(CCD only)

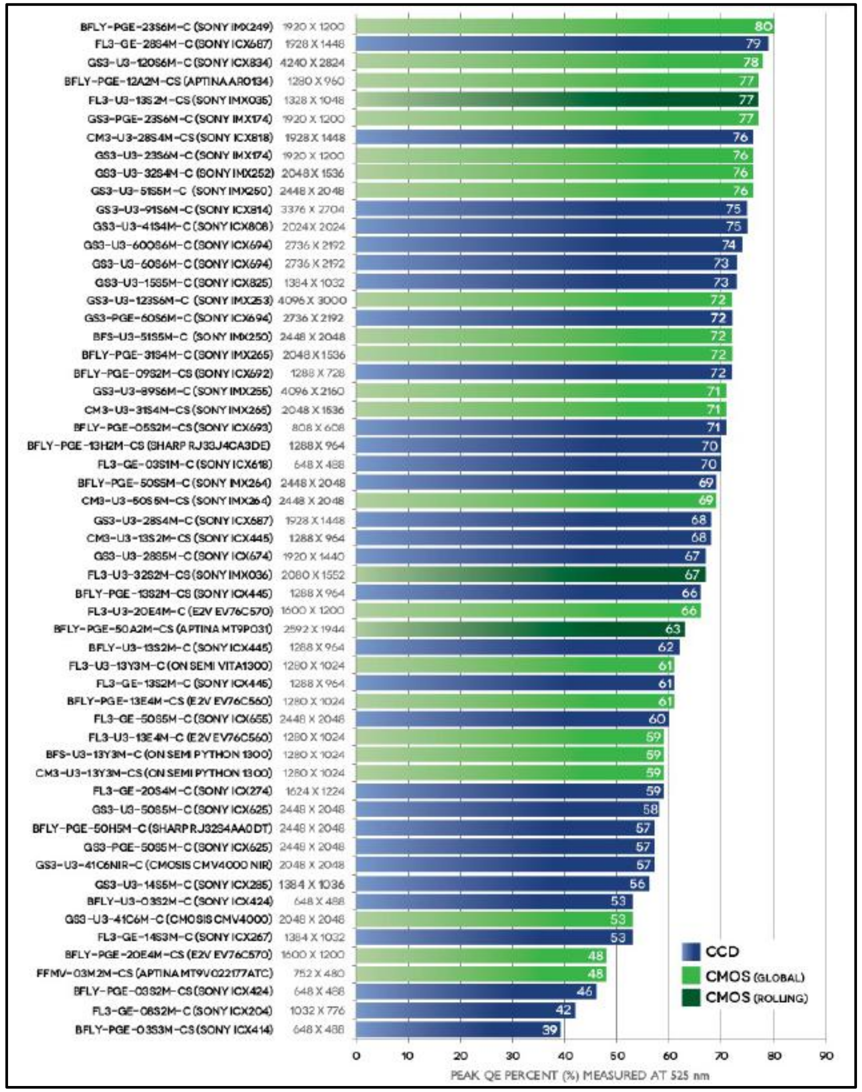
Overflow from saturated pixels

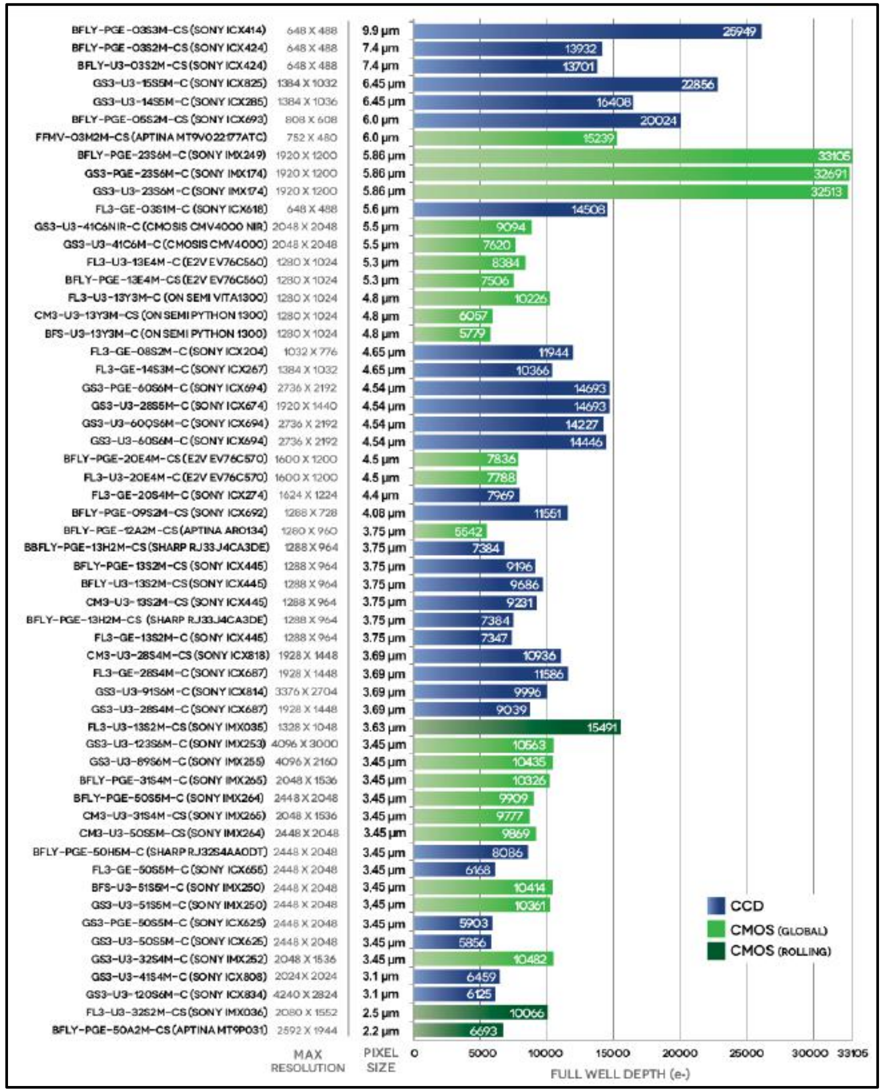
- mitigated by more electronics to contain charge
  (at the cost of photosensitive area)

CCD vs CMOS

- Modern CMOS sensors have optical performance comparable to CCD sensors.

- Most modern commercial and industrial cameras use CMOS sensors.





What does an imaging sensor do?

When the camera shutter opens, the sensor:

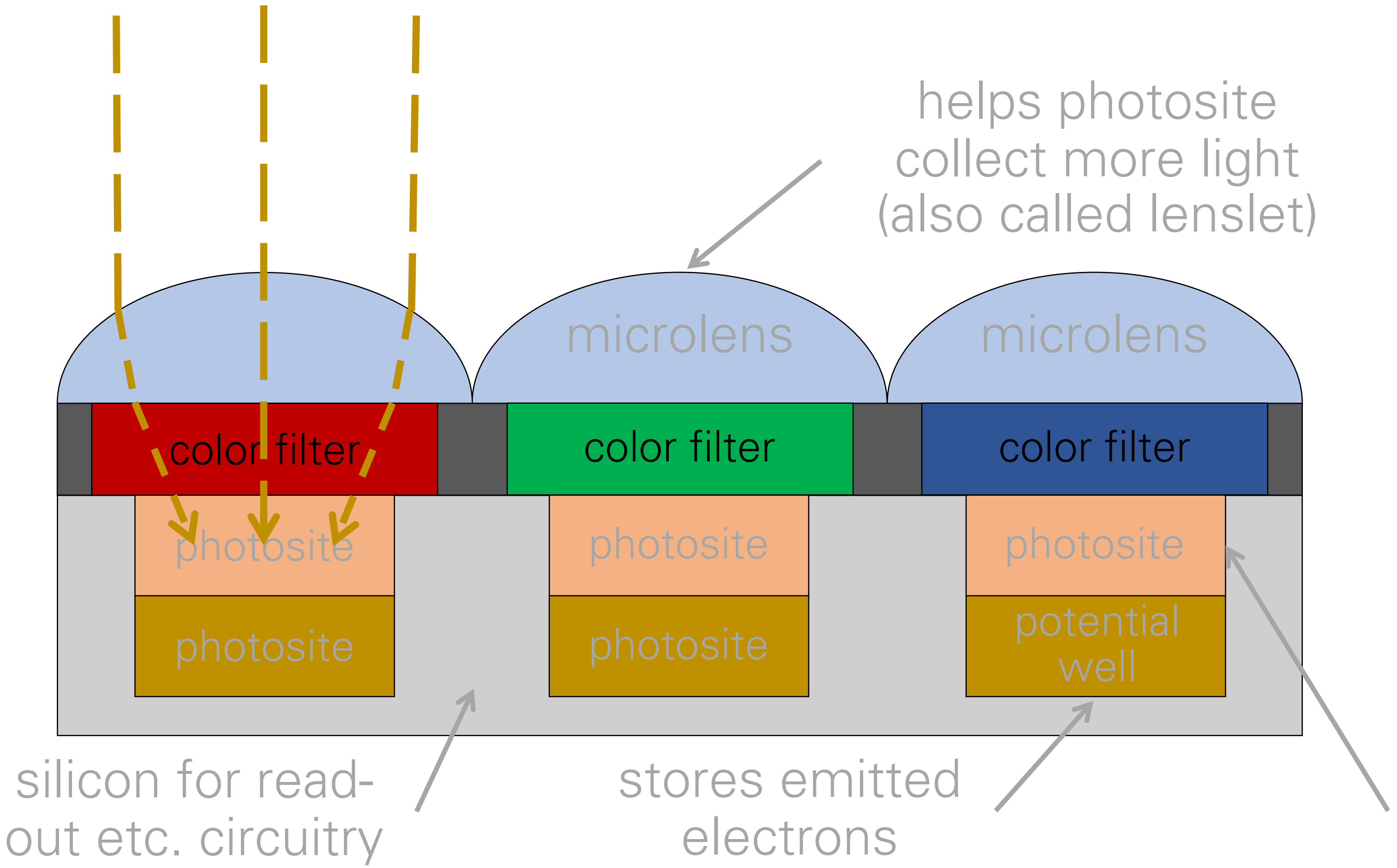
- at every photosite, converts incident photons into electrons
- stores electrons into the photosite’s potential well while it is not full

… until camera shutter closes. Then, the analog front-end:

- reads out photosites’ wells, row-by-row, and converts them to analog signals
- applies a (possibly non-uniform) gain to these analog signals
- converts them to digital signals
- corrects non-linearities

… and finally returns an image.

Remember these?



- Lenslets also filter the image to avoid resolution artifacts.
- Lenslets are problematic when working with coherent light.
- Many modern cameras do not have lenslet arrays.

We will see what the color filters are for later in this lecture.

Color primer

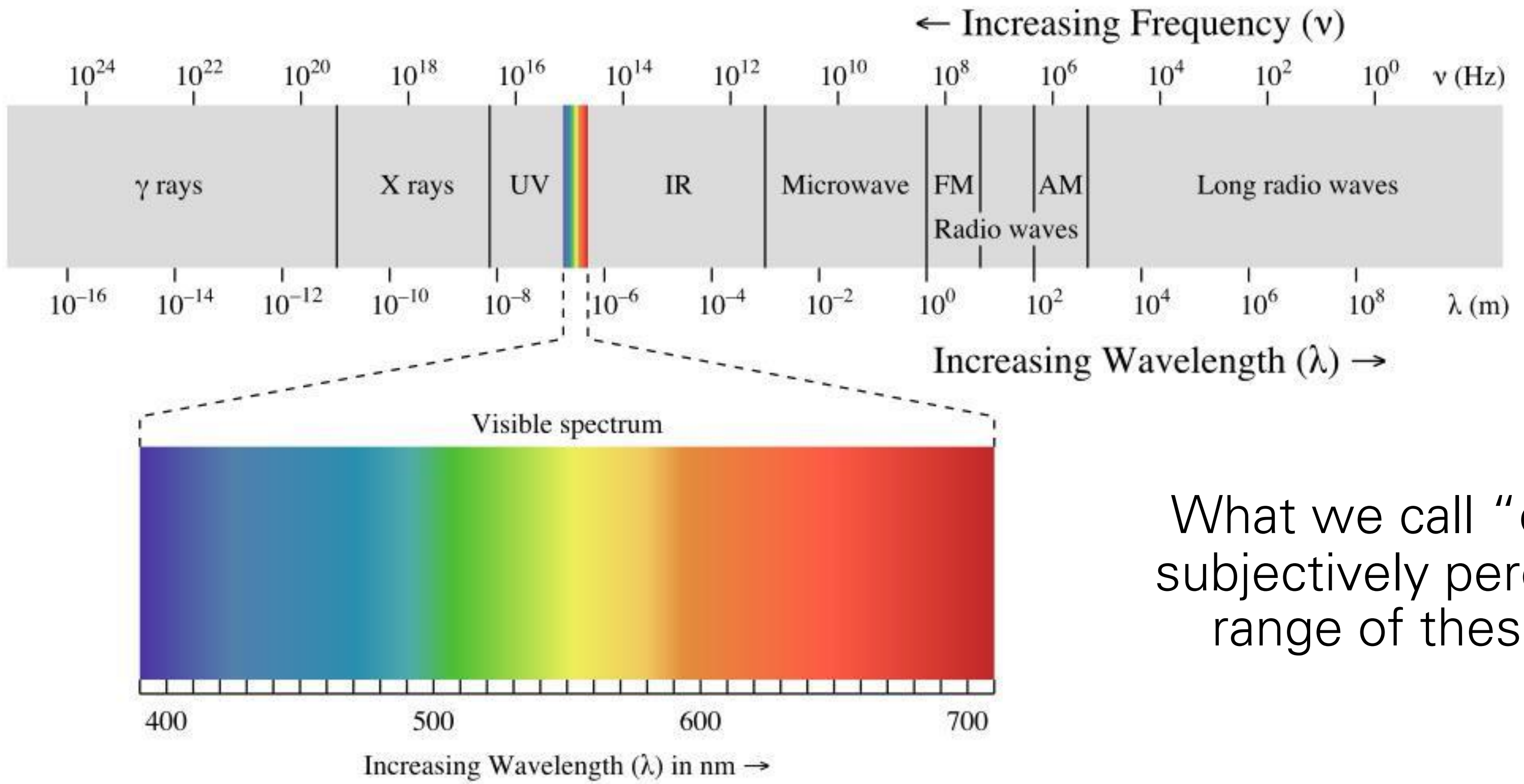
Color

- Very high-level of color as it relates to digital photography.
- We could spend an entire course covering color.
- We will discuss color in more detail in later today.

color is complicated

Color is an artifact of human perception

- "Color" is not an objective physical property of light (electromagnetic radiation).
- Instead, light is characterized by its wavelength.



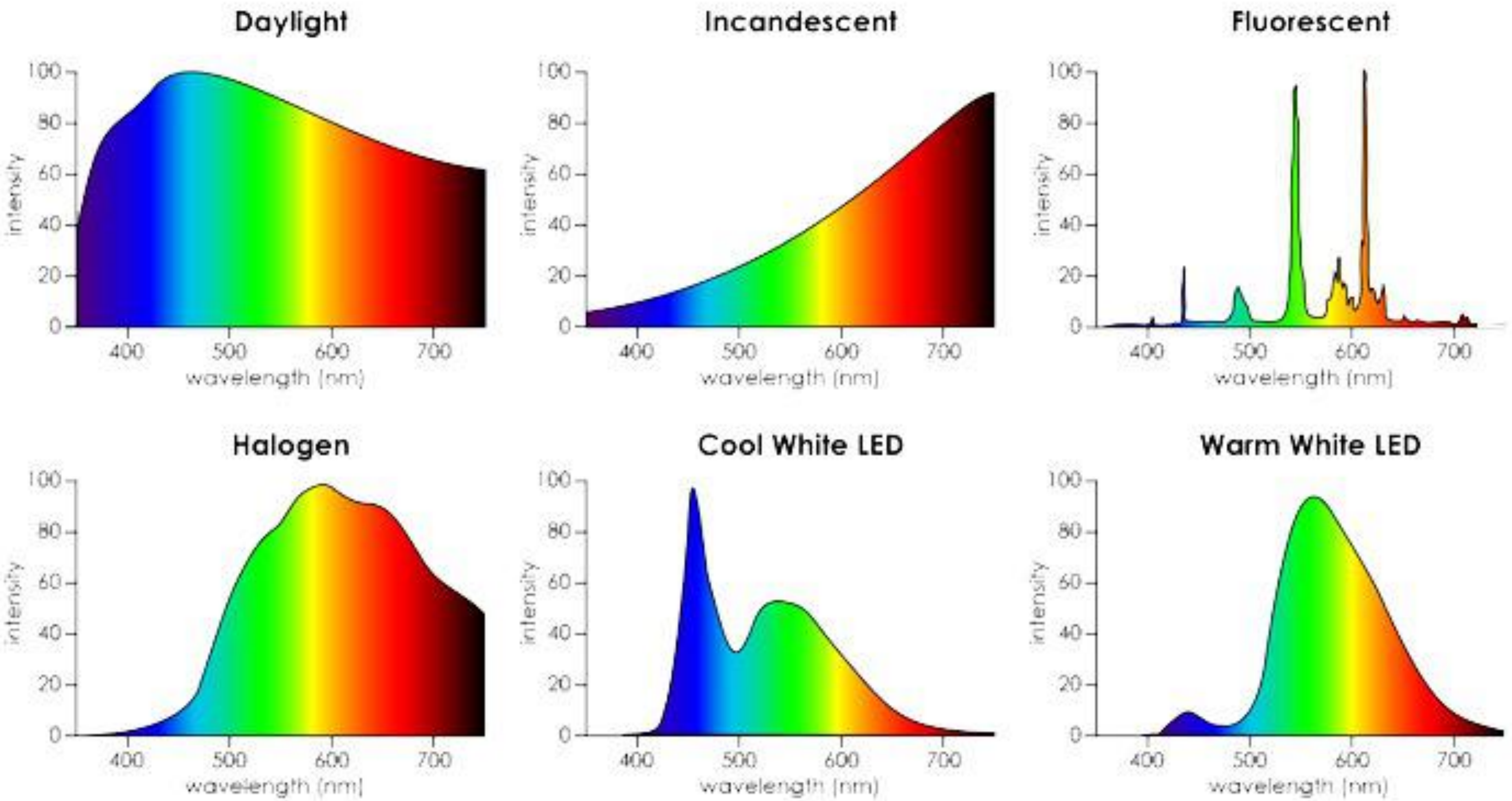
electromagnetic spectrum

What we call "color" is how we subjectively perceive a very small range of these wavelengths.

Spectral Power Distribution (SPD)

- Most types of light “contain” more than one wavelengths.
- We can describe light based on the distribution of power over different wavelengths.

We call our sensation of all of these distributions “white”.



Spectral Sensitivity Function (SSF)

- Any light sensor (digital or not) has different sensitivity to different wavelengths.
- This is described by the sensor's spectral sensitivity function $f(\lambda)$.
- When measuring light of a some SPD $\Phi(\lambda)$, the sensor produces a scalar response:

light SPD sensor SSF

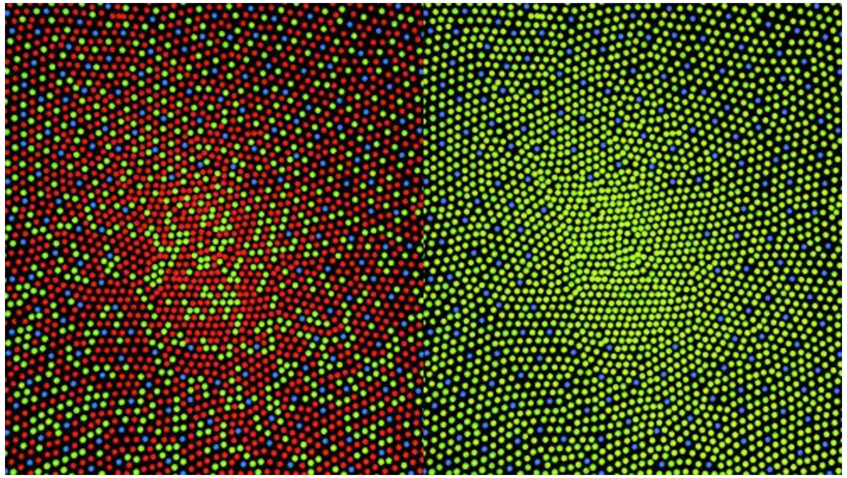
sensor response

$$R = \int_\lambda \Phi(\lambda) f(\lambda) d\lambda$$

Weighted combination of light's SPD: light contributes more at wavelengths where the sensor has higher sensitivity.

Spectral Sensitivity Function of Human Eye

- The human eye is a collection of light sensors called cone cells.
- There are three types of cells with different spectral sensitivity functions.
- Human color perception is three-dimensional (tristimulus color).

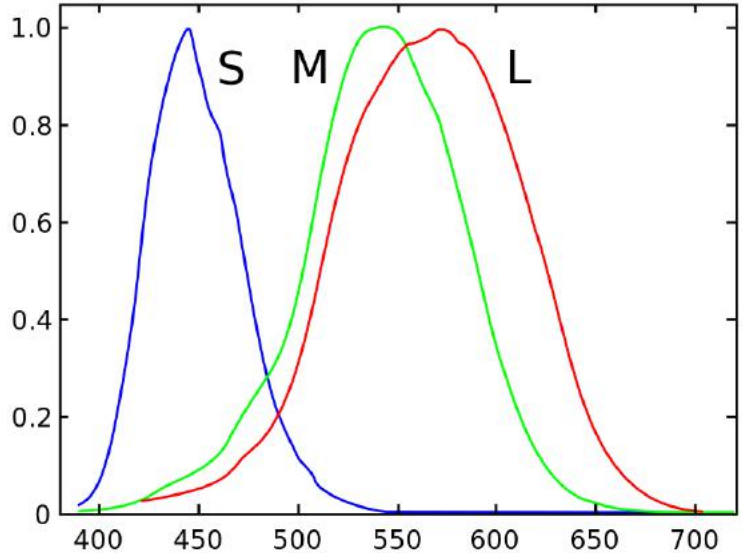


cone distribution
for normal vision
(64% L, 32% M)

"short" $S = \int_\lambda \Phi(\lambda) S(\lambda) d\lambda$

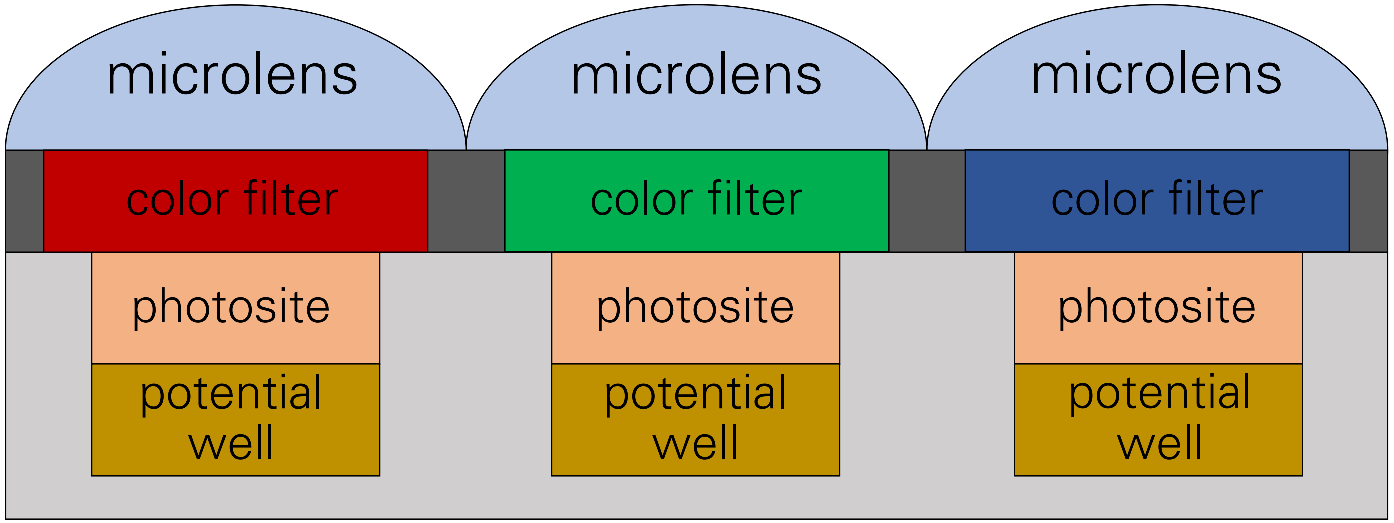
"medium" $M = \int_\lambda \Phi(\lambda) M(\lambda) d\lambda$

"long" $L = \int_\lambda \Phi(\lambda) L(\lambda) d\lambda$



Color filter arrays (CFA)

- To measure color with a digital sensor, mimic cone cells of human vision system.
- “Cones” correspond to pixels that are covered by different color filters, each with its own spectral sensitivity function.

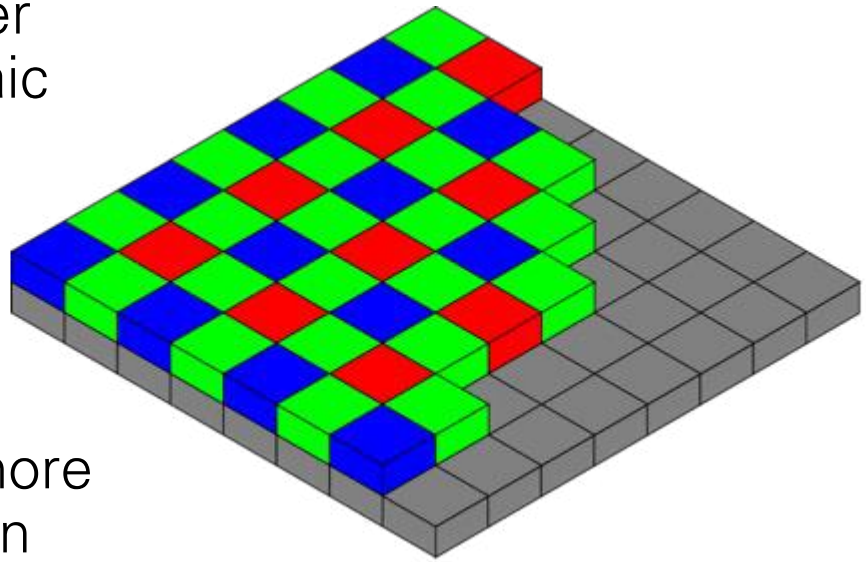


What color filters to use?

Two design choices:

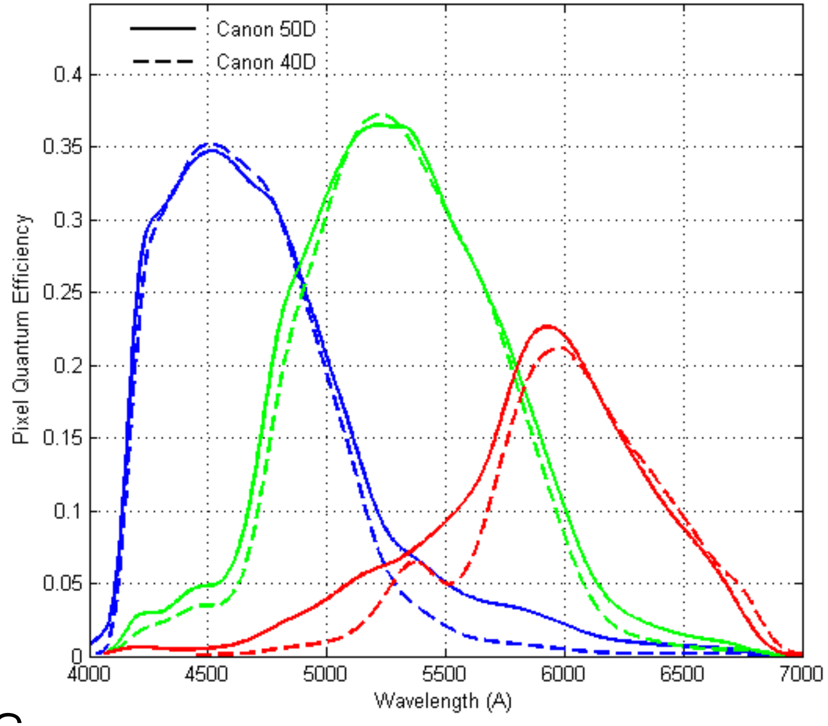
- What spectral sensitivity functions $f(\lambda)$ to use for each color filter?
- How to spatially arrange ("mosaic") different color filters

Bayer mosaic



Why more green pixels?

SSF for Canon 50D



Generally do not match human LMS.

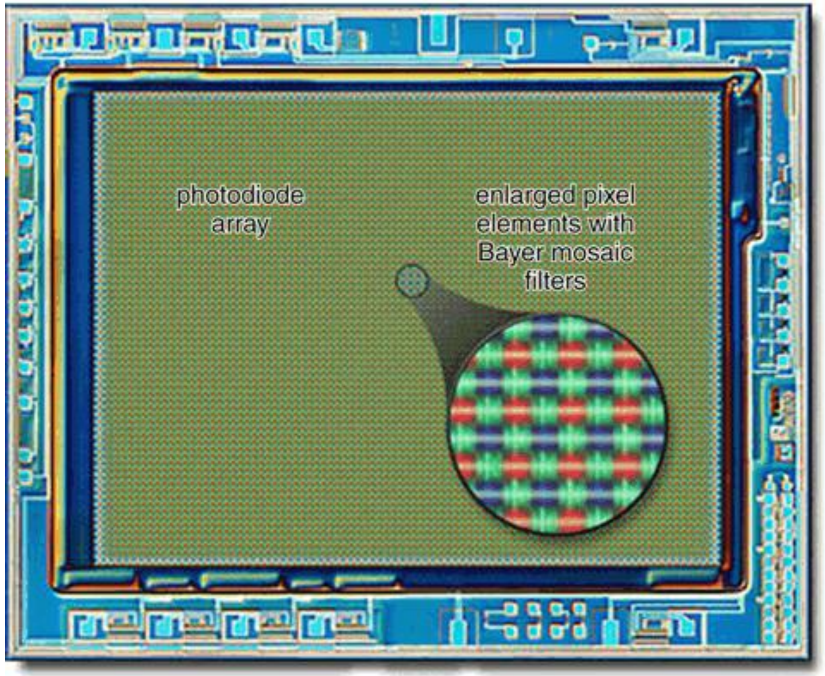
$f(\lambda)$

What color filters to use?

Two design choices:

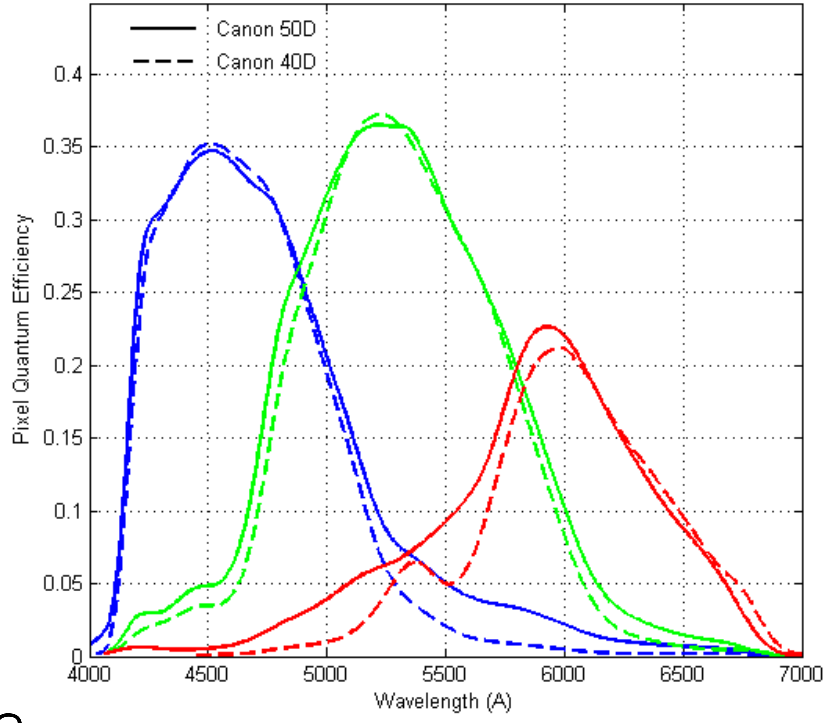
- What spectral sensitivity functions $f(\lambda)$ to use for each color filter?
- How to spatially arrange ("mosaic") different color filters

Bayer mosaic



http://www.currentprotocols.com/WileyCDA/CPUnit/refId-ns0204.html

SSF for Canon 50D

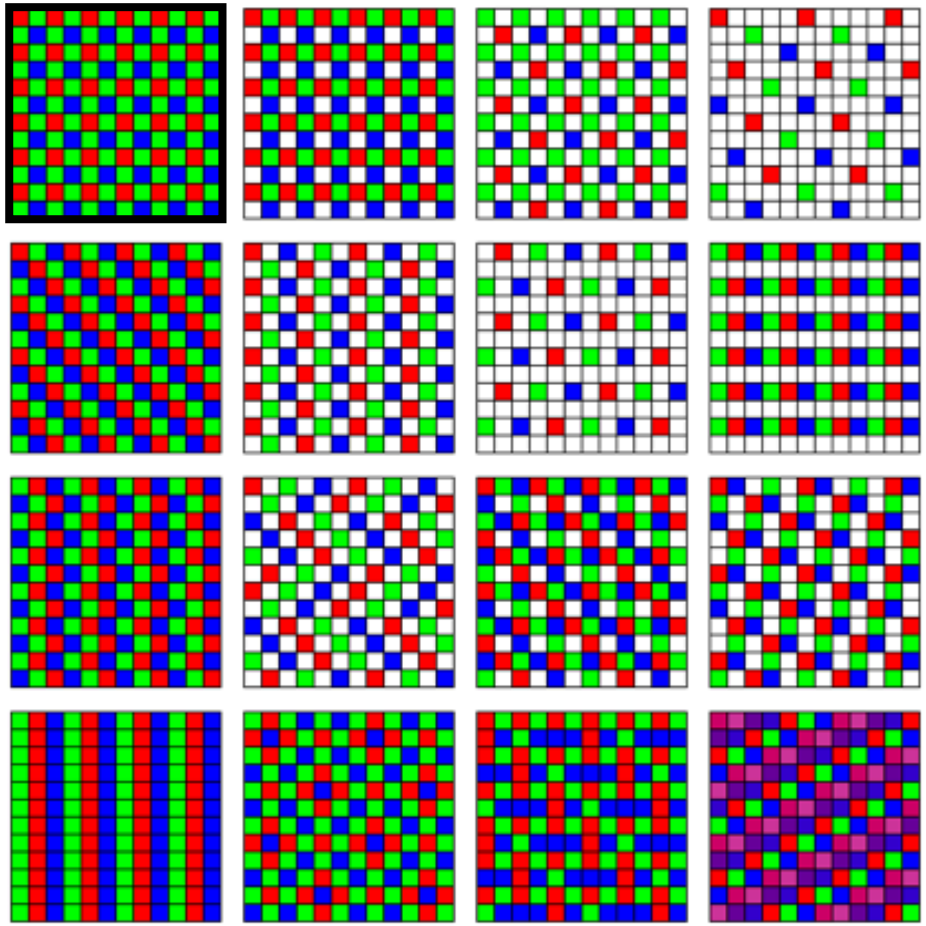


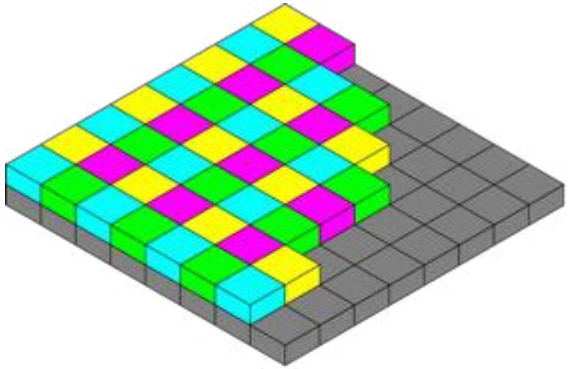
Generally do not match human LMS.

$f(\lambda)$

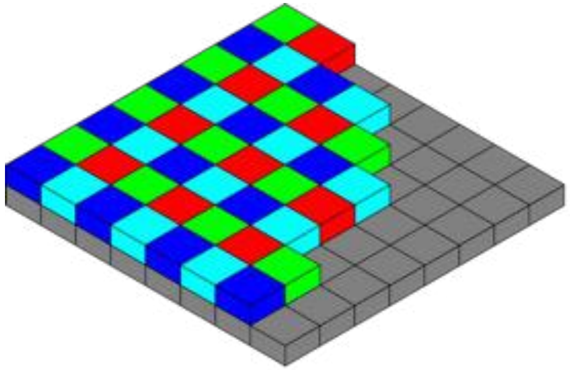
Many different CFAs

Finding the “best” CFA mosaic is an active research area.





CYGM
Canon IXUS, Powershot



RGBE
Sony Cyber-shot

How would you go about designing your own CFA? What criteria would you consider?

Many different spectral sensitivity functions

Each camera has its more or less unique, and most of the time secret, SSF.

- Makes it very difficult to correctly reproduce the color of sensor measurements.
- We will see more about this in the color lecture.

Images of the same scene captured using 3 different cameras with identical settings.

What does an imaging sensor do?

When the camera shutter opens, the sensor:

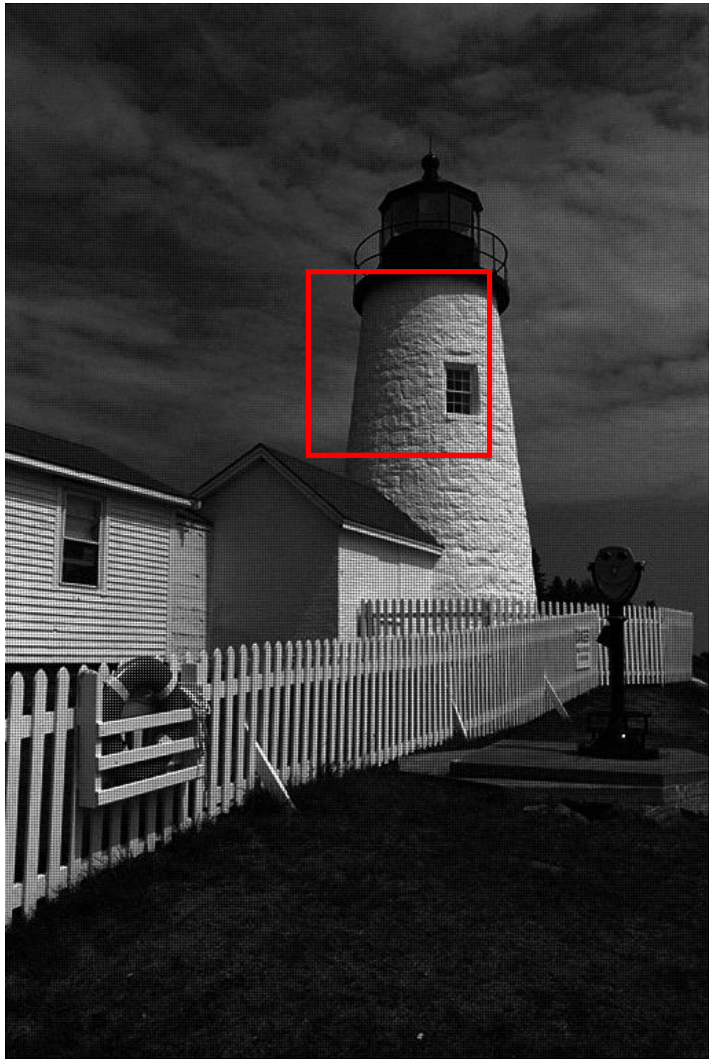
- at every photosite, converts incident photons into electrons using mosaic's SSF
- stores electrons into the photosite's potential well while it is not full

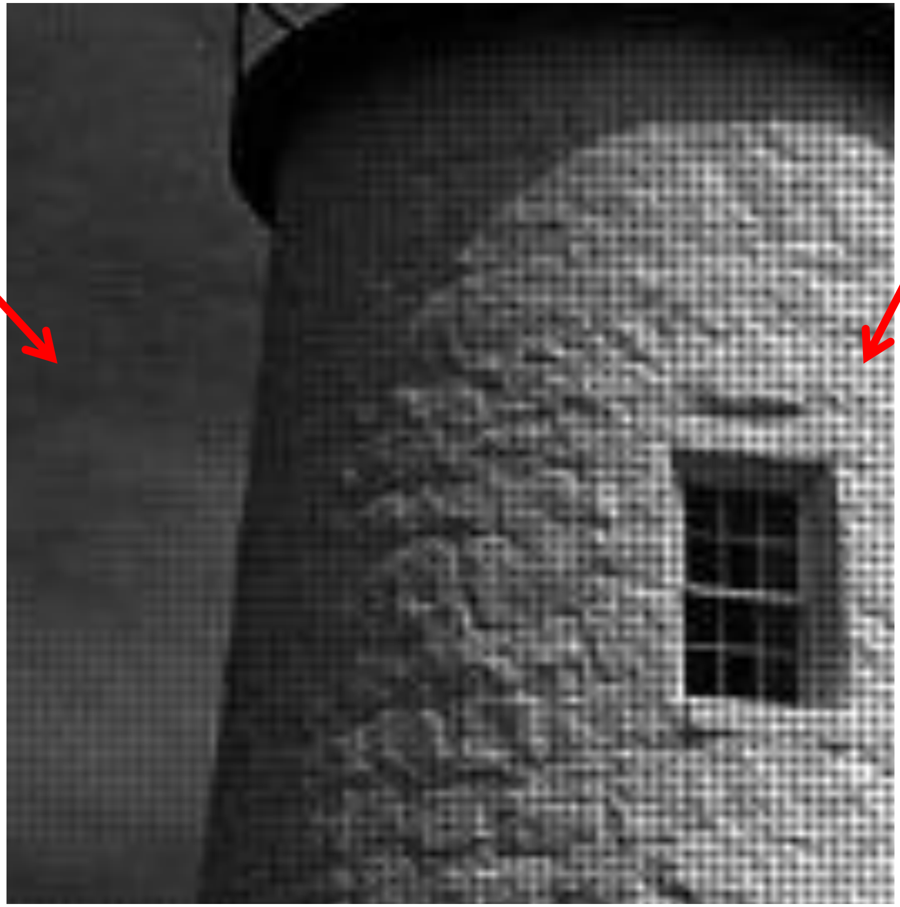
… until camera shutter closes. Then, the analog front-end:

- reads out photosites' wells, row-by-row, and converts them to analog signals
- applies a (possibly non-uniform) gain to these analog signals
- converts them to digital signals
- corrects non-linearities

… and finally returns an image.

After all of this, what does an image look like?





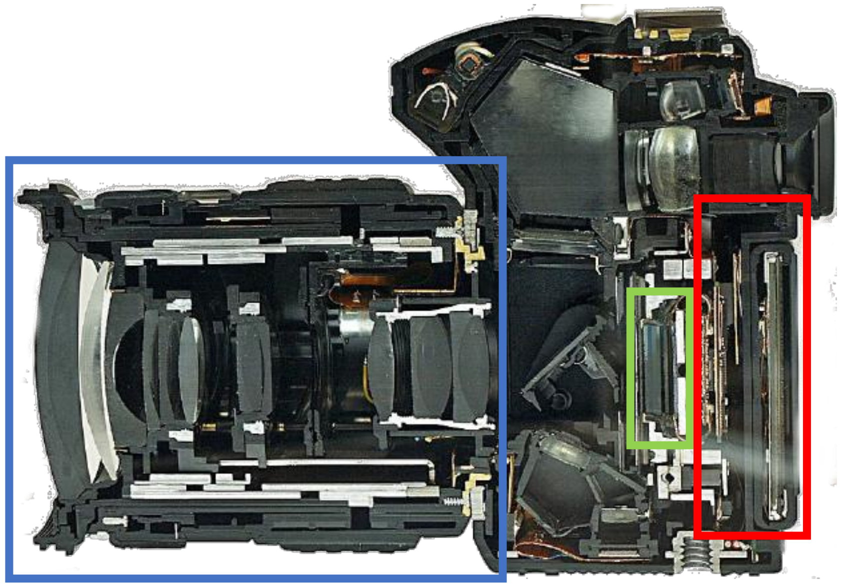
- Kind of disappointing.
- We call this the RAW image.

Today’s Lecture

- Digital photography
- Standard camera pipeline
- Noise
- Color

The modern photography pipeline

post-capture processing

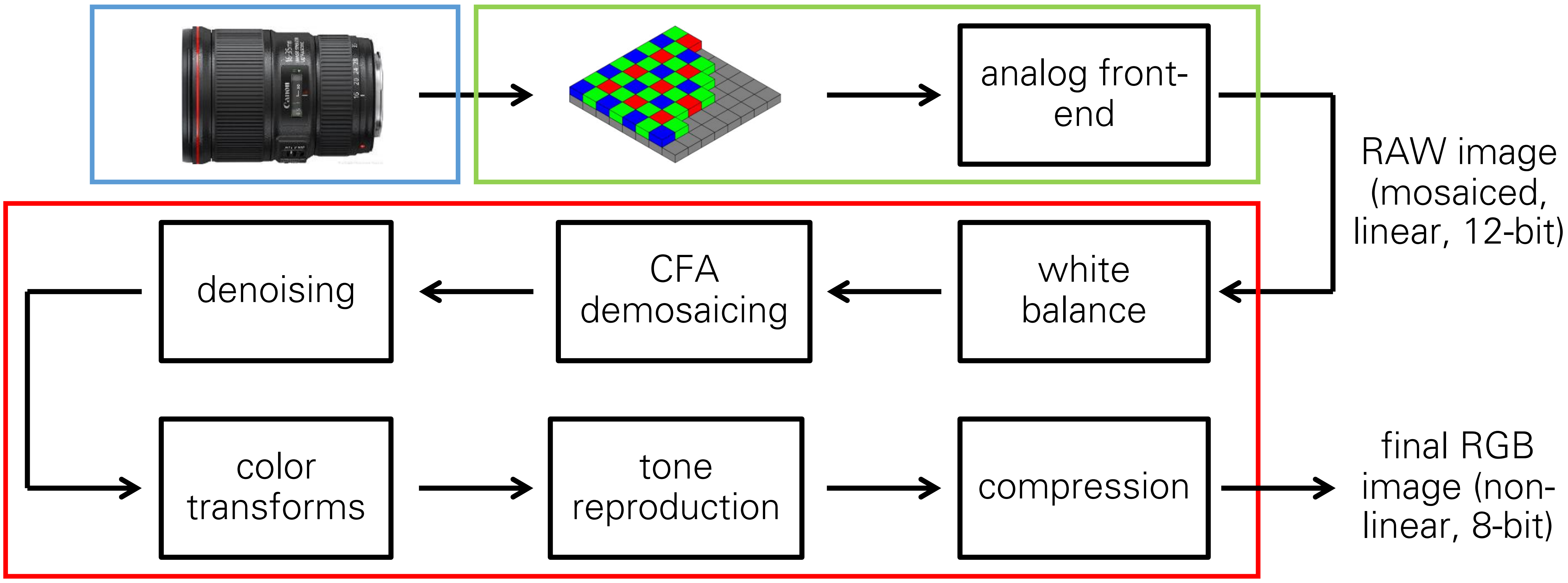


optics and optical controls → sensor, analog front-end, and color filter array → in-camera image processing pipeline

The in-camera image processing pipeline

The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Quick notes on terminology

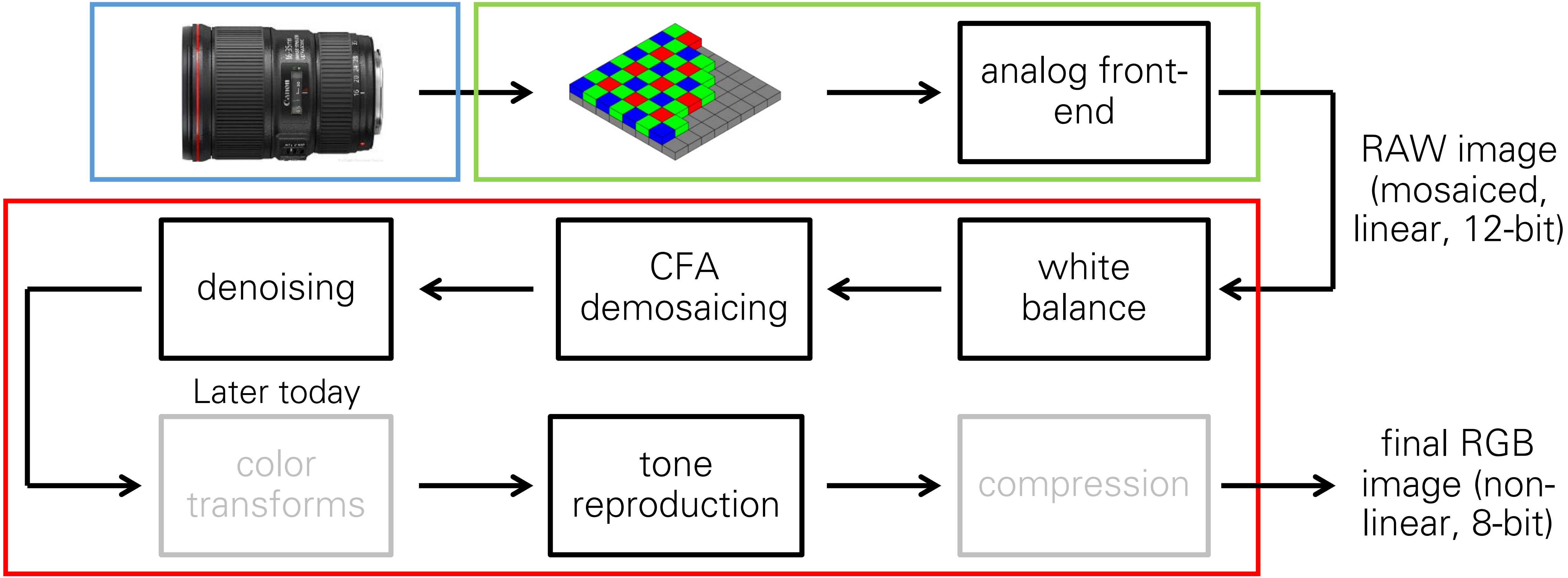
- Sometimes the term image signal processor (ISP) is used to refer to the image processing pipeline itself.

- The process of converting a RAW image to a "conventional" image is often called rendering (unrelated to the image synthesis procedure of the same name in graphics).

- The inverse process, going from a "conventional" image back to RAW is called derendering.

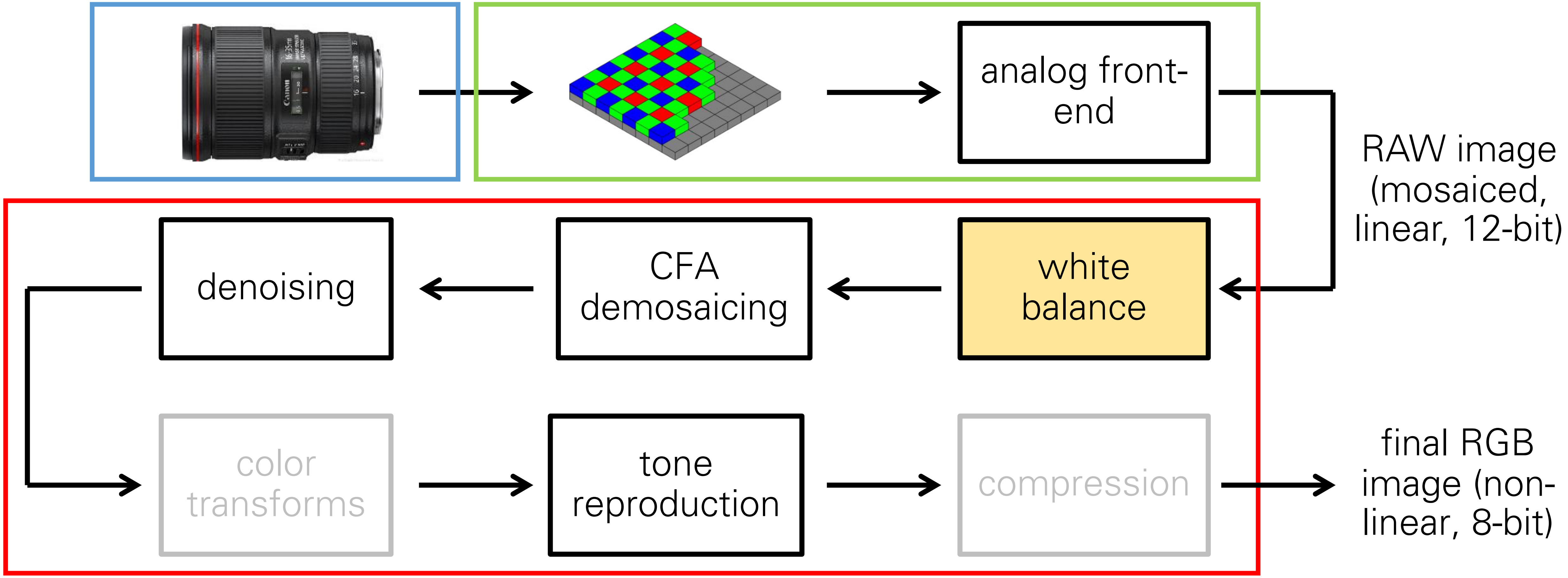
The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



White balancing

Human visual system has chromatic adaptation:

- We can perceive white (and other colors) correctly under different light sources.

Slide credit: Todd Zickler

White balancing

Human visual system has chromatic adaptation:
- We can perceive white (and other colors) correctly under different light sources.

Slide credit: Todd Zickler

White balancing

Human visual system has chromatic adaptation:
- We can perceive white (and other colors) correctly under different light sources.

Retinal vs perceived color.

Slide credit: Todd Zickler

White balancing

Human visual system has chromatic adaptation:

- We can perceive white (and other colors) correctly under different light sources.
- Cameras cannot do that (there is no "camera perception").

White balancing: The process of removing color casts so that colors that we would perceive as white are rendered as white in final image.

different whites

image captured under fluorescent

image white-balanced to daylight

White balancing presets

Cameras nowadays come with a large number of presets: You can select which light you are taking images under, and the appropriate white balancing is applied.

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

Manual vs automatic white balancing

Manual white balancing:

- Select a camera preset based on lighting.

Can you think of any other way to do manual white balancing?

Manual vs automatic white balancing

Manual white balancing:

- Select a camera preset based on lighting.
- Manually select object in photograph that is color-neutral and use it to normalize.

How can we do automatic white balancing?

Manual vs automatic white balancing

Manual white balancing:

- Select a camera preset based on lighting.
- Manually select object in photograph that is color-neutral and use it to normalize.

Automatic white balancing:

- Grey world assumption: force average color of scene to be grey.
- White world assumption: force brightest object in scene to be white.
- Sophisticated histogram-based algorithms (what most modern cameras do).

Automatic white balancing

Grey world assumption:

- Compute per-channel average.
- Normalize each channel by its average.
- Normalize by green channel average.

white-balanced RGB $\longrightarrow$
$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} G_{avg}/R_{avg} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & G_{avg}/B_{avg} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
$\longleftarrow$ sensor RGB

White world assumption:

- Compute per-channel maximum.
- Normalize each channel by its maximum.
- Normalize by green channel maximum.

white-balanced RGB $\longrightarrow$
$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} G_{max}/R_{max} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & G_{max}/B_{max} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
$\longleftarrow$ sensor RGB

Automatic white balancing example

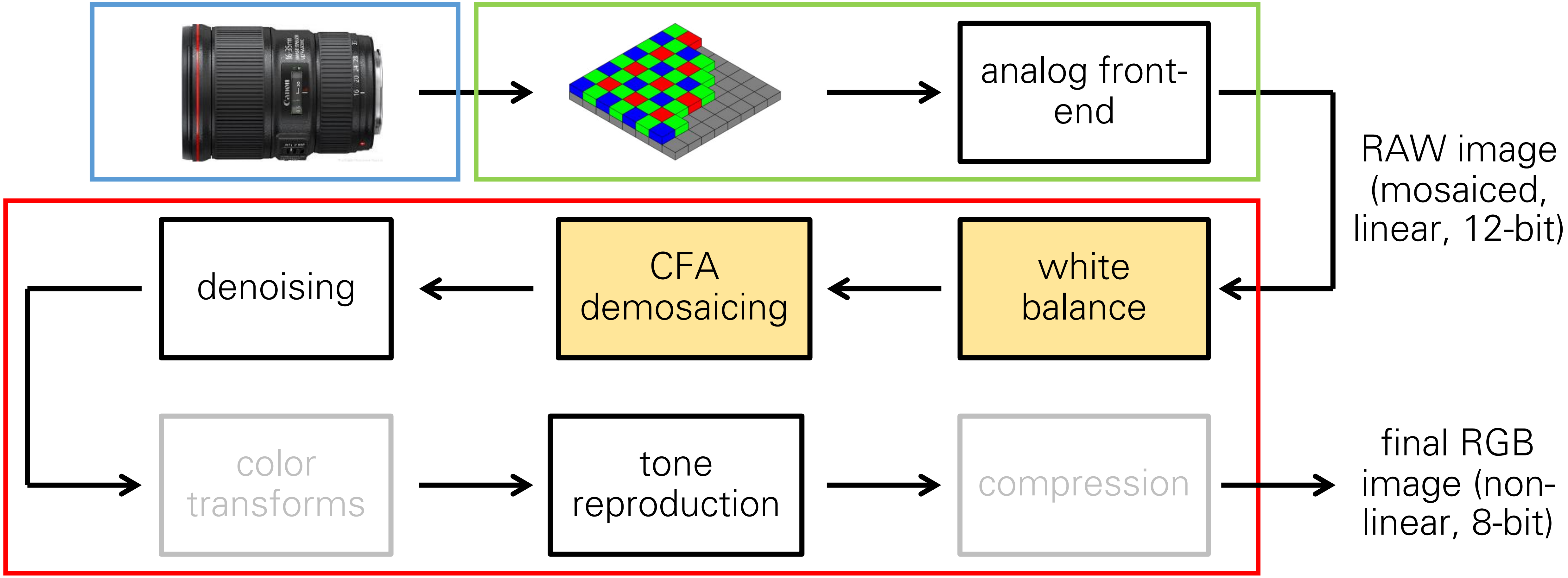
input image

grey world

white world

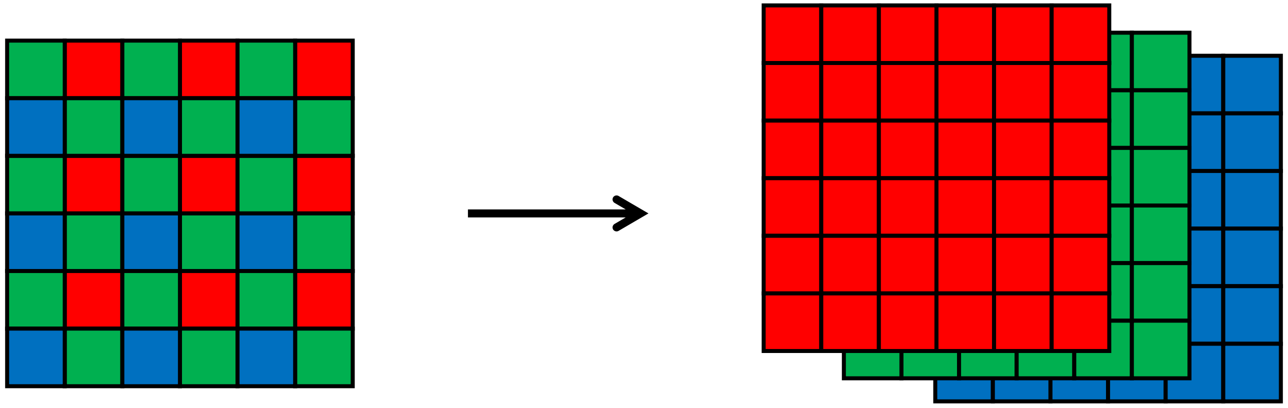
The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



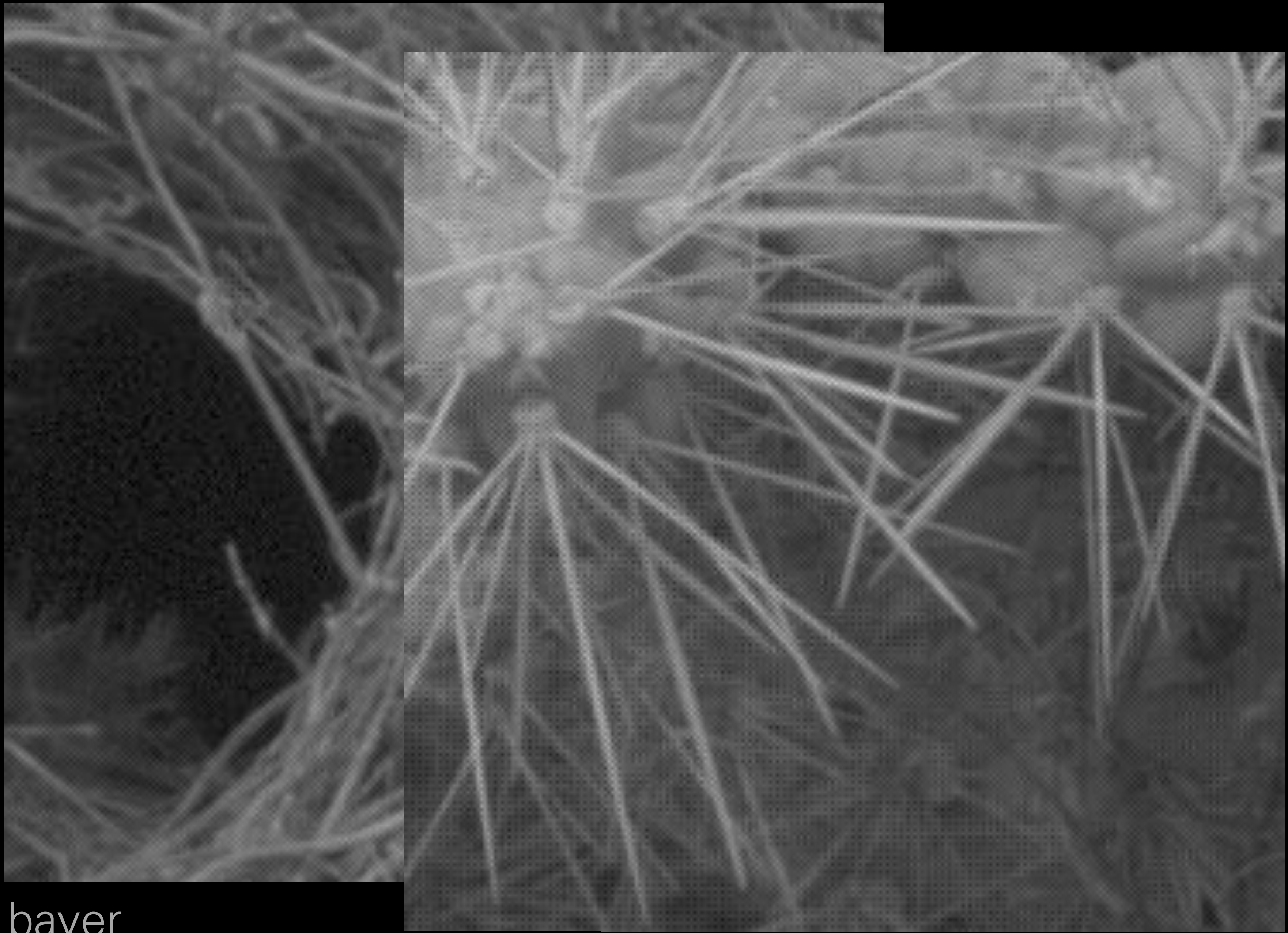
CFA demosaicing

Produce full RGB image from mosaiced sensor output.



Any ideas on how to do this?

bayer

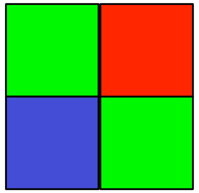


bayer

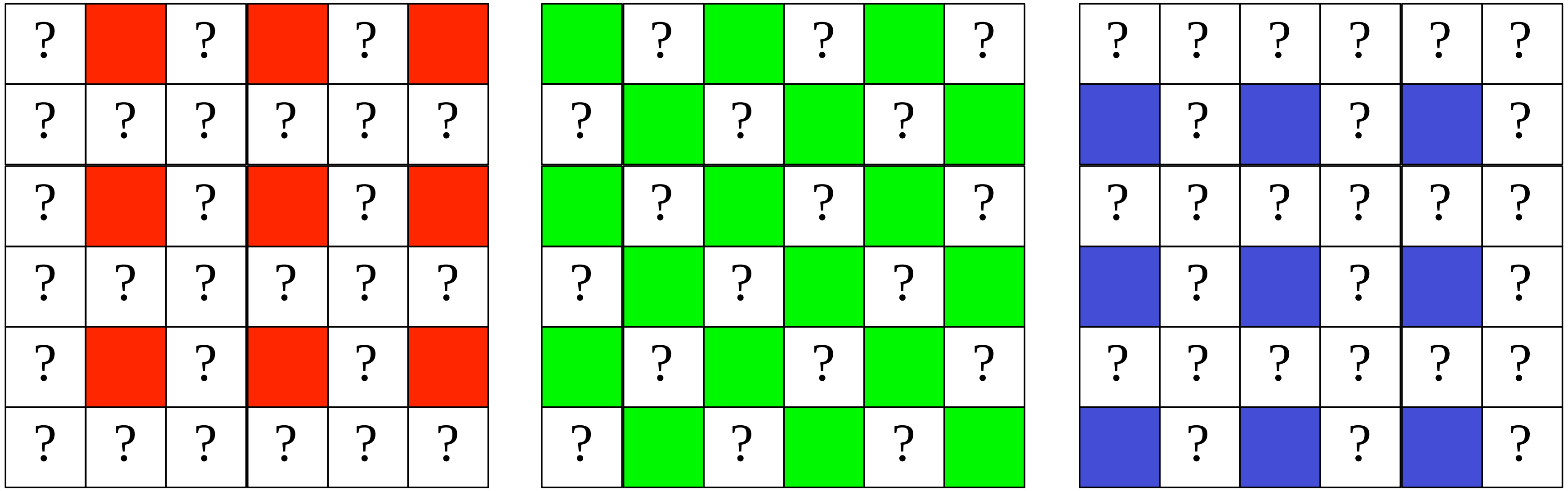
bayer

Half-resolution demosaic

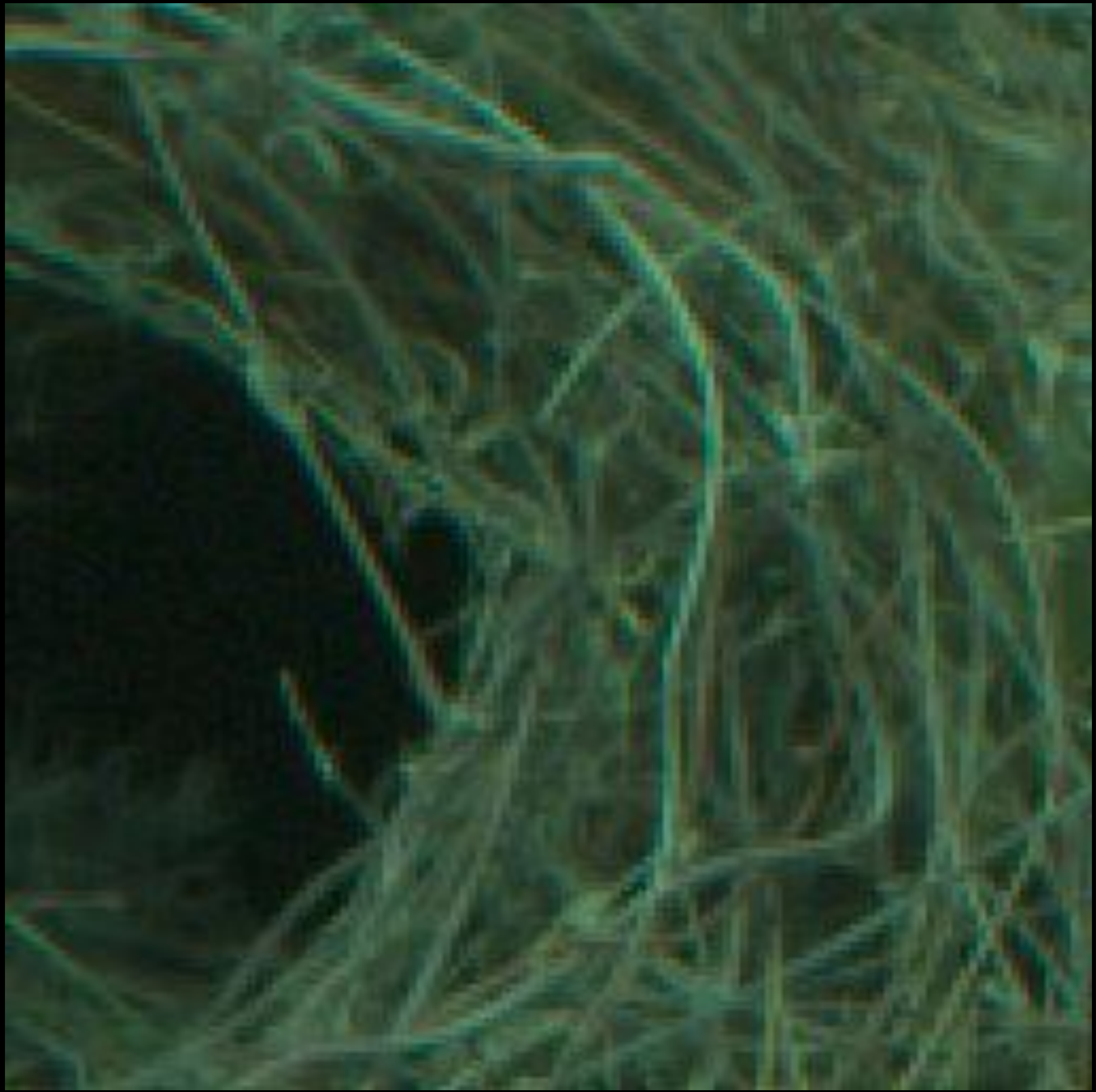
- Idea 1: treat each block of four pixels as a pixel



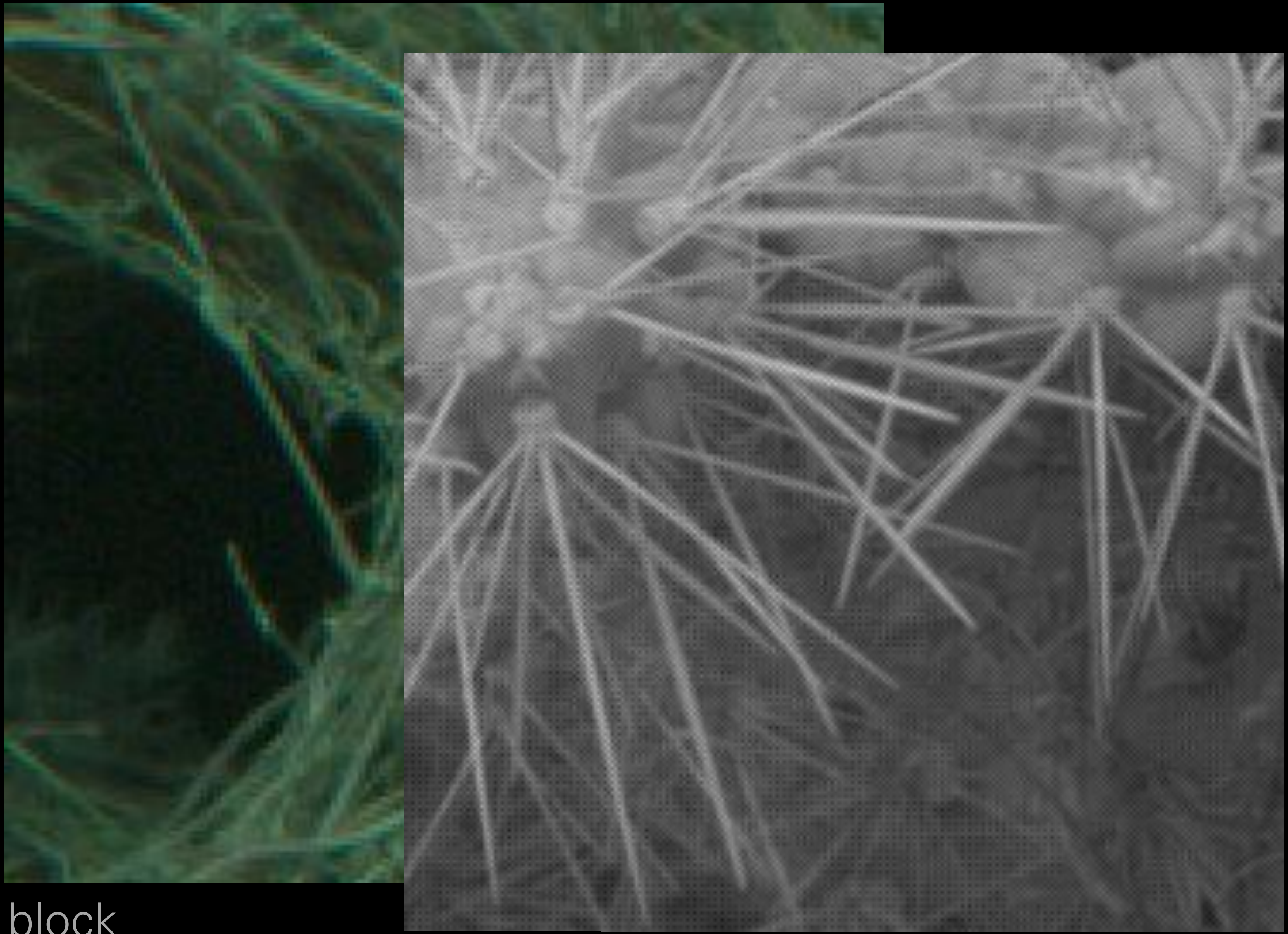
Easy to code up in one line of Matlab. But what is wrong with this?
1. throws away too much resolution -- make a half-resolution image
2. produces subpixel shifts in color planes!



bayer

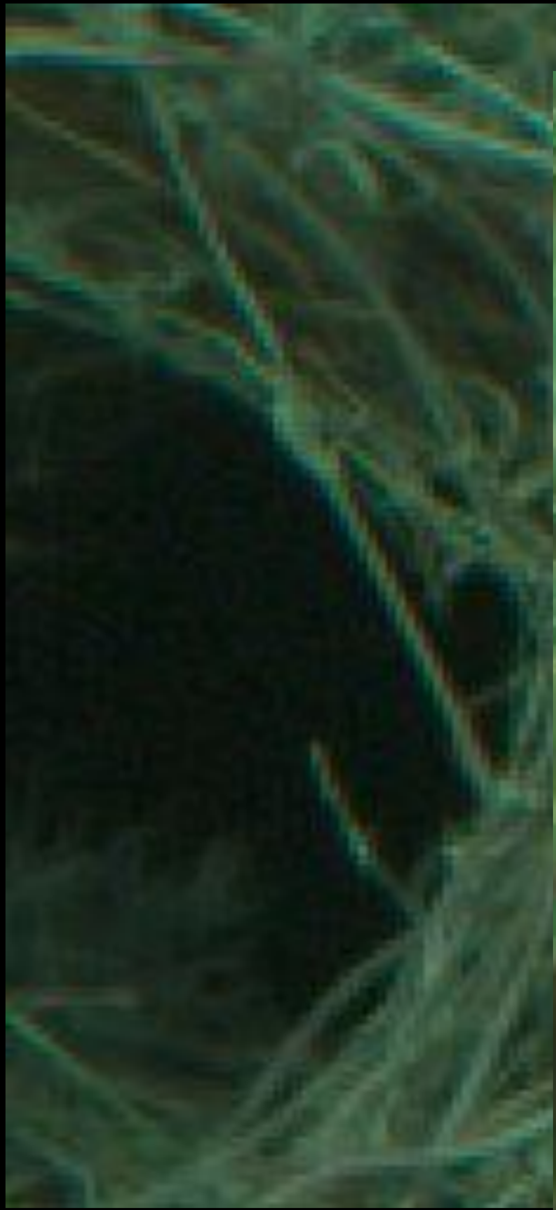


block



block

bayer

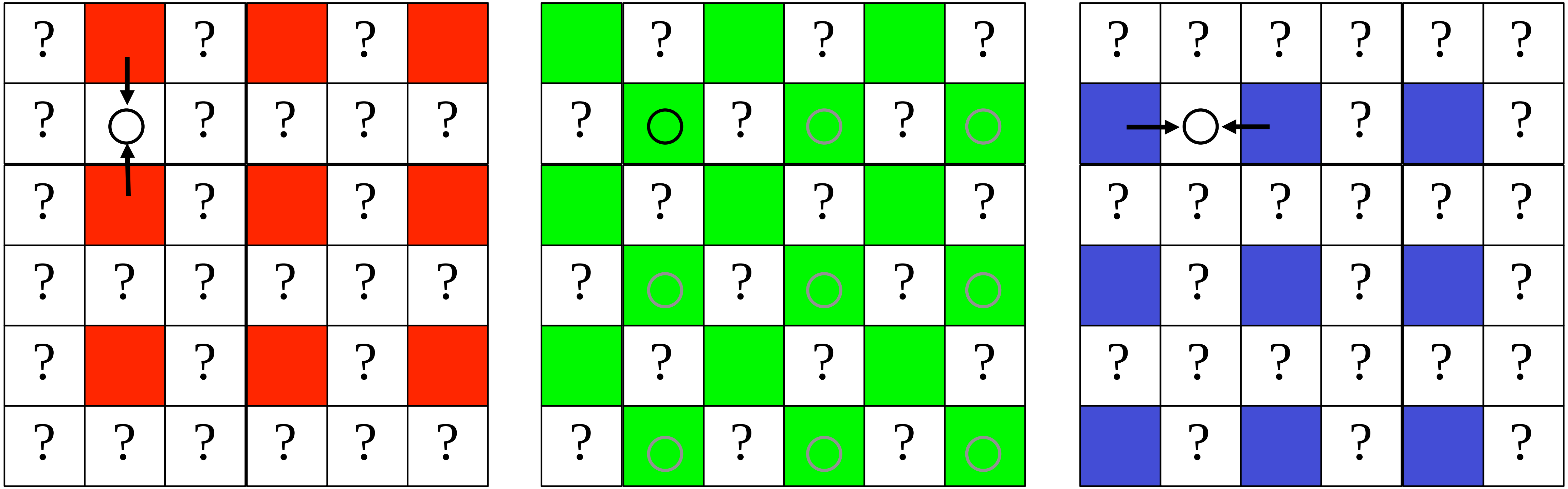


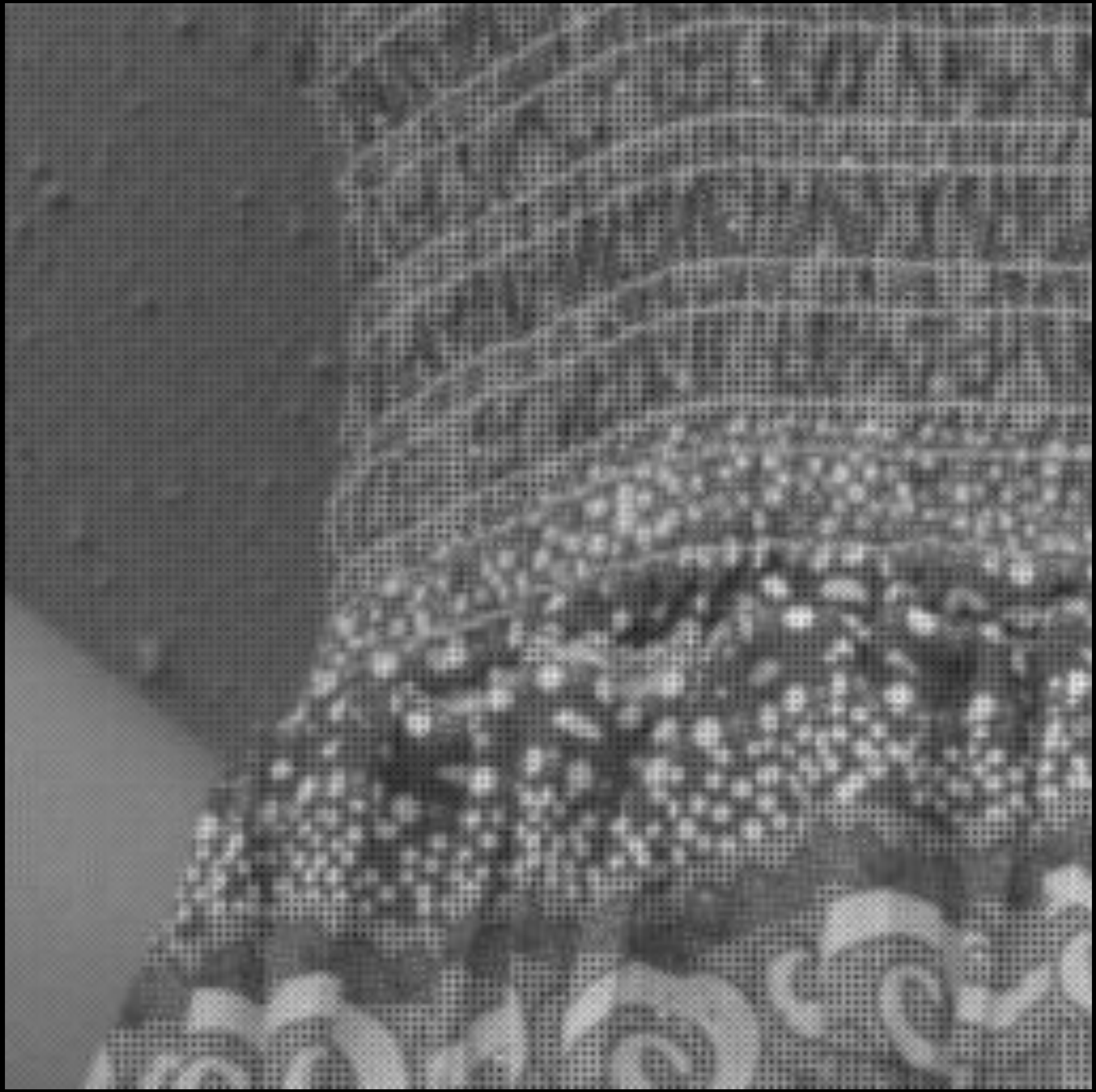
block

block

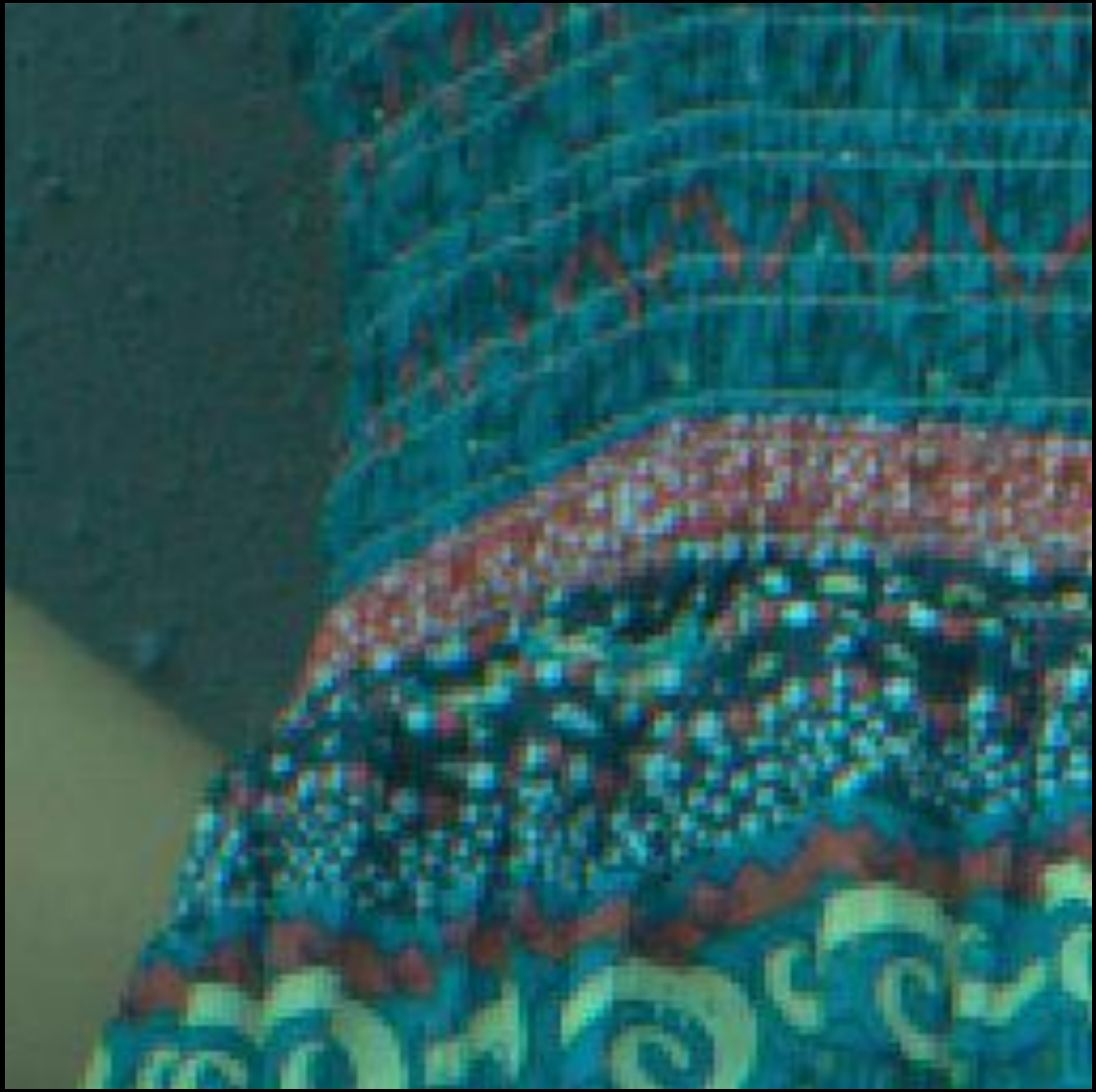
Centered half-resolution

- Average pixels in groups that all have the same "center of gravity"
  avoids major color fringing





bayer



block

centered

centered

bayer

centered

block

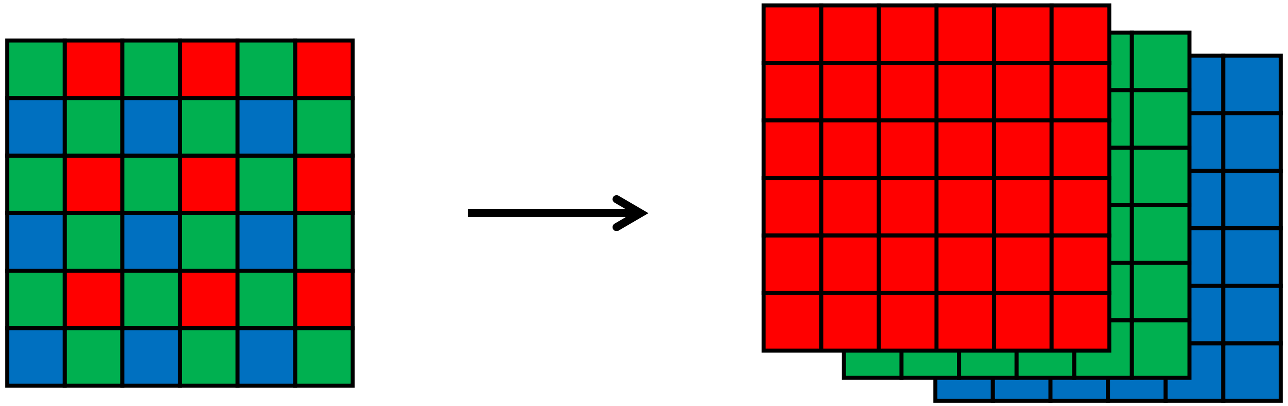
centered

centered

Naïve full-resolution interpolation

- What if we don't want to throw away so much sharpness?

Produce full RGB image from mosaiced sensor output.

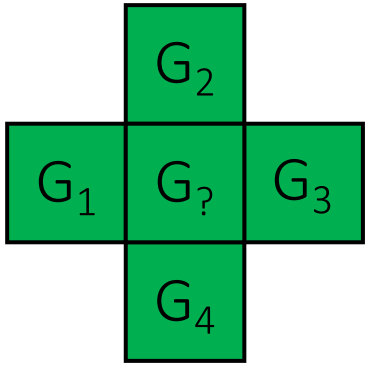


Interpolate from neighbors:

- Bilinear interpolation (needs 4 neighbors).
- Bicubic interpolation (needs more neighbors, may overblur).
- Edge-aware interpolation (more on this later).

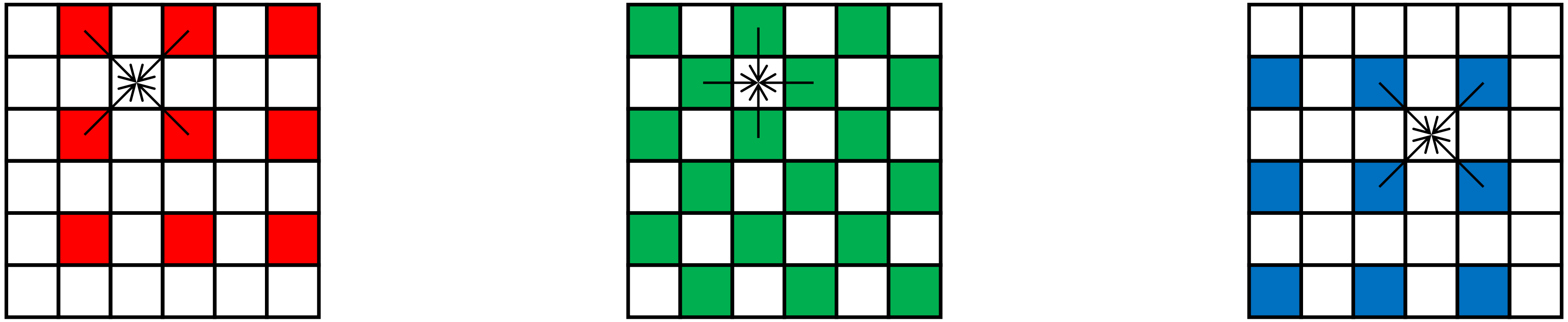
Demosaicing by bilinear interpolation

Bilinear interpolation: Simply average your 4 neighbors.

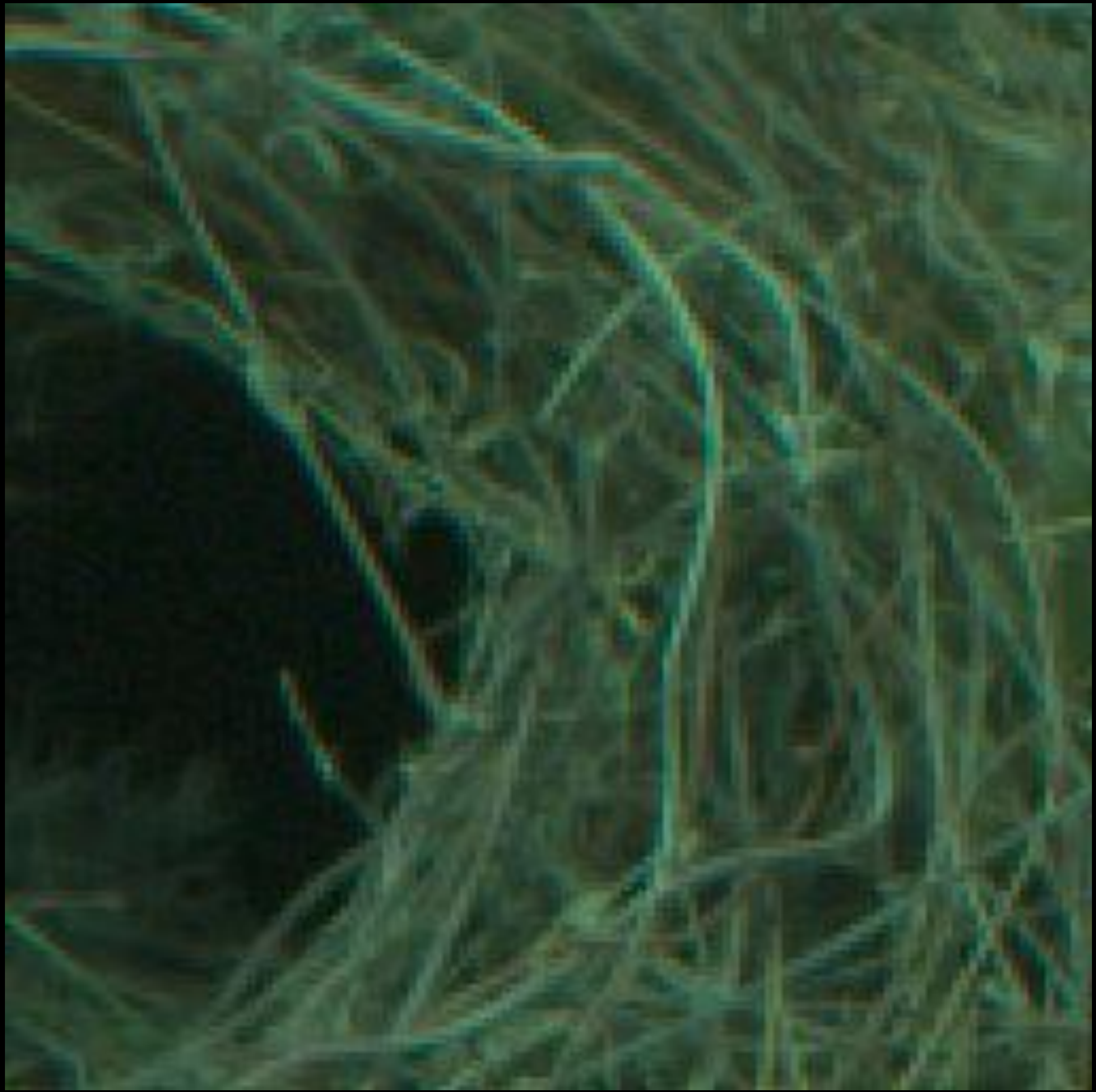


$$G_? = \frac{G_1 + G_2 + G_3 + G_4}{4}$$

Neighborhood changes for different channels:



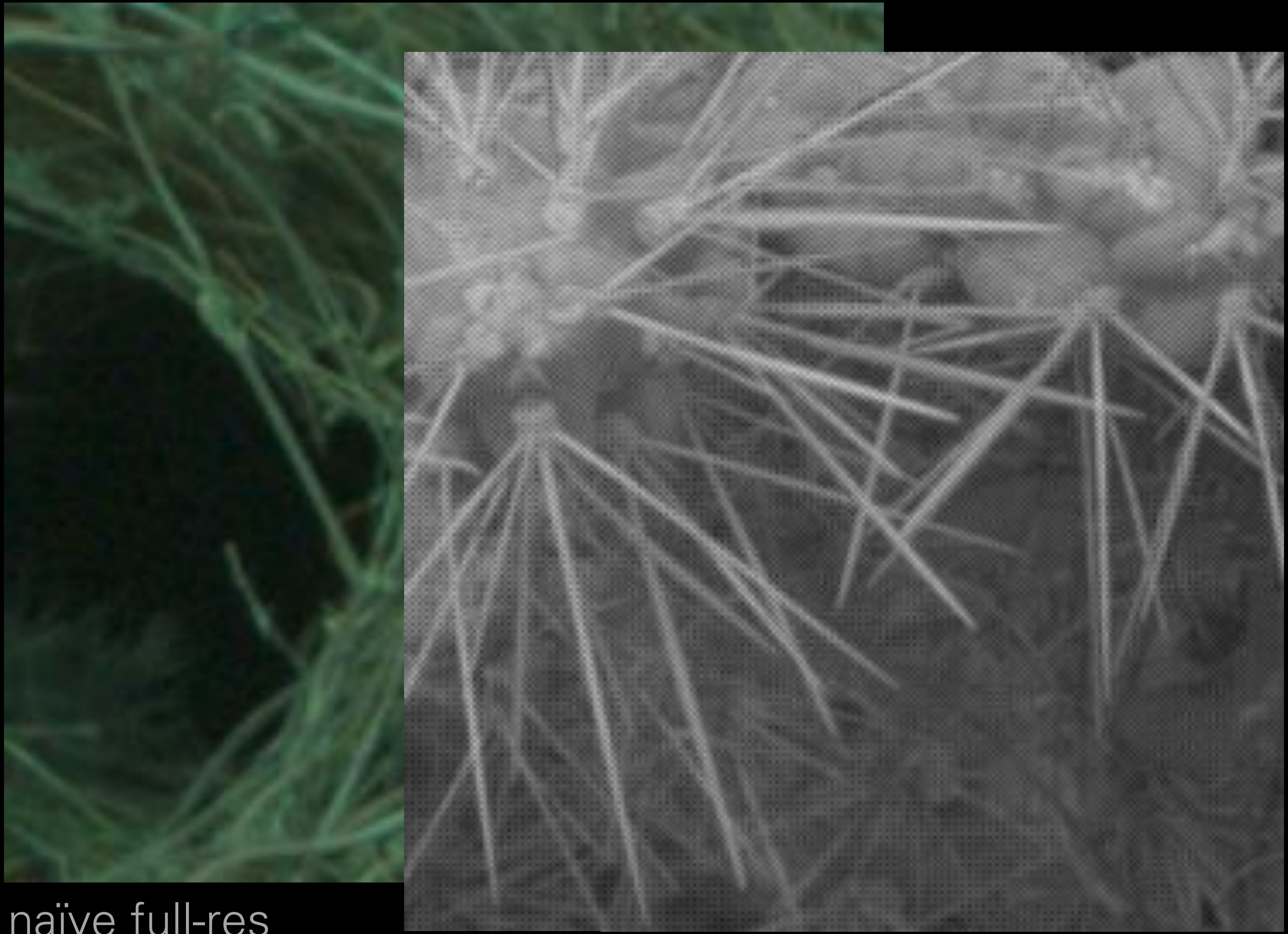
bayer



block

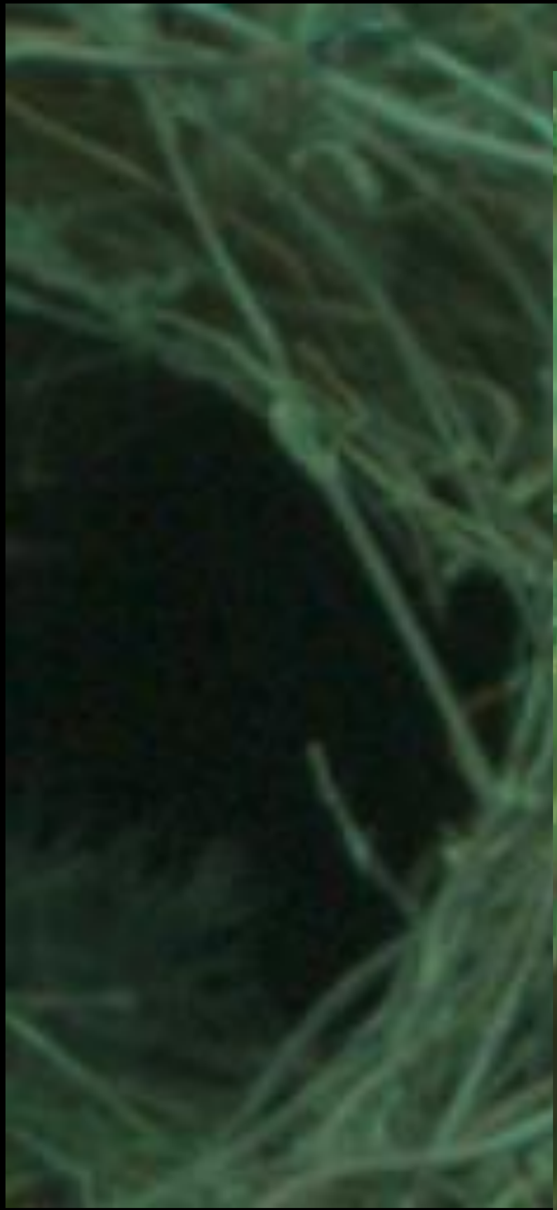
centered

naïve full-res



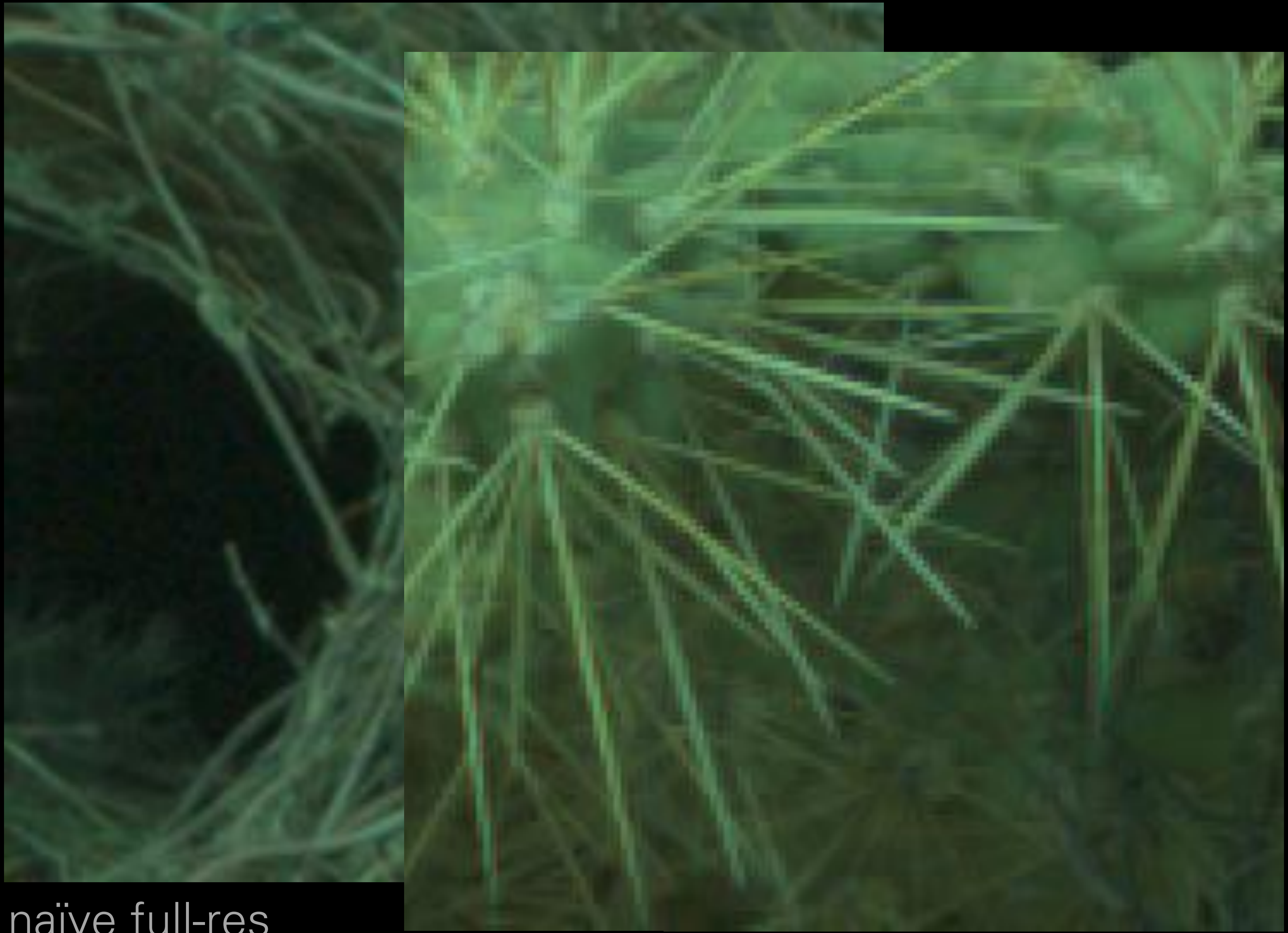
naïve full-res

bayer



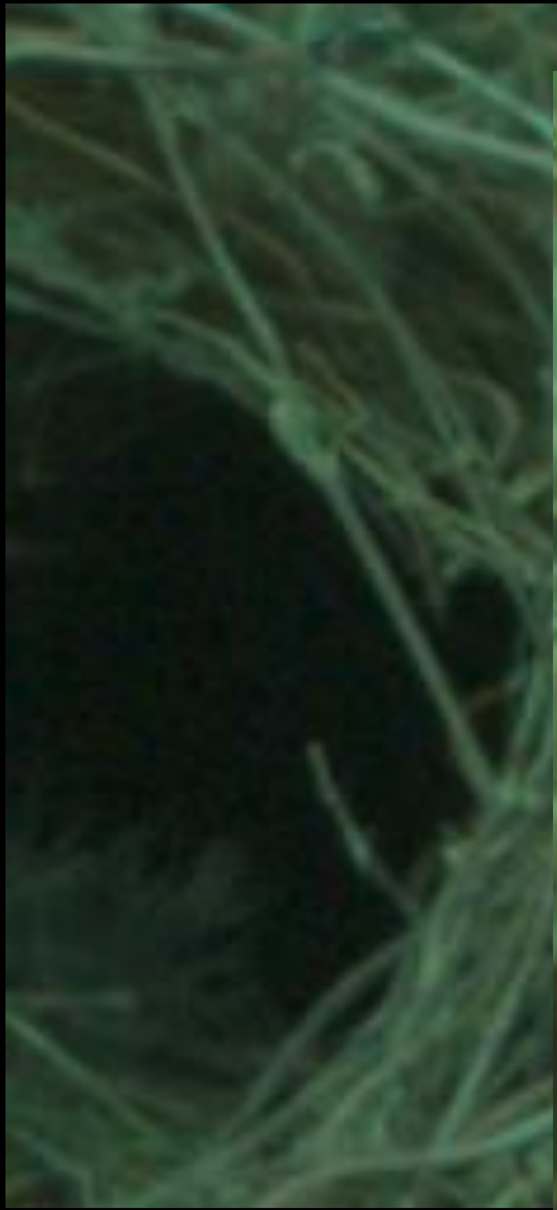
naïve full-res

block



naïve full-res

centered



naïve full-res

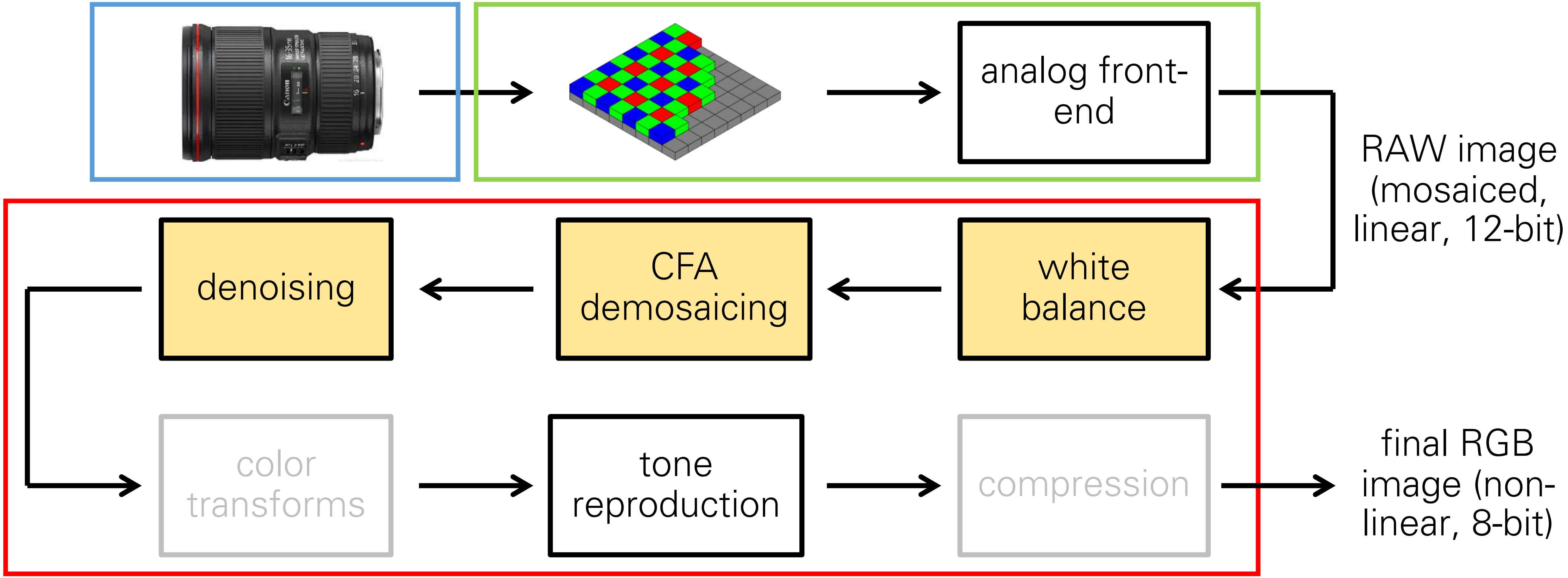
naïve full-res

dcraw

dcraw

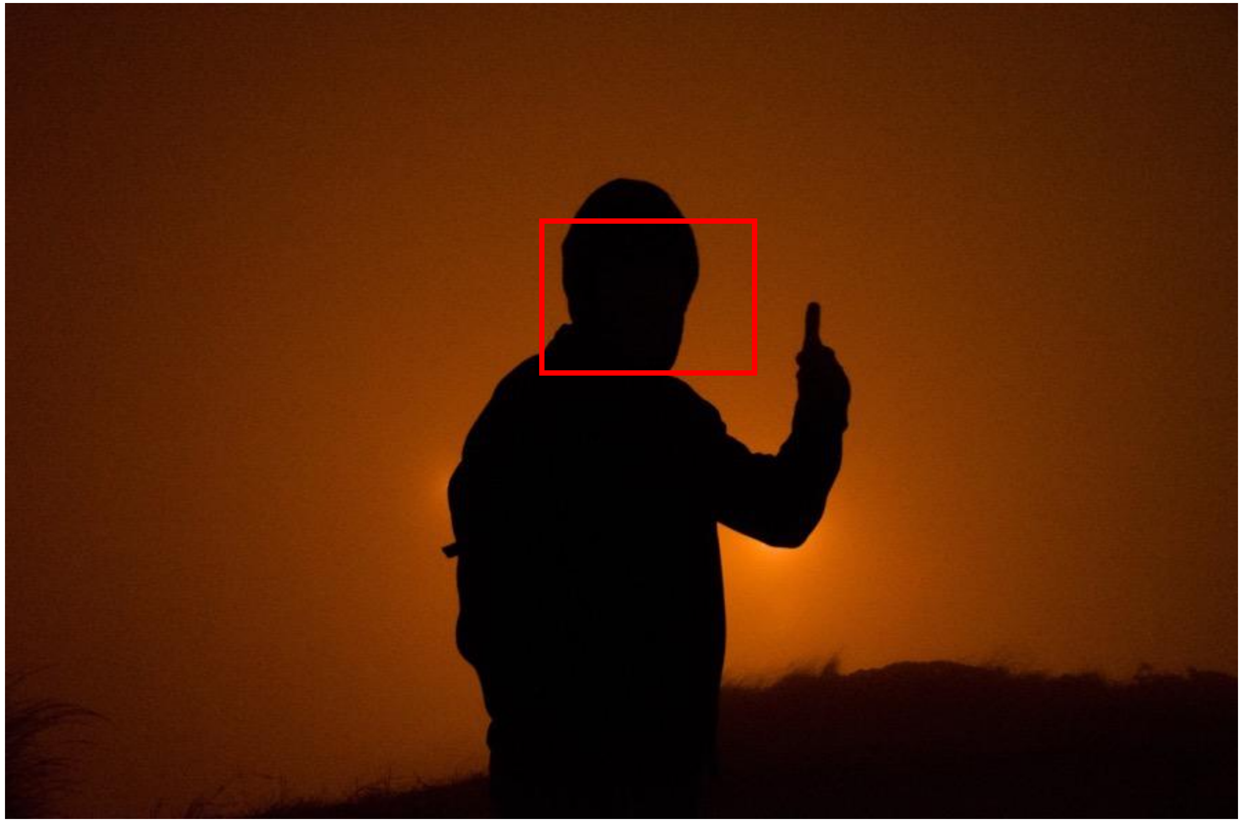
The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Noise in images

Can be very pronounced in low-light images.



Three types of sensor noise

1) (Photon) shot noise:

- Photon arrival rates are a random process (Poisson distribution).
- The brighter the scene, the larger the variance of the distribution.

2) Dark-shot noise:

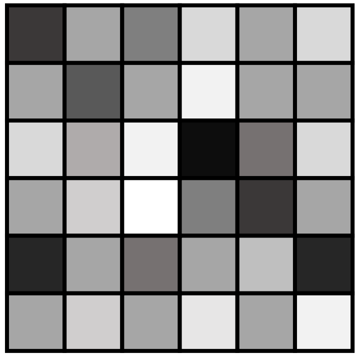
- Emitted electrons due to thermal activity (becomes worse as sensor gets hotter.)

3) Read noise:

- Caused by read-out and AFE electronics (e.g., gain, A/D converter).

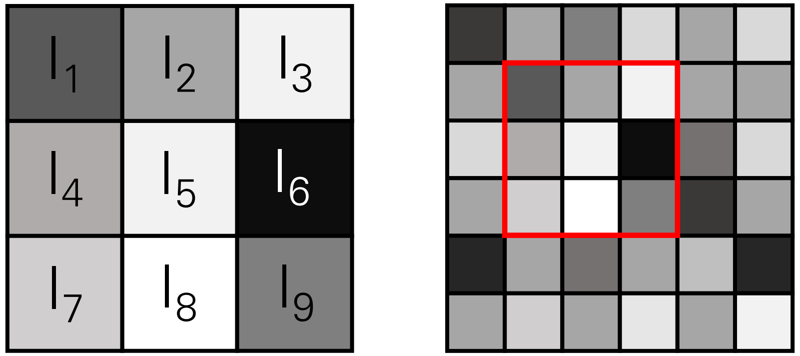
Bright scene and large pixels: photon shot noise is the main noise source.

How to denoise?



How to denoise?

Look at the neighborhood around you.



- Mean filtering (take average):

$$I'_5 = \frac{I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7 + I_8 + I_9}{9}$$

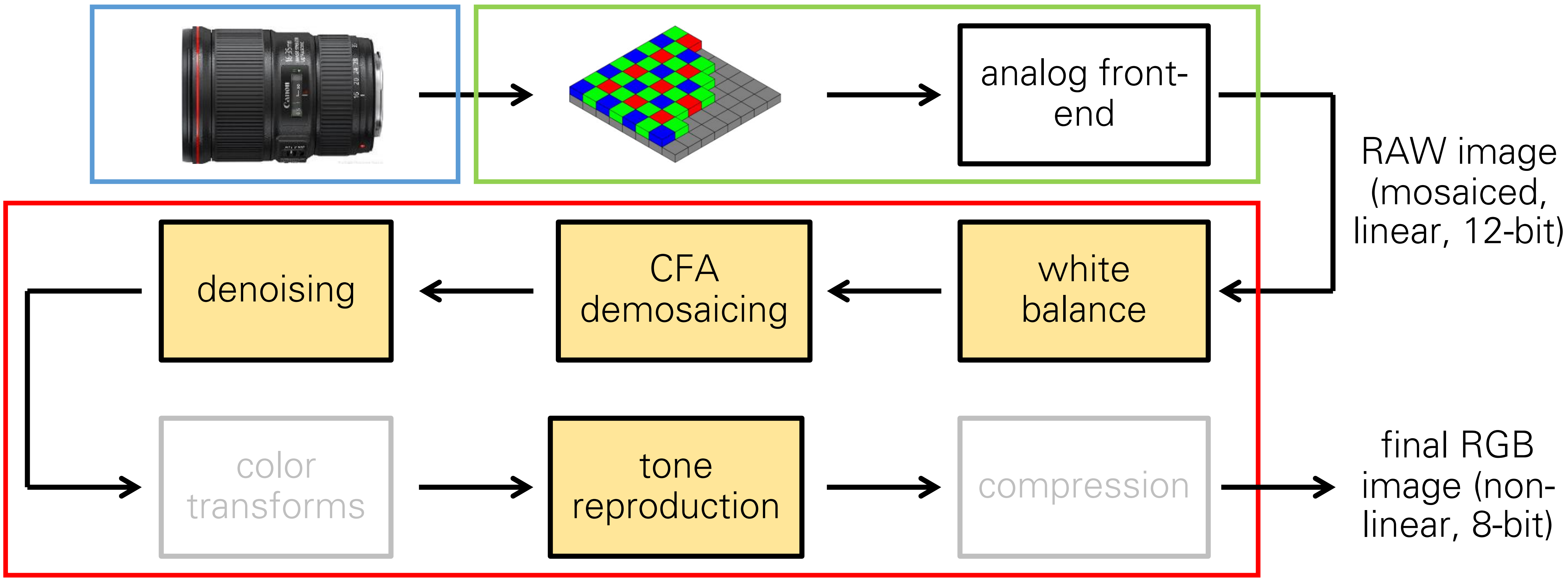
- Median filtering (take median):

$$I'_5 = \text{median}(I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9)$$

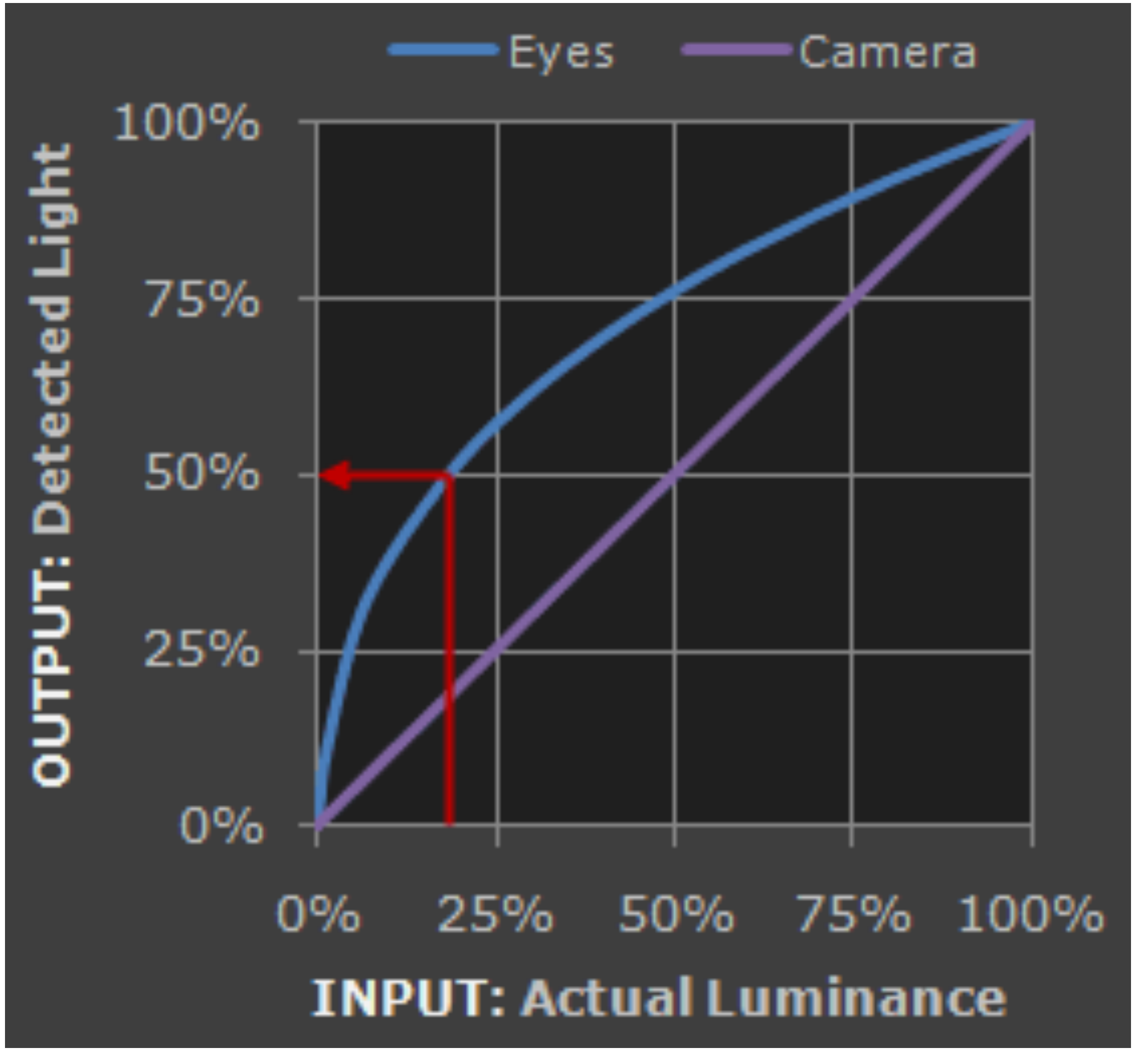
Large area of research. We will see some more about filtering in a later lecture.

The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Perceived vs measured brightness by human eye



We have already seen that sensor response is linear.

Human-eye response (measured brightness) is also linear.

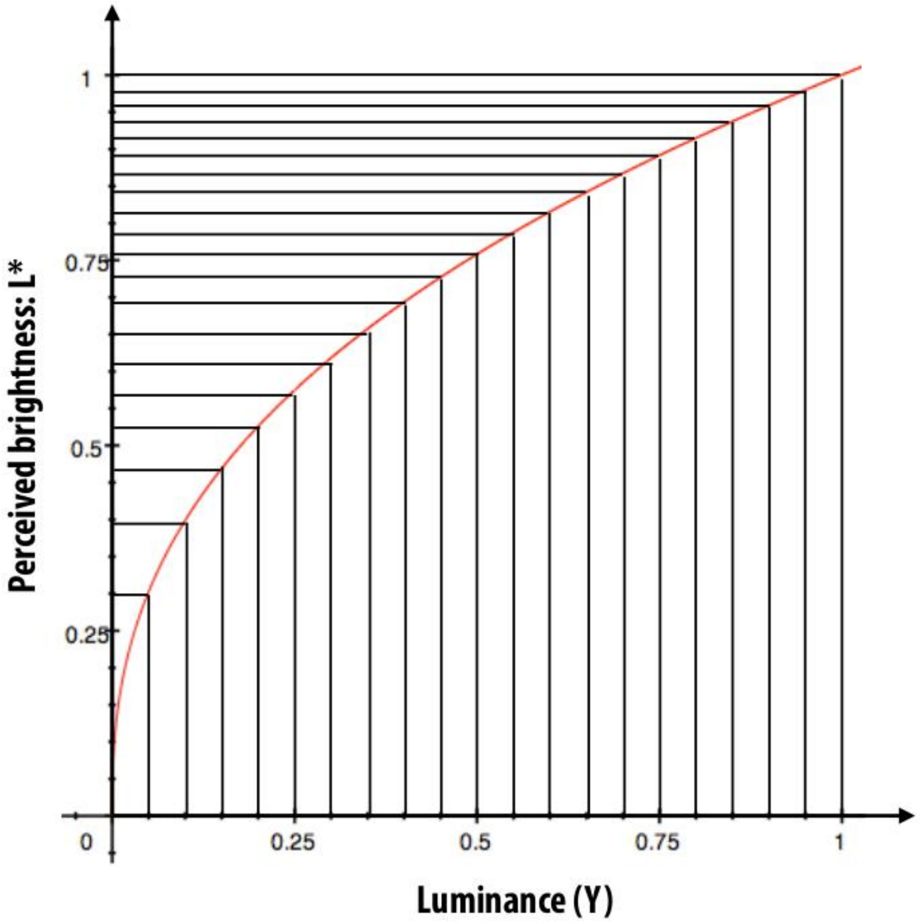
However, human-eye perception (perceived brightness) is non-linear:

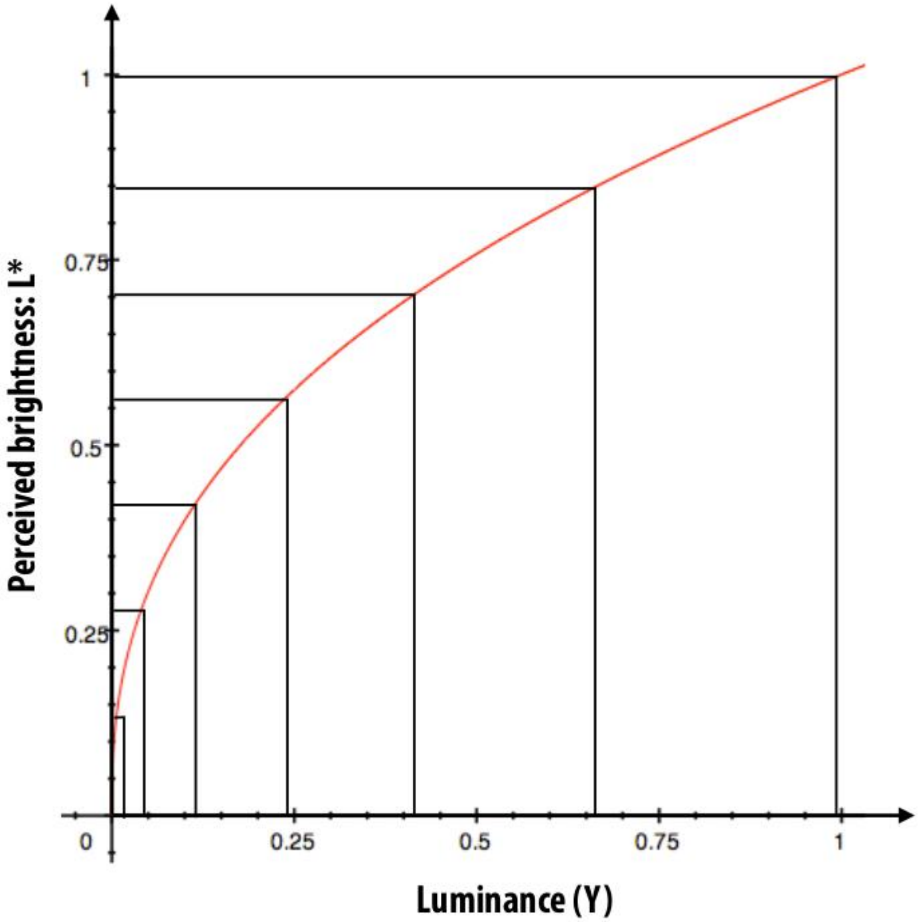
- More sensitive to dark tones.
- Approximately a I^{γ} function.

Gamma encoding

After this stage, we perform compression, which includes changing from 12 to 8 bits.

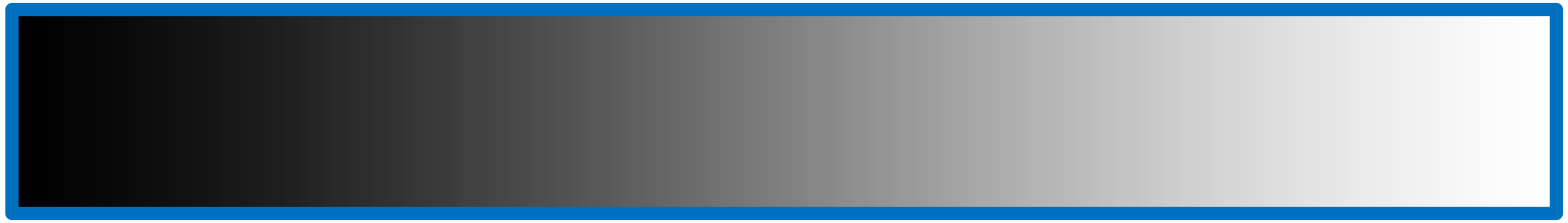
- Apply non-linear curve to use available bits to better encode the information human vision is more sensitive to.





Demonstration

original (8-bits, 256 tones)



Can you predict what will happen if we linearly encode this tone range with only 5 bits?

Can you predict what will happen if we gamma encode this tone range with only 5 bits?

Demonstration

original (8-bits, 256 tones)

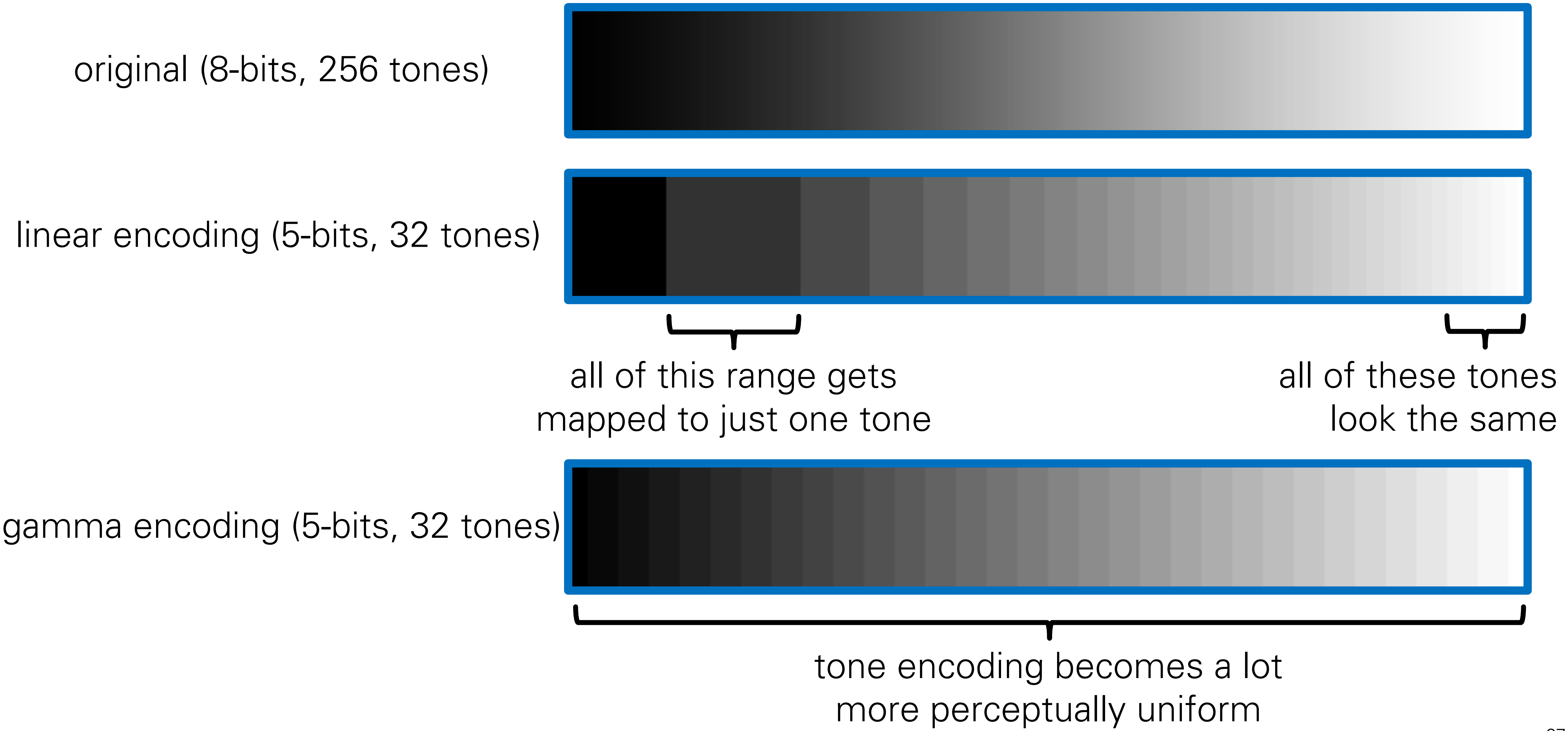
linear encoding (5-bits, 32 tones)

all of this range gets
mapped to just one tone

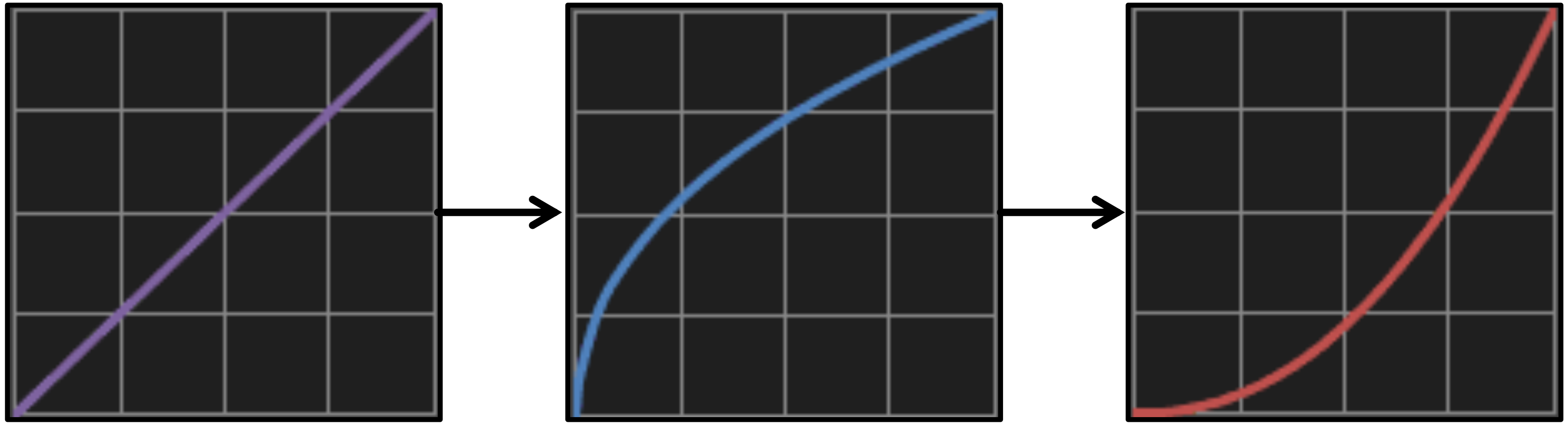
all of these tones
look the same

Can you predict what will happen if we gamma encode this tone range with only 5 bits?

Demonstration



Tone reproduction pipeline

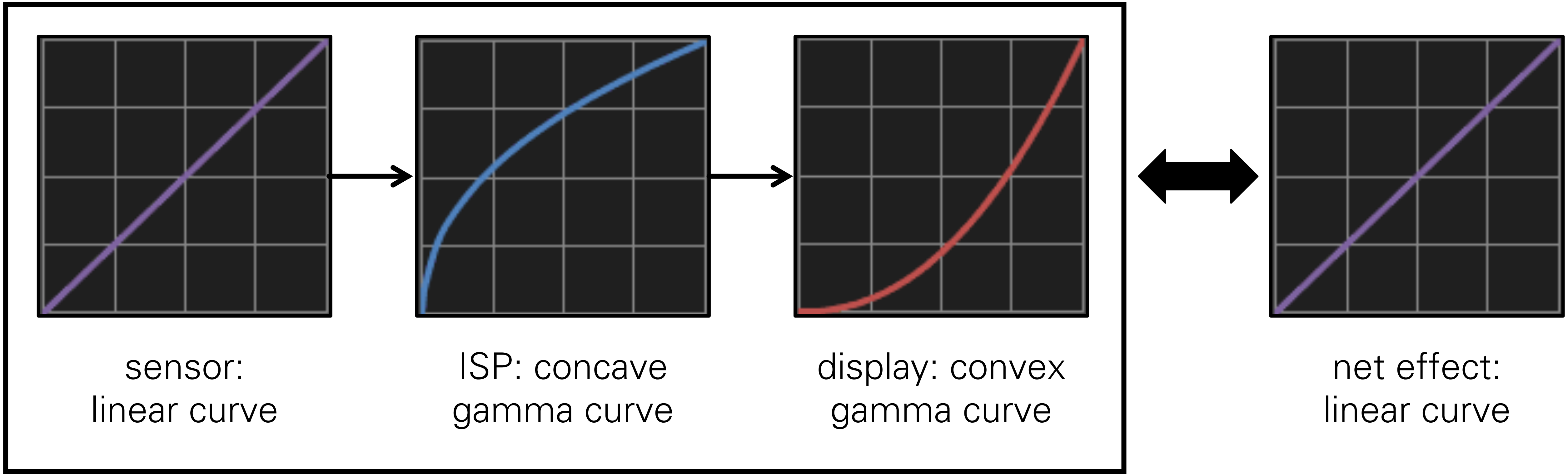


sensor:
linear curve

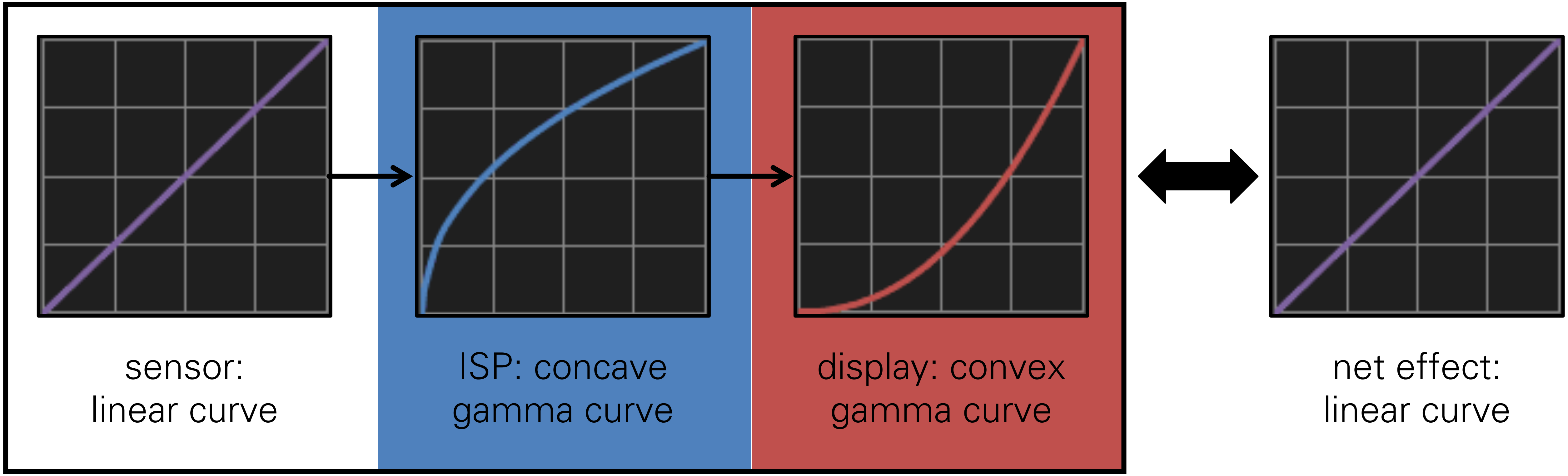
ISP: concave
gamma curve

display: convex
gamma curve

Tone reproduction pipeline



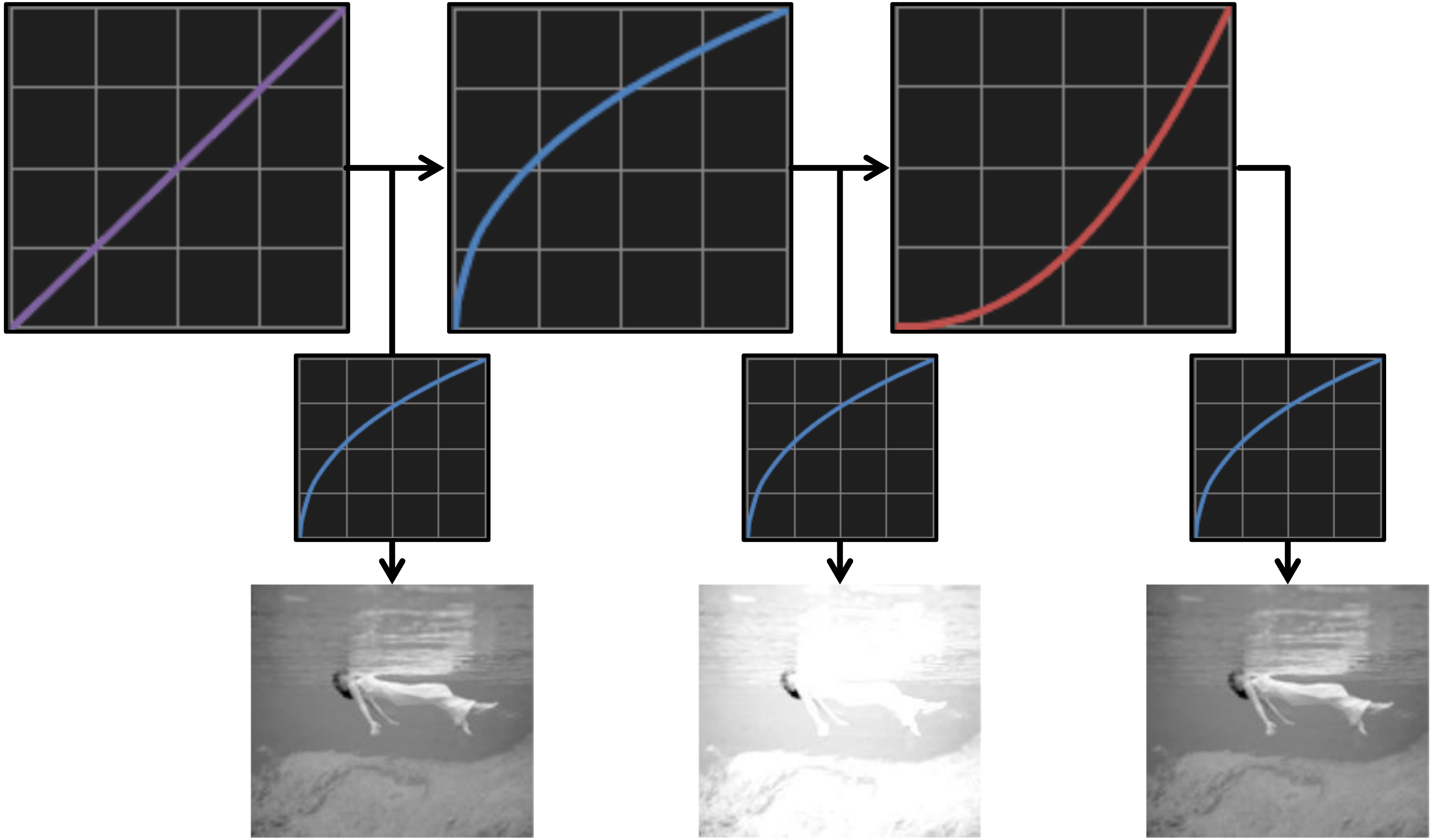
Tone reproduction pipeline



gamma encoding

gamma correction

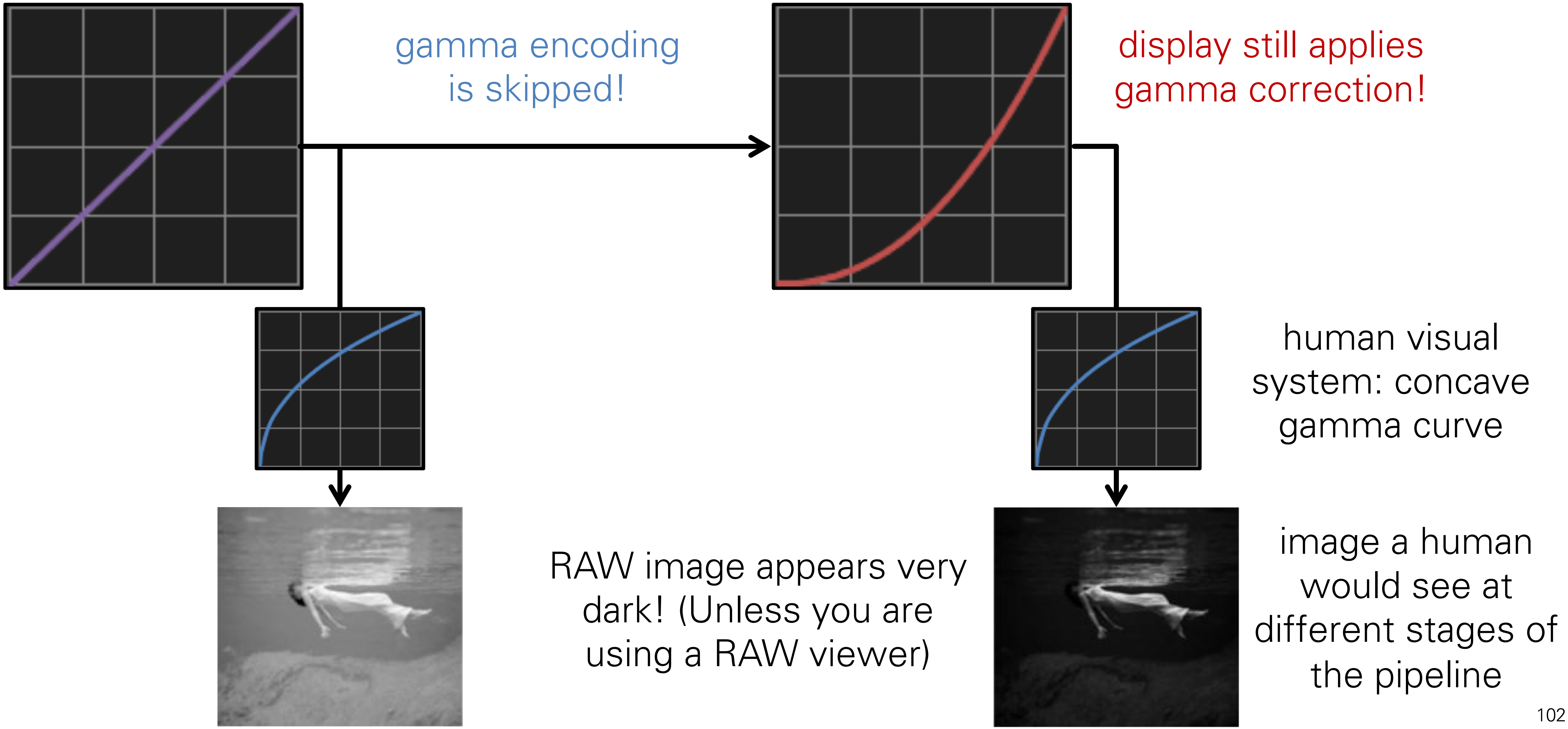
Tone reproduction pipeline



human visual system: concave gamma curve

image a human would see at different stages of the pipeline

RAW pipeline



Historical note

- CRT displays used to have a response curve that was (almost) exactly equal to the inverse of the human sensitivity curve. Therefore, displays could skip gamma correction and display directly the gamma-encoded images.

- It is sometimes mentioned that gamma encoding is done to undo the response curve of a display. This used to (?) be correct, but it is not true nowadays. Gamma encoding is performed to ensure a more perceptually-uniform use of the final image's 8 bits.

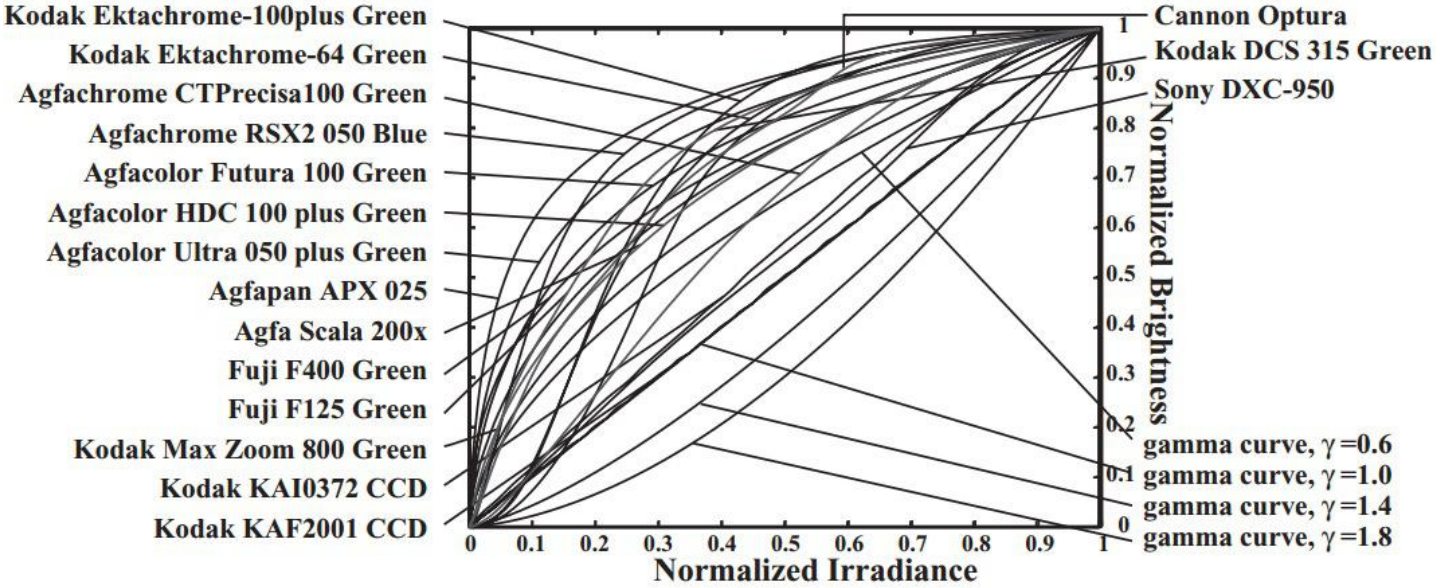
Gamma encoding curves

The exact gamma encoding curve depends on the camera.

- Often well approximated as L^γ, for different values of the power γ ("gamma").
- A good default is $\gamma = 1 / 2.2$.

before gamma

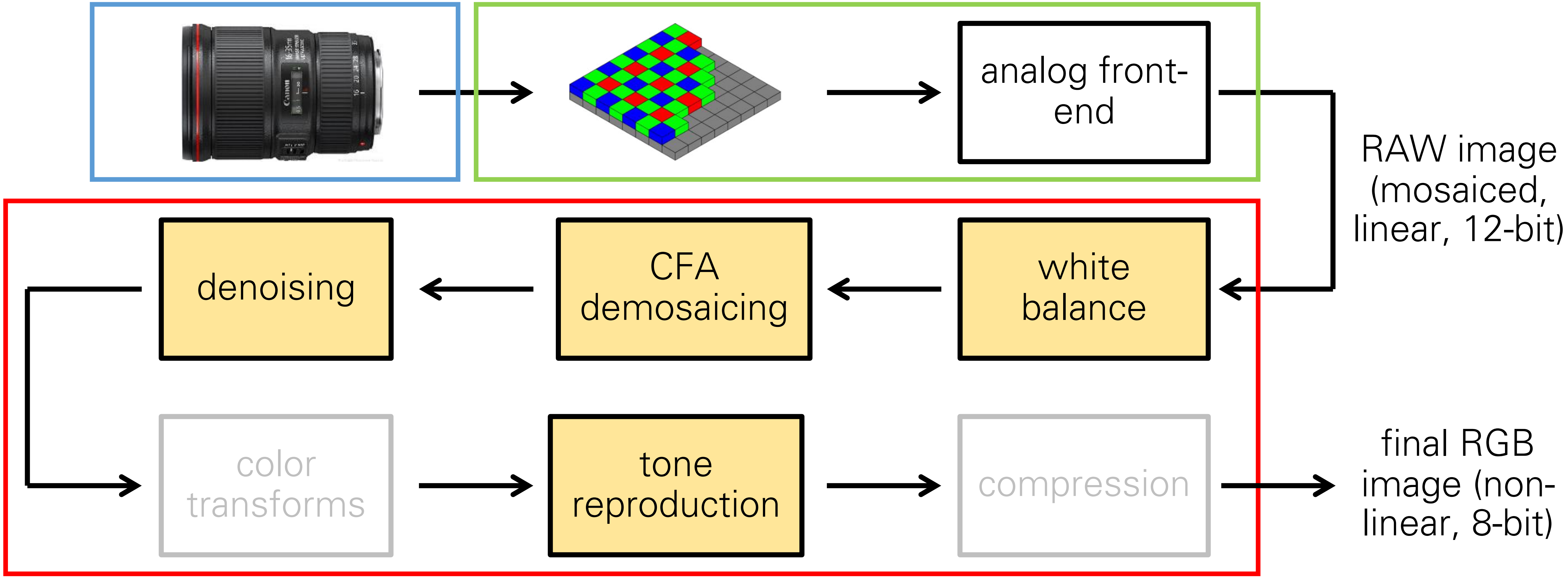
after gamma



Warning: Our values are no longer linear relative to scene radiance!

The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Some general thoughts on the image processing pipeline

Do I ever need to use RAW?

Do I ever need to use RAW?

Emphatic yes!

- Every time you use a physics-based computer vision algorithm, you need linear measurements of radiance.
- Examples: photometric stereo, shape from shading, image-based relighting, illumination estimation, anything to do with light transport and inverse rendering, etc.
- Applying the algorithms on non-linear (i.e., not RAW) images will produce completely invalid results.

What if I don't care about physics-based vision?

What if I don’t care about physics-based vision?

You often still want (rather than need) to use RAW!

- If you like re-finishing your photos (e.g., on Photoshop), RAW makes your life much easier and your edits much more flexible.

Are there any downsides to using RAW?

Are there any downsides to using RAW?

Image files are a lot bigger.

- You burn through multiple memory cards.
- Your camera will buffer more often when shooting in burst mode.
- Your computer needs to have sufficient memory to process RAW images.

Is it even possible to get access to RAW images?

Is it even possible to get access to RAW images?

Quite often yes!

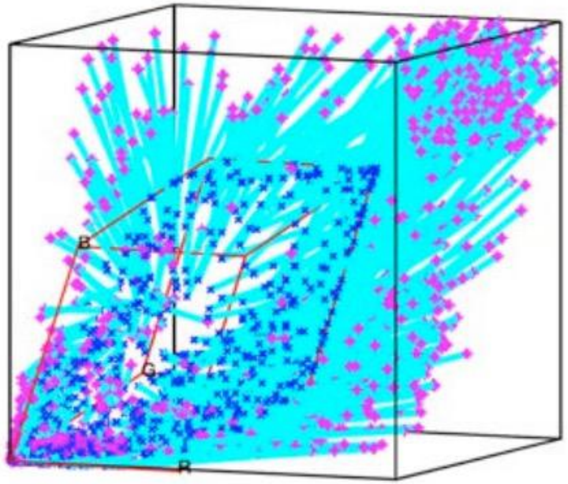
- Most high-end cameras provide an option to store RAW image files.
- Certain phone cameras allow, directly or indirectly, access to RAW.
- Sometimes, it may not be "fully" RAW. The Lightroom app provides images after demosaicking but before tone reproduction.

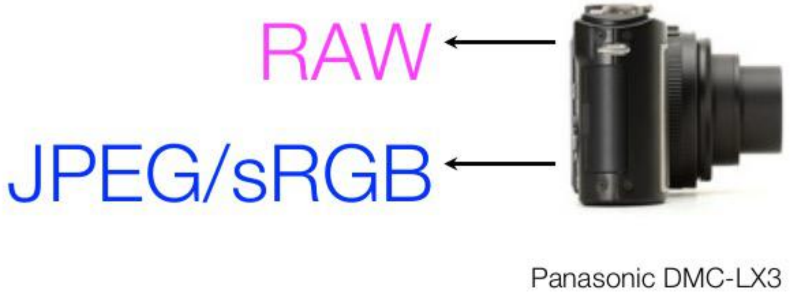
I forgot to set my camera to RAW, can I still get the RAW file?

Nope, tough luck.

- The image processing pipeline is lossy: After all the steps, information about the original image is lost.
- Sometimes we may be able to reverse a camera's image processing pipeline if we know exactly what it does (e.g., by using information from other similar RAW images).
- The conversion of PNG/JPG back to RAW is known as "derendering" and is an active research area.

Derendering





The image processing pipeline

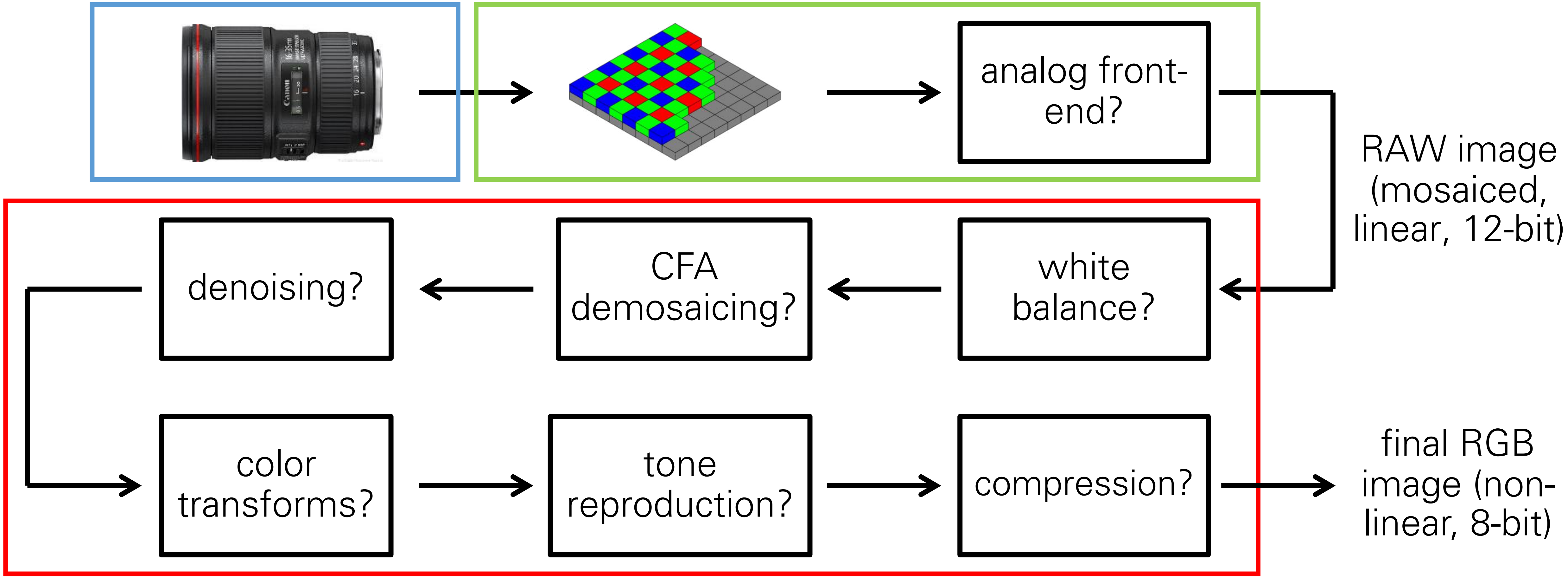
What I described today is an “idealized” version of what we think commercial cameras do.

- Almost all of the steps in both the sensor and image processing pipeline I described earlier are camera-dependent.

- Even if we know the basic steps, the implementation details are proprietary information that companies actively try to keep secret.

The hypothetical image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



How do I open a RAW file in Python?

You can't (not easily at least). You need to use one of the following:

- dcraw – tool for parsing camera-dependent RAW files (specification of file formats are also kept secret).

- Adobe DNG – recently(-ish) introduced file format that attempts to standardize RAW file handling.

See Homework 1 for more details.

Is this the best image processing pipeline?

It depends on how you define “best”. This definition is task-dependent.

- The standard image processing pipeline is designed to create “nice-looking” images.
- If you want to do physics-based vision, the best image processing pipeline is no pipeline at all (use RAW).
- What if you want to use images for, e.g., object recognition? Tracking? Robotics SLAM? Face identification? Forensics?

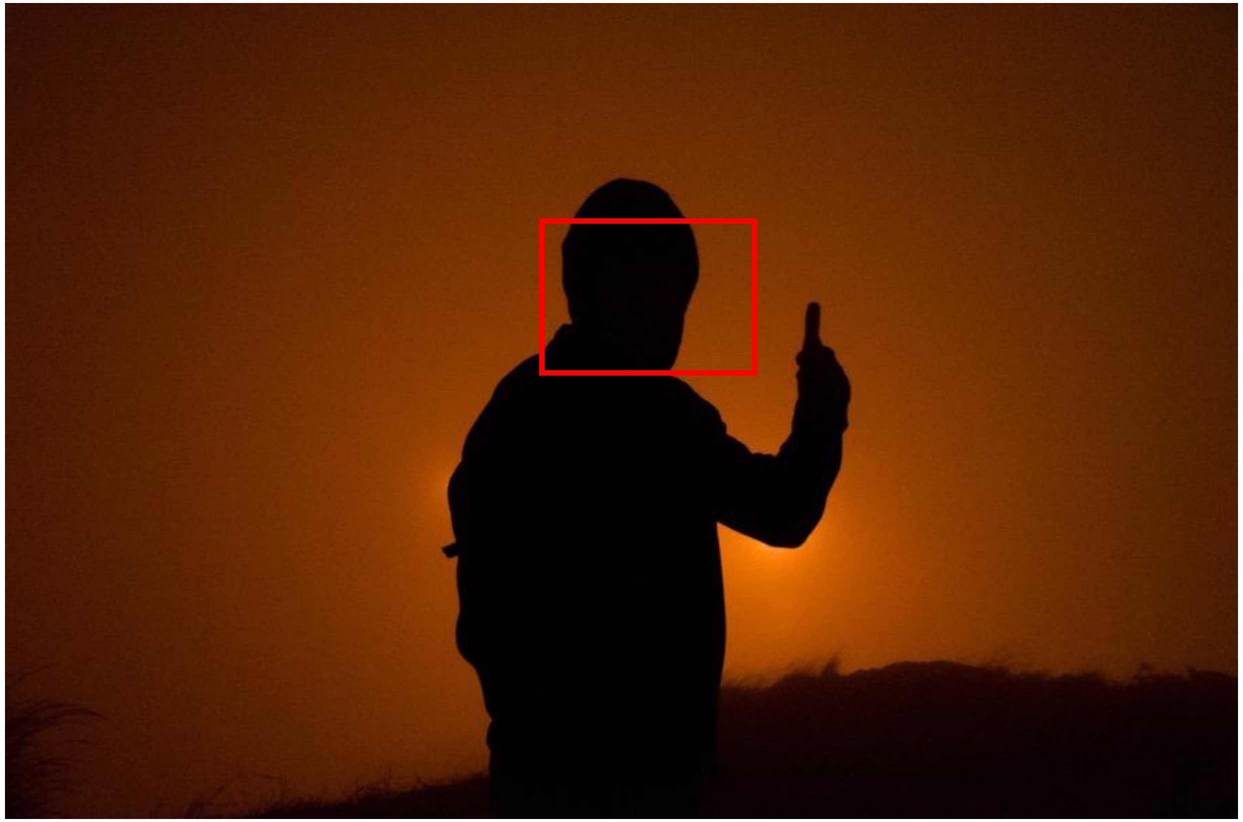
Developing task-adaptive image processing pipelines is an active area of research.

Today's Lecture

- Digital photography
- Standard camera pipeline
- Noise
- Color

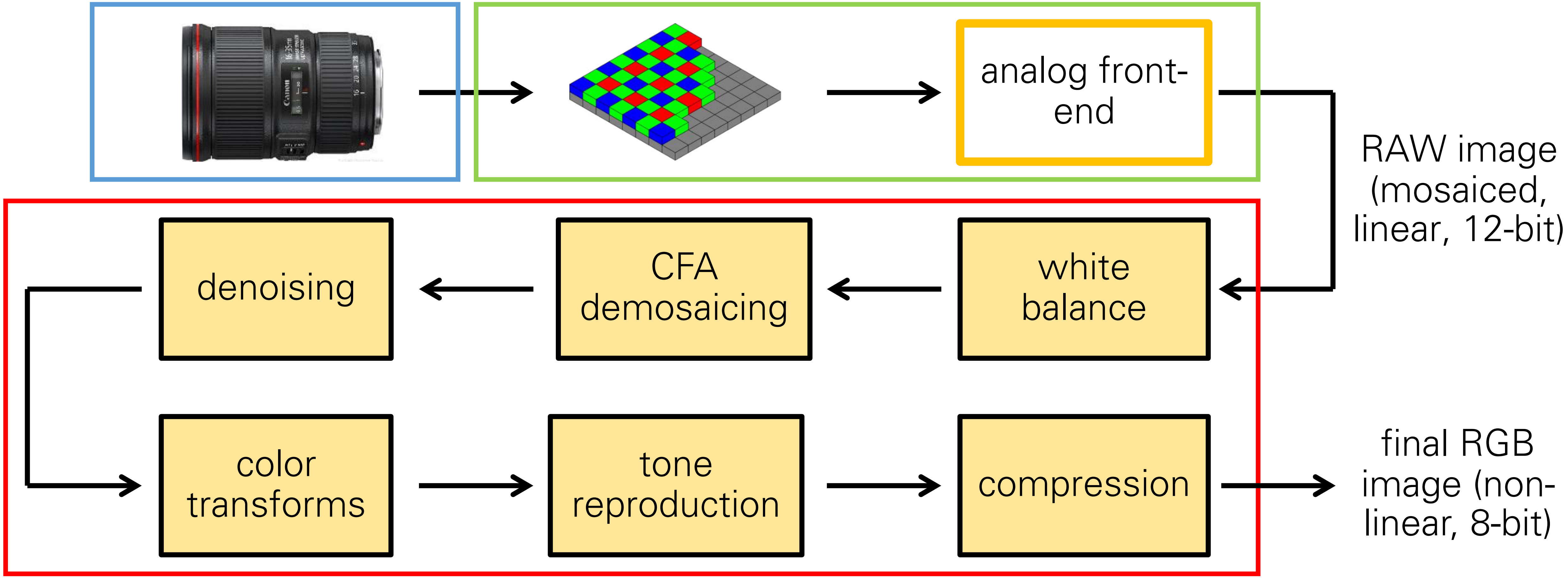
Noise in images

Results in “grainy” appearance.



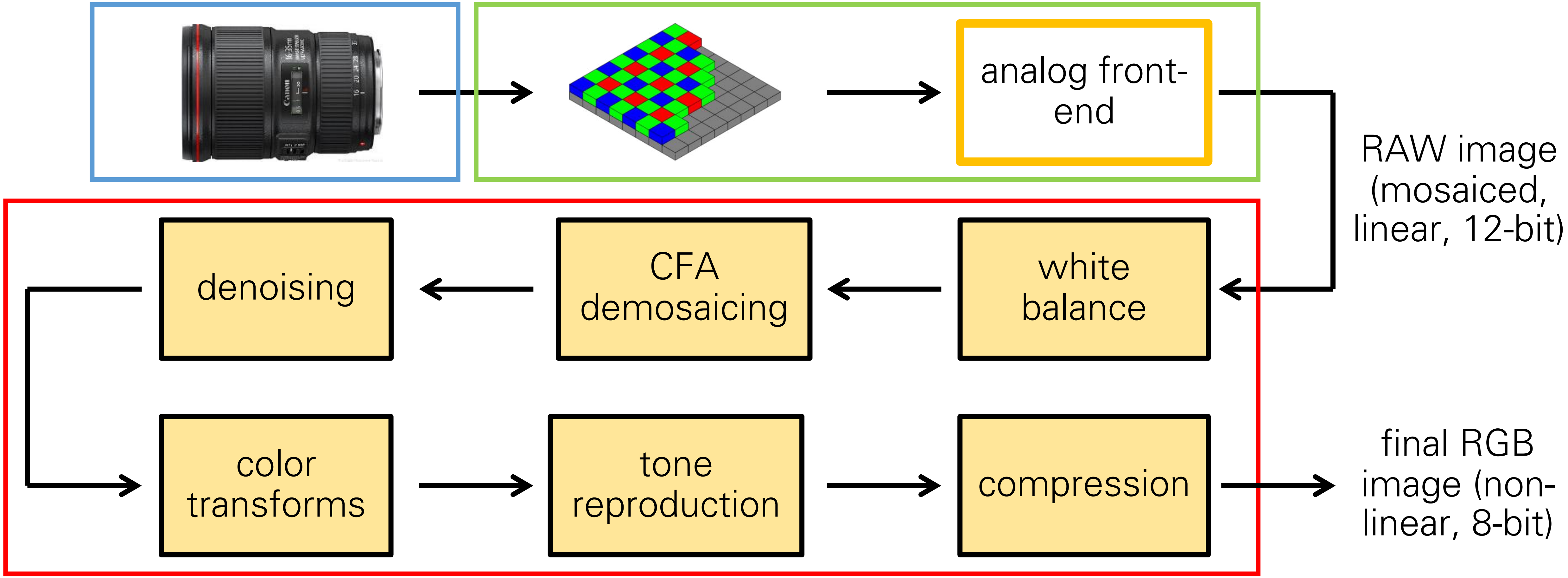
The (in-camera) image processing pipeline

Which part introduces noise?

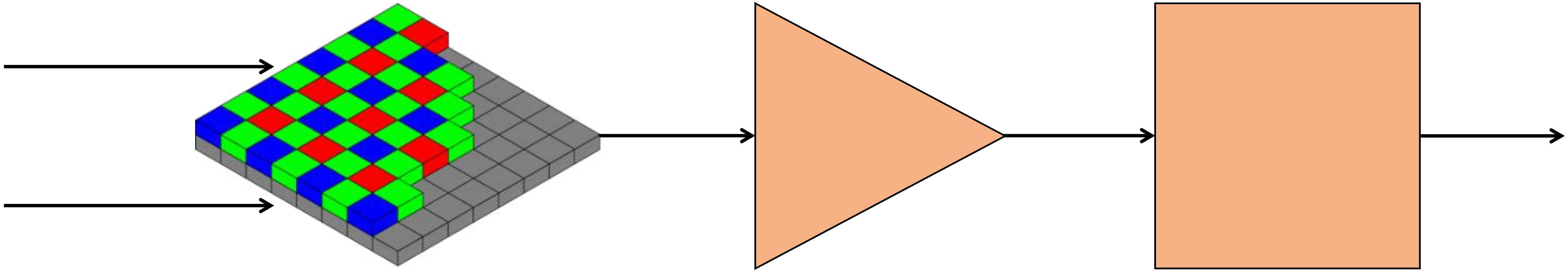


The (in-camera) image processing pipeline

Which part introduces noise?
- Noise is introduced in the green part.

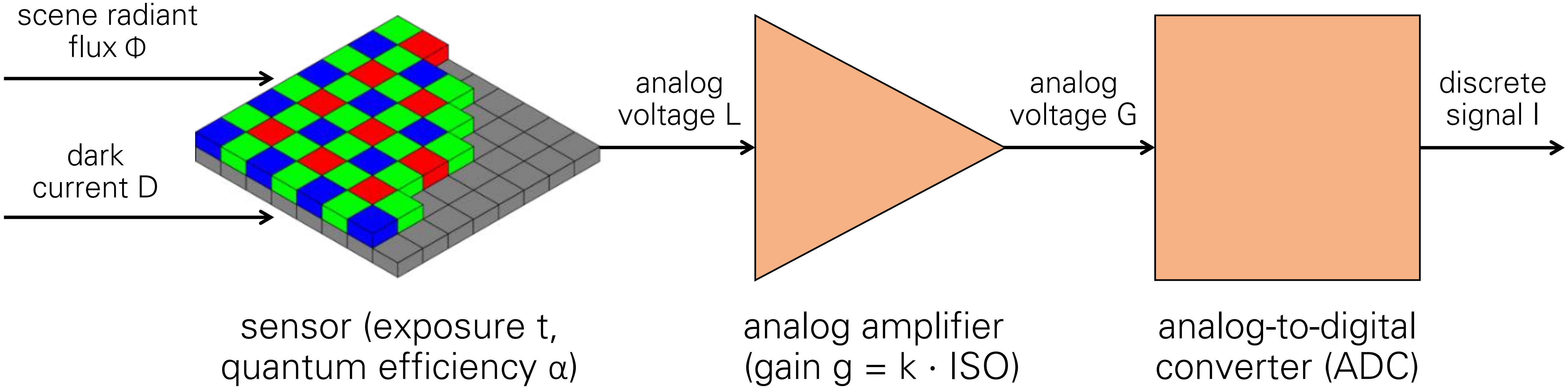


The noisy image formation process

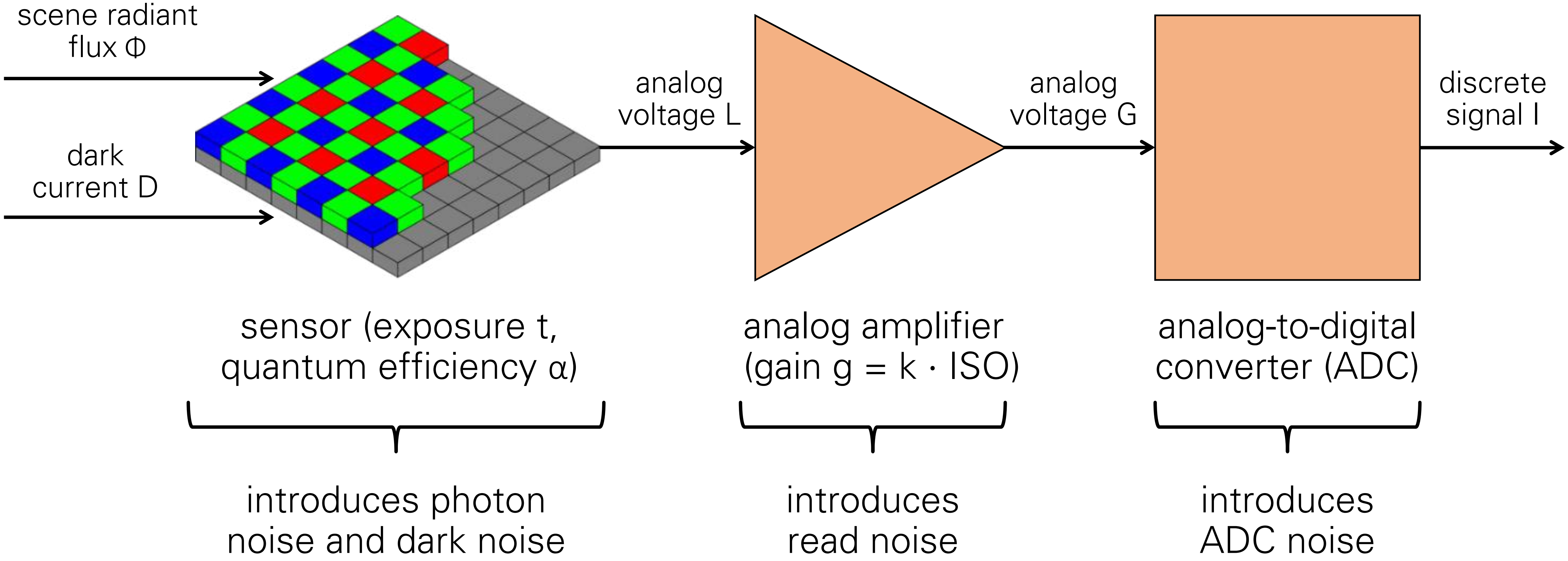


What are the various parts?

The noisy image formation process



The noisy image formation process



- We will be ignoring saturation, but it can be modeled using a clipping operation.

Background: Normal distribution

Is it a continuous or discrete probability distribution?

Background: Normal distribution

Is it a continuous or discrete probability distribution?

- It is continuous.

How many parameters does it depend on?

Background: Normal distribution

Is it a continuous or discrete probability distribution?

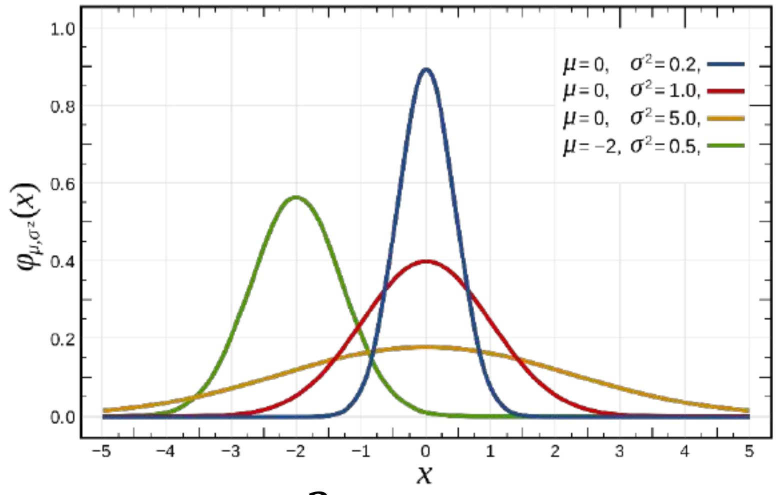
- It is continuous.

How many parameters does it depend on?

- Two parameters, the mean μ and the standard deviation σ.

What is its probability distribution function?

Background: Normal distribution



Is it a continuous or discrete probability distribution?

- It is continuous.

How many parameters does it depend on?

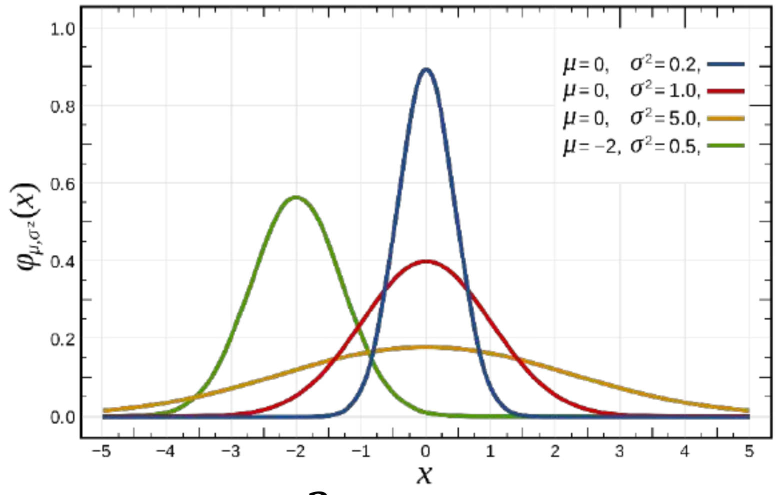
- Two parameters, the mean μ and the standard deviation σ.

What is its probability distribution function?

$$n \sim \mathrm{Normal}(\mu, \sigma) \Leftrightarrow p(n = x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What are its mean and variance?

Background: Normal distribution



Is it a continuous or discrete probability distribution?

- It is continuous.

How many parameters does it depend on?

- Two parameters, the mean μ and the standard deviation σ.

What is its probability distribution function?

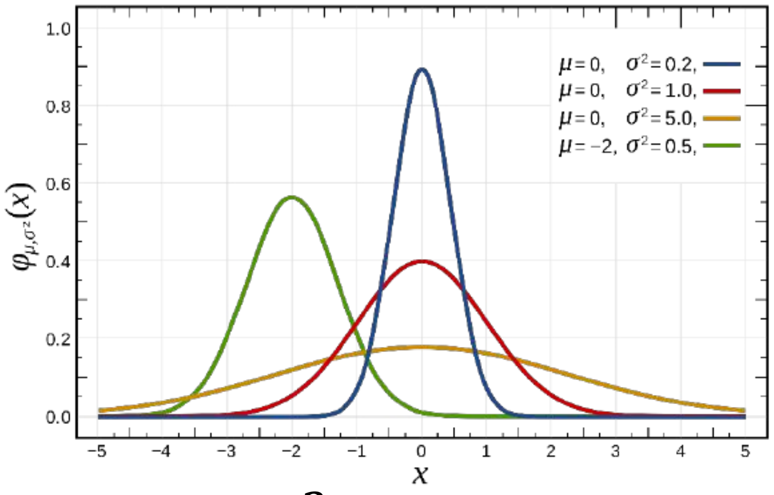
$$n \sim \mathrm{Normal}(\mu, \sigma) \Leftrightarrow p(n = x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What are its mean and variance?

- Mean: $\mu(n) = \mu$
- Variance: $\sigma(n)^2 = \sigma^2$

What is the distribution of the sum of two independent Normal random variables?

Background: Normal distribution



Is it a continuous or discrete probability distribution?

- It is continuous.

How many parameters does it depend on?

- Two parameters, the mean μ and the standard deviation σ.

What is its probability distribution function?

$$n \sim \mathrm{Normal}(\mu, \sigma) \Leftrightarrow p(n = x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What are its mean and variance?

- Mean: $\mu(n) = \mu$
- Variance: $\sigma(n)^2 = \sigma^2$

What is the distribution of the sum of two independent Normal random variables?

$$n_1 \sim \mathrm{Normal}(0, \sigma_1), n_2 \sim \mathrm{Normal}(0, \sigma_2) \Rightarrow n_1 + n_2 \sim \mathrm{Normal}\left(0, \sqrt{\sigma_1^2 + \sigma_2^2}\right)$$

Background: Poisson distribution

Is it a continuous or discrete probability distribution?

Background: Poisson distribution

Is it a continuous or discrete probability distribution?

- It is discrete.

How many parameters does it depend on?

Background: Poisson distribution

Is it a continuous or discrete probability distribution?

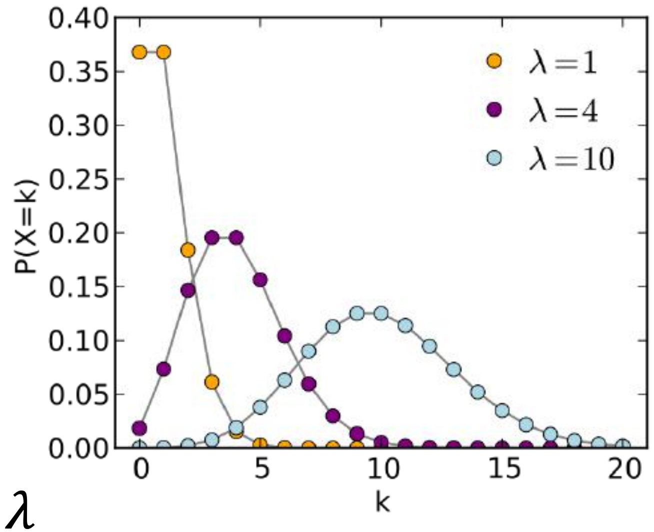
- It is discrete.

How many parameters does it depend on?

- One parameter, the rate λ.

What is its probability mass function?

Background: Poisson distribution



Is it a continuous or discrete probability distribution?

- It is discrete.

How many parameters does it depend on?

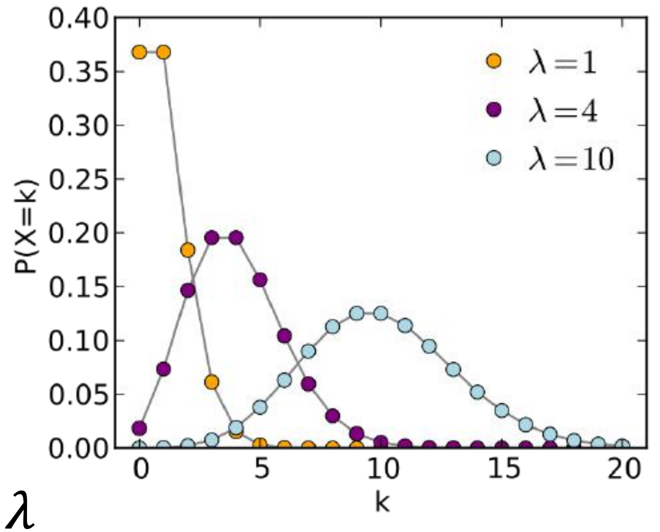
- One parameter, the rate λ.

What is its probability mass function?

$$N \sim \text{Poisson}(\lambda) \Leftrightarrow P(N = k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

What are its mean and variance?

Background: Poisson distribution



Is it a continuous or discrete probability distribution?

- It is discrete.

How many parameters does it depend on?

- One parameter, the rate λ.

What is its probability mass function?

$$N \sim \text{Poisson}(\lambda) \Leftrightarrow P(N = k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

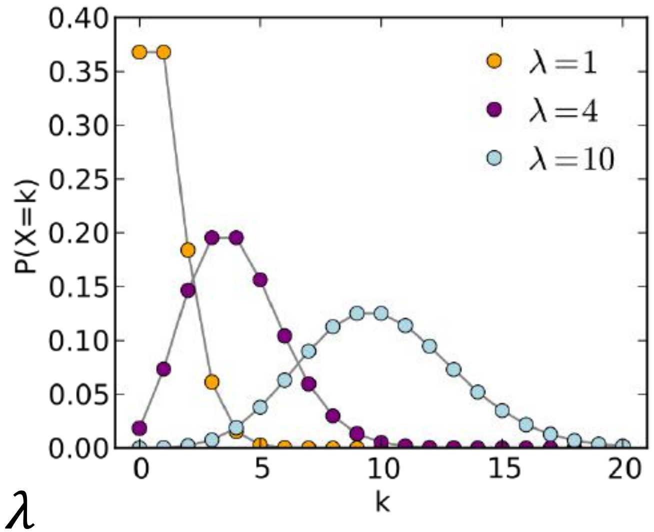
What are its mean and variance?

- Mean: $\mu(N) = \lambda$
- Variance: $\sigma(N)^2 = \lambda$

The mean and variance of a Poisson random variable both equal the rate λ.

What is the distribution of the sum of two independent Poisson random variables?

Background: Poisson distribution



Is it a continuous or discrete probability distribution?

- It is discrete.

How many parameters does it depend on?

- One parameter, the rate λ.

What is its probability mass function?

$$N \sim \text{Poisson}(\lambda) \Leftrightarrow P(N = k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

What are its mean and variance?

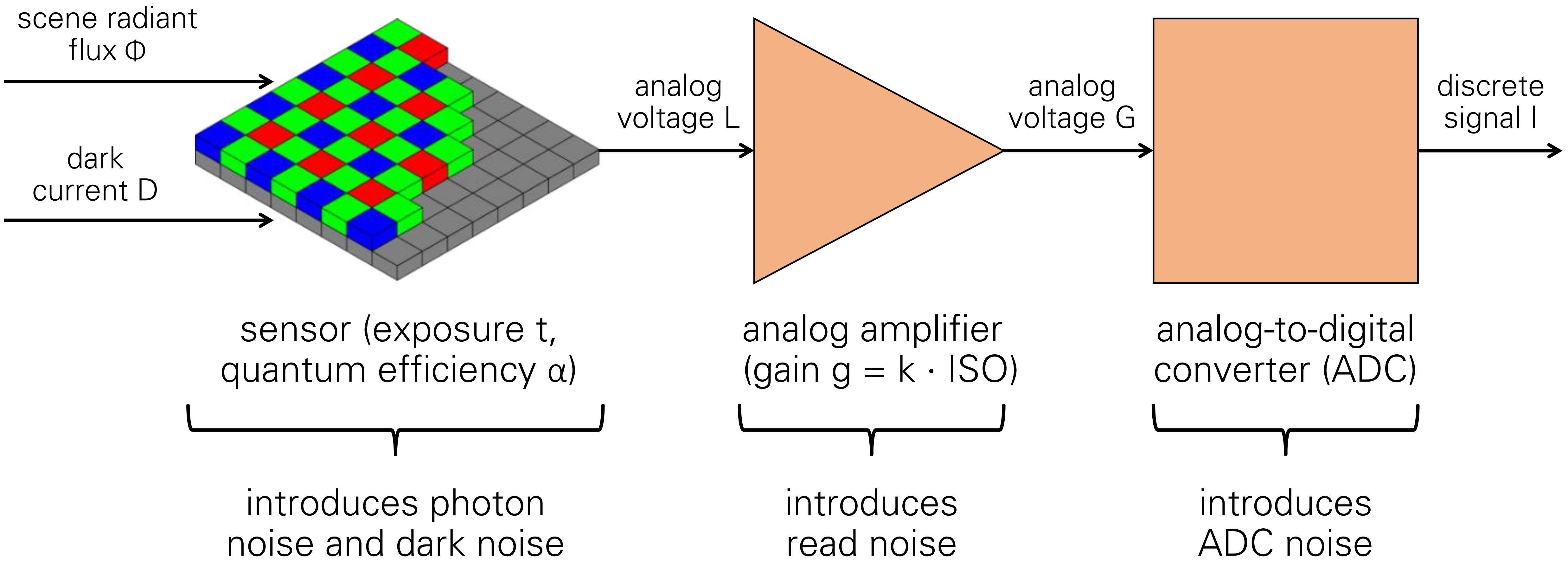
- Mean: $\mu(N) = \lambda$
- Variance: $\sigma(N)^2 = \lambda$

The mean and variance of a Poisson random variable both equal the rate λ.

What is the distribution of the sum of two independent Poisson random variables?

$$N_1 \sim \text{Poisson}(\lambda_1), N_2 \sim \text{Poisson}(\lambda_2) \Rightarrow N_1 + N_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$$

The noisy image formation process



- We will be ignoring saturation, but it can be modeled using a clipping operation.

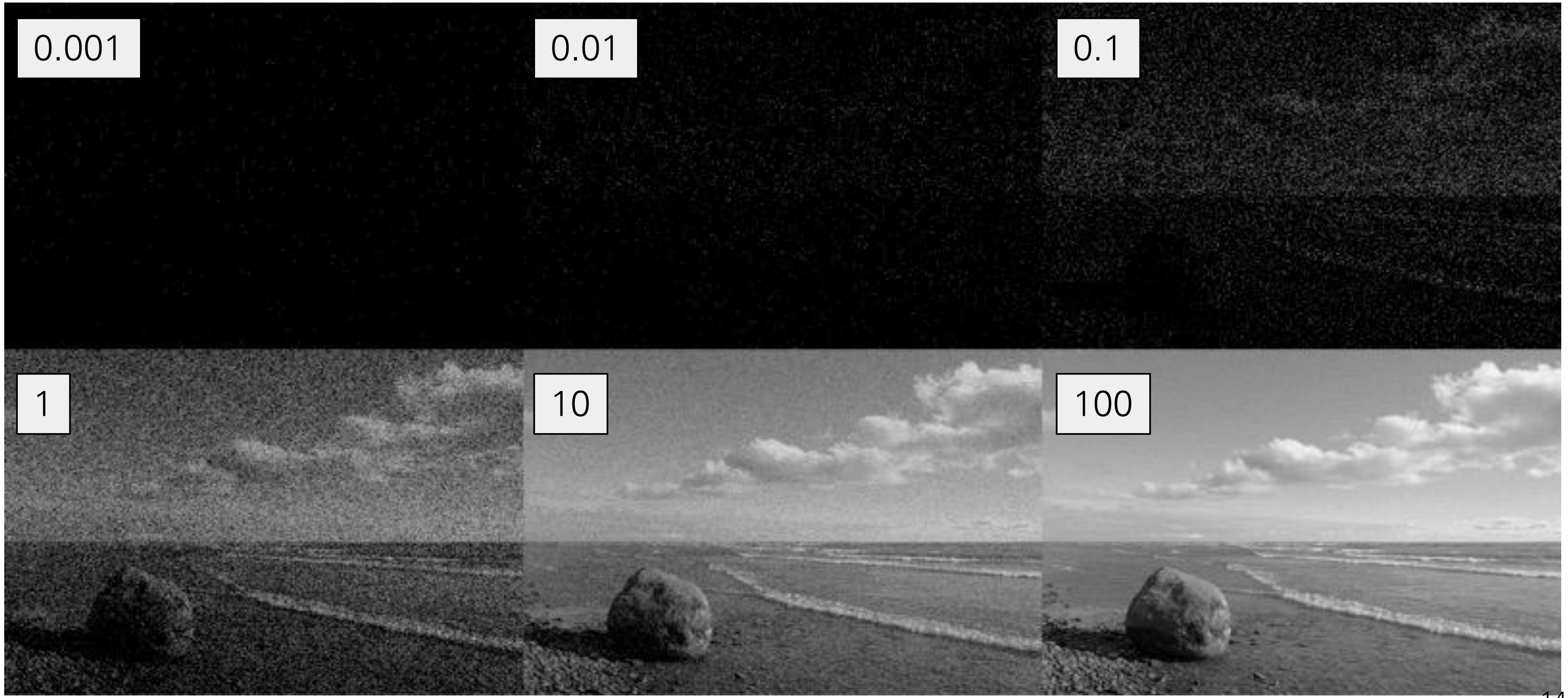
Photon noise

A consequence of the discrete (quantum) nature of light.

- Photon detections are independent random events.
- Total number of detections is Poisson distributed.
- Also known as shot noise and Schott noise.

$$N_{detections} \sim \mathrm{Poisson}[t \cdot \alpha \cdot \Phi]$$

simulated mean
#photons/pixel



Photon noise

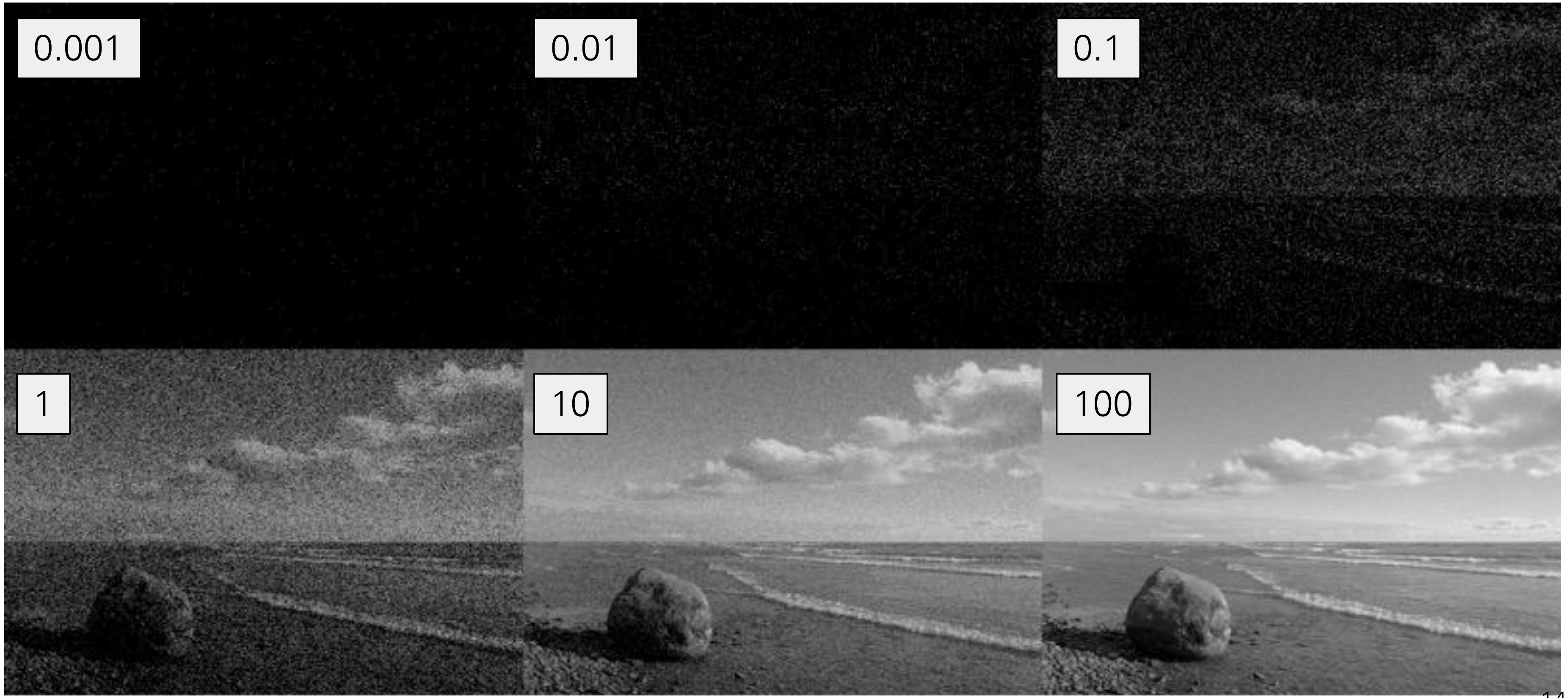
A consequence of the discrete (quantum) nature of light.

- Photon detections are independent random events.
- Total number of detections is Poisson distributed.
- Also known as shot noise and Schott noise.

photon noise depends on scene flux and exposure

$$N_{detections} \sim \text{Poisson}\boxed{[t \cdot \alpha \cdot \Phi]}$$

simulated mean #photons/pixel



Dark noise

A consequence of “phantom detections” by the sensor.

- Electrons are randomly released without any photons.
- Total number of detections is Poisson distributed.
- Increases exponentially with sensor temperature (+6°C ≈ doubling).

$$N_{detections} \sim \text{Poisson}[t \cdot D]$$

Dark noise

A consequence of “phantom detections” by the sensor.

- Electrons are randomly released without any photons.
- Total number of detections is Poisson distributed.
- Increases exponentially with sensor temperature (+6°C ≈ doubling).

dark noise depends on exposure but not on scene

$$N_{\text{detections}} \sim \text{Poisson}\left[\boxed{t \cdot D}\right]$$

Can you think of examples when dark noise is important?

Dark noise

dark noise depends on exposure but not on scene

$$N_{\text{detections}} \sim \text{Poisson}\left(\boxed{t \cdot D}\right)$$

A consequence of “phantom detections” by the sensor.

- Electrons are randomly released without any photons.
- Total number of detections is Poisson distributed.
- Increases exponentially with sensor temperature (+6°C ≈ doubling).

Can you think of examples when dark noise is important?

- Very long exposures (astrophotography, pinhole camera).

Can you think of ways to mitigate dark noise?

Dark noise

dark noise depends on exposure but not on scene

$$N_{\text{detections}} \sim \text{Poisson}\left(\boxed{t \cdot D}\right)$$

A consequence of "phantom detections" by the sensor.

- Electrons are randomly released without any photons.
- Total number of detections is Poisson distributed.
- Increases exponentially with sensor temperature (+6°C ≈ doubling).

Can you think of examples when dark noise is important?

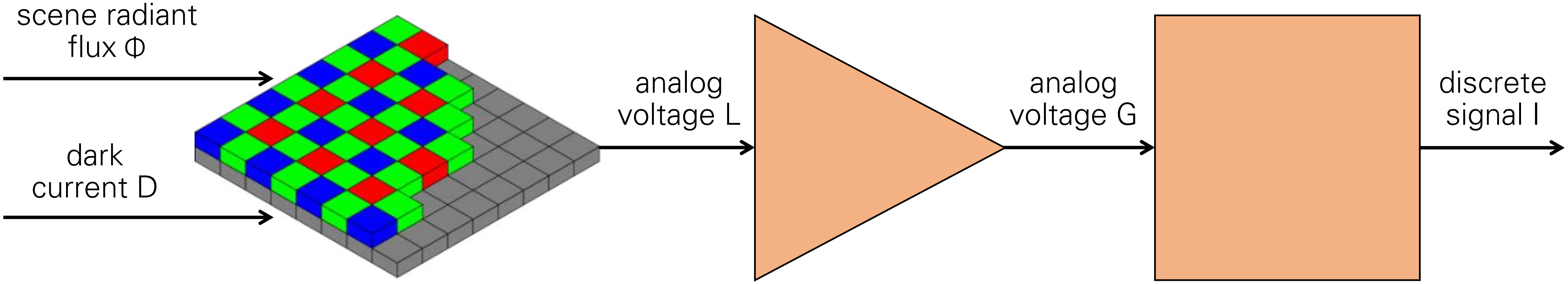
- Very long exposures (astrophotography, pinhole camera)

Can you think of ways to mitigate dark noise?

- Cool the sensor.

Passive and TE-Cooled Scientific CCD Cameras

The noisy image formation process



sensor (exposure t, quantum efficiency α) — introduces photon noise and dark noise

analog amplifier (gain $g = k \cdot \text{ISO}$) — introduces read noise

analog-to-digital converter (ADC) — introduces ADC noise

- What is the distribution of the sensor readout L?

The distribution of the sensor readout

We know that the sensor readout is the sum of all released electrons:

$$L = N_{\mathbf{photon\_detections}} + N_{\mathbf{phantom\_detections}}$$

What is the distribution of photon detections?

The distribution of the sensor readout

We know that the sensor readout is the sum of all released electrons:

$$L = N_{\mathbf{photon\_detections}} + N_{\mathbf{phantom\_detections}}$$

What is the distribution of photon detections?

$$N_{\mathbf{photon\_detections}} \sim \text{Poisson}(t \cdot \alpha \cdot \Phi)$$

What is the distribution of phantom detections?

The distribution of the sensor readout

We know that the sensor readout is the sum of all released electrons:

$$L = N_{\mathbf{photon\_detections}} + N_{\mathbf{phantom\_detections}}$$

What is the distribution of photon detections?

$$N_{\mathbf{photon\_detections}} \sim \mathrm{Poisson}(t \cdot \alpha \cdot \Phi)$$

What is the distribution of phantom detections?

$$N_{\mathbf{phantom\_detections}} \sim \mathrm{Poisson}(t \cdot D)$$

What is the distribution of the sensor readout?

The distribution of the sensor readout

We know that the sensor readout is the sum of all released electrons:

$$L = N_{\mathbf{photon\_detections}} + N_{\mathbf{phantom\_detections}}$$

What is the distribution of photon detections?

$$N_{\mathbf{photon\_detections}} \sim \text{Poisson}(t \cdot \alpha \cdot \Phi)$$

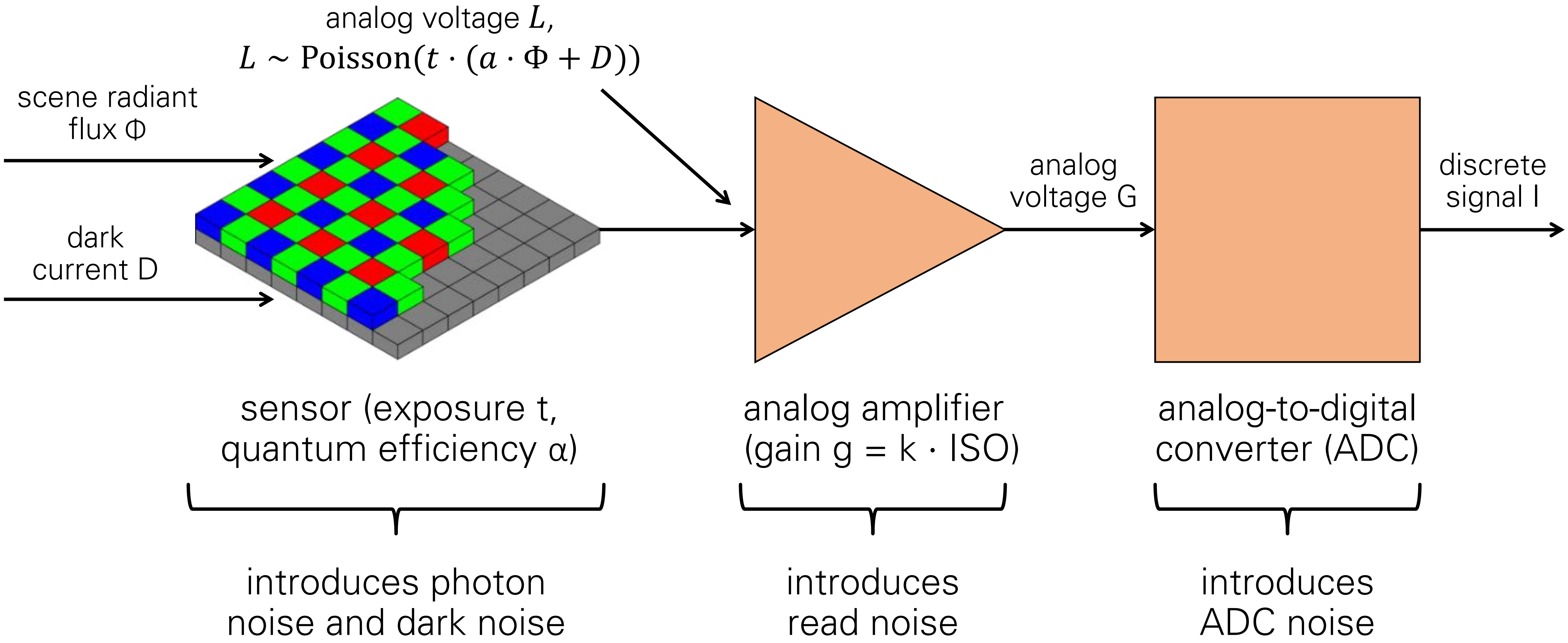
What is the distribution of phantom detections?

$$N_{\mathbf{phantom\_detections}} \sim \text{Poisson}(t \cdot D)$$

What is the distribution of the sensor readout?

$$L \sim \text{Poisson}(t \cdot (a \cdot \Phi + D))$$

The noisy image formation process



Read and ADC noise

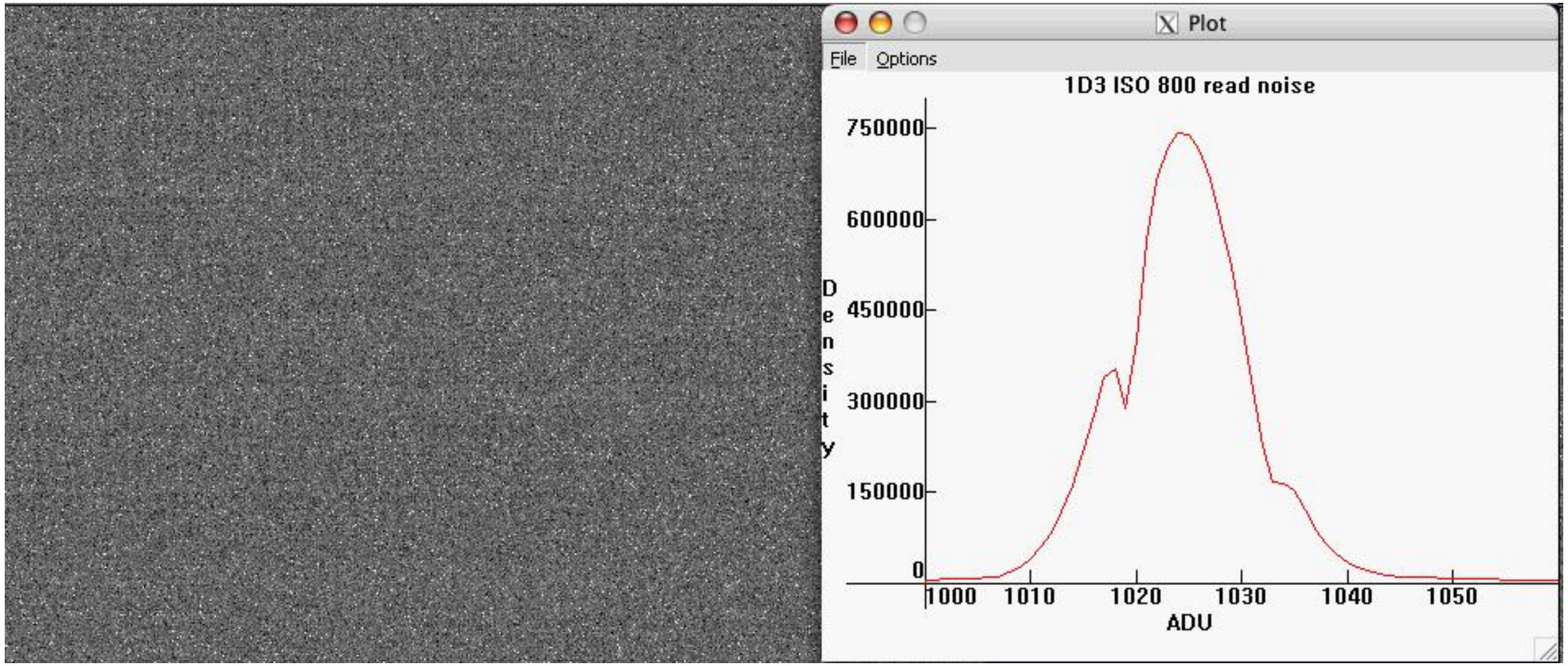
A consequence of random voltage fluctuations before and after amplifier.

- Both are independent of scene and exposure.
- Both are normally (zero-mean Guassian) distributed.
- ADC noise includes quantization errors.

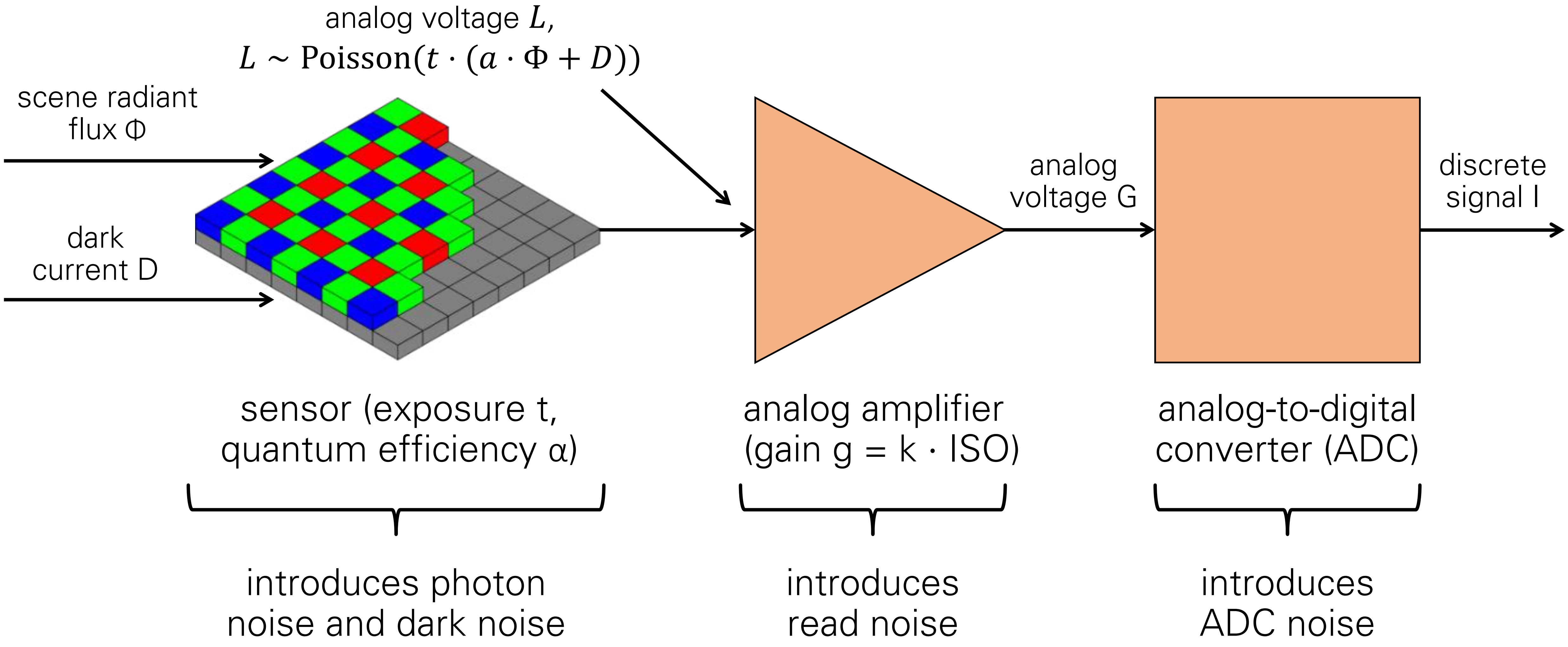
$$n_{read} \sim \text{Normal}(0, \sigma_{read})$$

$$n_{ADC} \sim \text{Normal}(0, \sigma_{ADC})$$

Very important for dark pixels.



The noisy image formation process



- How can we express the voltage G and discrete intensity I?

Expressions for the amplifier and ADC outputs

Both read noise and ADC noise are additive and zero-mean.

- How can we express the output of the amplifier?

Expressions for the amplifier and ADC outputs

Both read noise and ADC noise are additive and zero-mean.

- How can we express the output of the amplifier?

$$G = L \cdot g + n_{\mathbf{read}} \cdot g$$

don't forget to account for the ISO-dependent gain

- How can we express the output of the ADC?

Expressions for the amplifier and ADC outputs

Both read noise and ADC noise are additive and zero-mean.

- How can we express the output of the amplifier?

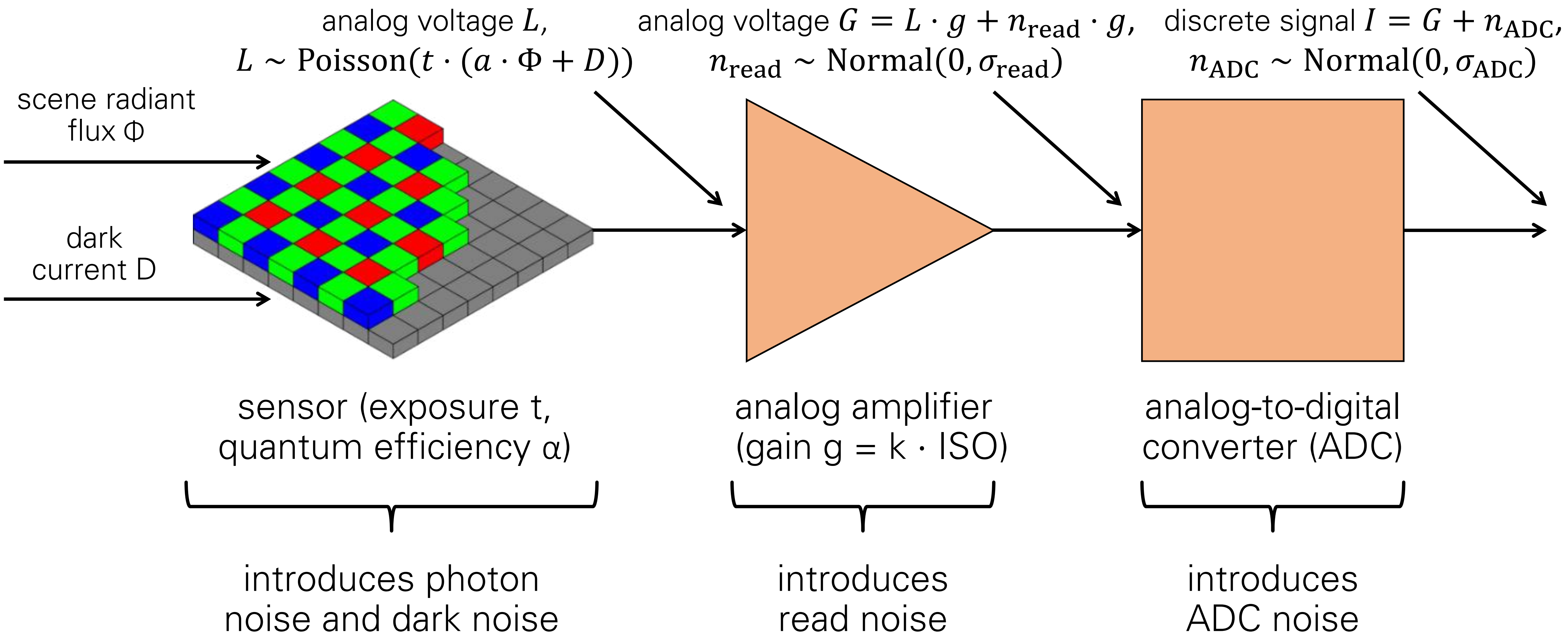
$$G = L \cdot g + n_{\mathbf{read}} \cdot g$$

don't forget to account for the ISO-dependent gain

- How can we express the output of the ADC?

$$I = G + n_{\mathrm{ADC}}$$

The noisy image formation process



Putting it all together

Without saturation, the digital intensity equals:

$$I = L \cdot g + n_{\mathrm{read}} \cdot g + n_{\mathrm{ADC}}$$

where

$$L \sim \mathrm{Poisson}\big(t \cdot (a \cdot \Phi + D)\big)$$
$$n_{\mathrm{read}} \sim \mathrm{Normal}(0, \sigma_{\mathrm{read}})$$
$$n_{\mathrm{ADC}} \sim \mathrm{Normal}(0, \sigma_{\mathrm{ADC}})$$

What is the mean of the digital intensity (assuming no saturation)?

$$E(I) =$$

Putting it all together

Without saturation, the digital intensity equals:

$$I = L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}$$

where

$$L \sim \text{Poisson}\big(t \cdot (a \cdot \Phi + D)\big)$$
$$n_{\text{read}} \sim \text{Normal}(0, \sigma_{\text{read}})$$
$$n_{\text{ADC}} \sim \text{Normal}(0, \sigma_{\text{ADC}})$$

What is the mean of the digital intensity (assuming no saturation)?

$$\begin{aligned} E(I) &= E(L \cdot g) + E(n_{\text{read}} \cdot g) + E(n_{\text{ADC}}) \\ &= \end{aligned}$$

Putting it all together

Without saturation, the digital intensity equals:

$$I = L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}$$ where $$\begin{aligned} L &\sim \text{Poisson}\big(t \cdot (a \cdot \Phi + D)\big) \\ n_{\text{read}} &\sim \text{Normal}(0, \sigma_{\text{read}}) \\ n_{\text{ADC}} &\sim \text{Normal}(0, \sigma_{\text{ADC}}) \end{aligned}$$

What is the mean of the digital intensity (assuming no saturation)?

$$\begin{aligned} E(I) &= E(L \cdot g) + E(n_{\text{read}} \cdot g) + E(n_{\text{ADC}}) \\ &= t \cdot (a \cdot \Phi + D) \cdot g \end{aligned}$$

What is the variance of the digital intensity (assuming no saturation)?

$$\sigma(I)^2 =$$

Putting it all together

Without saturation, the digital intensity equals:

$$I = L \cdot g + n_{\mathrm{read}} \cdot g + n_{\mathrm{ADC}}$$

where

$$L \sim \mathrm{Poisson}\big(t \cdot (a \cdot \Phi + D)\big)$$
$$n_{\mathrm{read}} \sim \mathrm{Normal}(0, \sigma_{\mathrm{read}})$$
$$n_{\mathrm{ADC}} \sim \mathrm{Normal}(0, \sigma_{\mathrm{ADC}})$$

What is the mean of the digital intensity (assuming no saturation)?

$$\begin{aligned} E(I) &= E(L \cdot g) + E(n_{\mathrm{read}} \cdot g) + E(n_{\mathrm{ADC}}) \\ &= t \cdot (a \cdot \Phi + D) \cdot g \end{aligned}$$

What is the variance of the digital intensity (assuming no saturation)?

$$\begin{aligned} \sigma(I)^2 &= \sigma(L \cdot g)^2 + \sigma(n_{\mathrm{read}} \cdot g)^2 + \sigma(n_{\mathrm{ADC}})^2 \\ &= \end{aligned}$$

Putting it all together

Without saturation, the digital intensity equals:

$$I = L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}$$

where

$$L \sim \text{Poisson}\big(t \cdot (a \cdot \Phi + D)\big)$$
$$n_{\text{read}} \sim \text{Normal}(0, \sigma_{\text{read}})$$
$$n_{\text{ADC}} \sim \text{Normal}(0, \sigma_{\text{ADC}})$$

What is the mean of the digital intensity (assuming no saturation)?

$$\begin{aligned} E(I) &= E(L \cdot g) + E(n_{\text{read}} \cdot g) + E(n_{\text{ADC}}) \\ &= t \cdot (a \cdot \Phi + D) \cdot g \end{aligned}$$

What is the variance of the digital intensity (assuming no saturation)?

$$\begin{aligned} \sigma(I)^2 &= \sigma(L \cdot g)^2 + \sigma(n_{\text{read}} \cdot g)^2 + \sigma(n_{\text{ADC}})^2 \\ &= t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2 \end{aligned}$$

How do we compute mean and variance in practice?

How do we compute mean and variance in practice?

Mean: capture multiple linear images with identical settings and average.

$$\bar{I} = \frac{1}{N}\sum_{n=1}^{N} I_n \xrightarrow{N\to\infty} E(I)$$

How do we compute mean and variance in practice?

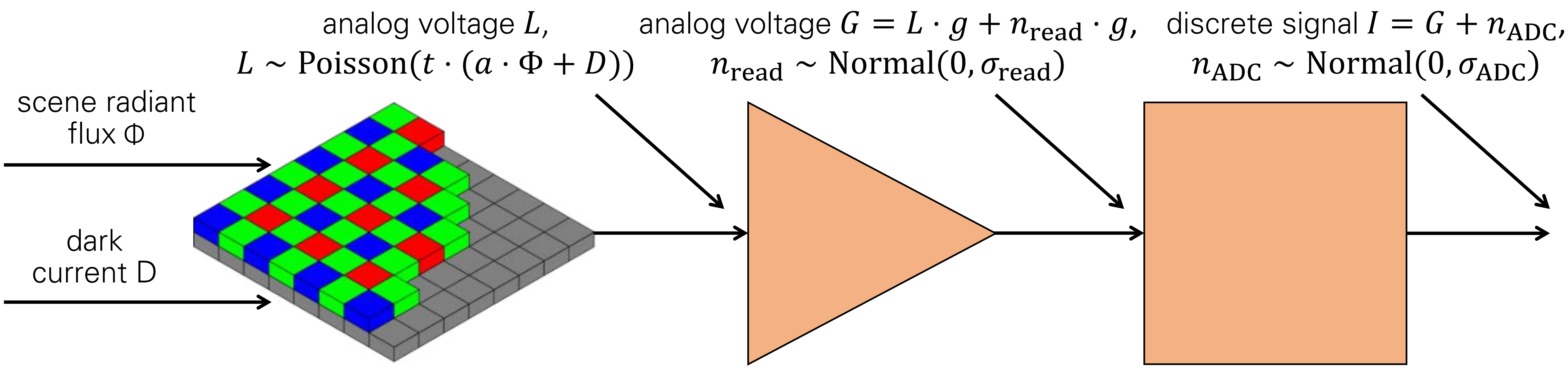
Mean: capture multiple linear images with identical settings and average.

$$\bar{I} = \frac{1}{N}\sum_{n=1}^{N} I_n \xrightarrow{N\to\infty} E(I)$$

Variance: capture multiple linear images with identical settings and form variance estimator.

$$\bar{\Sigma} = \frac{1}{N-1}\sum_{n=1}^{N} (I_n - \bar{I})^2 \xrightarrow{N\to\infty} \sigma(I)^2$$

The noisy image formation process



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\text{max}})$$

saturation level

intensity mean and variance (without saturation):

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Affine noise model

Combine read and ADC noise into a single additive noise term:

$$I = L \cdot g + n_{\mathbf{add}} \qquad \text{where} \qquad n_{\mathbf{add}} = n_{\mathbf{read}} \cdot g + n_{\mathbf{ADC}}$$

What is the distribution of the additive noise term?

Affine noise model

Combine read and ADC noise into a single additive noise term:

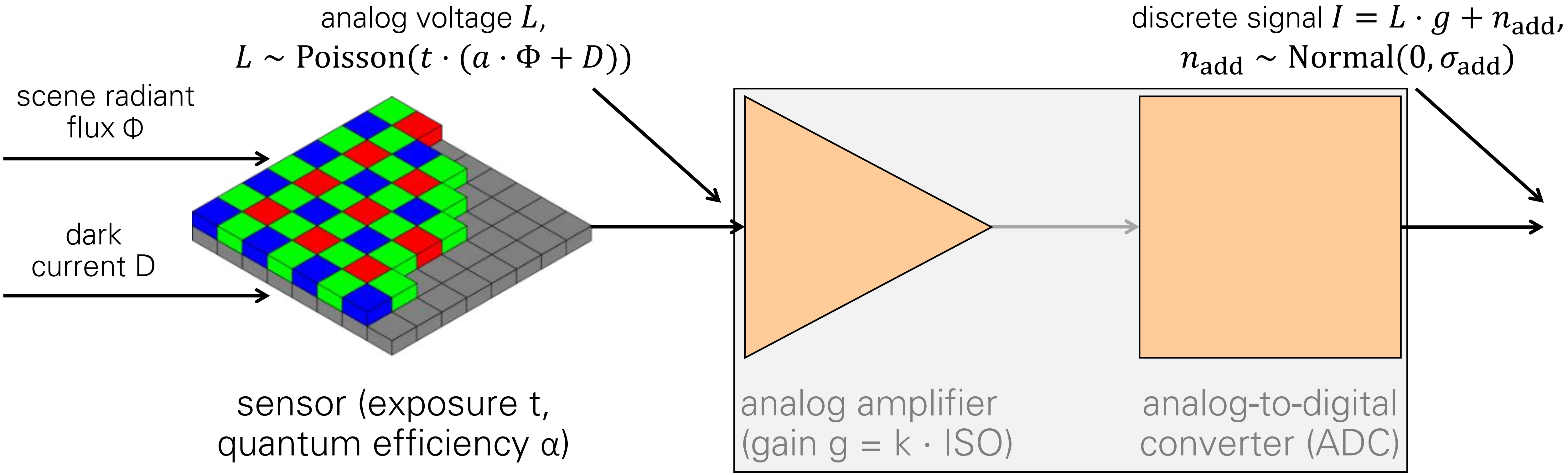
$$I = L \cdot g + n_{\mathrm{add}} \qquad \text{where} \qquad n_{\mathrm{add}} = n_{\mathrm{read}} \cdot g + n_{\mathrm{ADC}}$$

What is the distribution of the additive noise term?

- Sum of two independent, normal random variables.

$$n_{\mathrm{add}} \sim \mathrm{Normal}(0, \sqrt{\sigma_{\mathrm{read}}^2 \cdot g^2 + \sigma_{\mathrm{ADC}}^2})$$

Affine noise model



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{add}}, I_{\max})$$

intensity mean and variance (without saturation):

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{add}}^2$$

Some observations

Is image intensity an unbiased estimator of (scaled) scene radiant flux?

Some observations

Is image intensity an unbiased estimator of (scaled) scene radiant flux?

- No, because of dark noise (term $\mathrm{t} \cdot \mathrm{D} \cdot \mathrm{g}$ in the mean).
- Averaging multiple images cancels out read and ADC noise, but not dark noise.

When are photon noise and additive noise dominant?

Some observations

Is image intensity an unbiased estimator of (scaled) scene radiant flux?

- No, because of dark noise (term $\mathrm{t} \cdot \mathrm{D} \cdot \mathrm{g}$ in the mean).
- Averaging multiple images cancels out read and ADC noise, but not dark noise.

When are photon noise and additive noise dominant?

- Photon noise is dominant in very bright scenes.
- Additive noise is dominant in very dark scenes.

Can we ever completely remove noise?

Some observations

Is image intensity an unbiased estimator of (scaled) scene radiant flux?

- No, because of dark noise (term $\mathrm{t} \cdot \mathrm{D} \cdot \mathrm{g}$ in the mean).
- Averaging multiple images cancels out read and ADC noise, but not dark noise.

When are photon noise and additive noise dominant?

- Photon noise is dominant in very bright scenes.
- Additive noise is dominant in very dark scenes.

Can we ever completely remove noise?

- We cannot eliminate photon noise.
- Super-sensitive detectors have pure Poisson photon noise.

single-photon avalanche photodiode (SPAD)

Summary: noise regimes

regime	dominant noise	notes
bright pixels	photon noise	scene-dependent
dark pixels	read and ADC noise	scene-independent
low ISO	ADC noise	post-gain
high ISO	photon and read noise	pre-gain
long exposures dependence	dark noise	thermal

discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\max})$$

intensity mean and variance (without saturation):

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Summary: noise regimes

regime	dominant noise	notes
bright pixels	photon noise	scene-dependent
dark pixels	read and ADC noise	scene-independent
low ISO	ADC noise	post-gain
high ISO	photon and read noise	pre-gain
long exposures dependence	dark noise	thermal

Does this mean that using high exposure makes images more “noisy”?

discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\max})$$

intensity mean and variance (without saturation):

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Signal-to-noise ratio

Variance versus signal-to-noise ratio

Variance?

Variance versus signal-to-noise ratio

Variance is an absolute measure of the (squared) magnitude of noise:

$$\sigma(I)^2 = E\left(\left(I - E(I)\right)^2\right) = E(I^2) - E(I)^2$$

Signal-to-noise ratio (SNR)?

Variance versus signal-to-noise ratio

Variance is an absolute measure of the (squared) magnitude of noise:

$$\sigma(I)^2 = E\left(\left(I - E(I)\right)^2\right) = E(I^2) - E(I)^2$$

Signal-to-noise ratio (SNR) is a relative measure of the (inverse squared) magnitude of noise:

$$\mathrm{SNR} = \frac{E(I)^2}{\sigma(I)^2}$$

When noise decreases:

- The variance…
- The SNR…

Variance versus signal-to-noise ratio

Variance is an absolute measure of the (squared) magnitude of noise:

$$\sigma(I)^2 = E\left(\left(I - E(I)\right)^2\right) = E(I^2) - E(I)^2$$

Signal-to-noise ratio (SNR) is a relative measure of the (inverse squared) magnitude of noise:

$$\mathrm{SNR} = \frac{E(I)^2}{\sigma(I)^2}$$

When noise decreases:

- The variance decreases.
- The SNR increases.

The case of sensor noise

Assuming for simplicity that there is no dark current:

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2} \qquad \sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

What happens when the exposure time or flux are very large?

The case of sensor noise

Assuming for simplicity that there is no dark current:

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2} \qquad \sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

What happens when the exposure time or flux are very large?

- We can ignore additive (read and ADC) noise terms.

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2} = t \cdot a \cdot \Phi \qquad \sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2$$

What happens when the flux or exposure time are very small?

The case of sensor noise

Assuming for simplicity that there is no dark current:

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2} \qquad \sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

What happens when the exposure time or flux are very large?

- We can ignore additive (read and ADC) noise terms.

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2} = t \cdot a \cdot \Phi \qquad \sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2$$

What happens when the flux or exposure time are very small?

- We can ignore scene-dependent noise terms.

$$\text{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{\sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2} \qquad \sigma(I)^2 = \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

The case of sensor noise

Assuming for simplicity that there is no dark current:

$$\mathrm{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\mathrm{read}}^2 \cdot g^2 + \sigma_{\mathrm{ADC}}^2}$$

$$\sigma(I)^2 = t \cdot a \cdot \Phi \cdot g^2 + \sigma_{\mathrm{read}}^2 \cdot g^2 + \sigma_{\mathrm{ADC}}^2$$

What happens when the exposure time or flux are very large?

- We can ignore additive (read and ADC) noise terms.

$$\mathrm{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{t \cdot a \cdot \Phi \cdot g^2} = t \cdot a \cdot \Phi$$

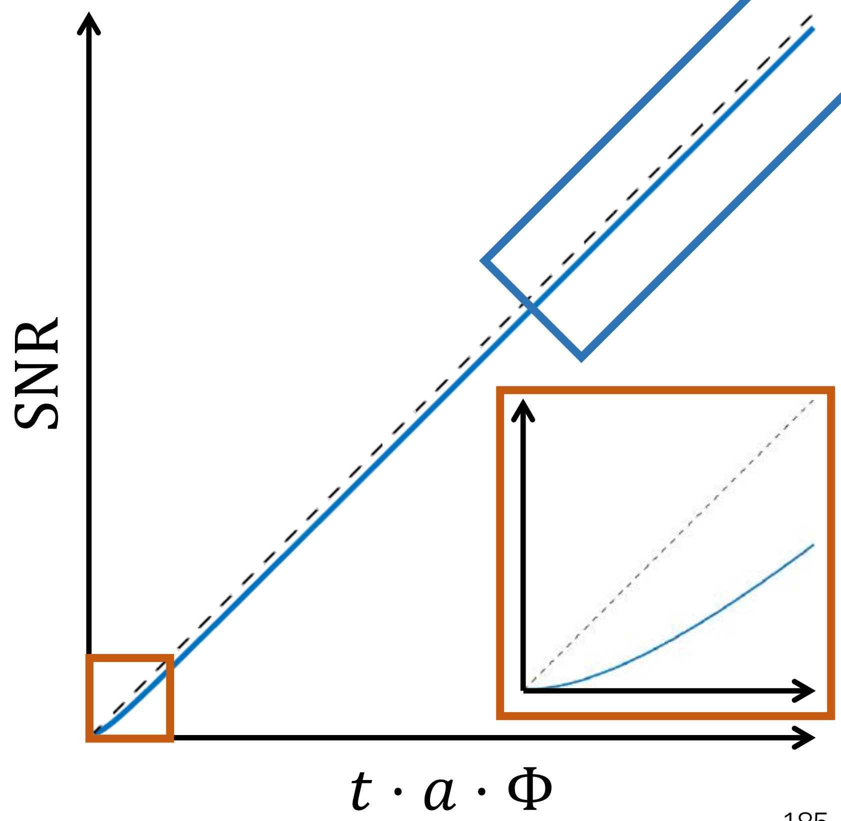
photon-noise-limited case

What happens when the flux or exposure time are very small?

- We can ignore scene-dependent noise terms.

$$\mathrm{SNR} = \frac{(t \cdot a \cdot \Phi \cdot g)^2}{\sigma_{\mathrm{read}}^2 \cdot g^2 + \sigma_{\mathrm{ADC}}^2}$$

additive-noise-limited case



The case of sensor noise

As flux or exposure time increase:

- The noise variance increases.
- The SNR also increases.

Even though the absolute magnitude of noise increases, its relative magnitude compared to the signal we are measuring decreases.

→ Our measurements become less noisy as flux or exposure time increase.

(For the case of exposure time, we need to be careful to also take into account dark noise.)

Effects of exposure time

fewer photons captured →

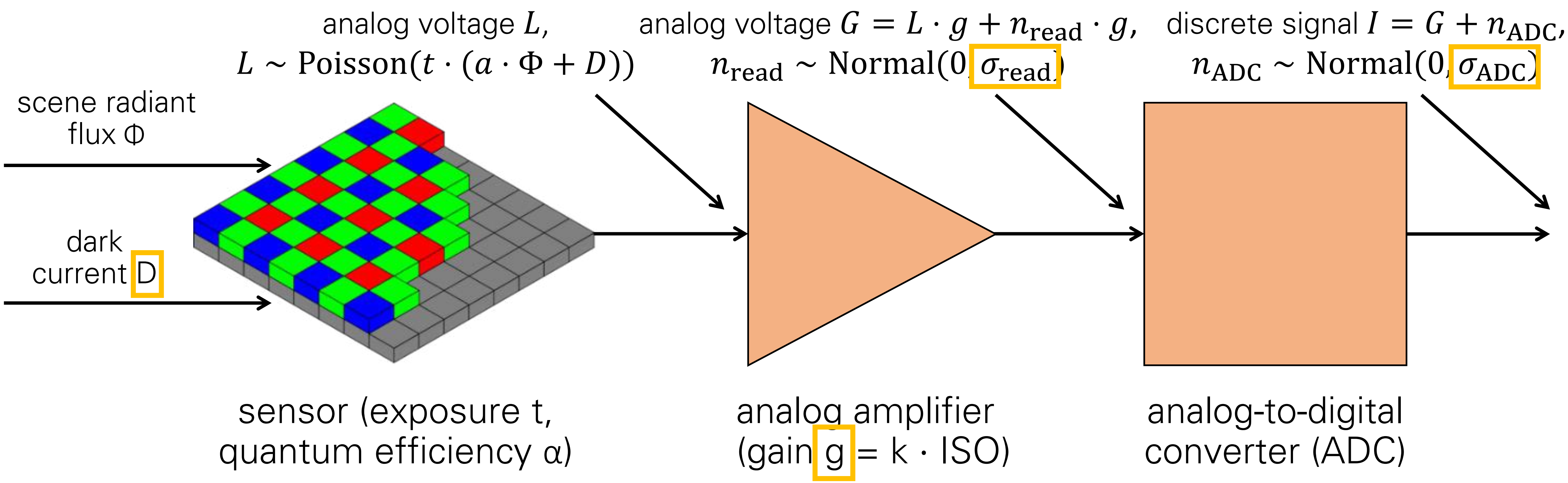
Slide credit: Chen et al.

Effects of exposure time

Slide credit: Chen et al.

Noise calibration

How can we estimate the various parameters?



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\max})$$

saturation level

intensity mean and variance:

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Estimating the dark current

Can you think of a procedure for estimating the dark current D?

Estimating the dark current

Can you think of a procedure for estimating the dark current D?

- Capture multiple images with the sensor completely blocked and average to form the dark frame.

Why is the dark frame a valid estimator of the dark current D?

Estimating the dark current

Can you think of a procedure for estimating the dark current D?

- Capture multiple images with the sensor completely blocked and average to form the dark frame.

Why is the dark frame a valid estimator of the dark current D?

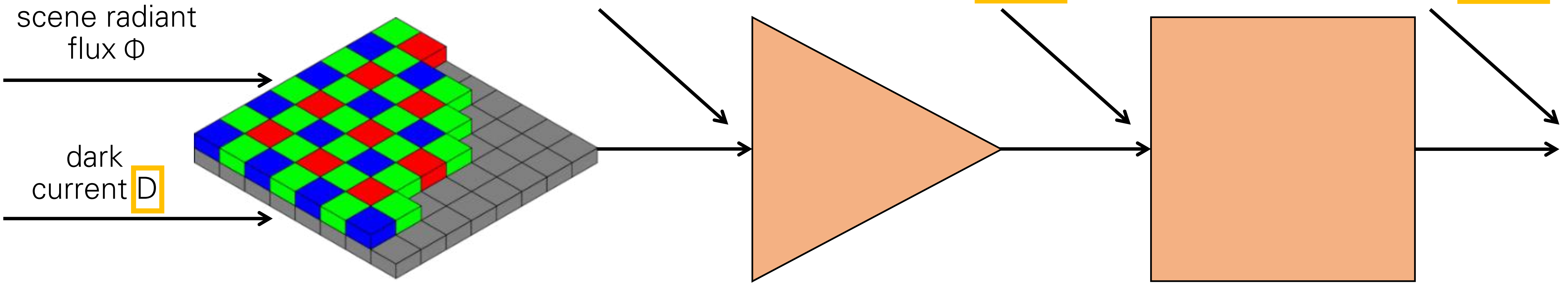
- By blocking the sensor, we effectively set $\Phi = 0$.
- Average intensity becomes:

$$E(I) = t \cdot (a \cdot 0 + D) \cdot g = t \cdot D \cdot g$$

- The dark frame needs to be computed separately for each ISO setting, unless we can also calibrate the gain g.

For the rest of these slides, we assume that we have calibrated D and removed it from captured images (by subtracting from them the dark frame).

Noise model before dark frame subtraction



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\max})$$

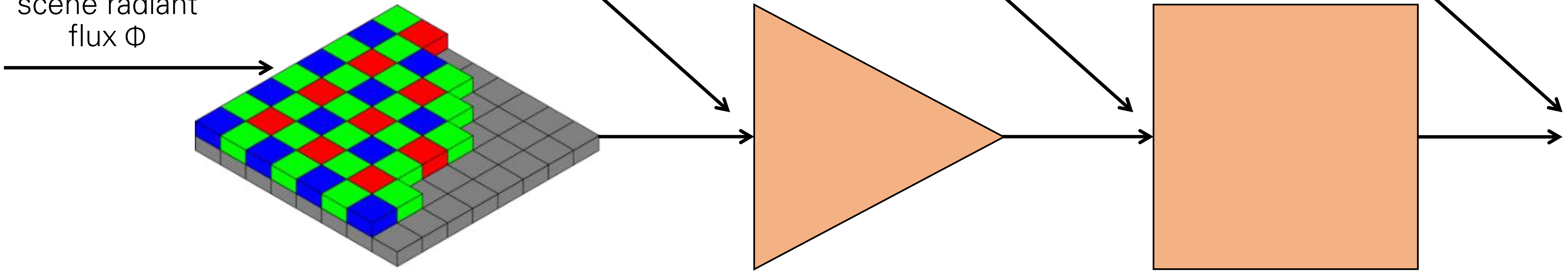
saturation level

intensity mean and variance:

$$E(I) = t \cdot (a \cdot \Phi + D) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Noise model before dark frame subtraction



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{read}} \cdot g + n_{\text{ADC}}, I_{\max})$$

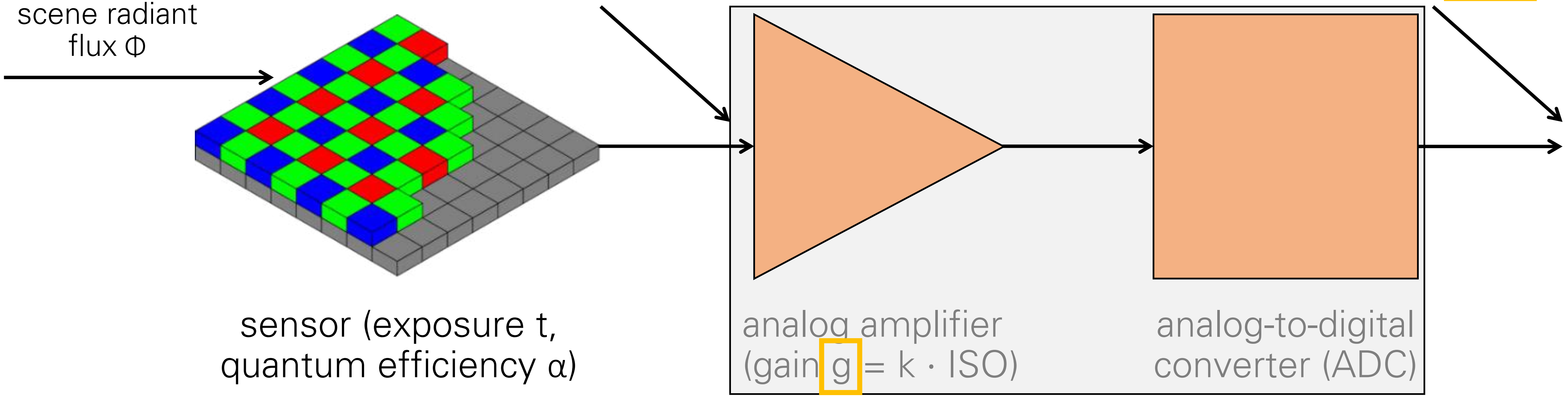
saturation level

intensity mean and variance:

$$E(I) = t \cdot (a \cdot \Phi) \cdot g$$

$$\sigma(I)^2 = t \cdot (a \cdot \Phi) \cdot g^2 + \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$$

Affine noise model after dark frame subtraction



discrete image intensity (with saturation):

$$I = \min(L \cdot g + n_{\text{add}}, I_{\max})$$

intensity mean and variance:

$$E(I) = t \cdot (a \cdot \Phi) \cdot g$$

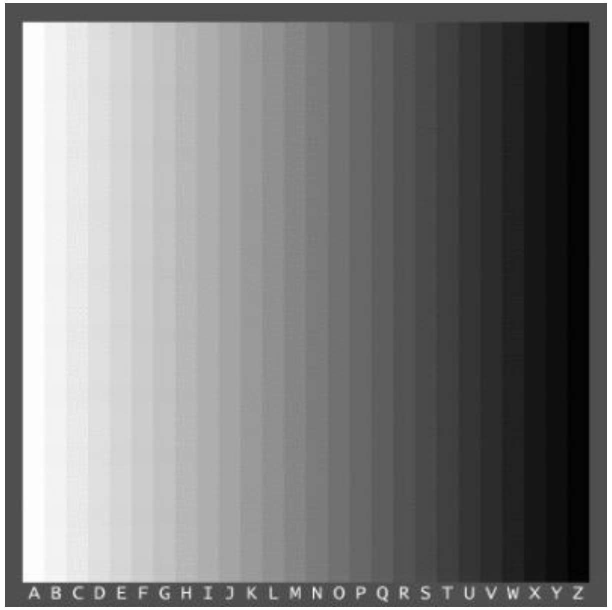
$$\sigma(I)^2 = t \cdot (a \cdot \Phi) \cdot g^2 + \sigma_{\text{add}}^2$$

Estimating the gain and additive noise variance

Can you think of a procedure for estimating these quantities?

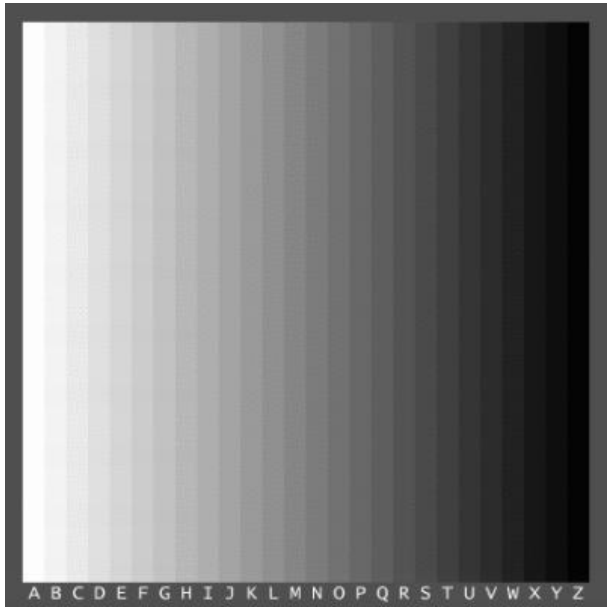
Estimating the gain and additive noise variance

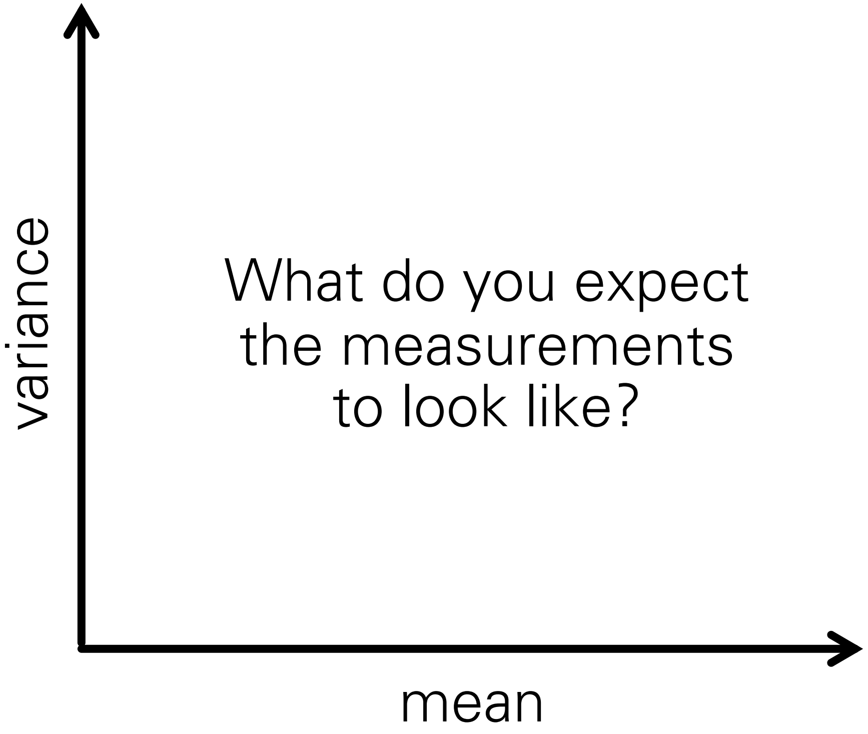
1. Capture a large number of images of a grayscale target.



Estimating the gain and additive noise variance

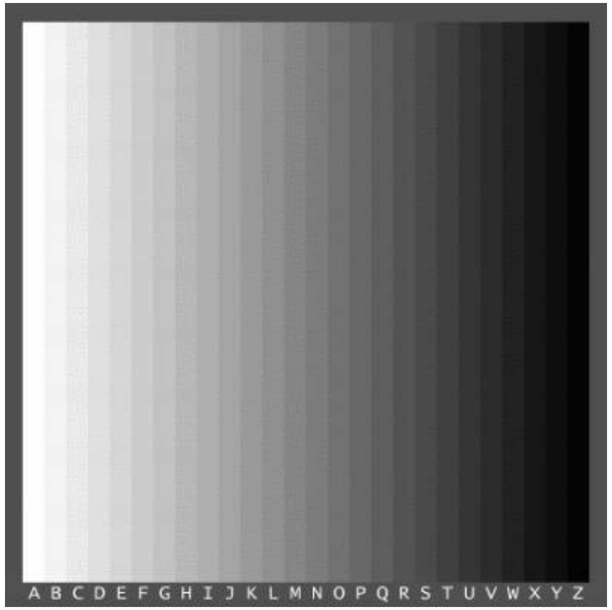
1. Capture a large number of images of a grayscale target.
2. Compute the empirical mean and variance for each pixel, then form a mean-variance plot.

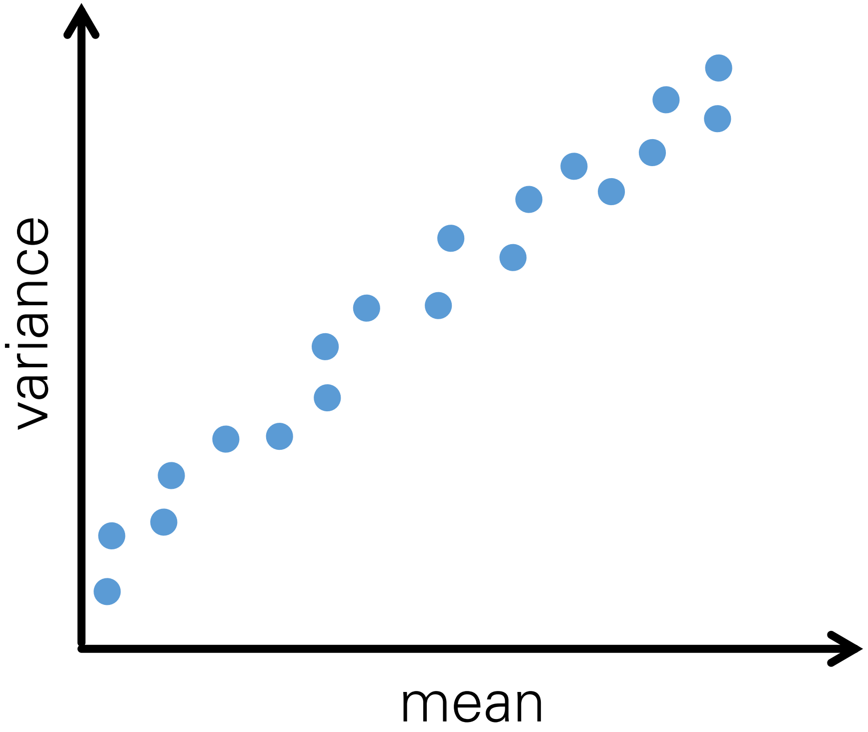




Estimating the gain and additive noise variance

1. Capture a large number of images of a grayscale target.
2. Compute the empirical mean and variance for each pixel, then form a mean-variance plot.





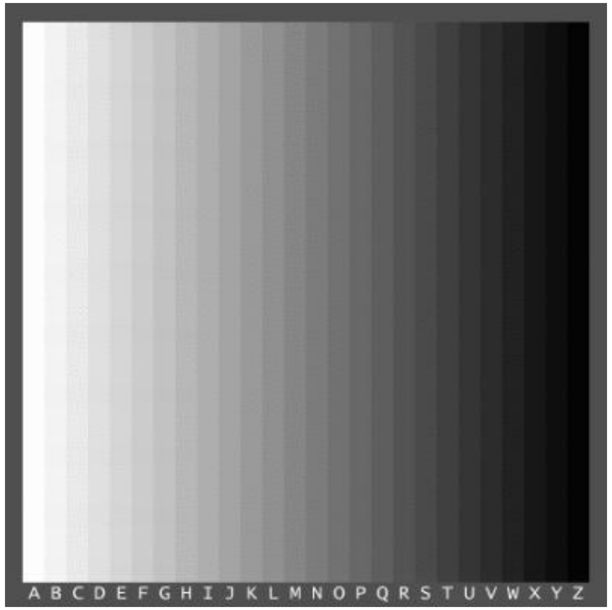
$$E(I) = t \cdot (a \cdot \Phi) \cdot g$$

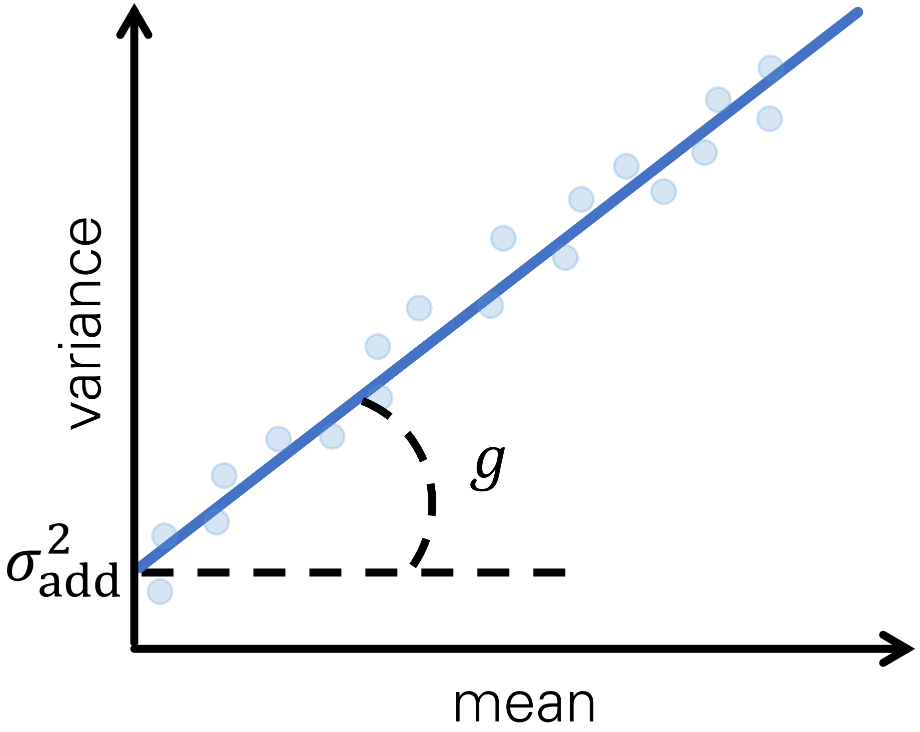
$$\sigma(I)^2 = t \cdot (a \cdot \Phi) \cdot g^2 + \sigma_{\text{add}}^2$$

$$\Rightarrow \sigma(I)^2 = E(I) \cdot g + \sigma_{\text{add}}^2$$

Estimating the gain and additive noise variance

1. Capture a large number of images of a grayscale target.
2. Compute the empirical mean and variance for each pixel, then form a mean-variance plot.
3. Fit a line and use slope and intercept to estimate the gain and variance.





equal to line slope

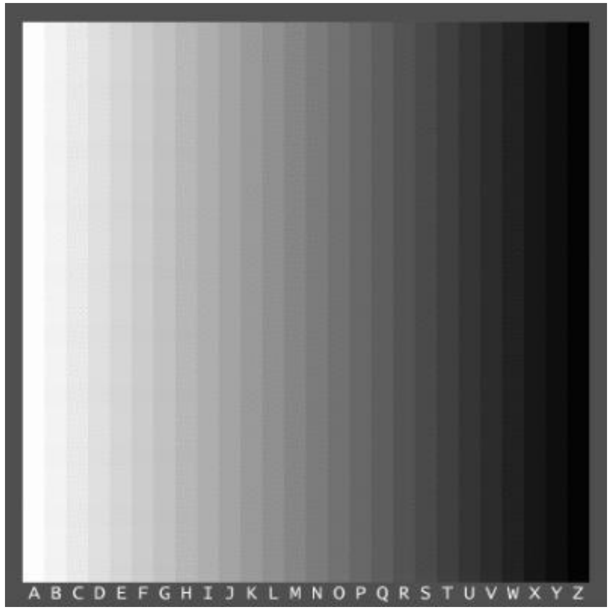
$$\sigma(I)^2 = E(I) \cdot g + \sigma_{\text{add}}^2$$

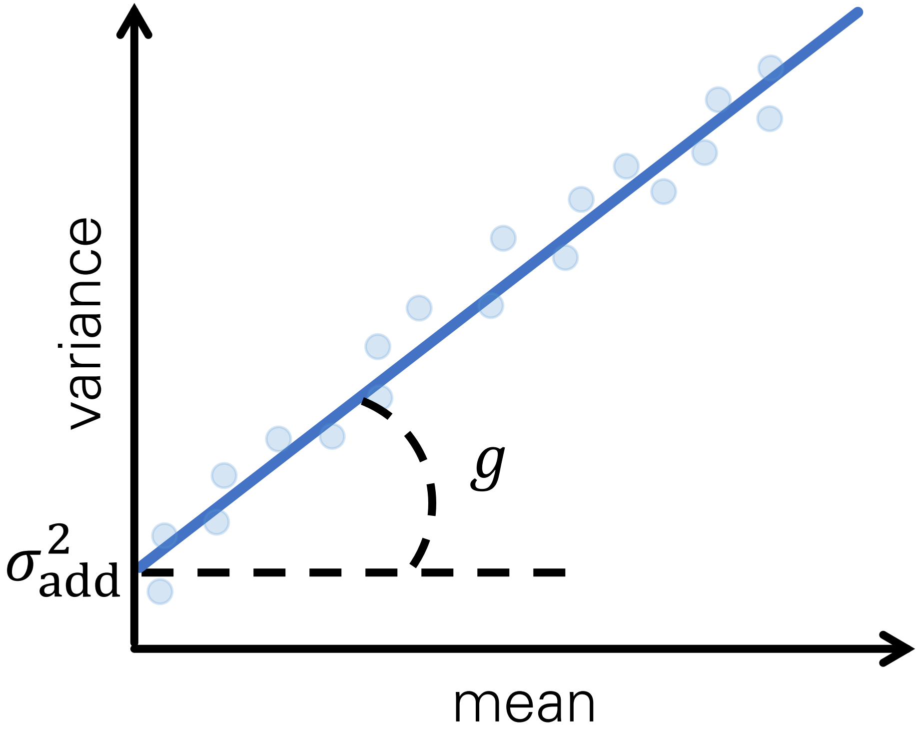
equal to line intercept

How would you modify this procedure to separately estimate read and ADC noise?

Estimating the gain and additive noise variance

1. Capture a large number of images of a grayscale target.
2. Compute the empirical mean and variance for each pixel, then form a mean-variance plot.
3. Fit a line and use slope and intercept to estimate the gain and variance.





equal to line slope

$$\sigma(I)^2 = E(I) \cdot g + \sigma^2_{\text{add}}$$

equal to line intercept

How would you modify this procedure to separately estimate read and ADC noise?

- Perform it for a few different ISO settings (i.e., gains g).

Important notes

Noise calibration should be performed with RAW images!

The above procedure assumes that all pixels have the same noise characteristics.

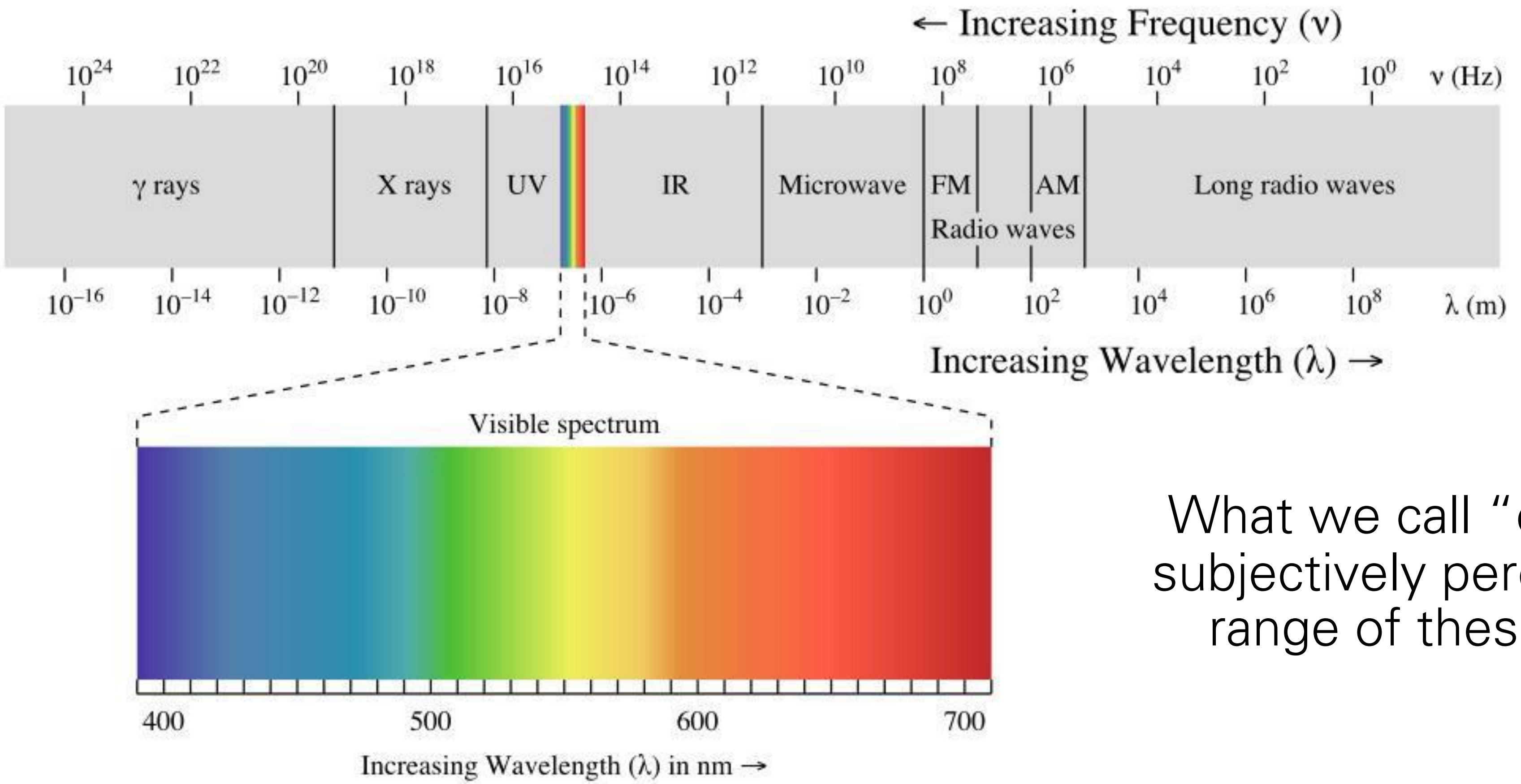
- If that is not the case, then you need to capture multiple images under multiple exposure times, and use those to form the mean-variance plot for each pixel.

Today's Lecture

- Digital photography
- Standard camera pipeline
- Noise
- Color

Color is an artifact of human perception

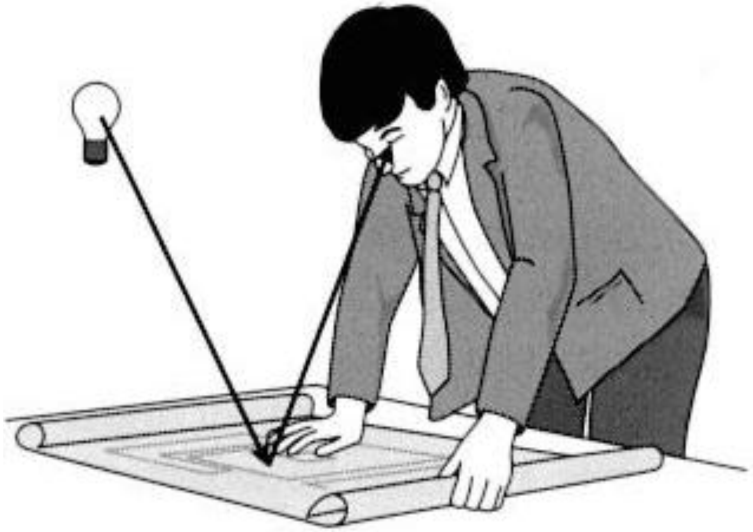
- "Color" is not an objective physical property of light (electromagnetic radiation).
- Instead, light is characterized by its wavelength.



electromagnetic spectrum

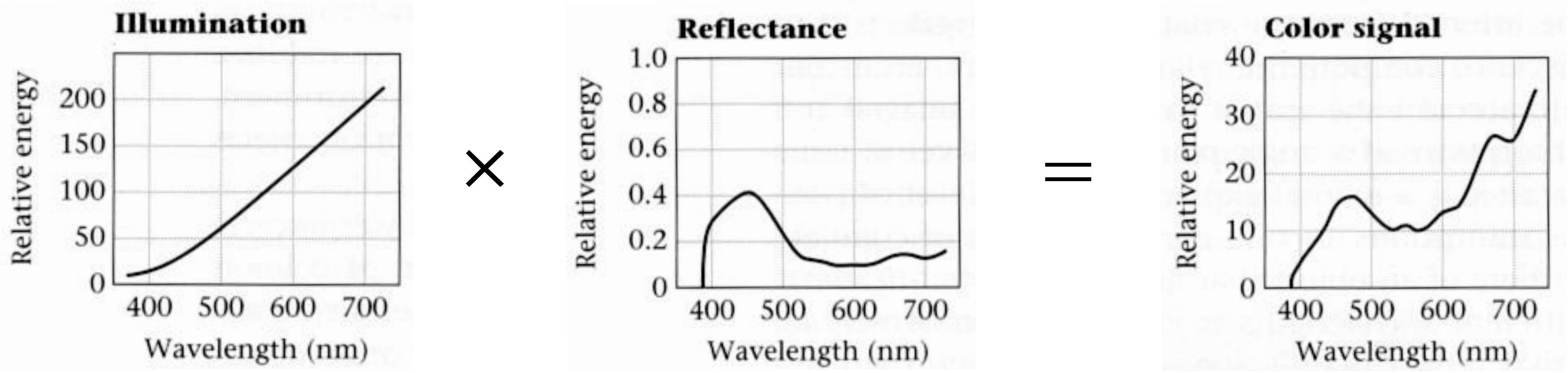
What we call "color" is how we subjectively perceive a very small range of these wavelengths.

Light-matter interaction

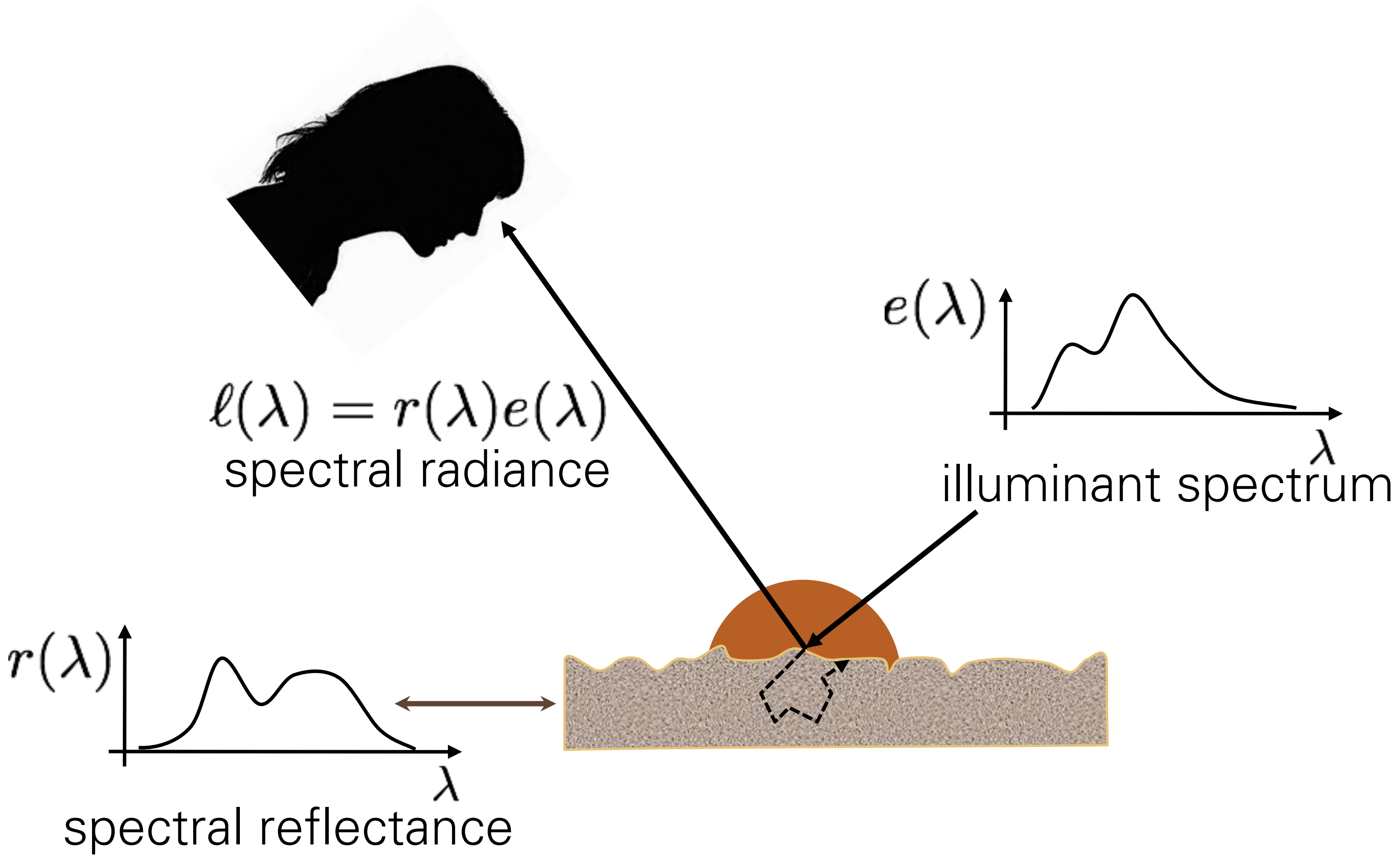


- Where spectra come from:
  - light source spectrum
  - object reflectance (aka spectral albedo)
  - multiplied wavelength by wavelength

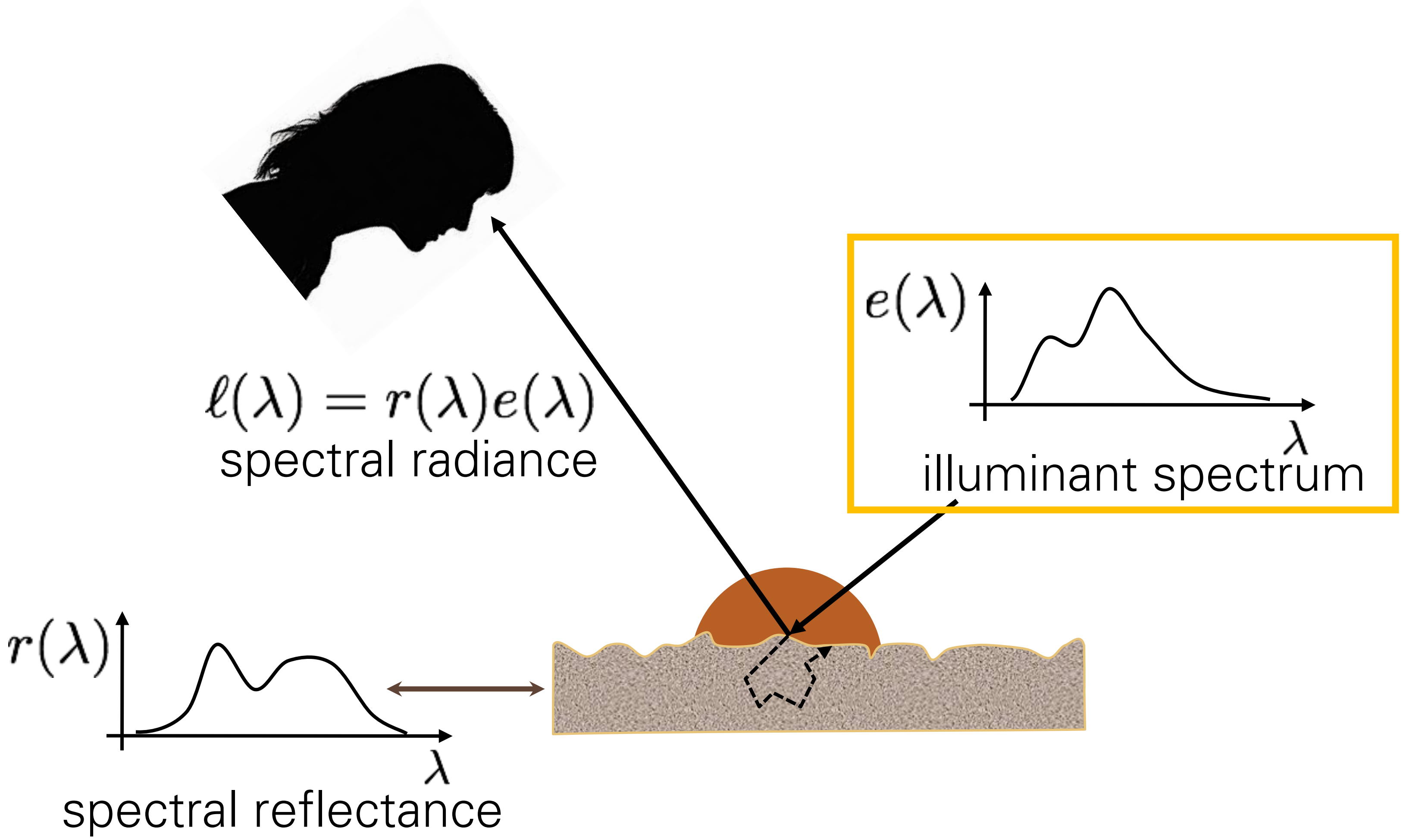
- There are different physical processes that explain this multiplication e.g. absorption, interferences



Light-material interaction



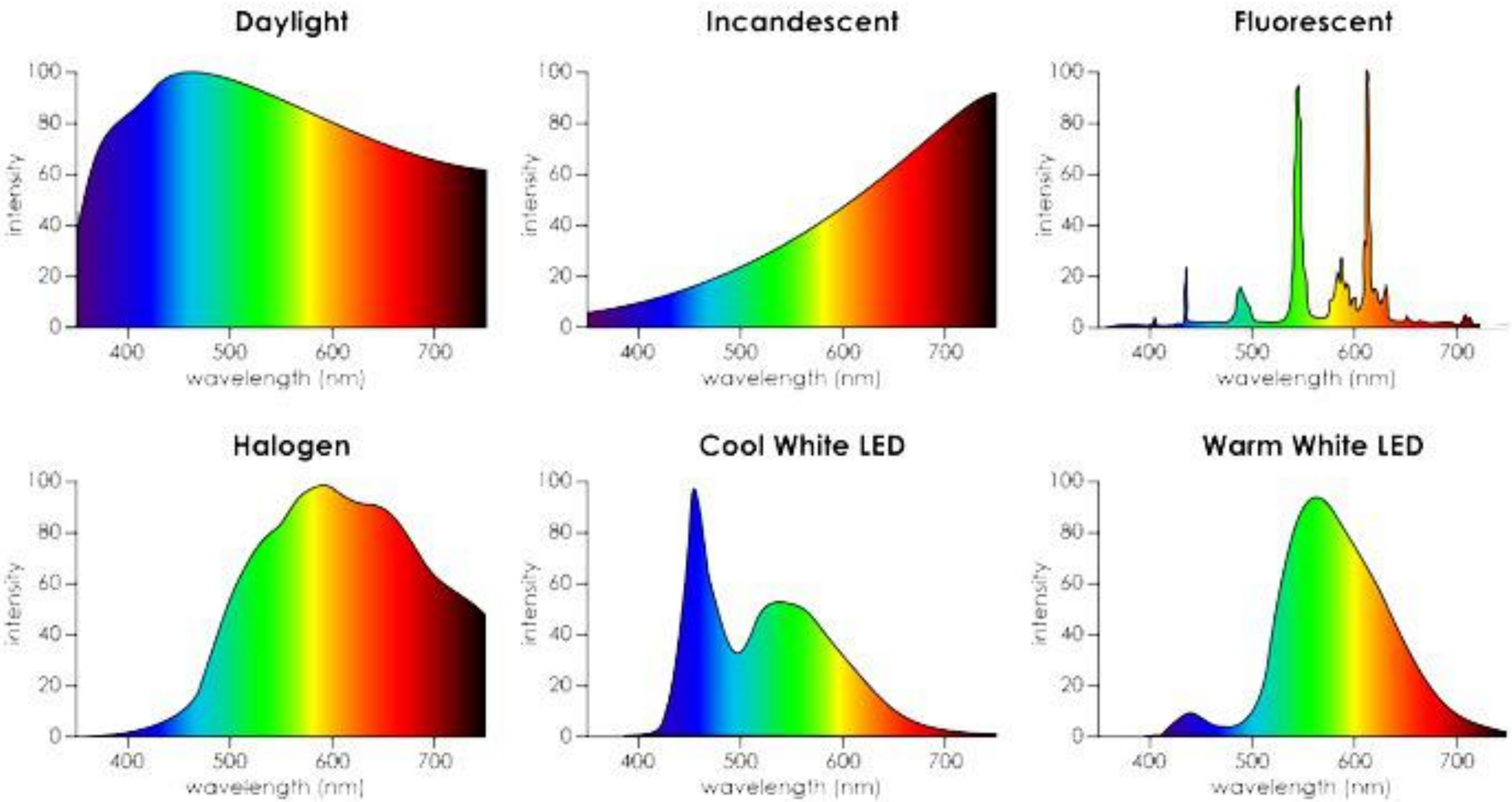
Light-material interaction



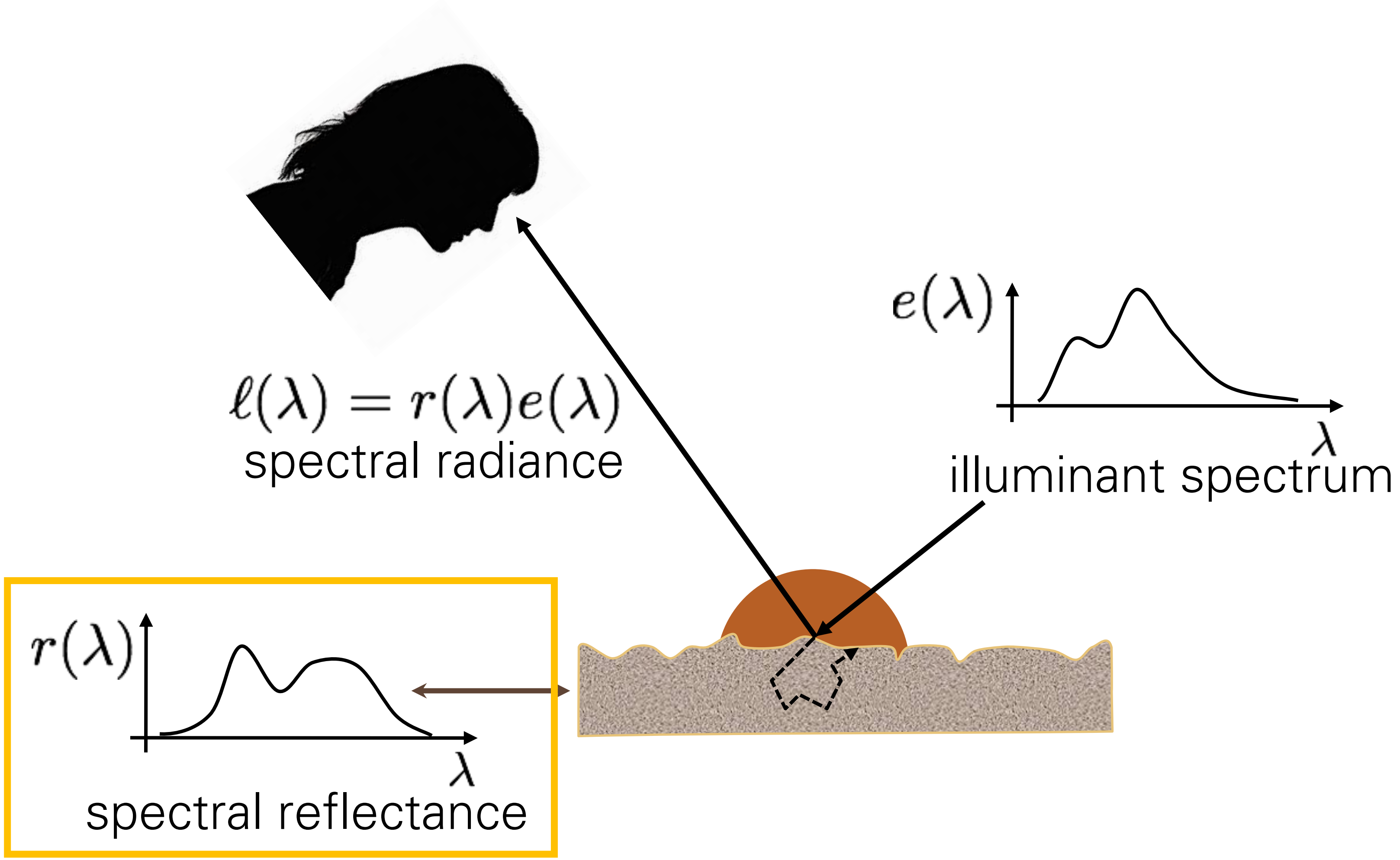
Illuminant Spectral Power Distribution (SPD)

- Most types of light “contain” more than one wavelengths.
- We can describe light based on the distribution of power over different wavelengths.

We call our sensation of all of these distributions “white”.

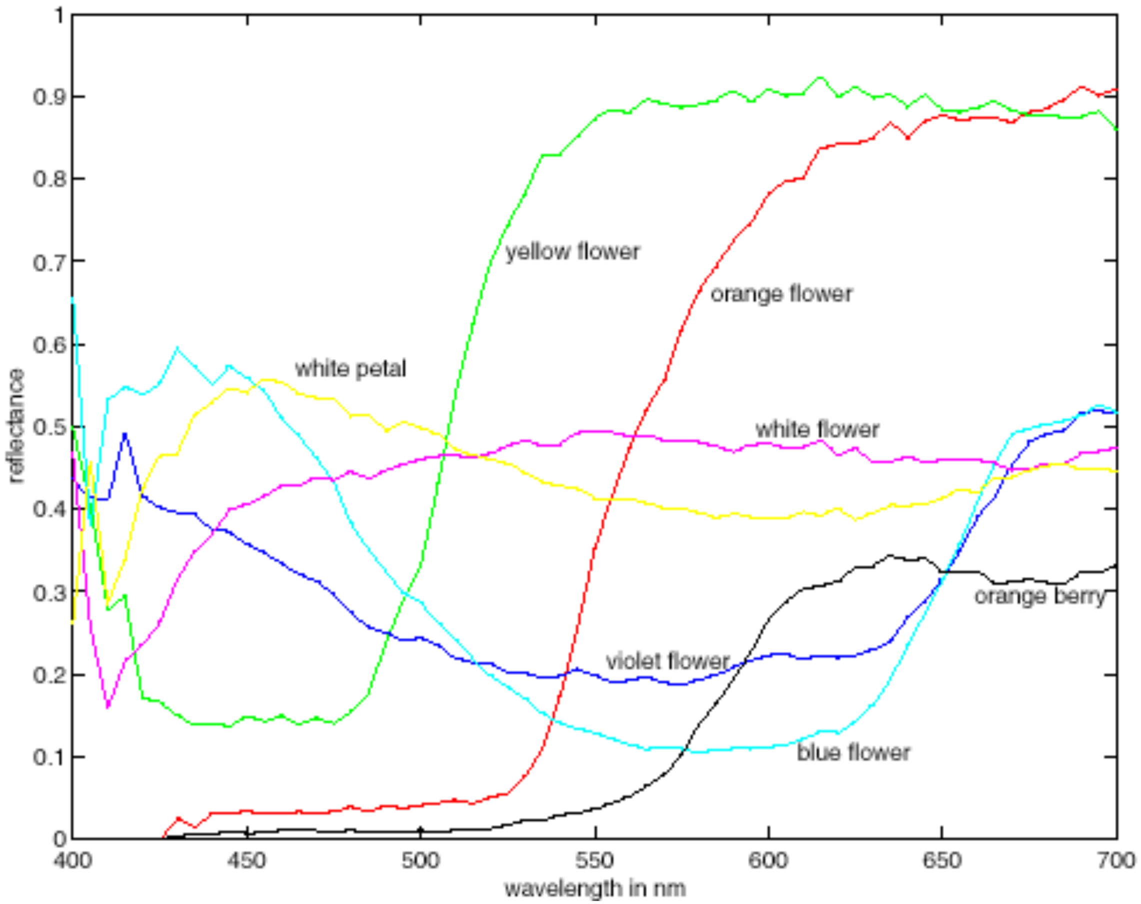


Light-material interaction

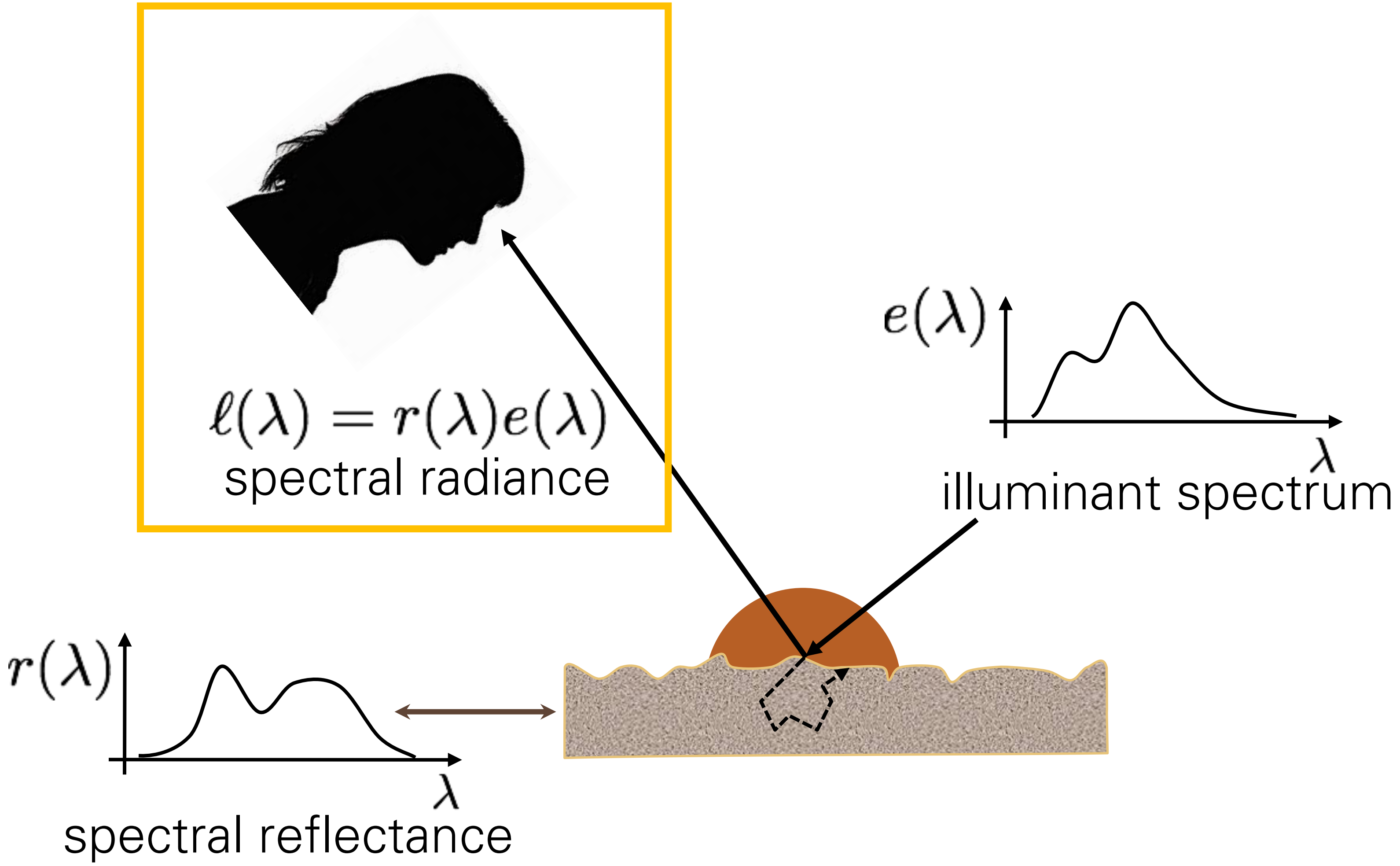


Spectral reflectance

- Most materials absorb and reflect light differently at different wavelengths.
- We can describe this as a ratio of reflected vs incident light over different wavelengths.

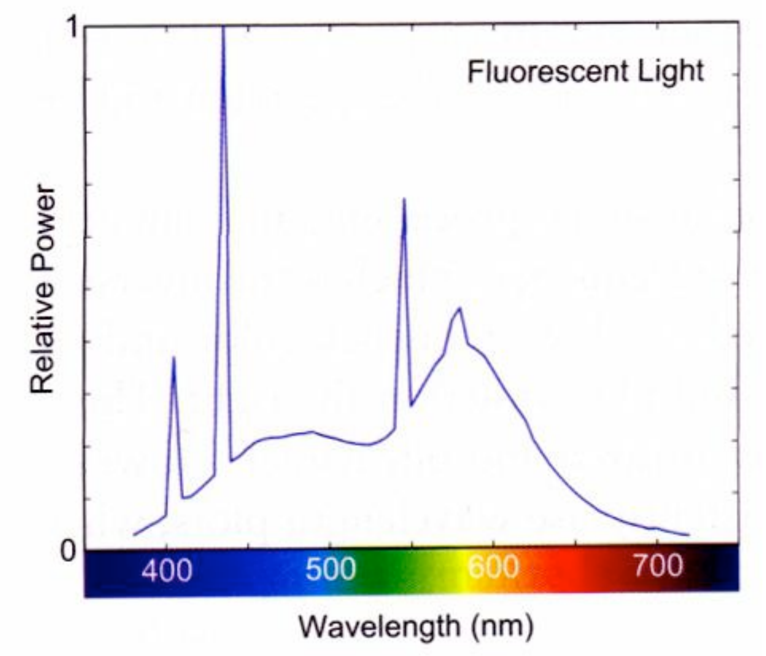


Light-material interaction

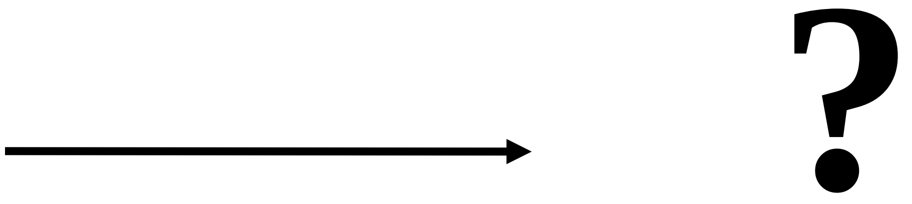


The problem of color science

- Build a model for human color perception
- That is, map a physical light description to a perceptual color sensation



Physical



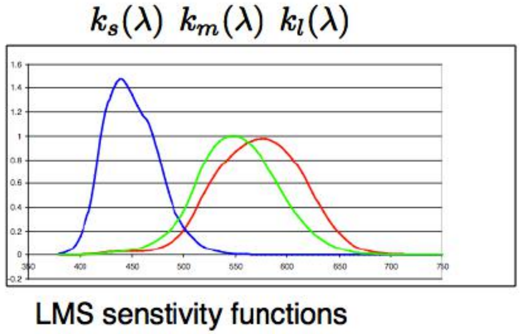
Perceptual

Human color vision

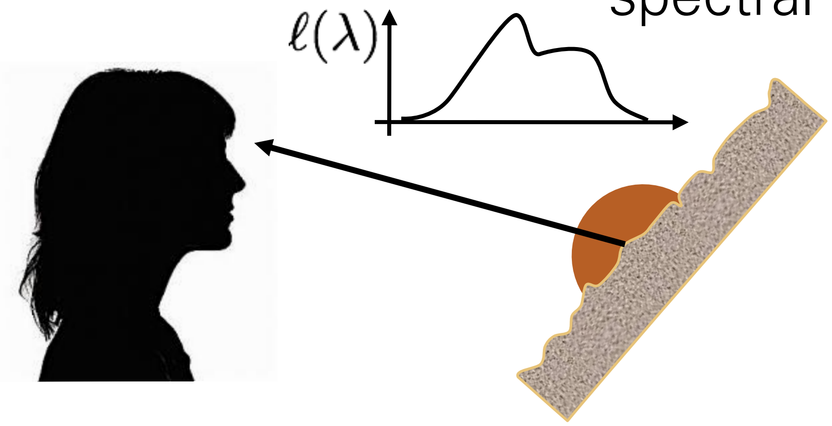
retinal color

$$\mathbf{c}(\ell(\lambda)) = (c_s, c_m, c_l)$$

$$c_s = \int k_s(\lambda)\ell(\lambda)d\lambda$$



spectral radiance



perceived color

object color

color names

Retinal vs perceived color

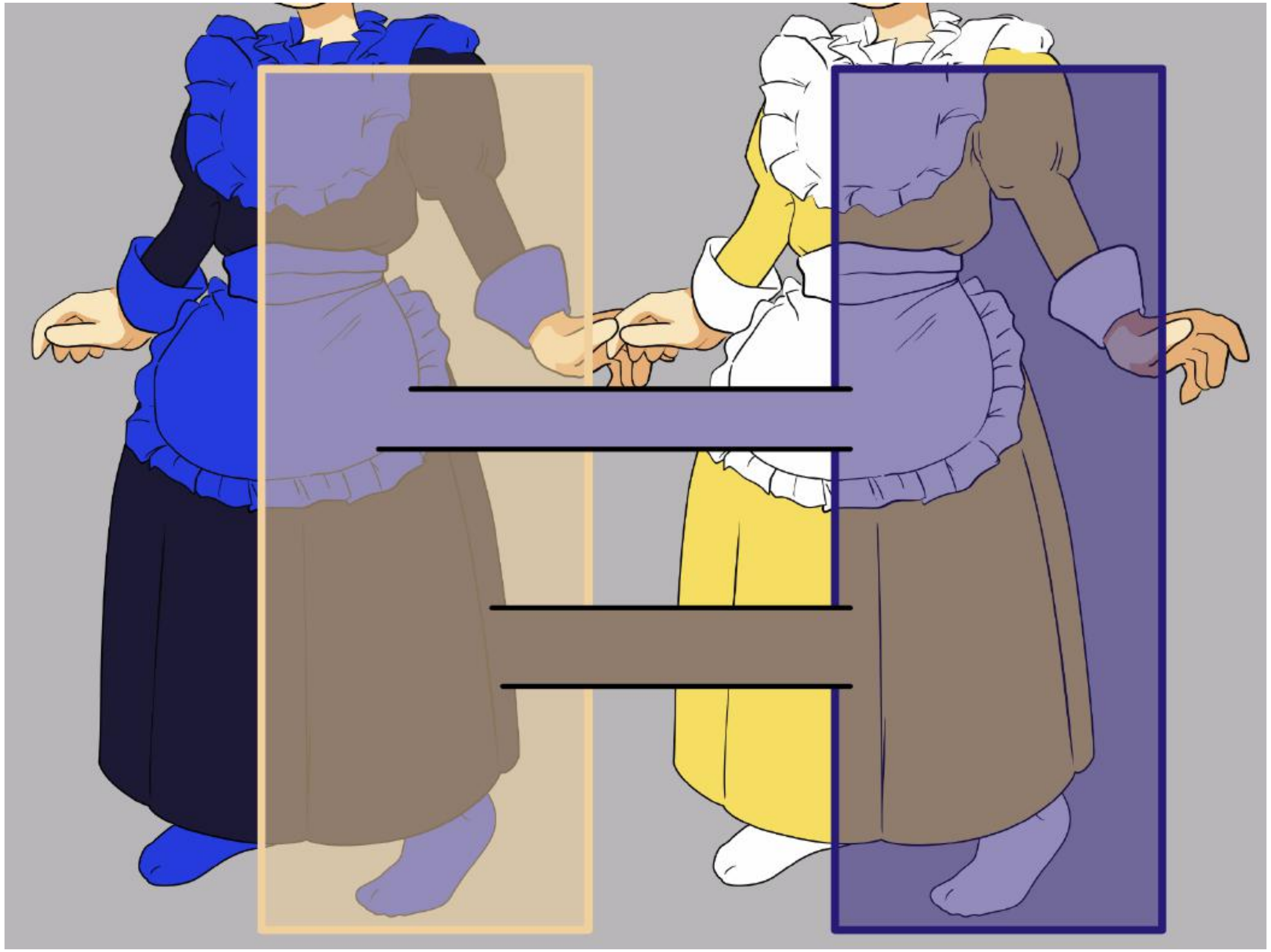
Retinal vs perceived color

Retinal vs perceived color.

Retinal vs perceived color

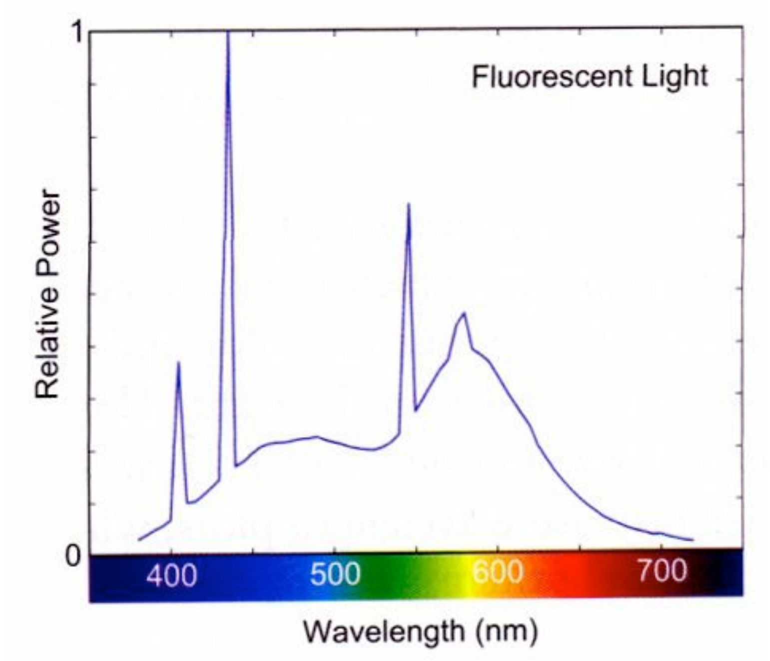
- Our visual system tries to "adapt" to illuminant.
- We may interpret the same retinal color very differently.

Retinal vs perceived color

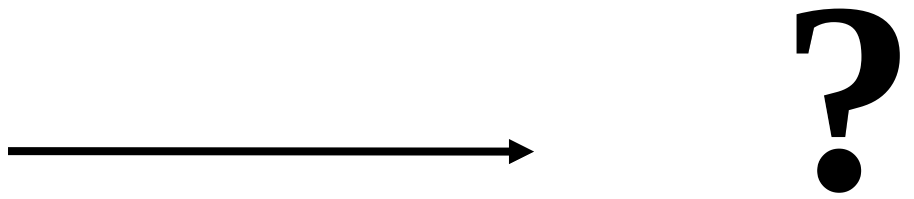


The problem of color science

- Build a model for human color perception
- That is, map a physical light description to a perceptual color sensation



Physical



Perceptual

Human color vision

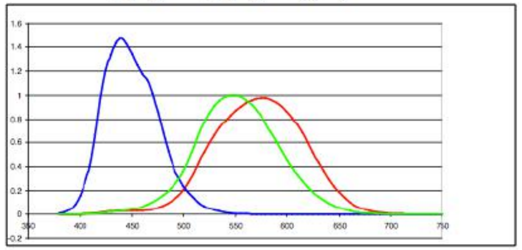
We will exclusively discuss retinal color in this course

retinal color

$$\mathbf{c}(\ell(\lambda)) = (c_s, c_m, c_l)$$

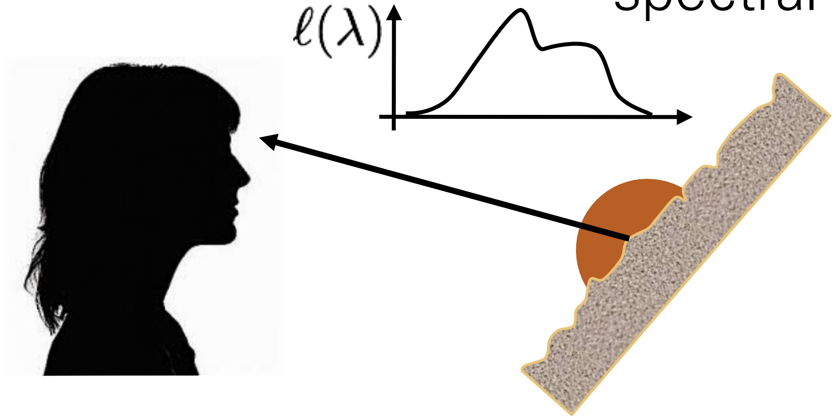
$$c_s = \int k_s(\lambda)\ell(\lambda)d\lambda$$

$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions

spectral radiance



perceived color

object color

color names

Retinal color space

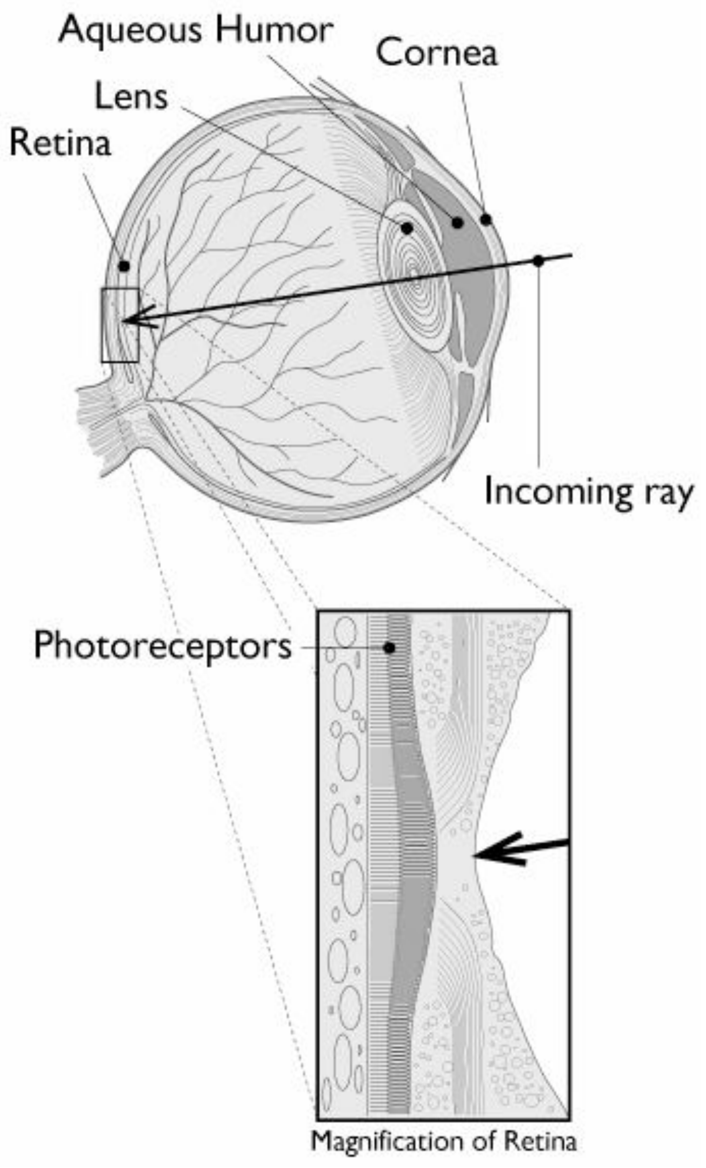
Spectral Sensitivity Function (SSF)

- Any light sensor (digital or not) has different sensitivity to different wavelengths.
- This is described by the sensor's spectral sensitivity function $f(\lambda)$.
- When measuring light of some SPD $\Phi(\lambda)$, the sensor produces a scalar response:

$$\underset{\text{sensor response}}{R} = \int_\lambda \underbrace{\Phi(\lambda)}_{\text{light SPD}} \underbrace{f(\lambda)}_{\text{sensor SSF}} d\lambda$$

Weighted combination of light's SPD: light contributes more at wavelengths where the sensor has higher sensitivity.

The eye as a measurement device



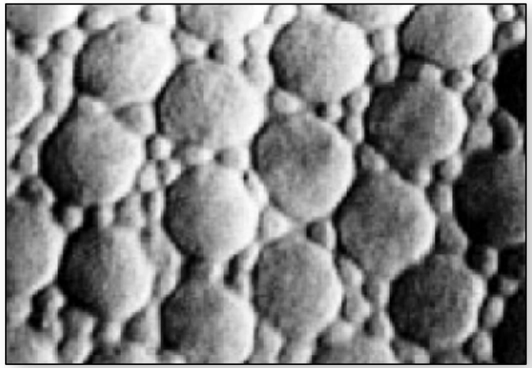
Greger et al. 1995

- We can model the low-level behavior of the eye by thinking of it as a light-measuring machine
  - its optics are much like a camera
  - its detection mechanism is also much like a camera
- Light is measured by the photoreceptors in the retina
  - they respond to visible light
  - different types respond to different wavelengths

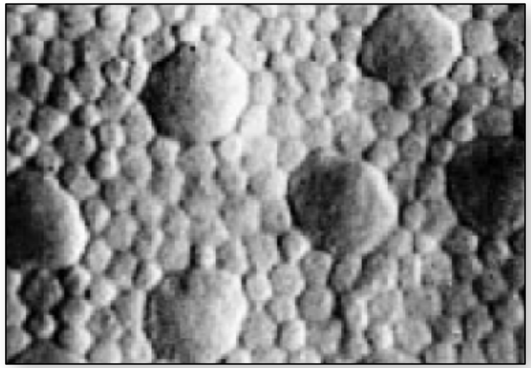
Retinal composition: two kinds of cells

- **Cones** are concentrated in fovea
  - high acuity, require more light
  - “respond to color”

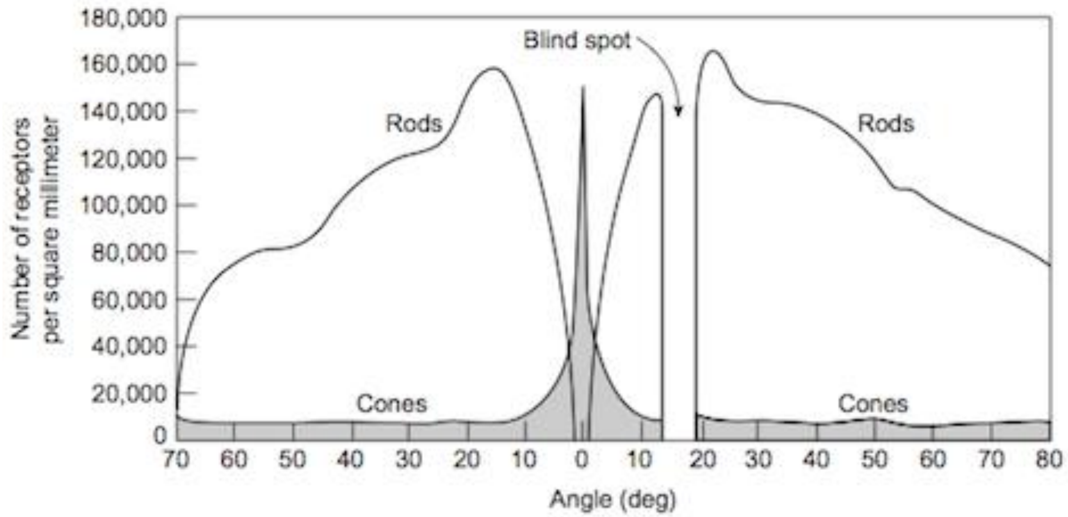
- **Rods** concentrated outside fovea
  - lower acuity, require less light
  - roughly 10x more sensitive
  - “respond to intensity only”



near fovea



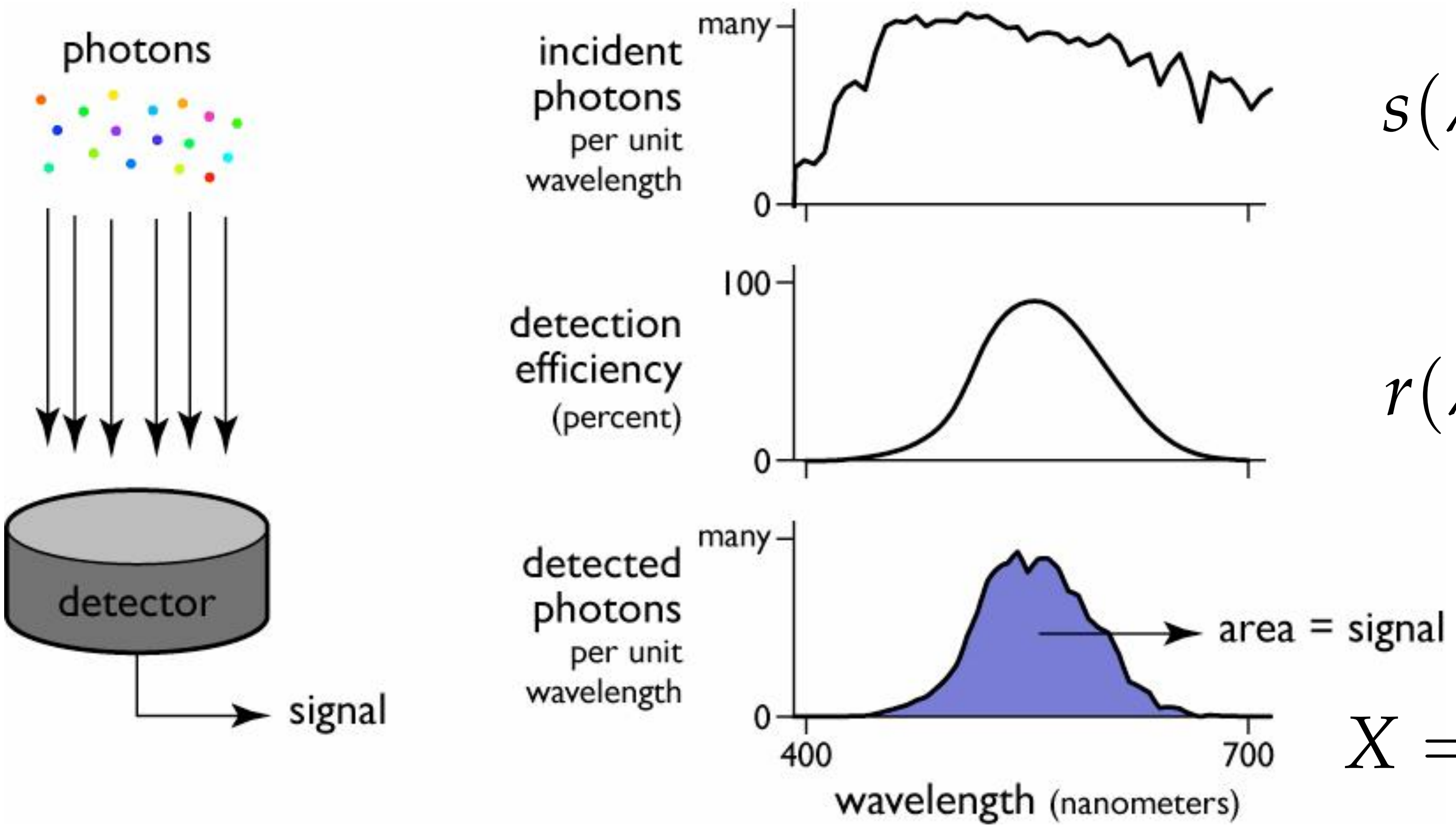
away from fovea



A simple light detector

- Produces a scalar value (a number) when photons land on it
  - this value depends strictly on the number of photons detected
  - each photon has a probability of being detected that depends on the wavelength
  - there is no way to tell the difference between signals caused by light of different wavelengths: there is just a number
- This is a reasonable model for many detectors:
  - based on semiconductors (such as in a digital camera)
  - based on visual photopigments (such as in human eyes)

A simple light detector



$$s(\lambda)$$

$$r(\lambda)$$

$$X = \int s(\lambda)\, r(\lambda)\, \mathrm{d}\lambda$$

A simple light detector

- Light entering the detector has a spectral power distribution (SPD), s(λ)
- Detector has a spectral sensitivity/spectral response, r(λ)

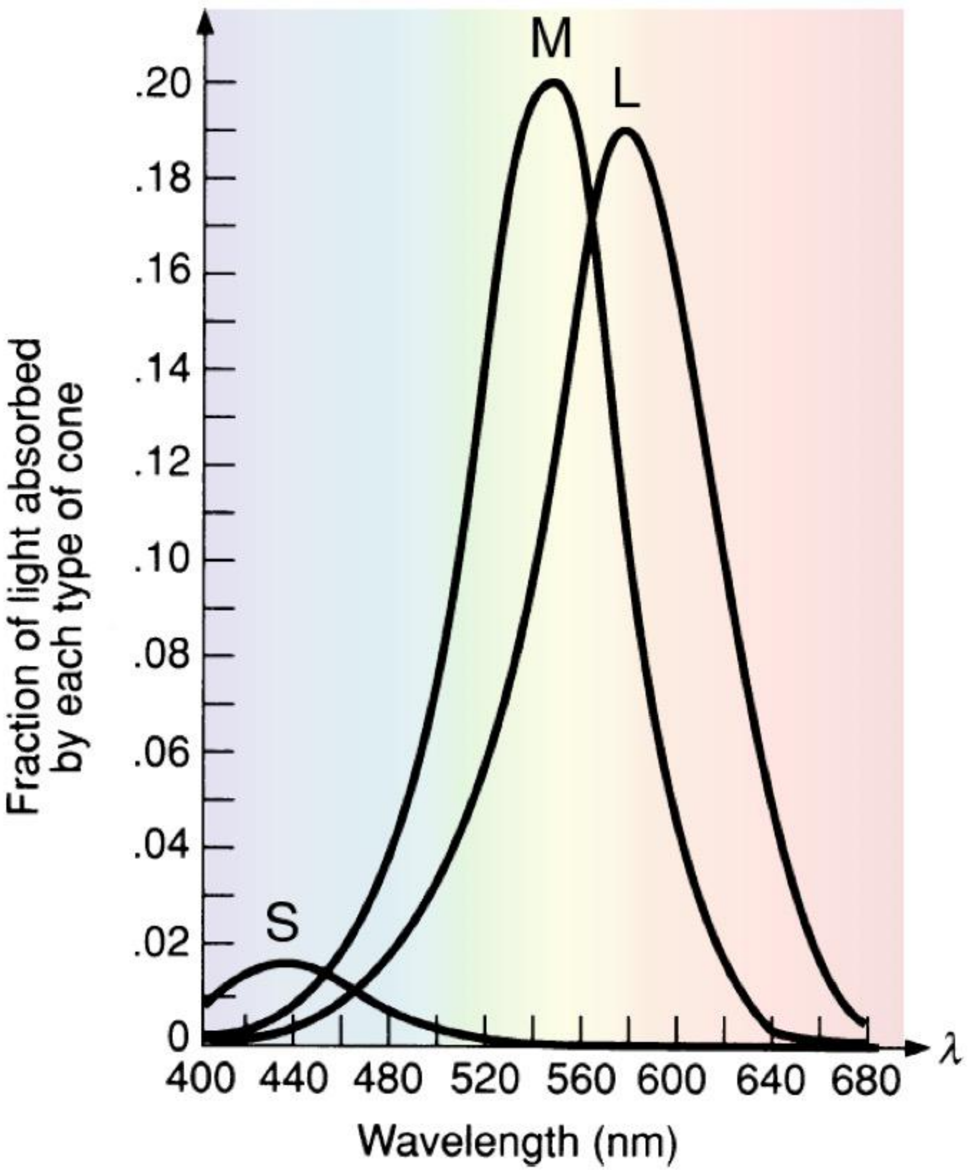
$$X = \int \underbrace{s(\lambda)}_{\text{input spectrum}}\, \underbrace{r(\lambda)}_{\text{detector's sensitivity}}\, \mathrm{d}\lambda$$

measured signal

input spectrum

detector's sensitivity

Cone responses



- **Three types of cones with broadband spectral sensitivity**
  - S cones respond to short-wavelengths ("blue")
  - M cones respond to medium-wavelengths ("green")
  - L cones respond to long-wavelengths ("red")
  - Experimentally determined in the 1980s
- **S,M,L neural response is integrated w.r.t. λ**
  - we'll call the response functions r_S, r_M, r_L
- **Results in a trichromatic visual system**
- **S, M, and L are tristimulus values**

Cone responses to a spectrum s (Math)

$$S = \int r_S(\lambda)\, s(\lambda)\, \mathrm{d}\lambda$$

$$M = \int r_M(\lambda)\, s(\lambda)\, \mathrm{d}\lambda$$

$$L = \int r_L(\lambda)\, s(\lambda)\, \mathrm{d}\lambda$$

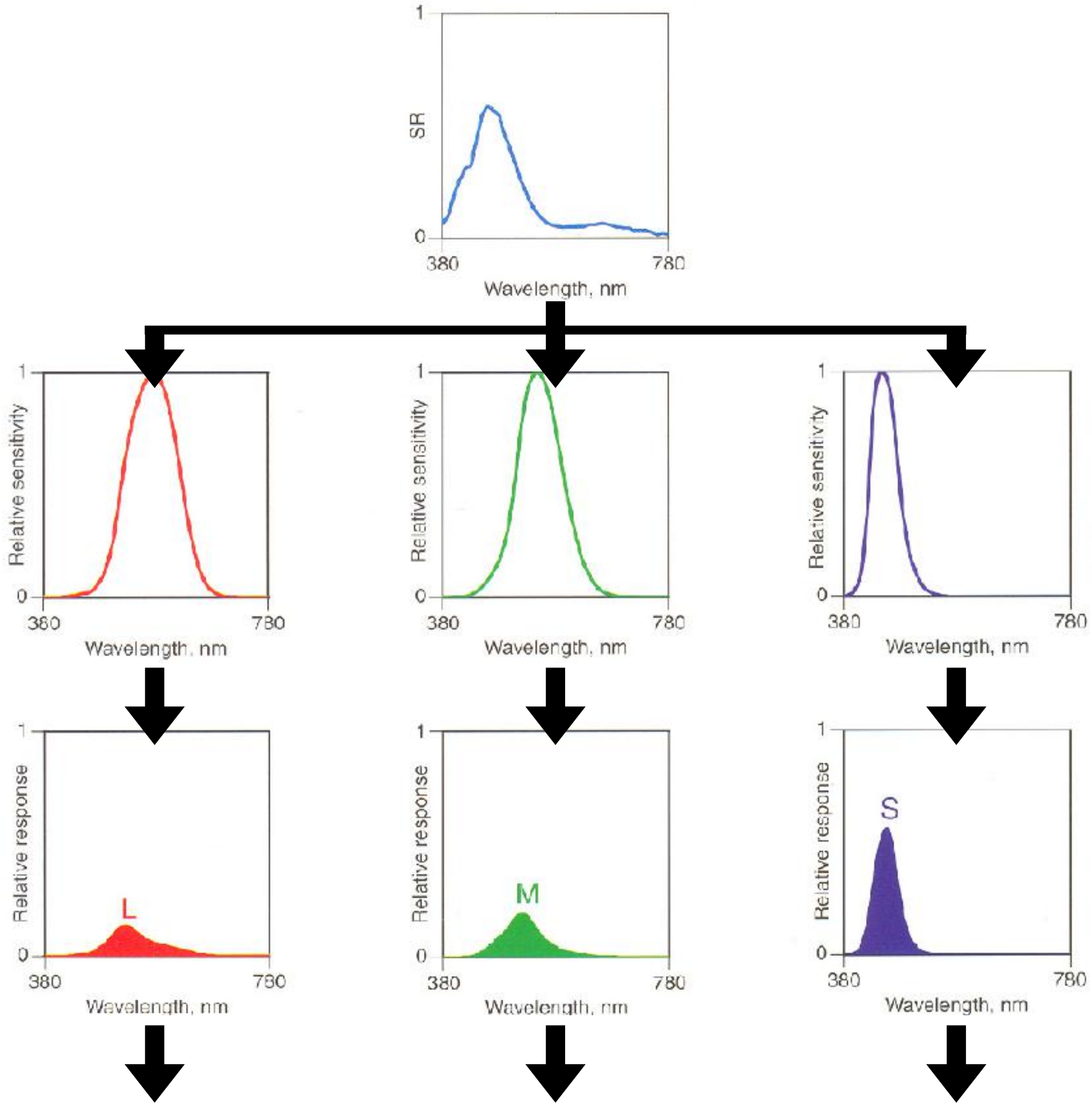
Cone responses to a spectrum *s* (Math)

Stimulus
(arbitrary spectrum)

Response curves

Multiply

Integrate

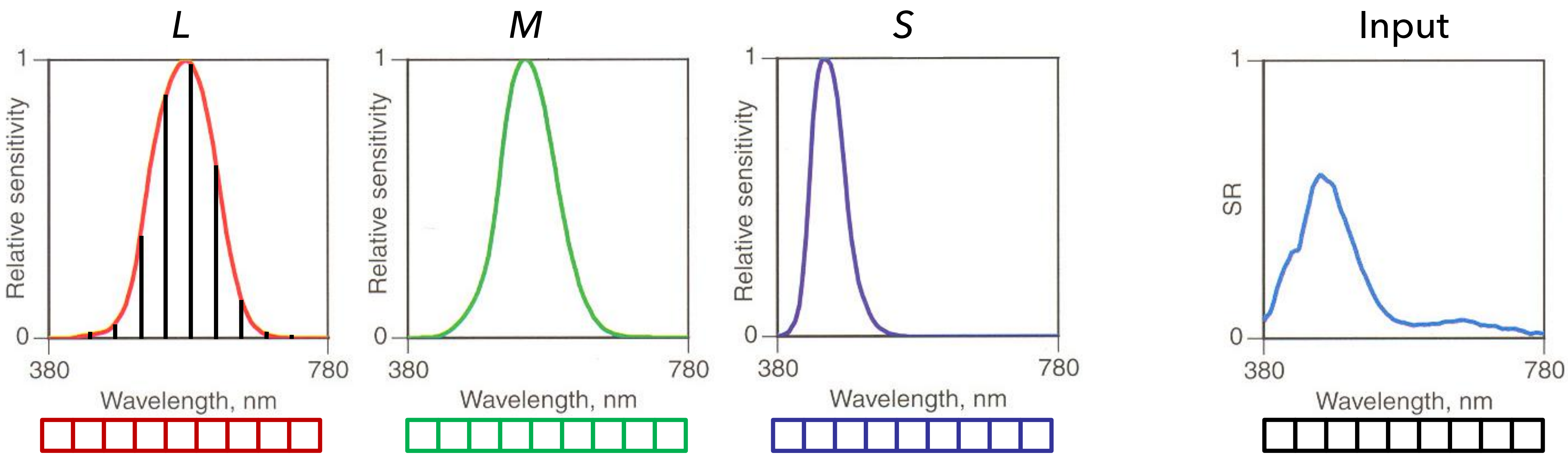


1 number 1 number 1 number

Start with infinite number of values (one per wavelength)

End up with 3 values (one per cone type)

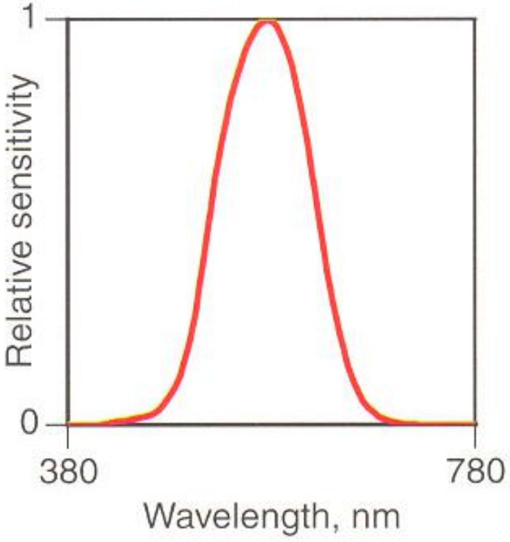
Linear algebra interpretation



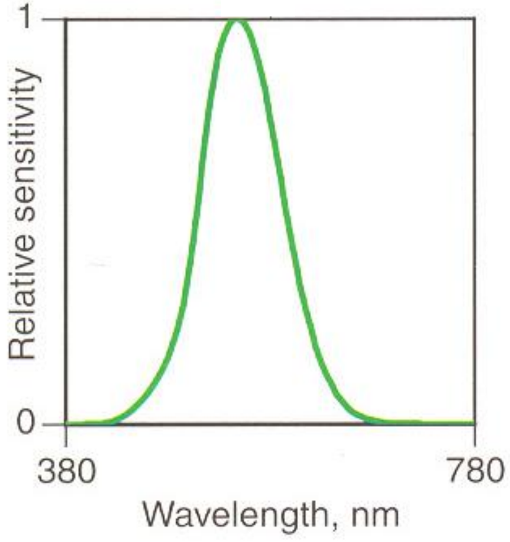
- Sample response curves and input spectra at discrete wavelengths to obtain vectors

Linear algebra interpretation

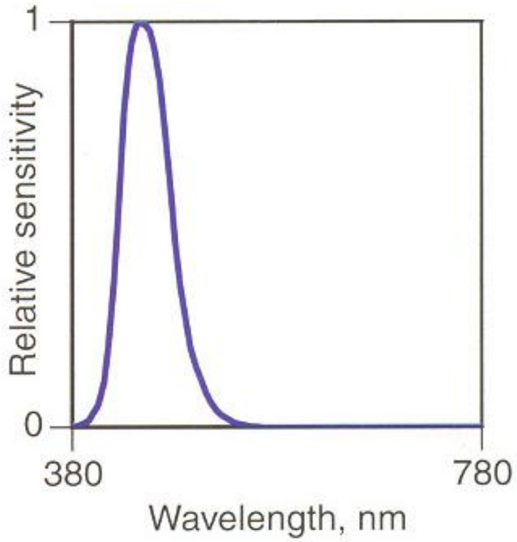
L



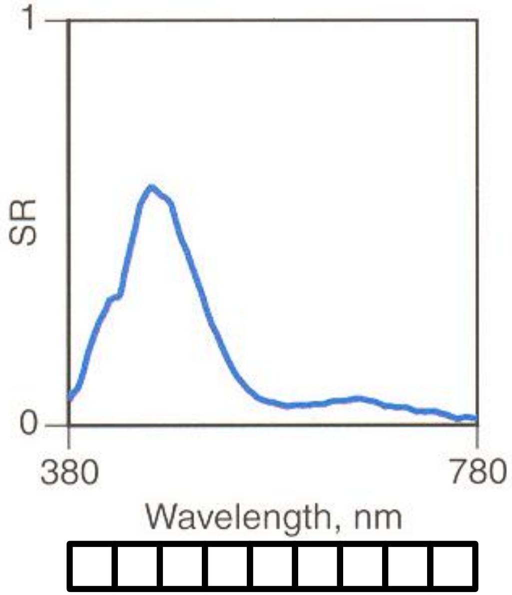
M

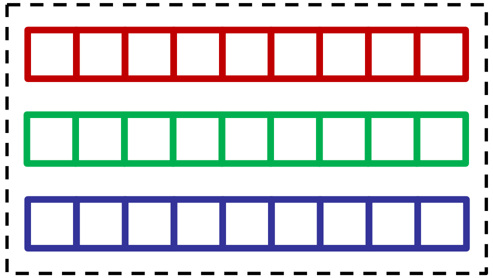


S



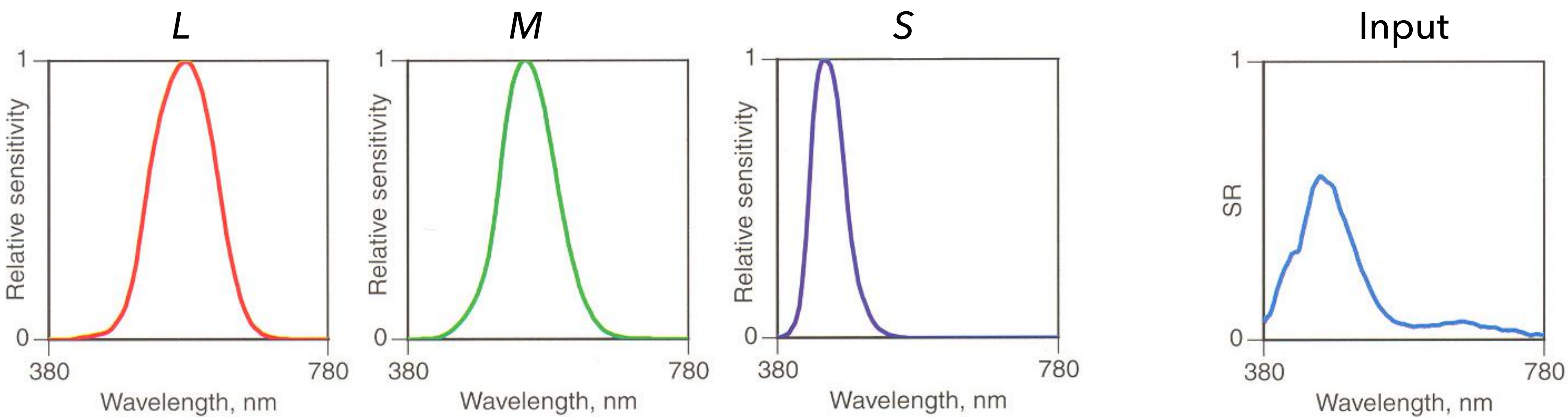
Input

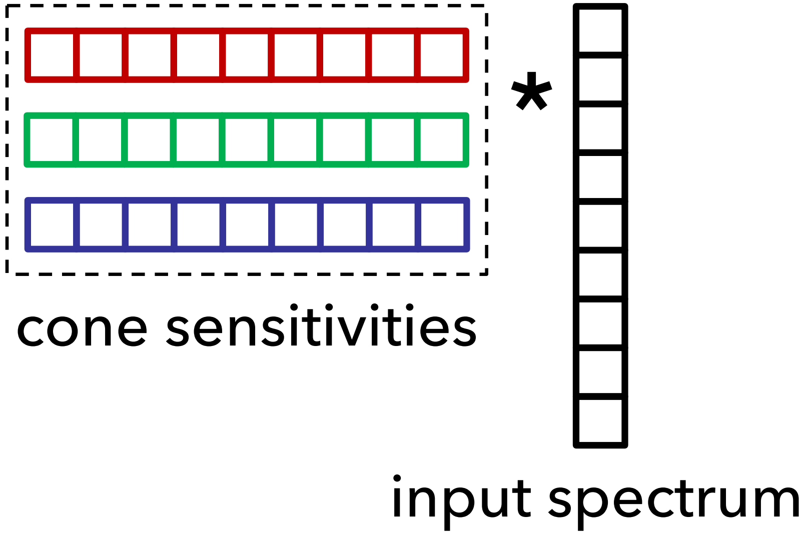




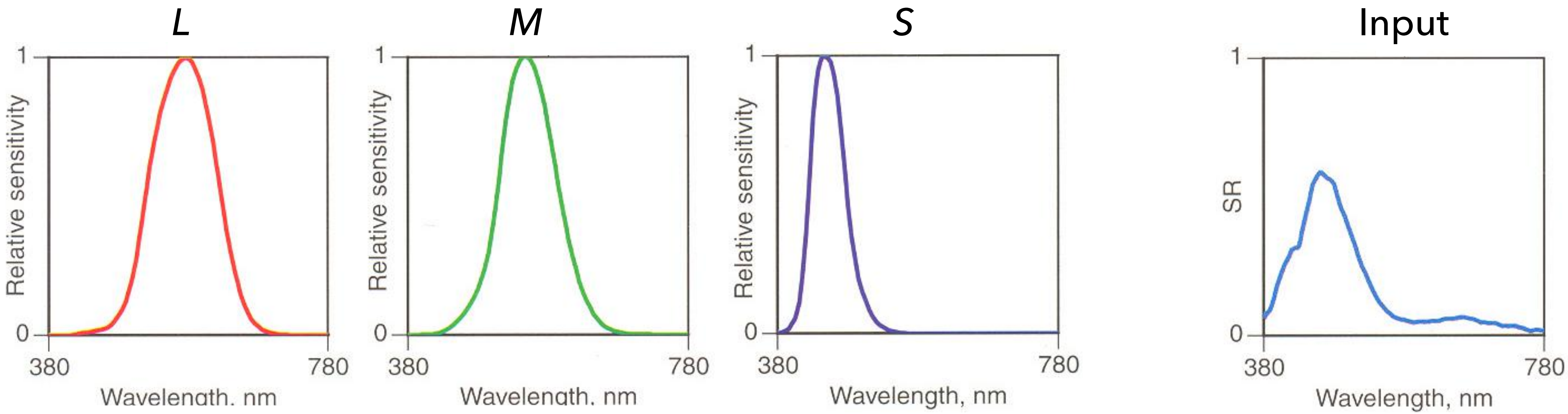
cone sensitivities

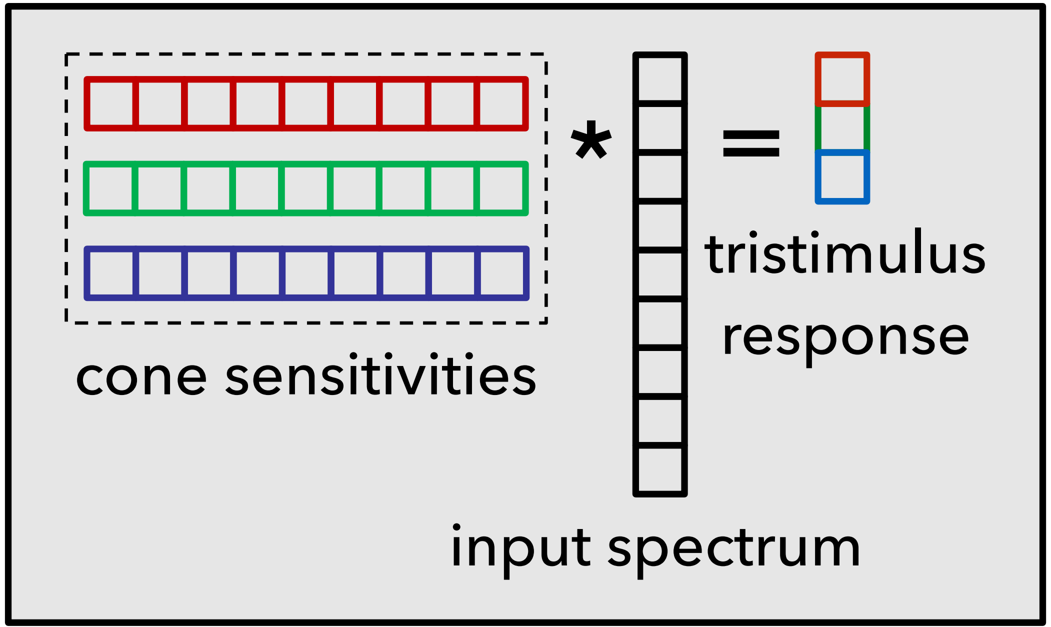
Linear algebra interpretation





Linear algebra interpretation





Tristimulus response is a matrix-vector multiplication

Integration is now summation

Cone responses to a spectrum s

Integral notation:

$$S = \int r_S(\lambda)\, s(\lambda)\, \mathrm{d}\lambda = r_S \cdot s$$

$$M = \int r_M(\lambda)\, s(\lambda)\, \mathrm{d}\lambda = r_M \cdot s$$

$$L = \int r_L(\lambda)\, s(\lambda)\, \mathrm{d}\lambda = r_L \cdot s$$

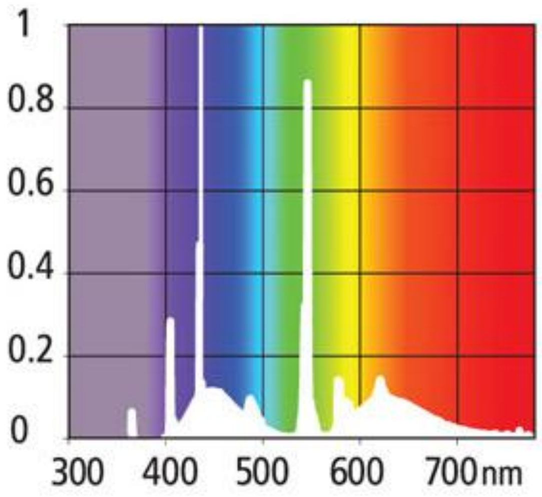
Matrix notation:

$$\begin{bmatrix} S \\ M \\ L \end{bmatrix} = \begin{bmatrix} — & r_S & — \\ — & r_M & — \\ — & r_L & — \end{bmatrix} \begin{bmatrix} | \\ s \\ | \end{bmatrix}$$

r_S, r_M and r_L are N-dimensional vectors, where $N = \infty$

Colorimetry: an answer to the problem

- Wanted to map a physical light description to a perceptual color sensation
- Basic solution was known and standardized by 1930
  - Though not quite in this form — more on that later



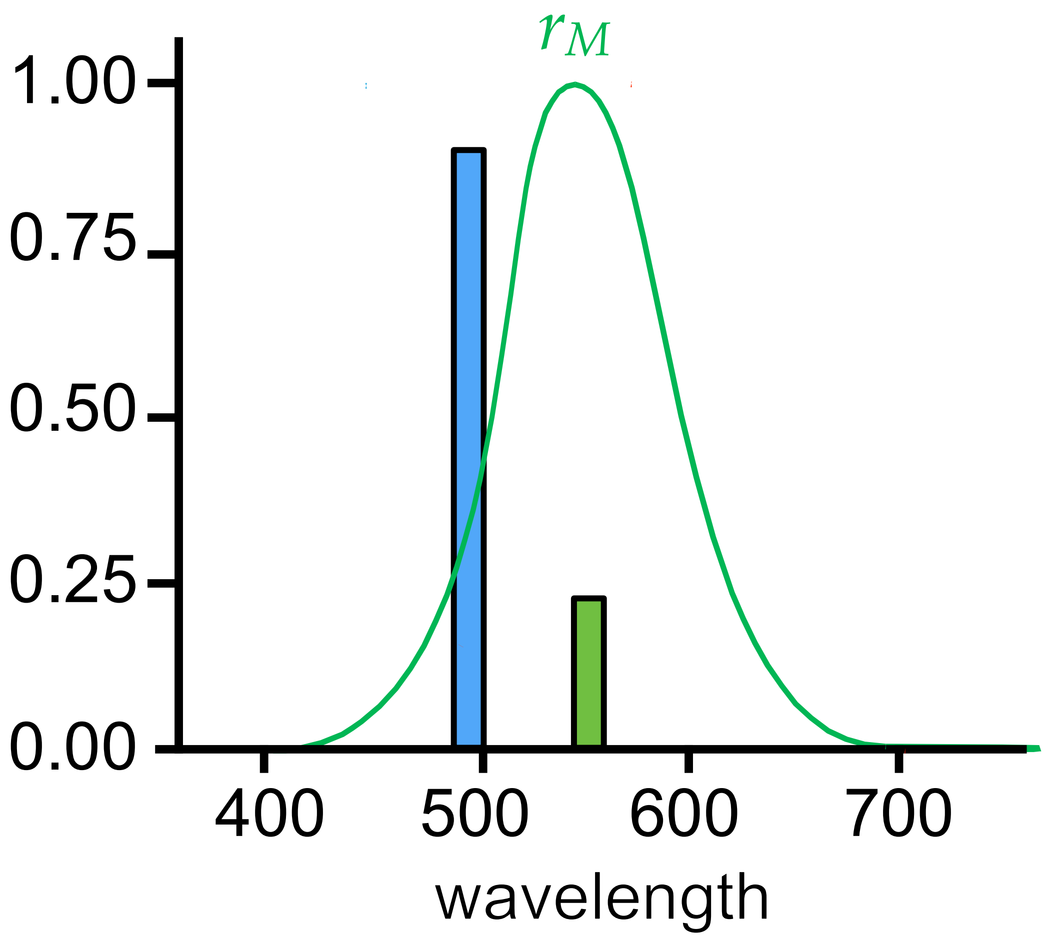
Physical

$$S = r_S \cdot s$$
$$M = r_M \cdot s$$
$$L = r_L \cdot s$$

Perceptual

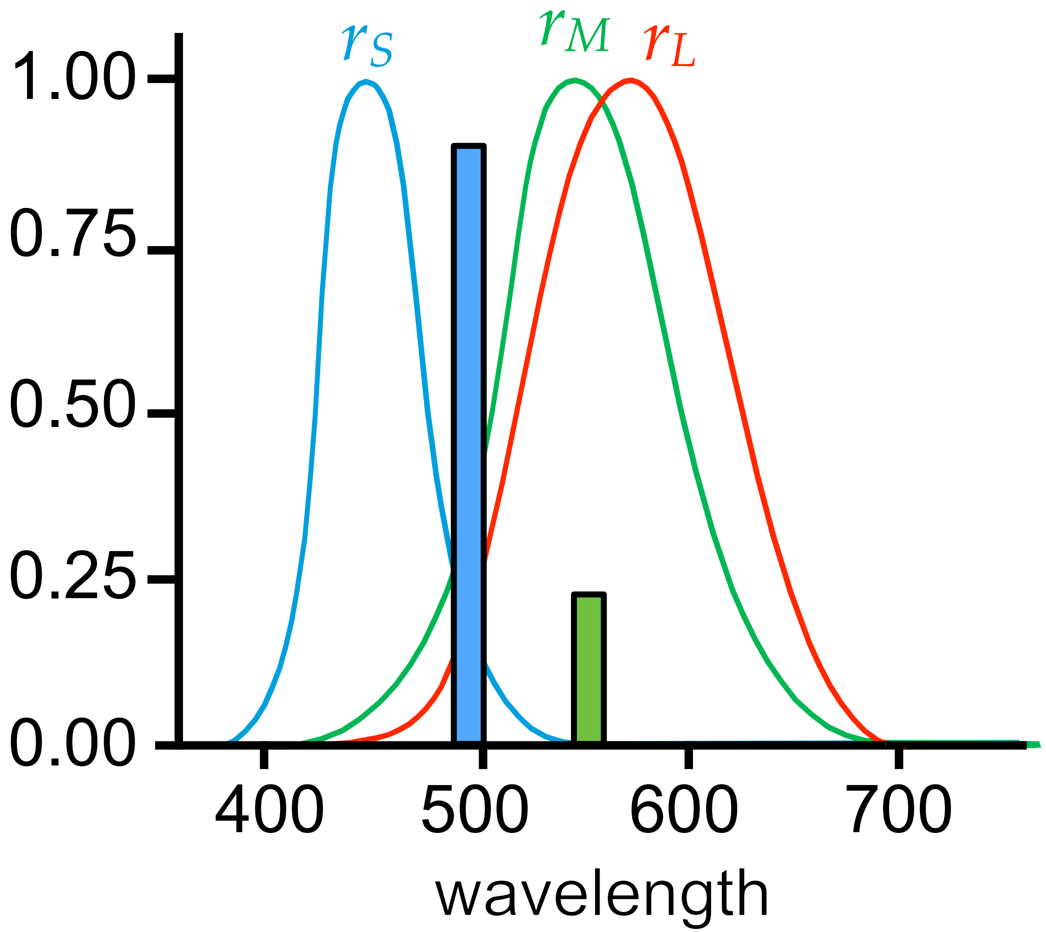
Colorimetry: an answer to the problem

- Different wavelength, different intensity
- Same response



Response comparison

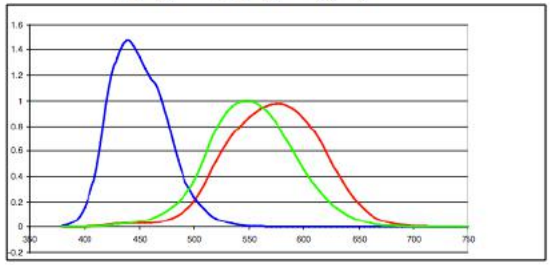
- Different wavelength, different intensity
- But different response for different cones



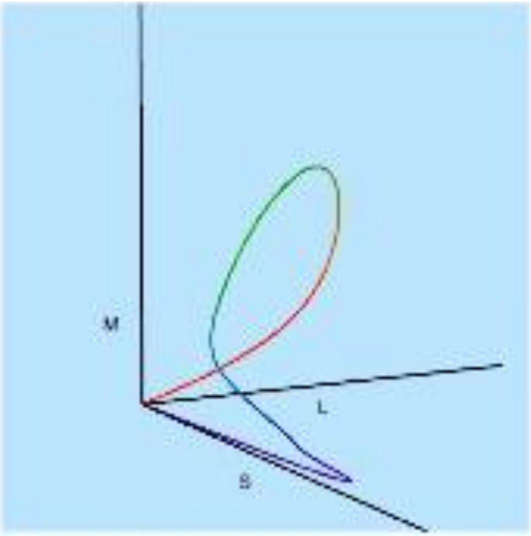
The retinal color space

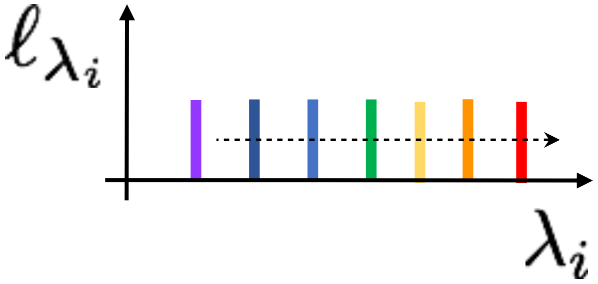
$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions



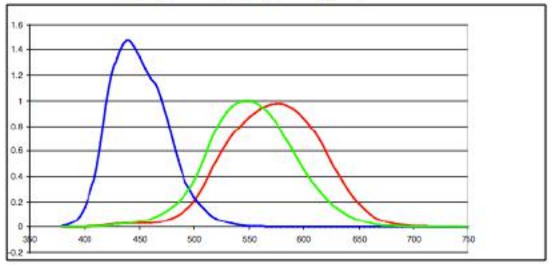


“pure beam” (laser)

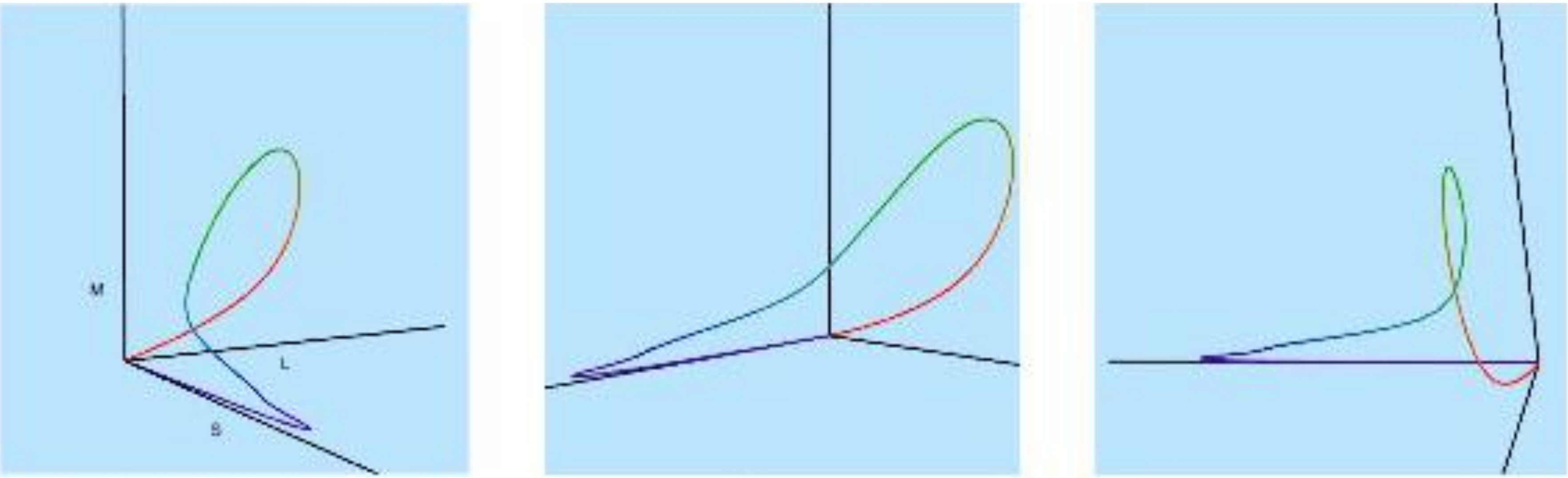
The retinal color space

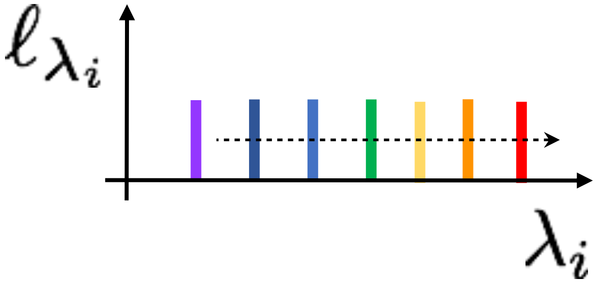
$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions





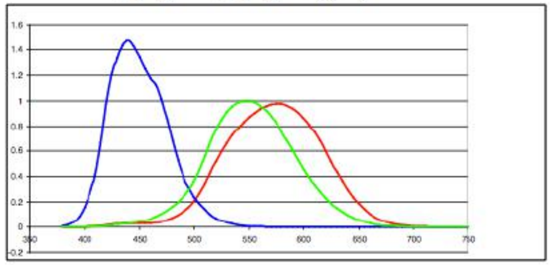
"pure beam" (laser)

- "lasso curve"
- contained in positive octant
- parameterized by wavelength
- starts and ends at origin ← why?
- never comes close to M axis ← why?

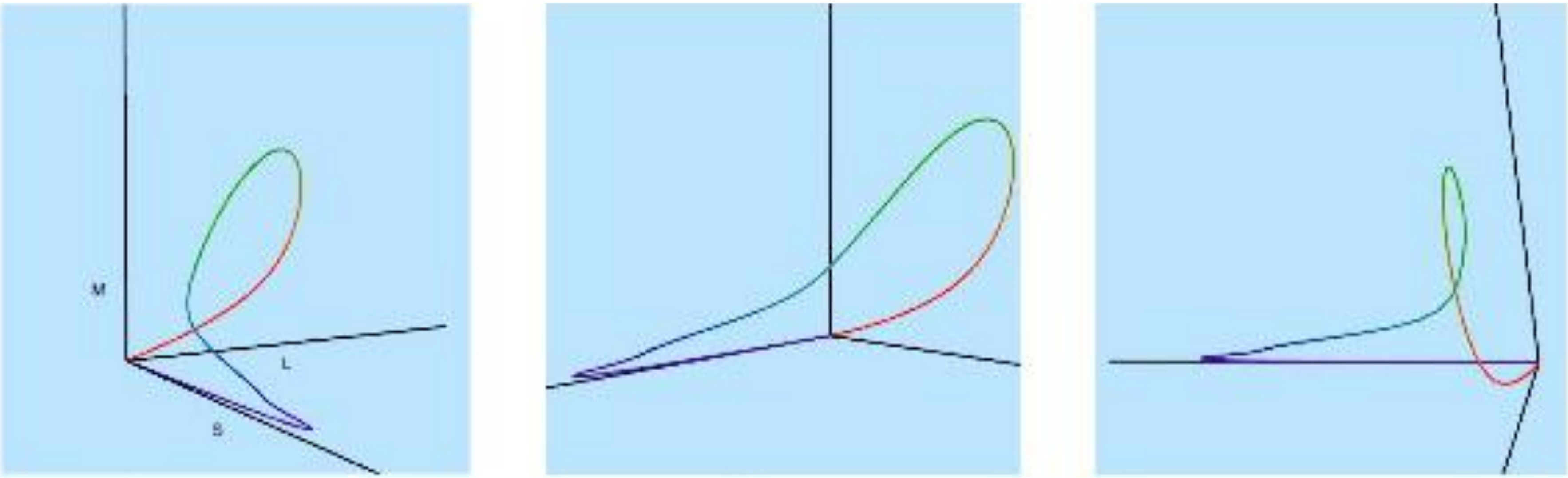
The retinal color space

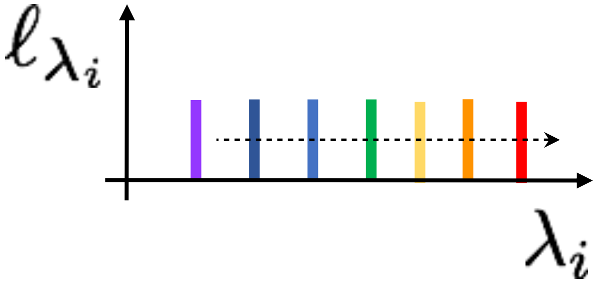
$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions





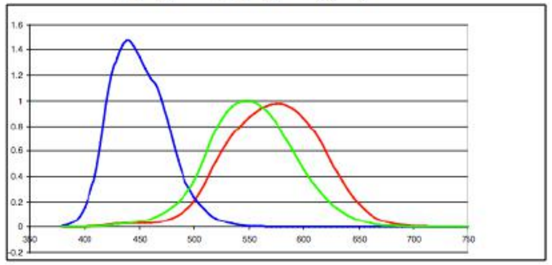
"pure beam" (laser)

- starts and ends at origin
  These extreme wavelengths are at the boundaries of the visible region, beyond which the responses are zero.

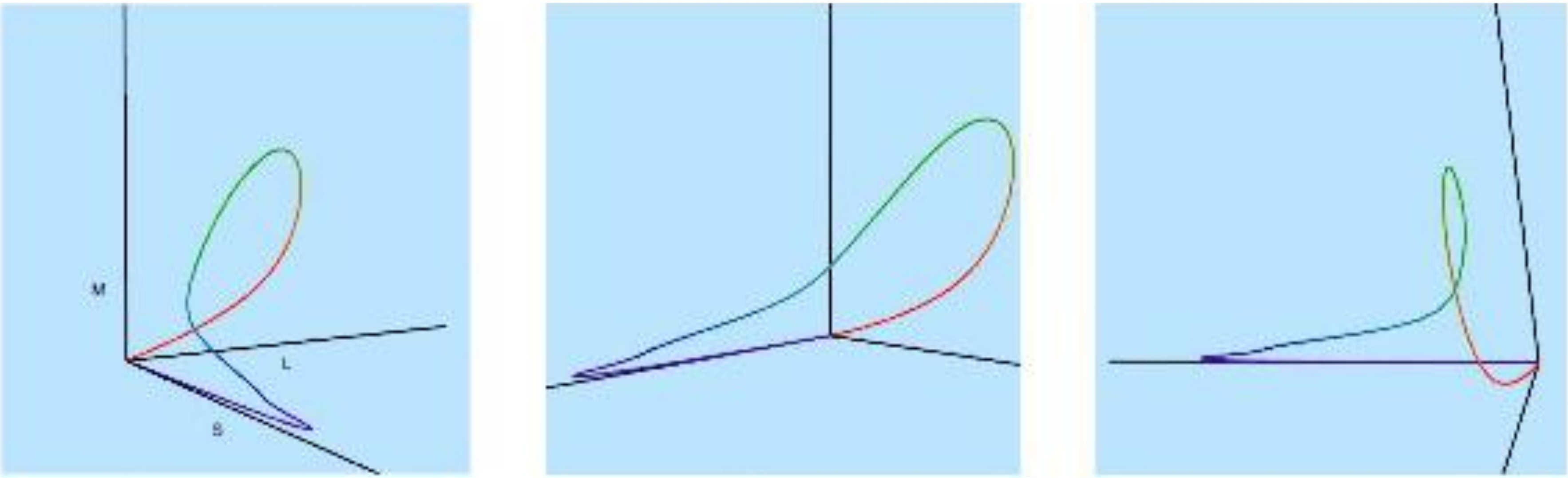
The retinal color space

$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

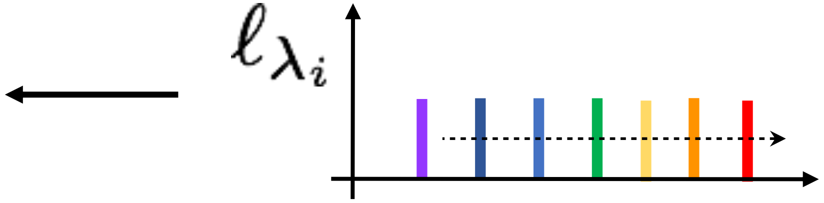
$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions



ℓ_{λ_i}



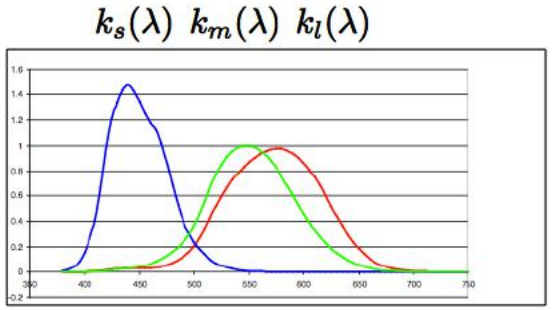
λ_i

“pure beam” (laser)

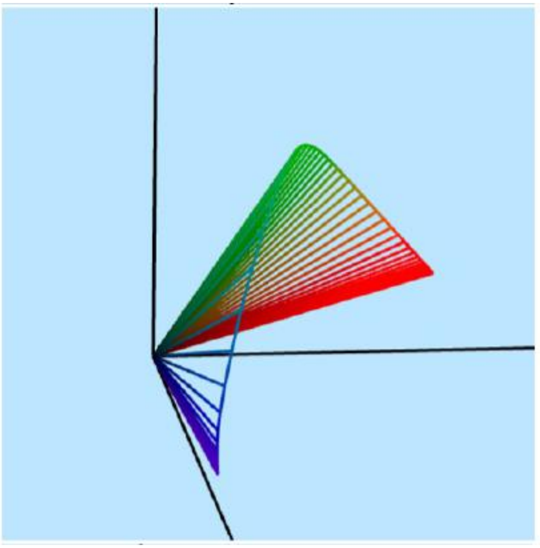
- never comes close to M axis
  There is no light that stimulates these cones alone.

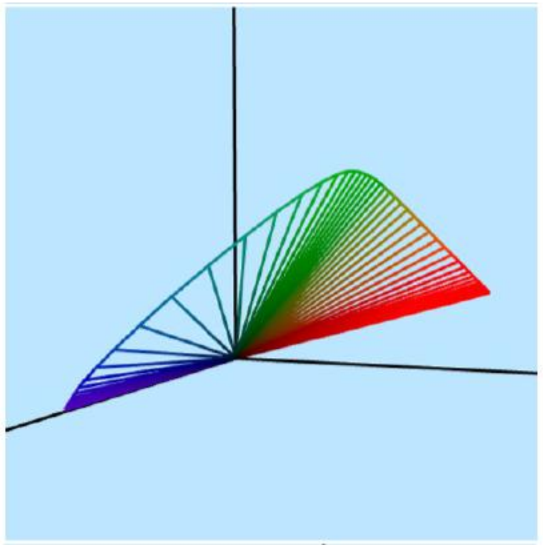
The retinal color space

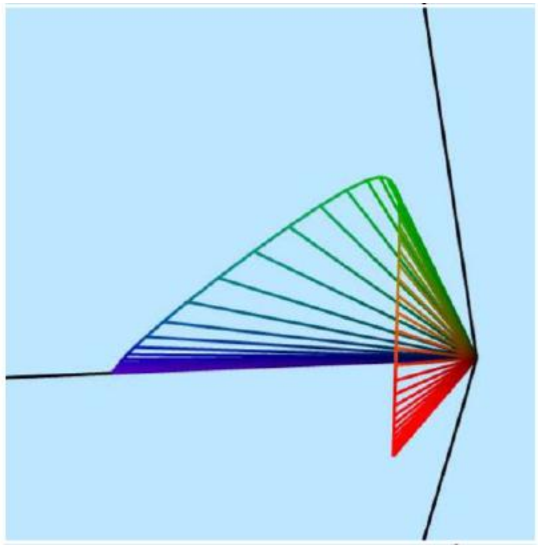
$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

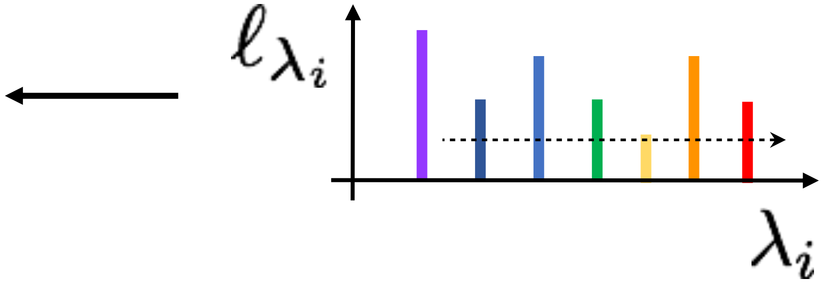


LMS senstivity functions









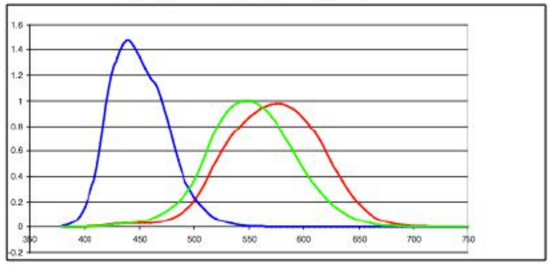
"pure beam" (laser)

if we also consider variations in the strength of the laser this "lasso" turns into (convex!) radial cone with a "horse-shoe shaped" radial cross-section

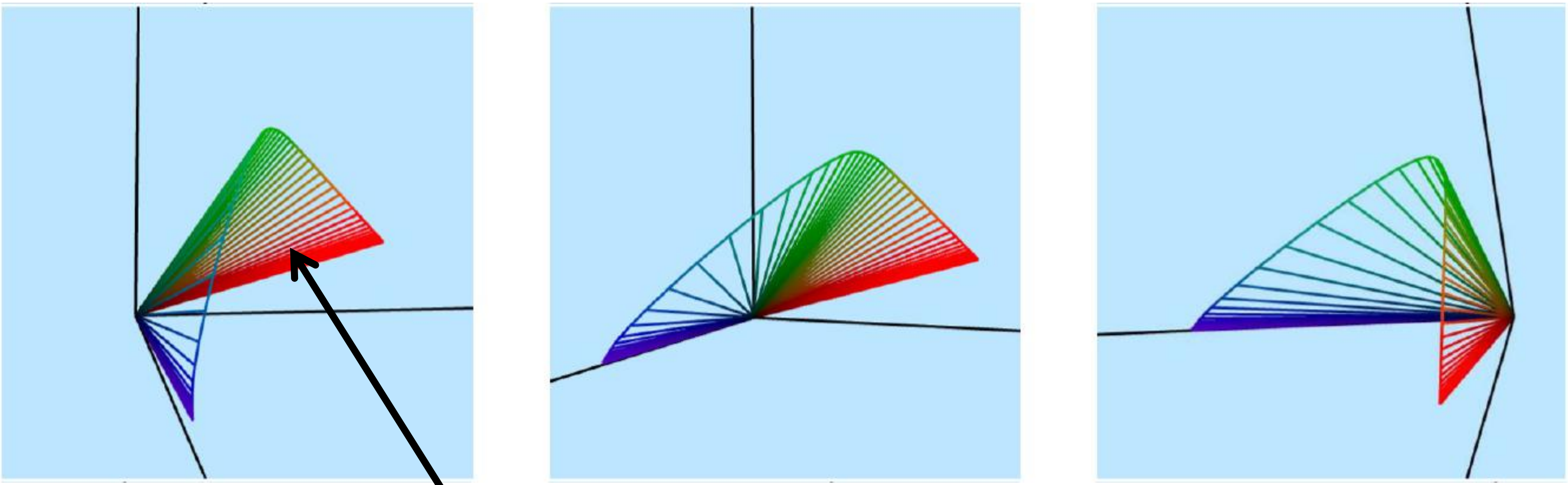
The retinal color space

$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$

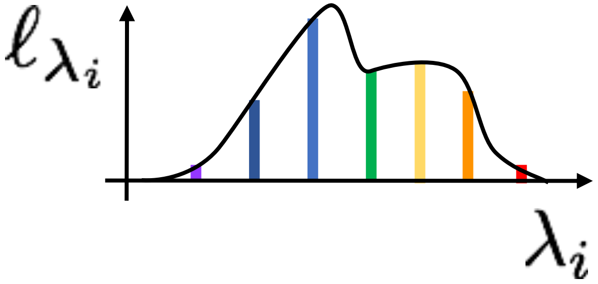
$k_s(\lambda)$ $k_m(\lambda)$ $k_l(\lambda)$



LMS senstivity functions



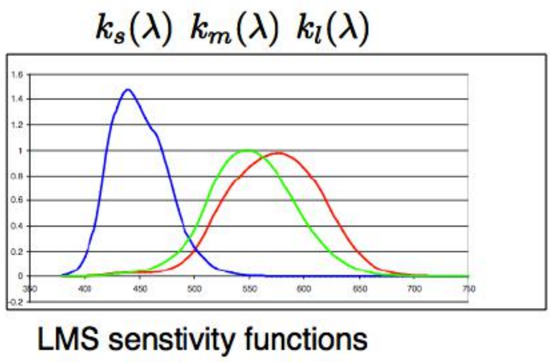
colors of mixed beams are <u>at the interior</u> of the convex cone with boundary the surface produced by monochromatic lights

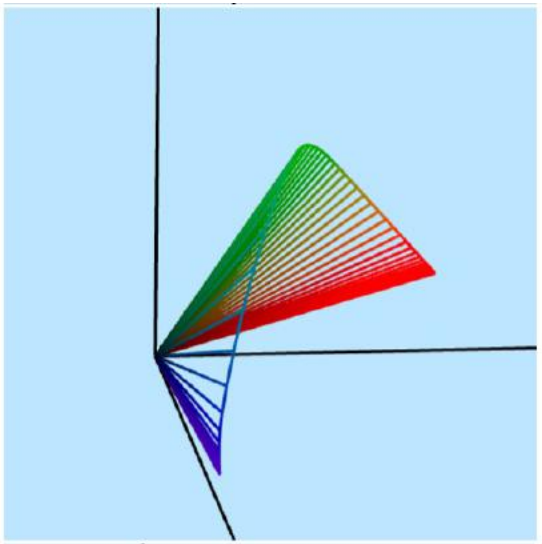


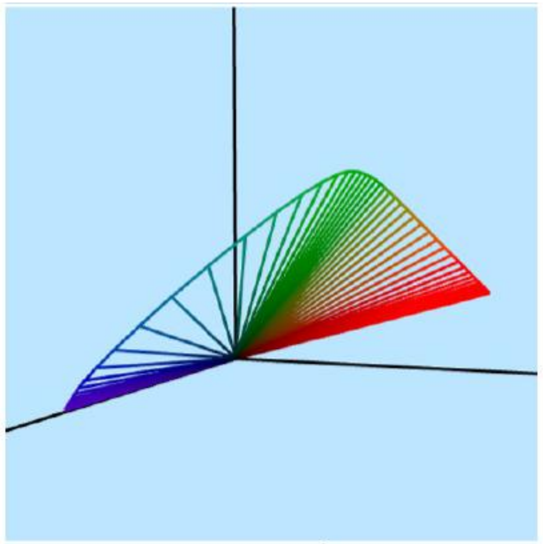
“mixed beam”
= convex combination of pure colors

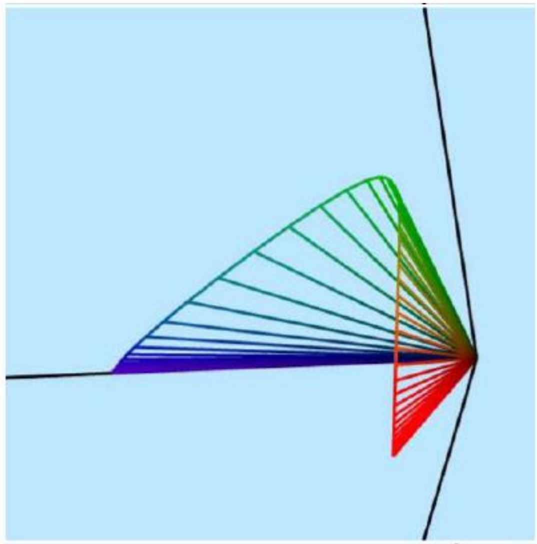
The retinal color space

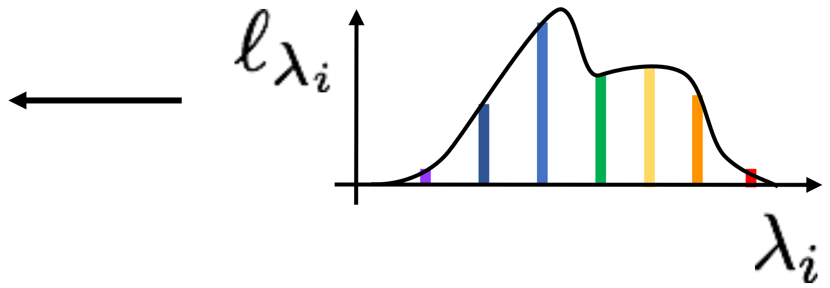
$$\mathbf{c}(\ell_{\lambda_i}) = (c_s, c_m, c_l)$$











"mixed beam"
= convex combination of pure colors

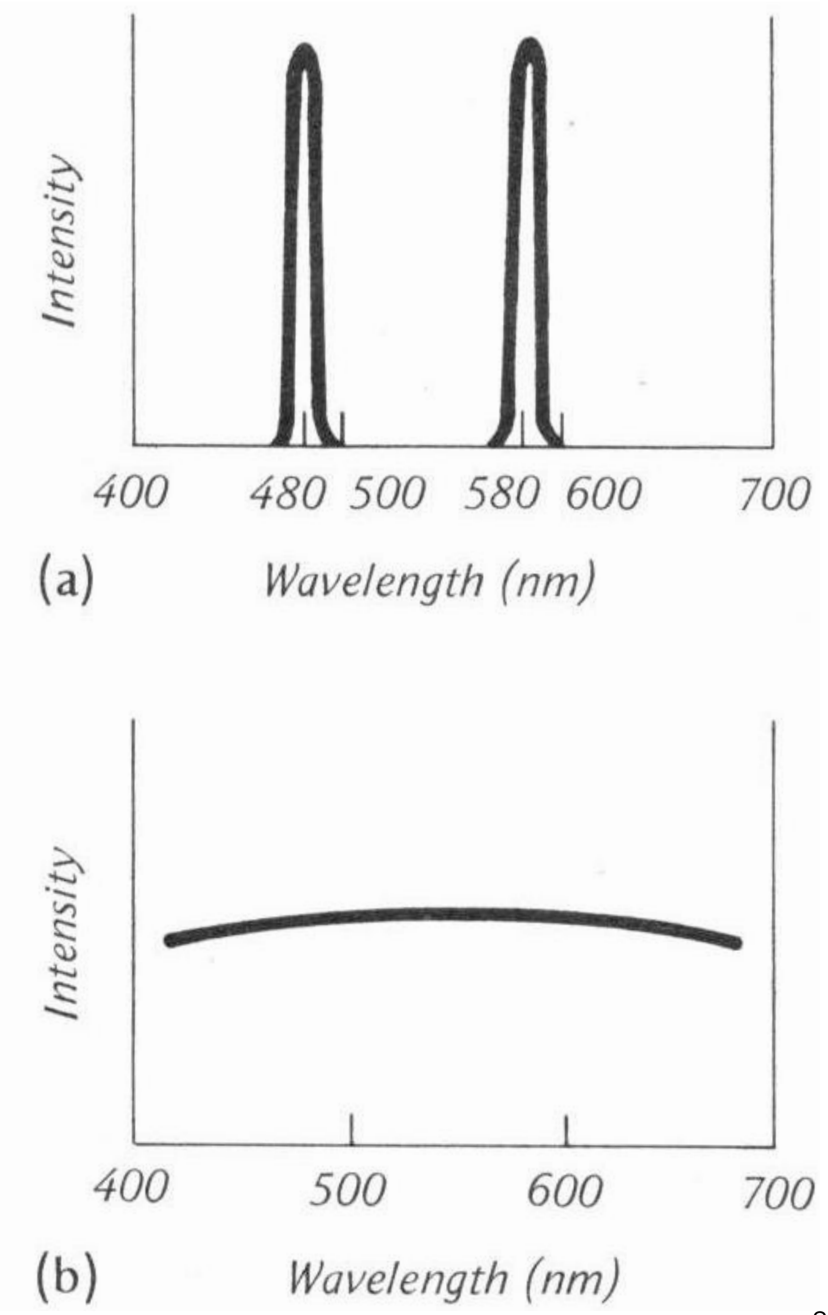
- distinct mixed beams can produce the same retinal color
- these beams are called metamers

Metamers

- We are all color blind!

1. Take a spectrum (which is a function)
2. Eye produces three numbers
3. This throws away a lot of information!
    - Two different spectra can produce the same three values/visual responses (right)

- Called **metamers**

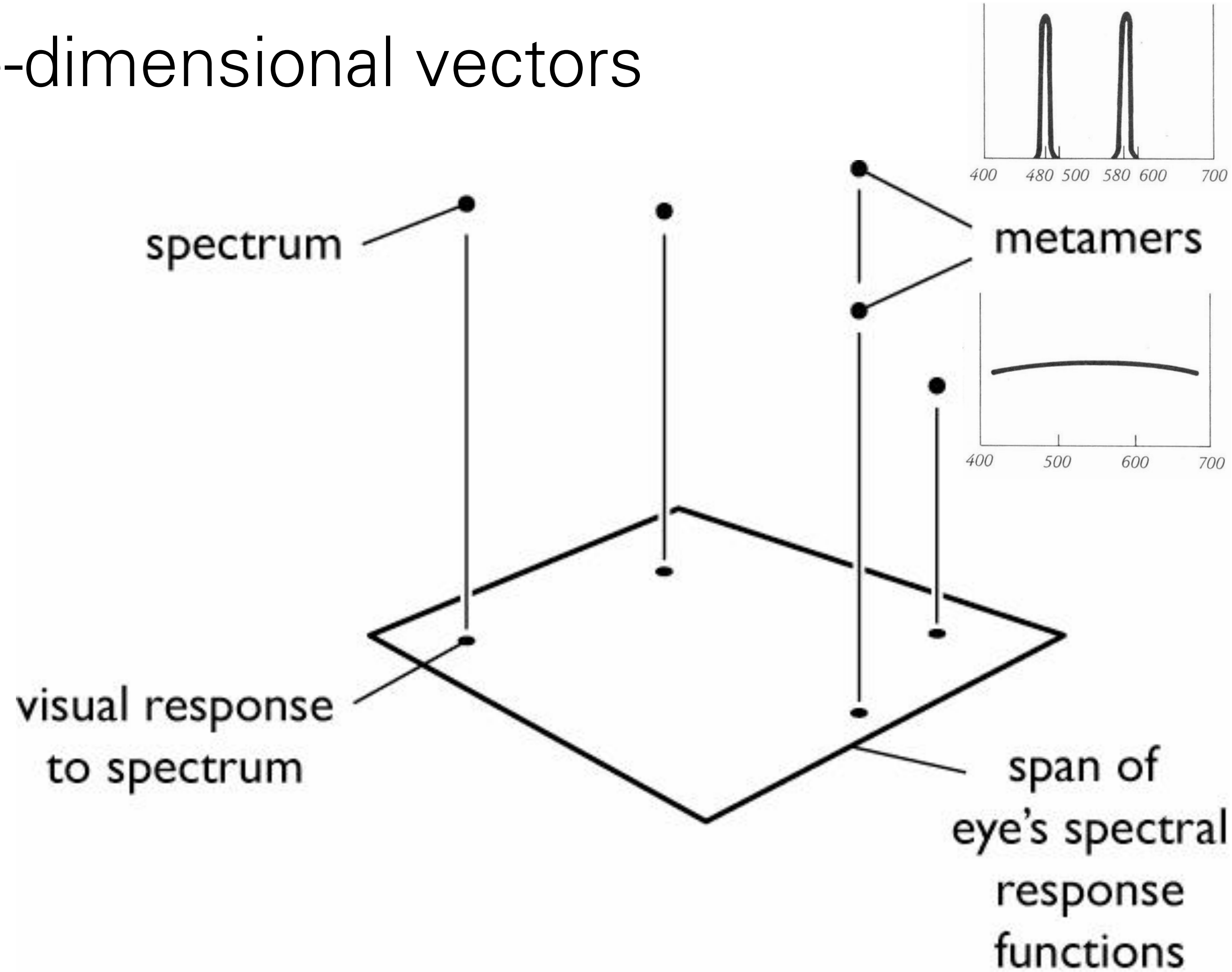


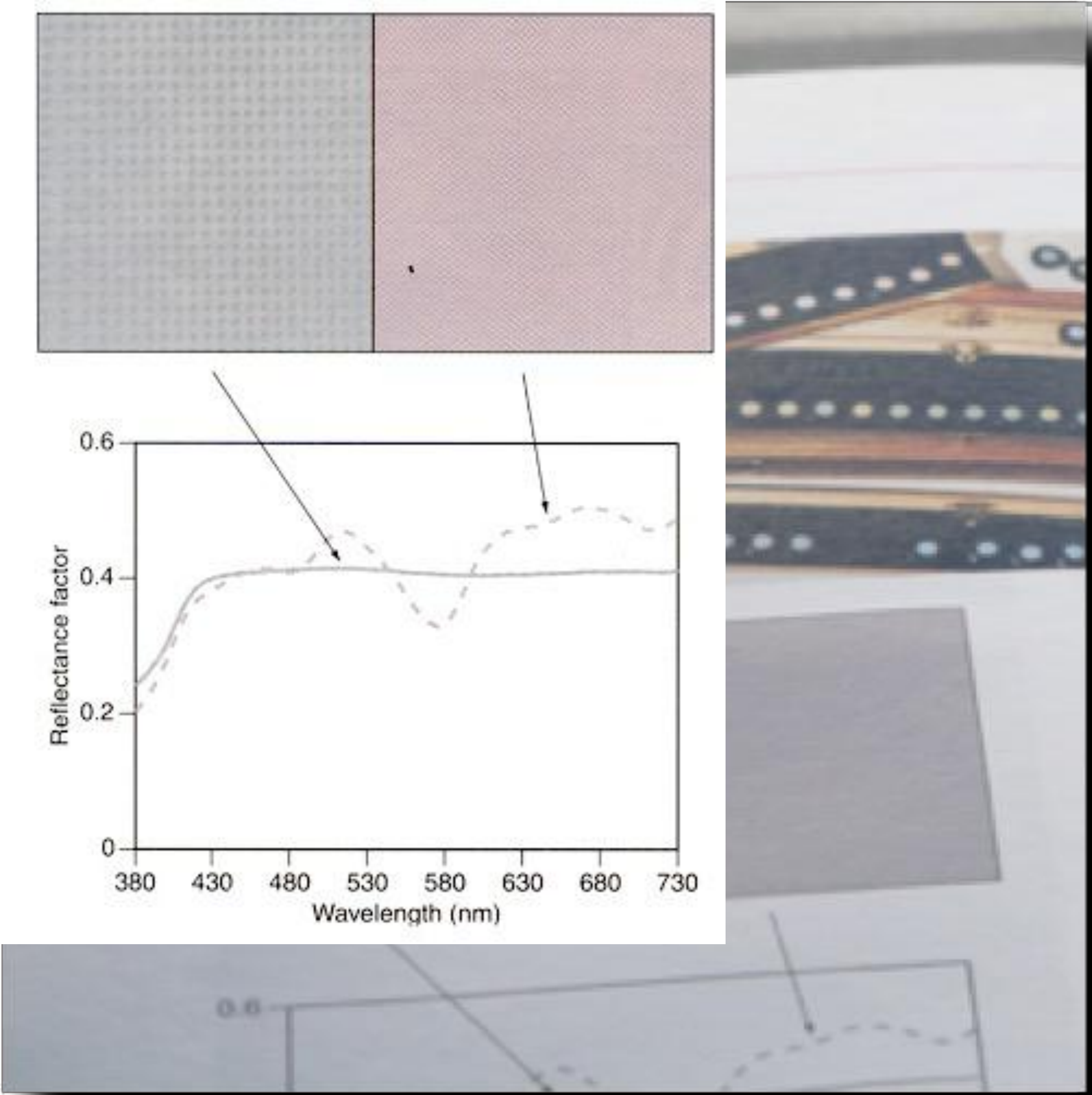
Pseudo-geometric interpretation

- A dot product is a projection
- Humans project an infinite dimensional vector (the SPD) onto a 3-D subspace
  - differences that are perpendicular to all 3 vectors are not detectable

Pseudo-geometric interpretation

- For intuition, we can imagine a 3D analog
  - 3D stands in for the infinite-dimensional vectors
  - 2D stands in for 3D
  - Then color perception is just projection onto a plane



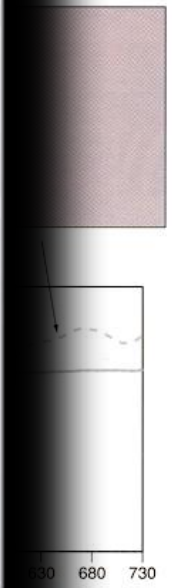


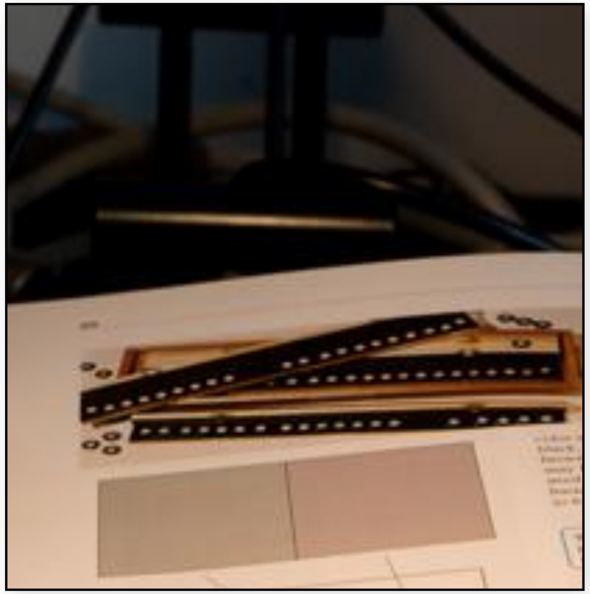
urces

ce, may not be metamers under a

es from a book.

but different under neon or halogen





Daylight

Scan (neon)

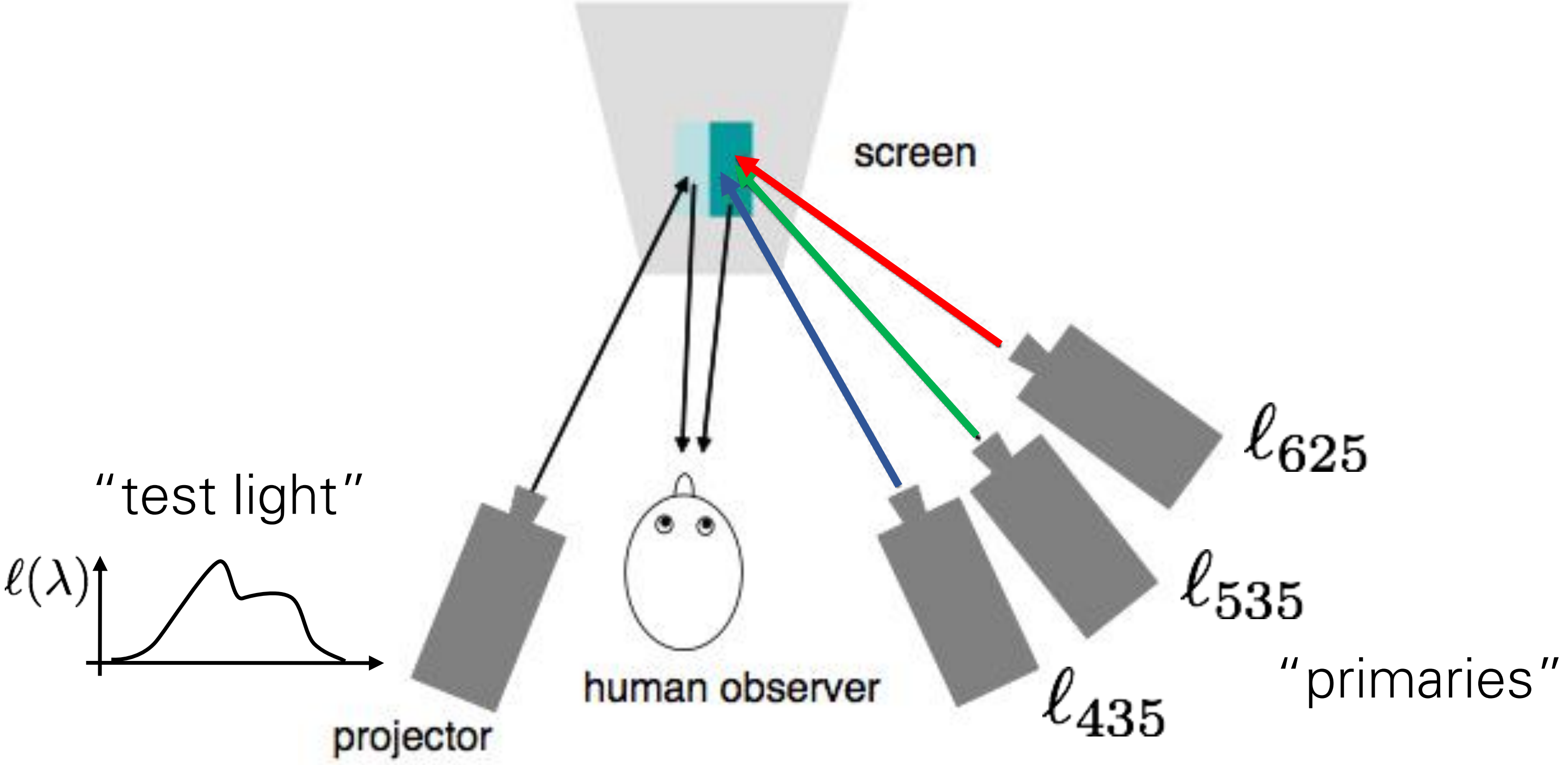
Hallogen

Recap

- Spectrum is an infinity of numbers
- Projected to 3D cone-response space
  - for each cone, multiply per wavelength and integrate
  - a.k.a. dot product
- Metamerism: infinite-D points projected to the same 3D point (different spectrum, same perceived color)
  - affected by illuminant
  - enables color reproduction with onlu 3 primaries

Color matching

CIE color matching

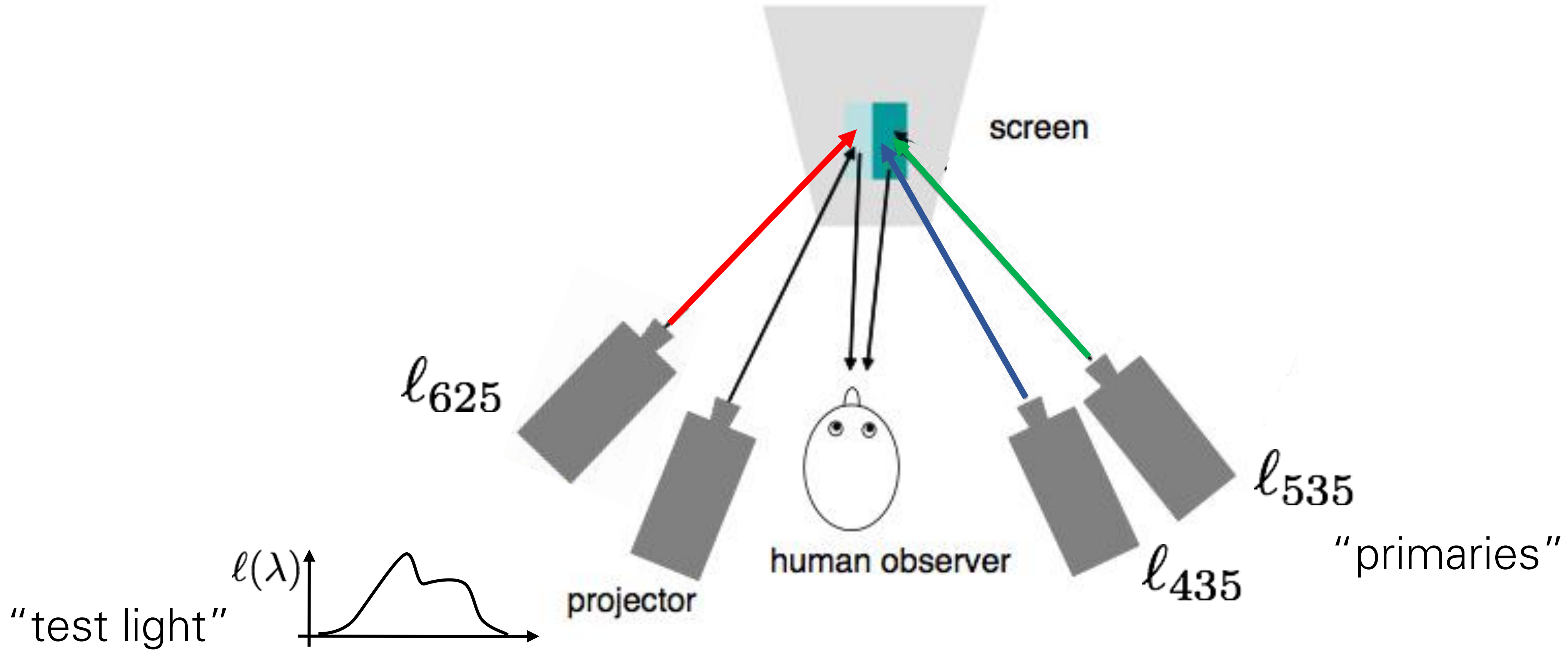


Adjust the strengths of the primaries until they re-produce the test color. Then:

$$\mathbf{c}(\ell(\lambda)) = \alpha\mathbf{c}(\ell_{435}) + \beta\mathbf{c}(\ell_{535}) + \gamma\mathbf{c}(\ell_{625})$$

equality symbol means "has the same retinal color as" or "is metameric to"

CIE color matching

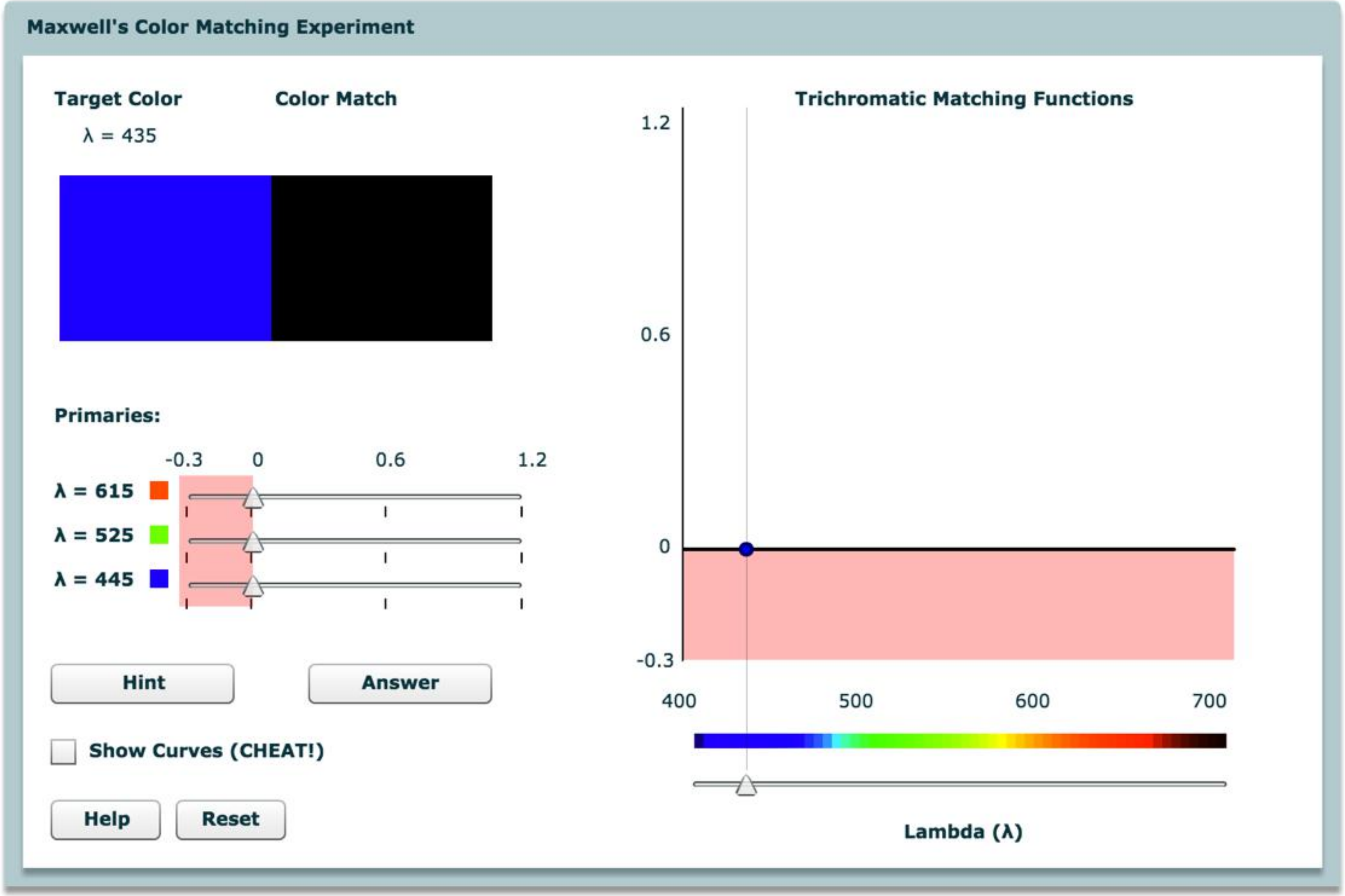


To match some test colors, you need to add some primary beam on the left (same as "subtracting light" from the right) $\longrightarrow$

$$\mathbf{c}(\ell(\lambda)) + \gamma\mathbf{c}(\ell_{625}) = \alpha\mathbf{c}(\ell_{435}) + \beta\mathbf{c}(\ell_{535})$$

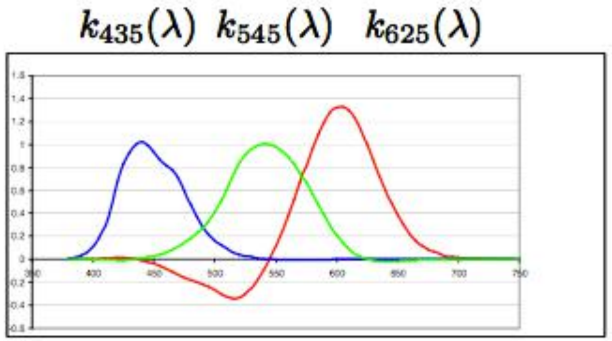
$$\mathbf{c}(\ell(\lambda)) = \alpha\mathbf{c}(\ell_{435}) + \beta\mathbf{c}(\ell_{535}) - \gamma\mathbf{c}(\ell_{625})$$

Color matching demo

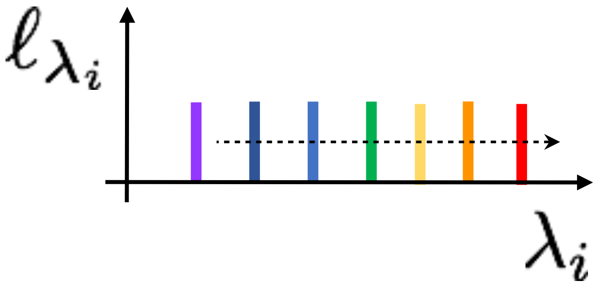


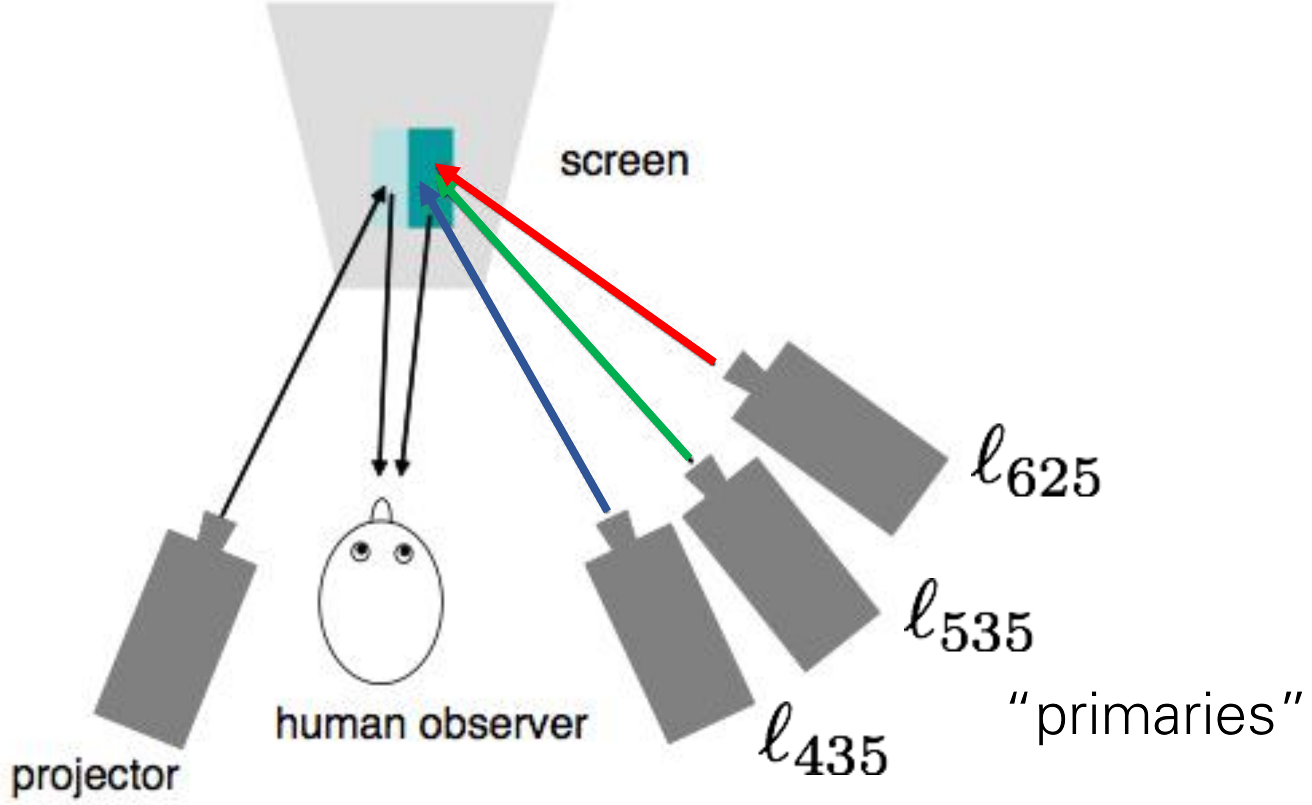
http://graphics.stanford.edu/courses/cs178/applets/colormatching.html

CIE color matching



Matching experiment matching functions



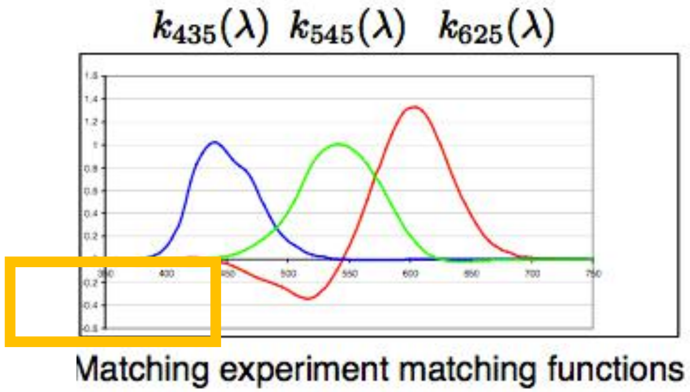


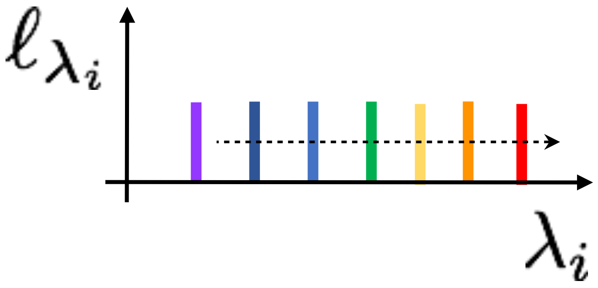
Repeat this matching experiments for pure test beams at wavelengths λ_i and keep track of the coefficients (negative or positive) required to reproduce each pure test beam.

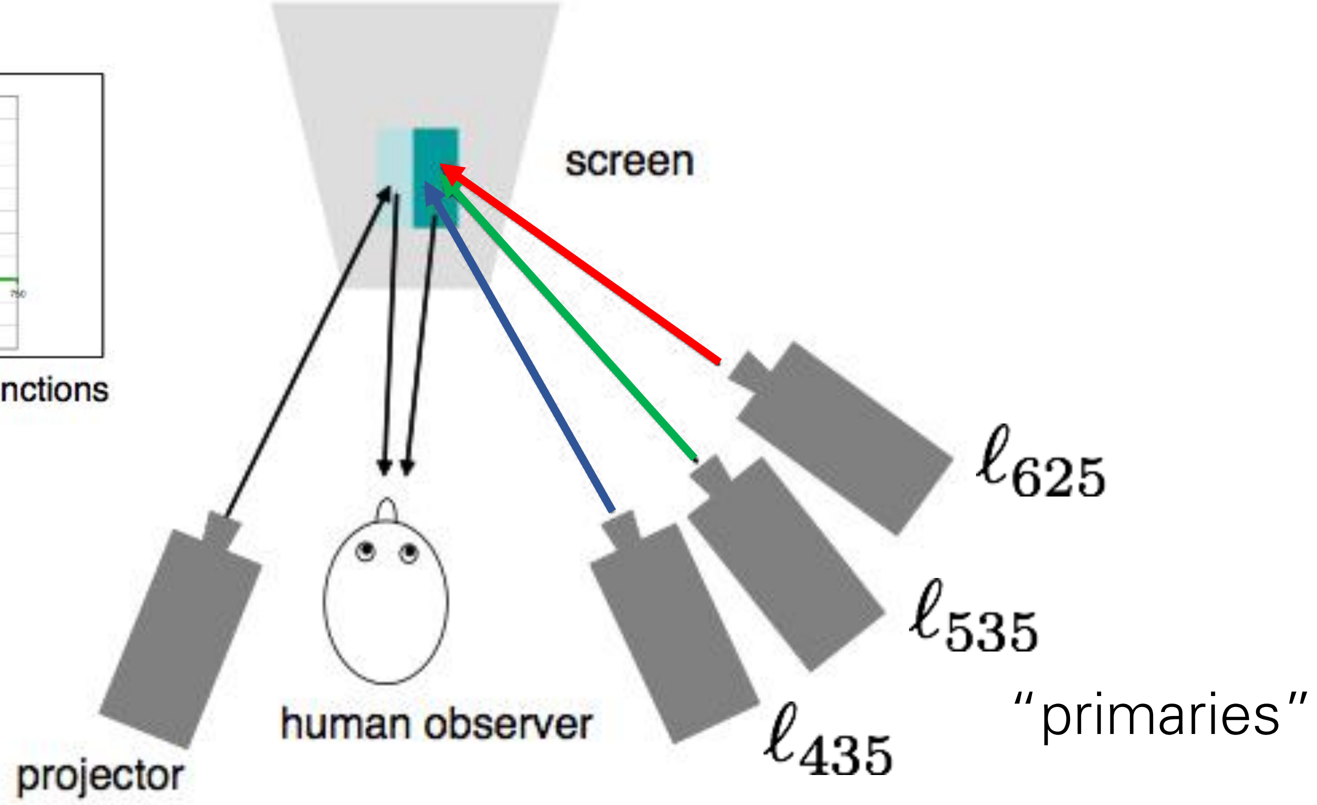
$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

CIE color matching

note the negative values



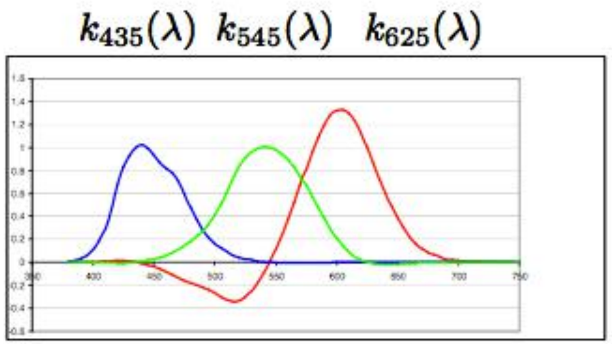




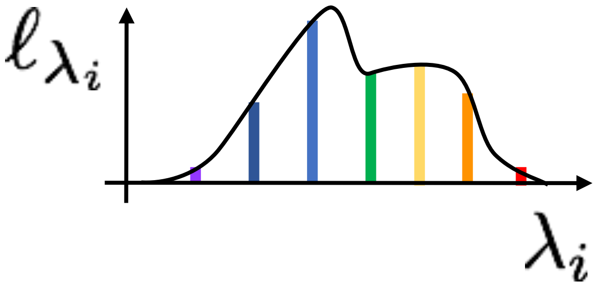
Repeat this matching experiments for pure test beams at wavelengths λ_i and keep track of the coefficients (negative or positive) required to reproduce each pure test beam.

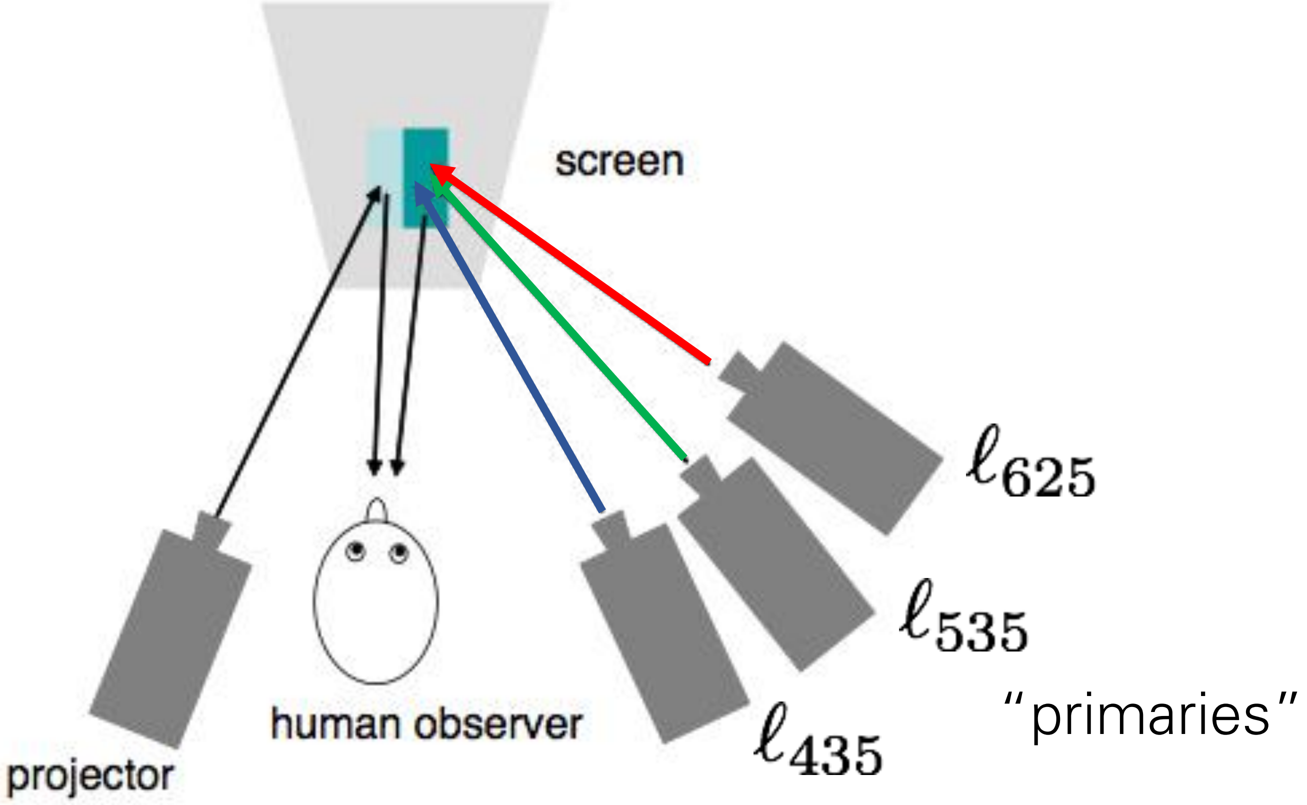
$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

CIE color matching



Matching experiment matching functions

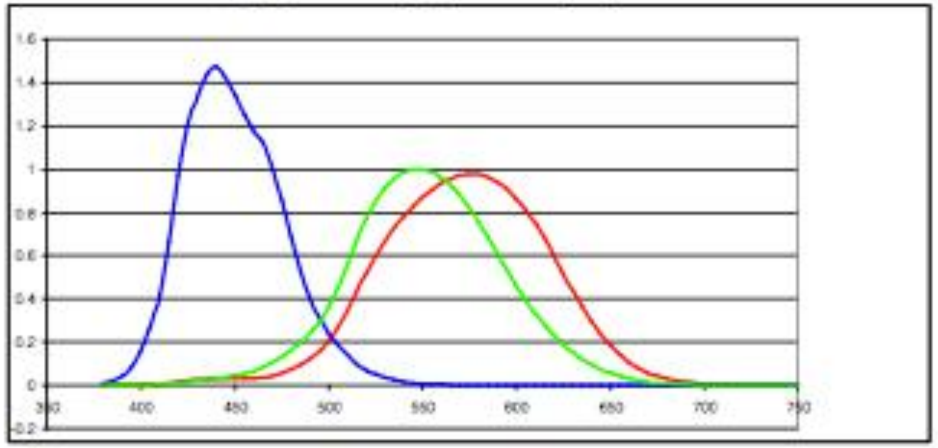




What about "mixed beams"?

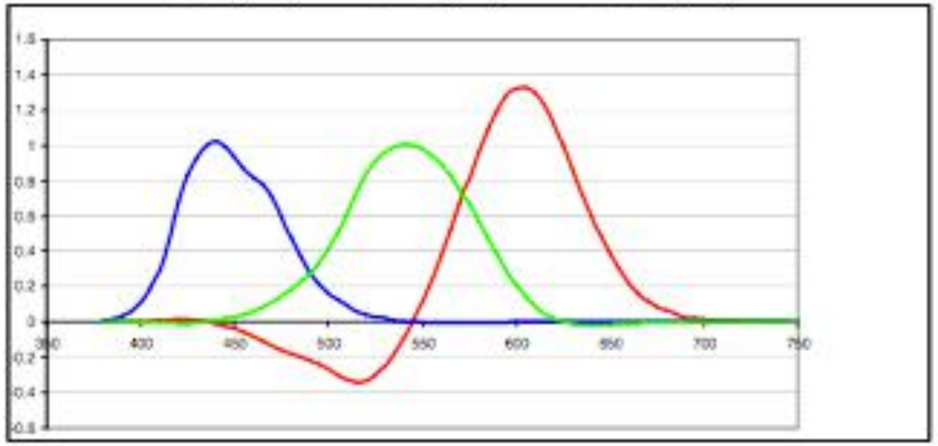
Two views of retinal color

$k_s(\lambda)\ \ k_m(\lambda)\ \ k_l(\lambda)$



LMS senstivity functions

$k_{435}(\lambda)\ \ k_{545}(\lambda)\ \ k_{625}(\lambda)$



Matching experiment matching functions

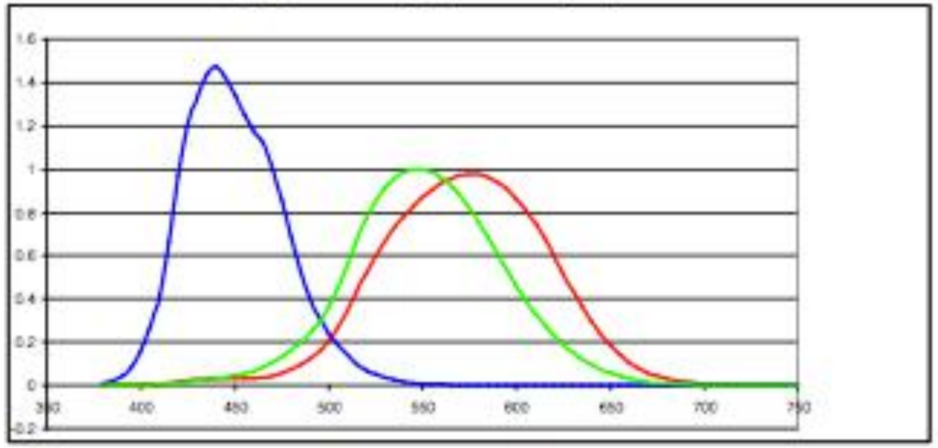
<u>Analytic:</u> Retinal color is produced by analyzing spectral power distributions using the color sensitivity functions.

<u>Synthetic:</u> Retinal color is produced by synthesizing color primaries using the color matching functions.

What is each view of retinal color best suited for?

Two views of retinal color

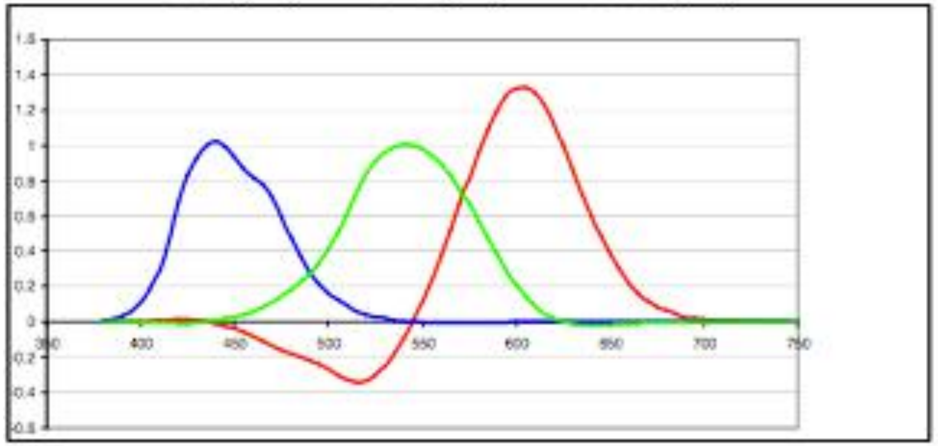
$k_s(\lambda)\ \ k_m(\lambda)\ \ k_l(\lambda)$



LMS senstivity functions

?

$k_{435}(\lambda)\ \ k_{545}(\lambda)\ \ k_{625}(\lambda)$



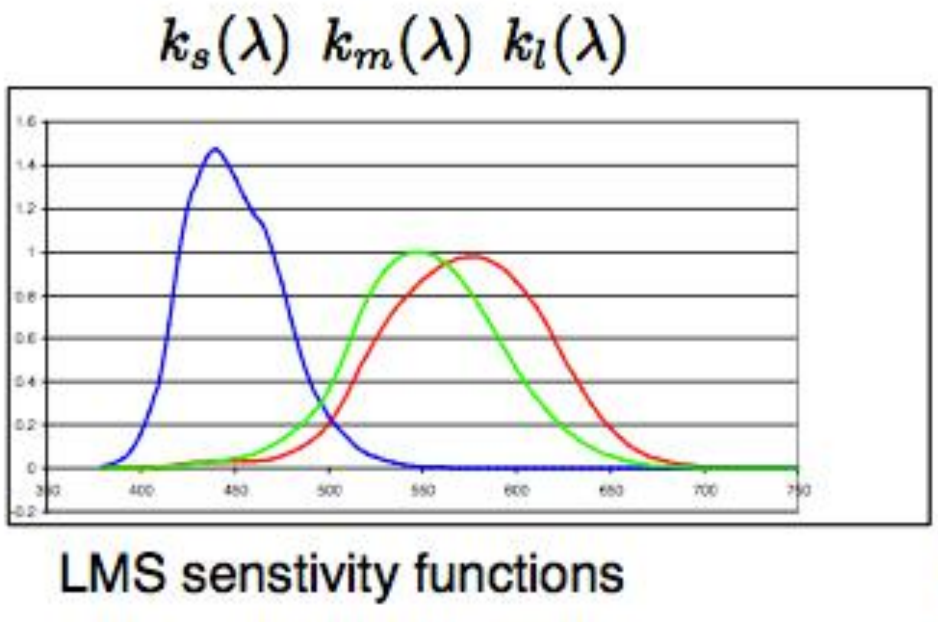
Matching experiment matching functions

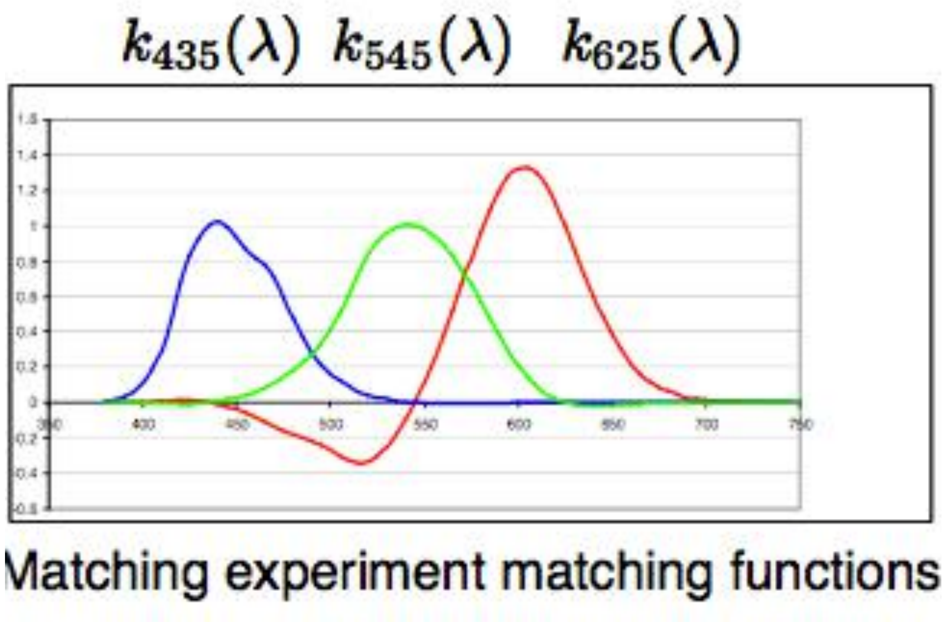
Analytic: Retinal color is produced by analyzing spectral power distributions using the color sensitivity functions.

Synthetic: Retinal color is produced by synthesizing color primaries using the color matching functions.

How do they relate to each other?

Two views of retinal color





<u>Analytic:</u> Retinal color is produced by analyzing spectral power distributions using the color sensitivity functions.

<u>Synthetic:</u> Retinal color is produced by synthesizing color primaries using the color matching functions.

<u>The two views are equivalent:</u> Color matching functions are also color sensitivity functions. For each set of color sensitivity functions, there are corresponding color primaries.

Linear color spaces

Linear color spaces

1) Color matching experimental outcome:

$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

same in matrix form:

$$\begin{bmatrix} | \\ \mathbf{c}(\lambda_\mathrm{i}) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} k_{435} \\ k_{535} \\ k_{625} \end{bmatrix}$$

how is this matrix formed?

Linear color spaces

1) Color matching experimental outcome:

$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

same in matrix form:

$$\begin{bmatrix} | \\ \mathbf{c}(\lambda_\mathrm{i}) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} k_{435} \\ k_{535} \\ k_{625} \end{bmatrix}$$

2) Implication for arbitrary mixed beams:

$$\begin{bmatrix} | \\ \mathbf{c}(\ell(\lambda)) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} \int k_{435}(\lambda)\ell(\lambda)d\lambda \\ \int k_{535}(\lambda)\ell(\lambda)d\lambda \\ \int k_{625}(\lambda)\ell(\lambda)d\lambda \end{bmatrix}$$

where do these terms come from?

Linear color spaces

1) Color matching experimental outcome:

$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

same in matrix form:

$$\begin{bmatrix} | \\ \mathbf{c}(\lambda_\mathrm{i}) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} k_{435} \\ k_{535} \\ k_{625} \end{bmatrix}$$

2) Implication for arbitrary mixed beams:

$$\begin{bmatrix} | \\ \mathbf{c}(\ell(\lambda)) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} \int k_{435}(\lambda)\ell(\lambda)d\lambda \\ \int k_{535}(\lambda)\ell(\lambda)d\lambda \\ \int k_{625}(\lambda)\ell(\lambda)d\lambda \end{bmatrix}$$

what is this similar to?

Linear color spaces

1) Color matching experimental outcome:

$$\mathbf{c}(\lambda_i) = k_{435}(\lambda)\mathbf{c}(\ell_{435}) + k_{535}(\lambda)\mathbf{c}(\ell_{535}) + k_{625}(\lambda)\mathbf{c}(\ell_{625})$$

same in matrix form:

$$\begin{bmatrix} | \\ \mathbf{c}(\lambda_\mathrm{i}) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\boldsymbol{\ell_{435}}) & \mathbf{c}(\boldsymbol{\ell_{545}}) & \mathbf{c}(\boldsymbol{\ell_{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} k_{435} \\ k_{535} \\ k_{625} \end{bmatrix}$$

2) Implication for arbitrary mixed beams:

$$\begin{bmatrix} | \\ \mathbf{c}(\ell(\lambda)) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\boldsymbol{\ell_{435}}) & \mathbf{c}(\boldsymbol{\ell_{545}}) & \mathbf{c}(\boldsymbol{\ell_{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} \int k_{435}(\lambda)\ell(\lambda)d\lambda \\ \int k_{535}(\lambda)\ell(\lambda)d\lambda \\ \int k_{625}(\lambda)\ell(\lambda)d\lambda \end{bmatrix}$$

representation of retinal color in LMS space

change of basis matrix

representation of retinal color in space of primaries

Linear color spaces

basis for retinal color ⇔ color matching functions ⇔ primary colors ⇔ color space

$$\begin{bmatrix} | \\ \mathbf{c}(\ell(\lambda)) \\ | \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 \\ | & | & | \end{bmatrix}}_{\begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \mathbf{M}^{-1}} \begin{bmatrix} \int k_1(\lambda)\ell(\lambda)d\lambda \\ \int k_2(\lambda)\ell(\lambda)d\lambda \\ \int k_3(\lambda)\ell(\lambda)d\lambda \end{bmatrix}$$

$$\begin{bmatrix} k_1(\lambda) \\ k_2(\lambda) \\ k_3(\lambda) \end{bmatrix} = \mathbf{M} \begin{bmatrix} k_{435}(\lambda) \\ k_{545}(\lambda) \\ k_{625}(\lambda) \end{bmatrix}$$

$\mathbf{M}^{-1}\mathbf{M}$ can insert any invertible M

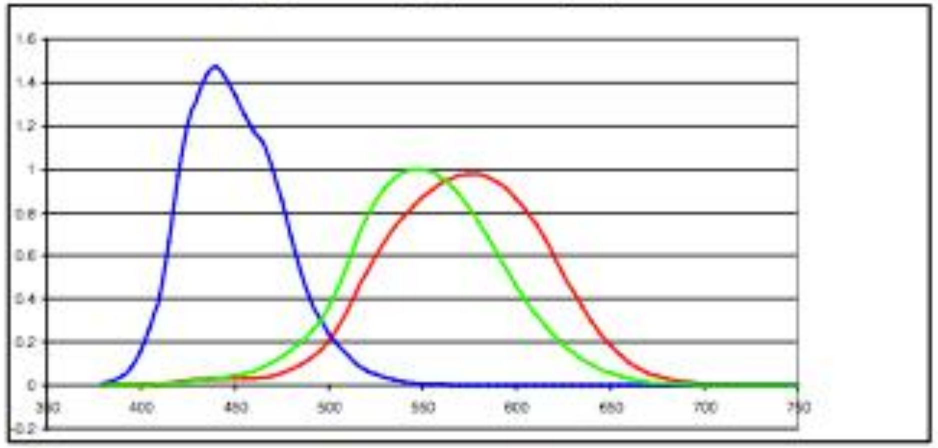
$$\begin{bmatrix} | \\ \mathbf{c}(\ell(\lambda)) \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{c}(\ell_{\mathbf{435}}) & \mathbf{c}(\ell_{\mathbf{545}}) & \mathbf{c}(\ell_{\mathbf{625}}) \\ | & | & | \end{bmatrix} \begin{bmatrix} \int k_{435}(\lambda)\ell(\lambda)d\lambda \\ \int k_{535}(\lambda)\ell(\lambda)d\lambda \\ \int k_{625}(\lambda)\ell(\lambda)d\lambda \end{bmatrix}$$

representation of retinal color in LMS space

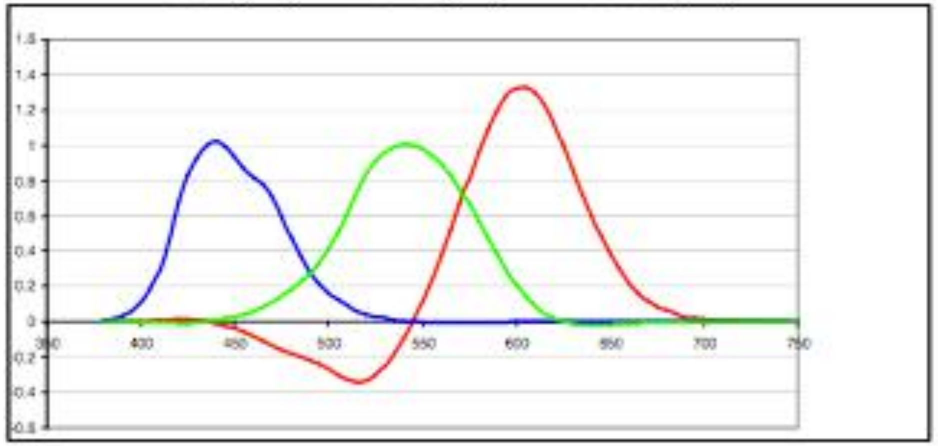
change of basis matrix

representation of retinal color in space of primaries

A few important color spaces



LMS color space



CIE RGB color space

not the "usual" RGB color
space encountered in practice

Two views of retinal color

Analytic: Retinal color is three numbers formed by taking the dot product of a power spectral distribution with three color matching/sensitivity functions.

Synthetic: Retinal color is three numbers formed by assigning weights to three color primaries to match the perception of a power spectral distribution.

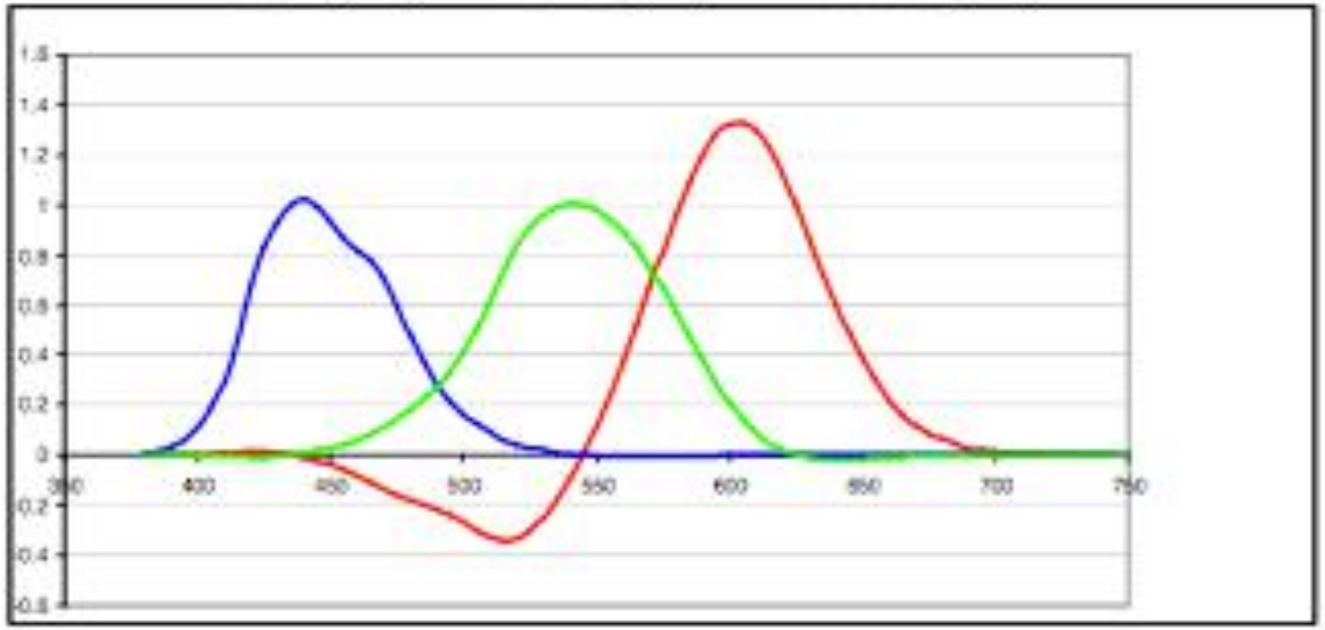
How would you make a color measurement device?

How would you make a color measurement device?

Do what the eye does:

- Select three spectral filters (i.e., three color matching functions.).
- Capture three measurements.

Can we use the CIE RGB color matching functions?

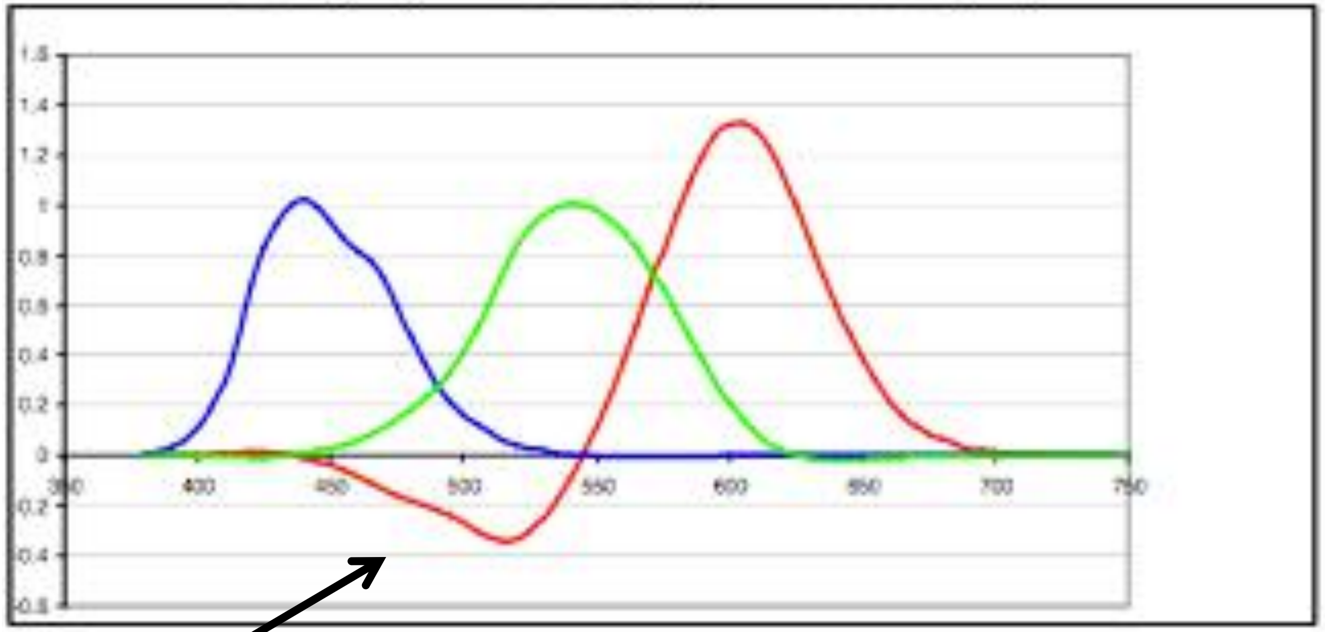


CIE RGB color space

How would you make a color measurement device?

Do what the eye does:

- Select three spectral filters (i.e., three color matching functions.).
- Capture three measurements.



Can we use the CIE RGB color matching functions?

CIE RGB color space

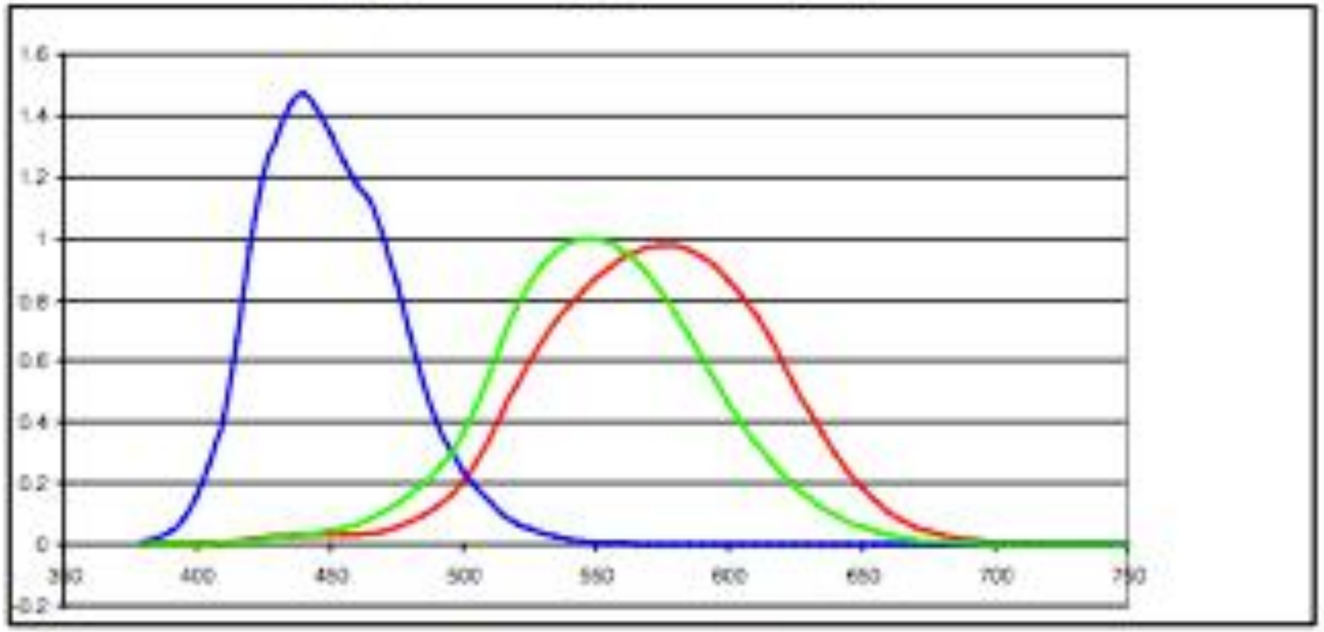
Negative values are an issue
(we can't "subtract" light at a sensor)

How would you make a color measurement device?

Do what the eye does:

- Select three spectral filters (i.e., three color matching functions).
- Capture three measurements.

Can we use the LMS color matching functions?

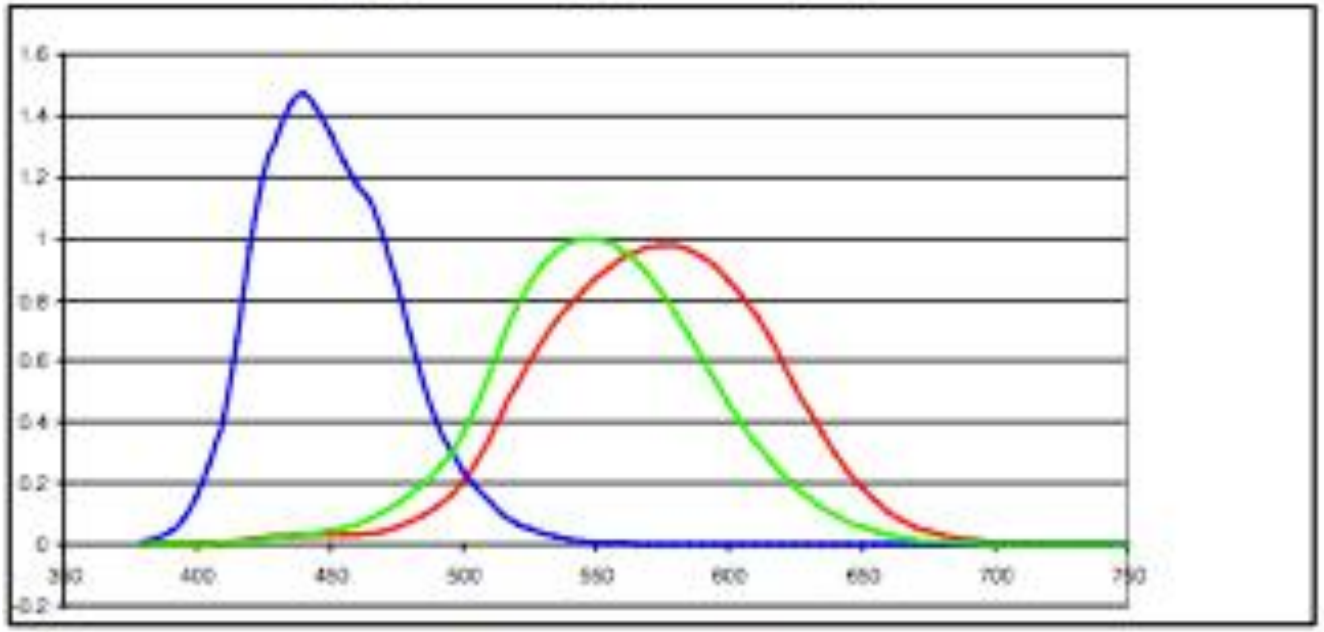


LMS color space

How would you make a color measurement device?

Do what the eye does:

- Select three spectral filters (i.e., three color matching functions).
- Capture three measurements.



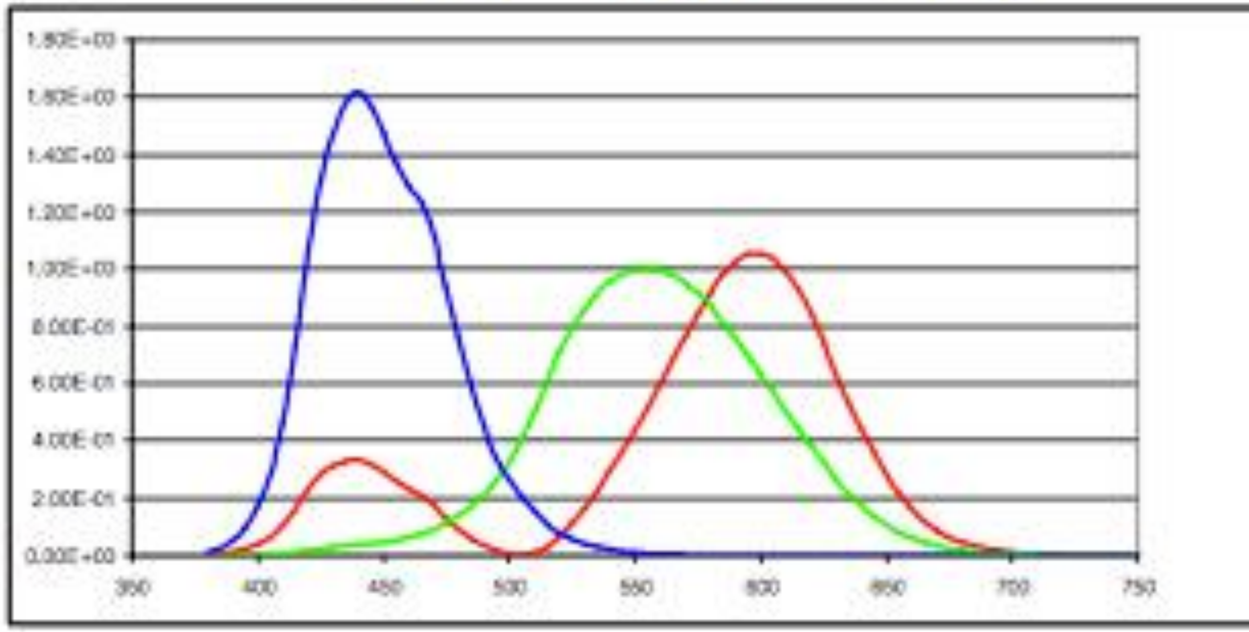
LMS color space

Can we use the LMS color matching functions?

- They weren't known when CIE was doing their color matching experiments.

How would you make a color measurement device?

- Derived from CIE RGB by adding enough blue and green to make the red positive.
- Probably the most important reference (i.e., device independent) color space.

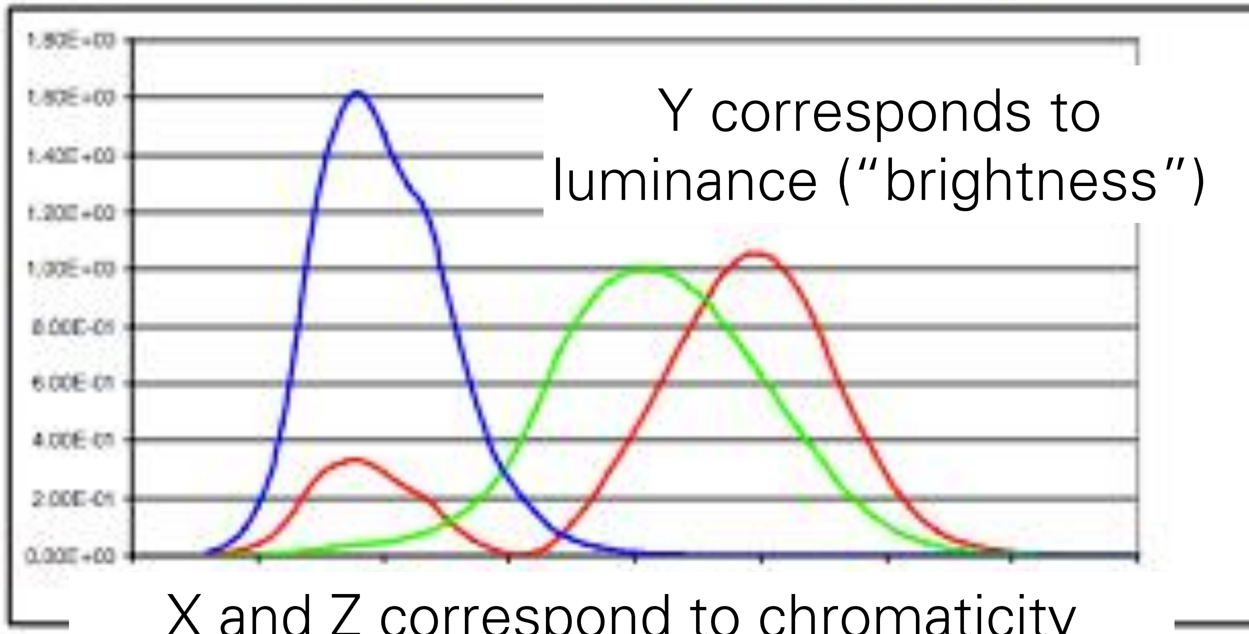


CIE XYZ color space

Remarkable and/or scary: 80+ years of CIE XYZ is all down to color matching experiments done with 12 “standard observers”.

The CIE XYZ color space

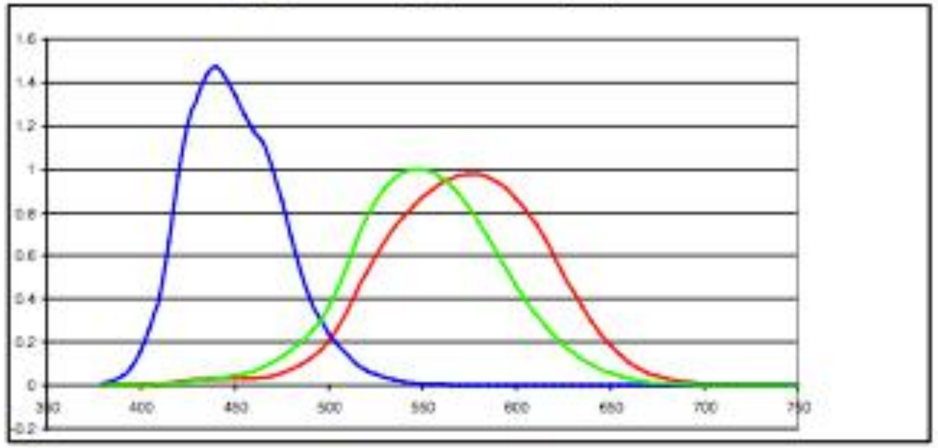
- Derived from CIE RGB by adding enough blue and green to make the red positive.
- Probably the most important reference (i.e., device independent) color space.



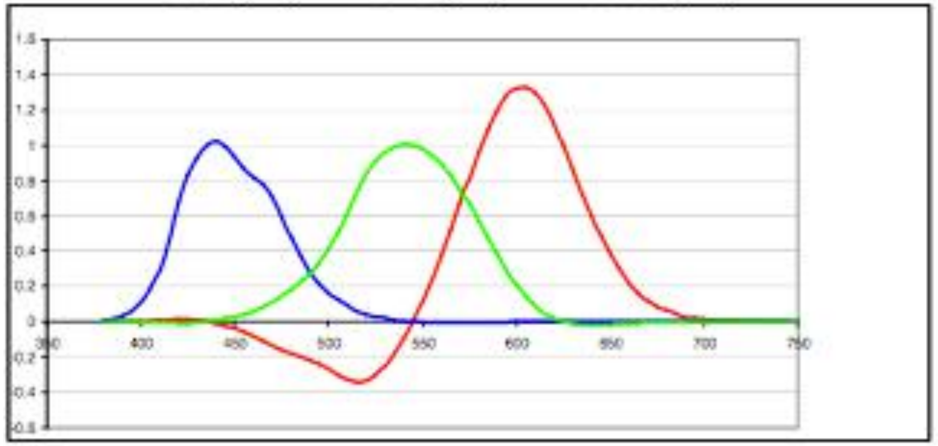
CIE XYZ color space

How would you convert a color image to grayscale?

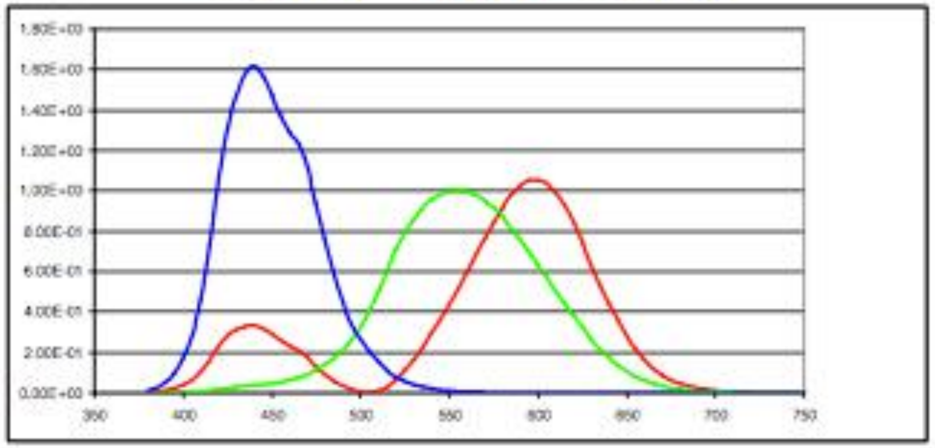
A few important color spaces



LMS color space



CIE RGB color space



CIE XYZ color space

Two views of retinal color

Analytic: Retinal color is three numbers formed by taking the dot product of a power spectral distribution with three color matching/sensitivity functions.

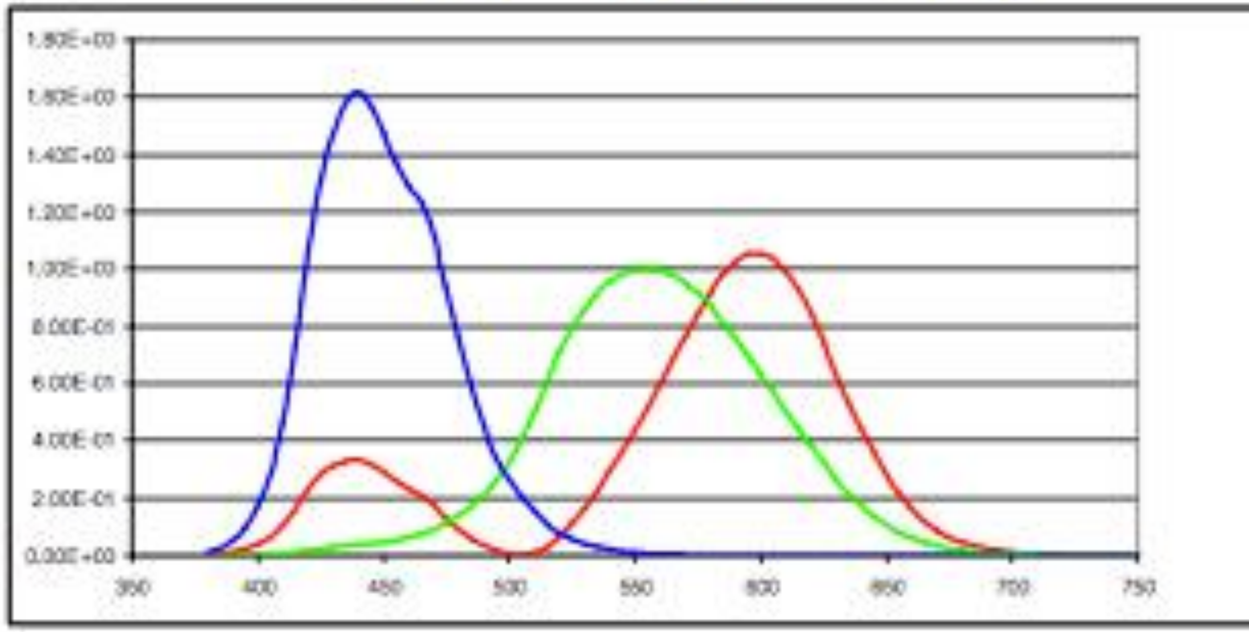
Synthetic: Retinal color is three numbers formed by assigning weights to three color primaries to match the perception of a power spectral distribution.

How would you make a color reproduction device?

How would you make a color reproduction device?

Do what color matching does:
- Select three color primaries.
- Represent all colors as mixtures of these three primaries.

Can we use the XYZ color primaries?

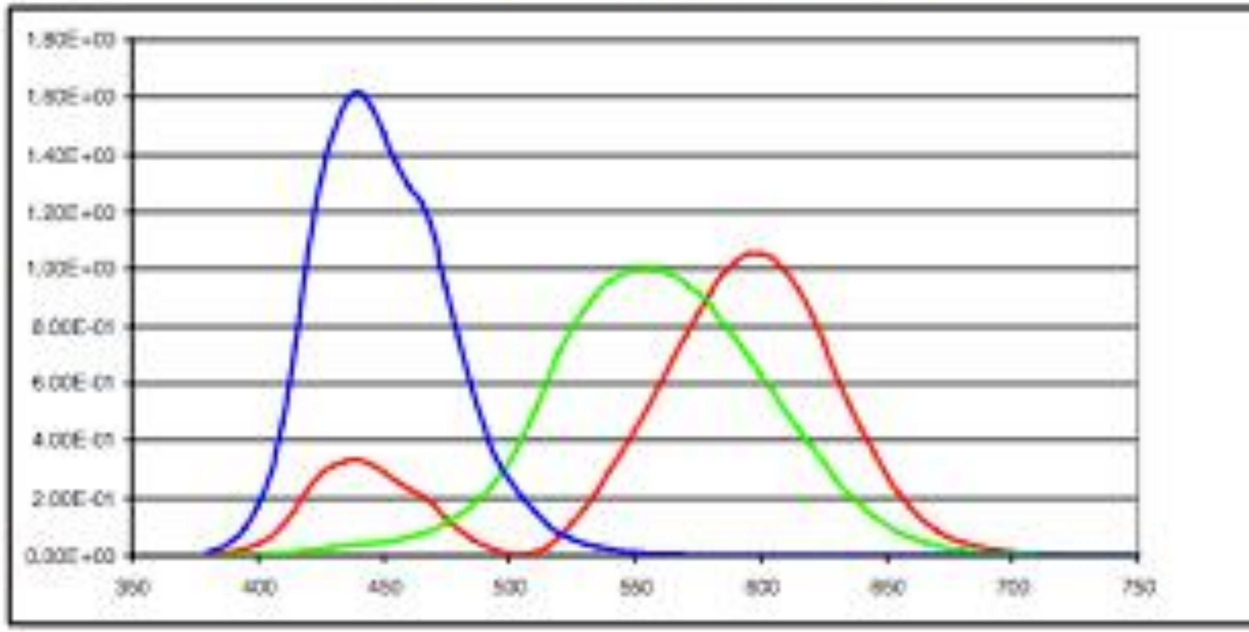


CIE XYZ color space

How would you make a color reproduction device?

Do what color matching does:

- Select three color primaries.
- Represent all colors as mixtures of these three primaries.



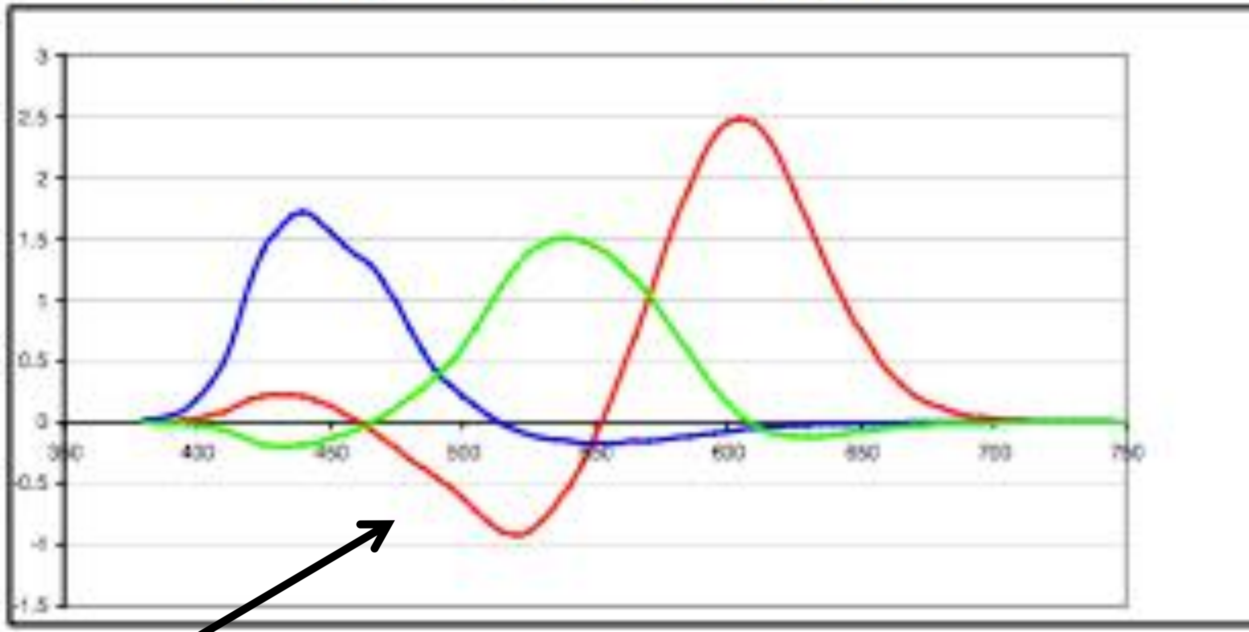
CIE XYZ color space

Can we use the XYZ color primaries?

- No, because they are not "real" colors (they require an SPD with negative values).
- Same goes for LMS color primaries.

The Standard RGB (sRGB) color space

- Derived by Microsoft and HP in 1996, based on CRT displays used at the time.
- Similar but not equivalent to CIE RGB.



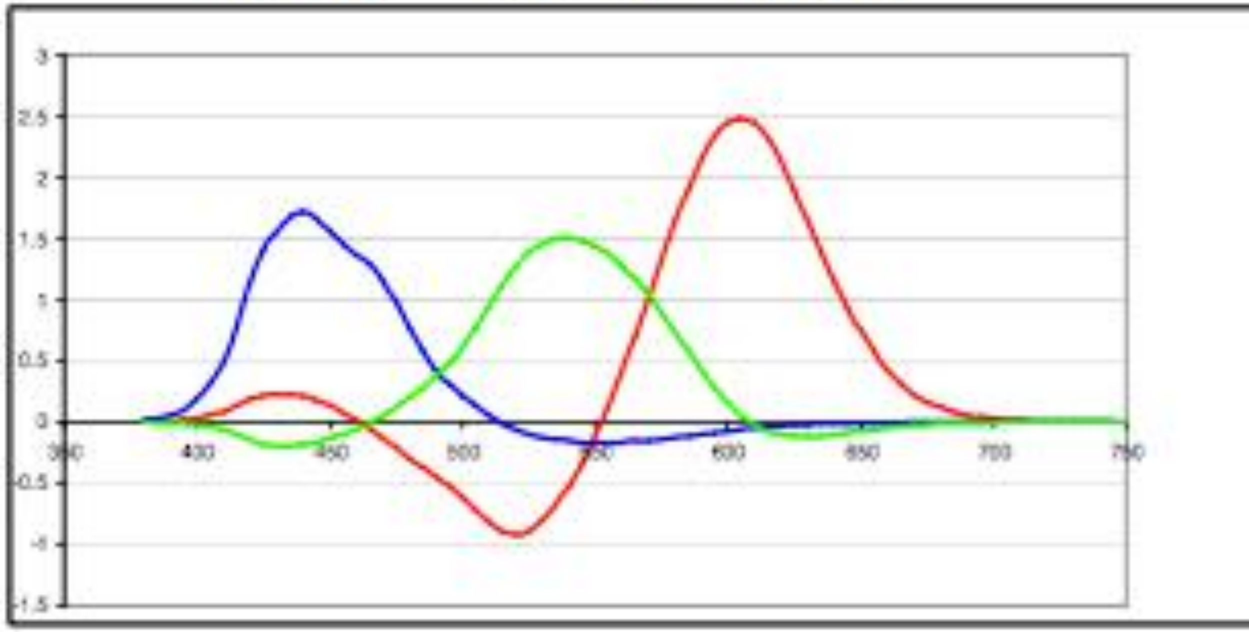
Note the negative values

sRGB color space

While it is called “standard”, when you grab an “RGB” image, it is highly likely it is in a different RGB color space…

The Standard RGB (sRGB) color space

- Derived by Microsoft and HP in 1996, based on CRT displays used at the time.
- Similar but not equivalent to CIE RGB.



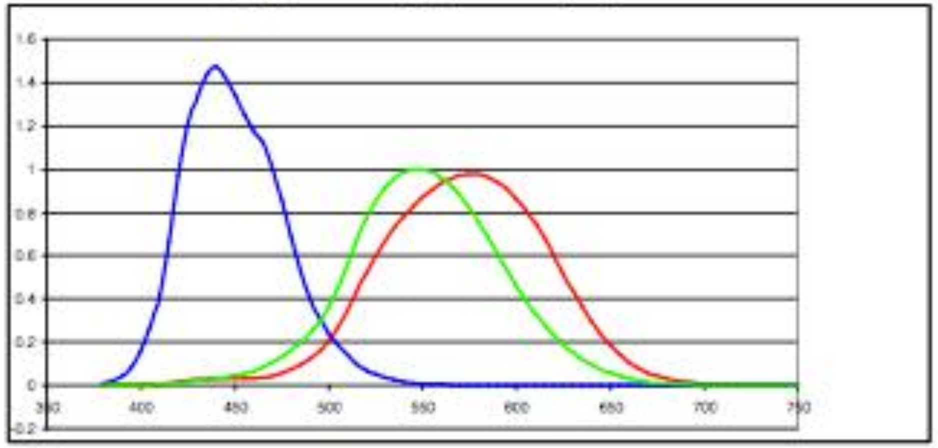
sRGB color space

There are really two kinds of sRGB color spaces: linear and non-linear.

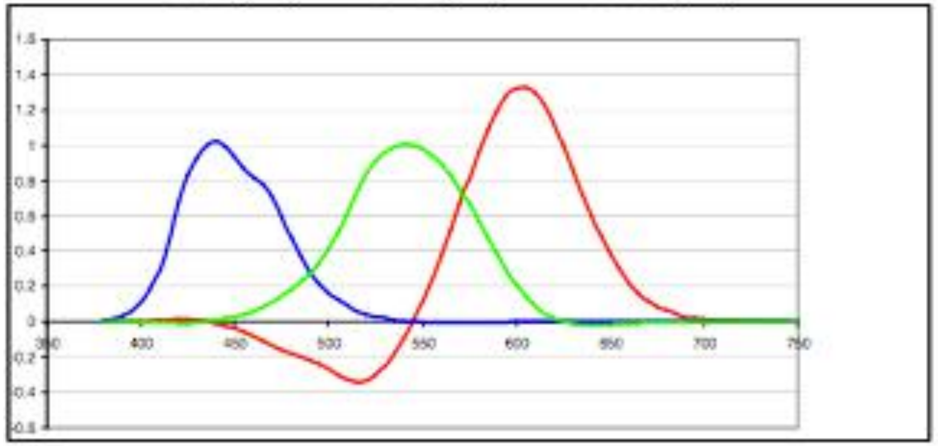
- Non-linear sRGB images have the following tone reproduction curve applied to them.

$$C_{\text{non-linear}} = \begin{cases} 12.92 \cdot C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1+0.055) \cdot C_{\text{linear}}^{\frac{1}{2.4}} - 0.055, & C_{\text{linear}} \geq 0.0031308 \end{cases}$$

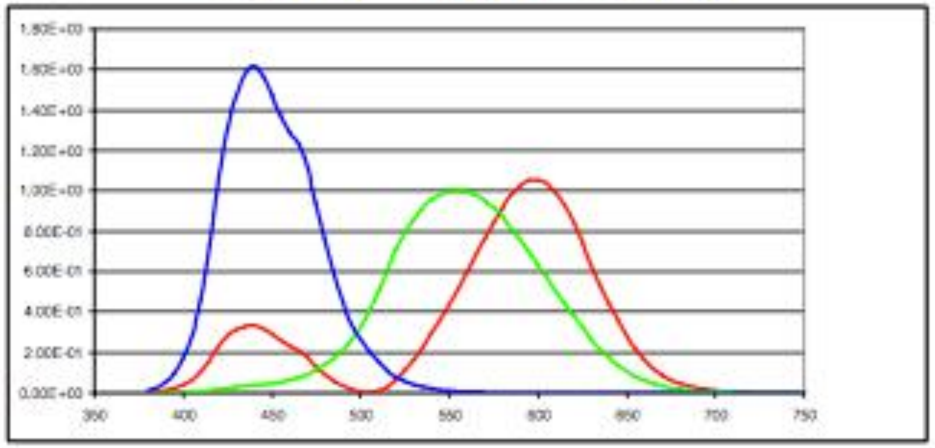
A few important color spaces



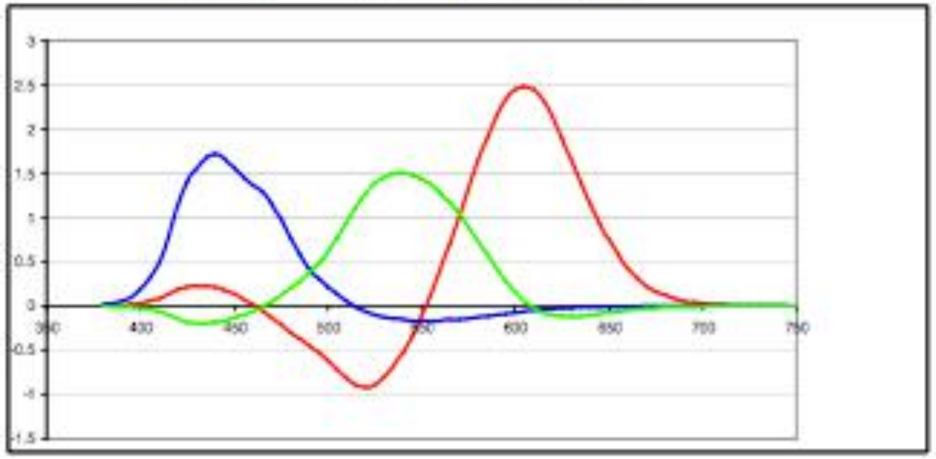
LMS color space



CIE RGB color space

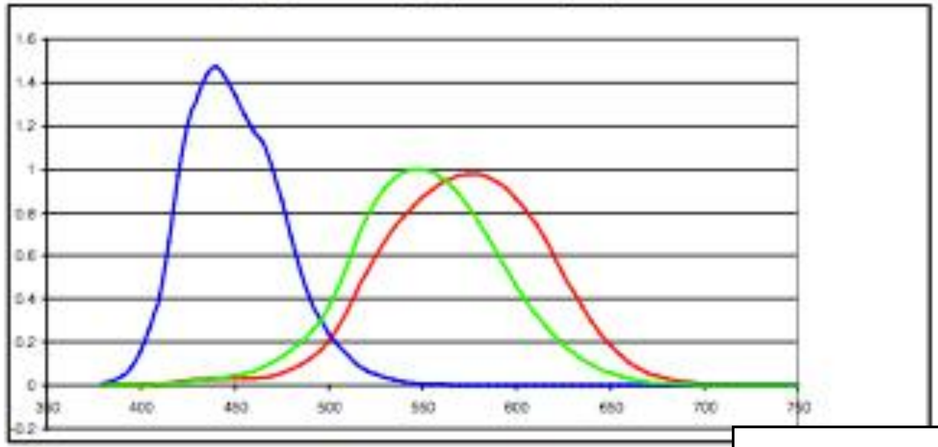


CIE XYZ color space



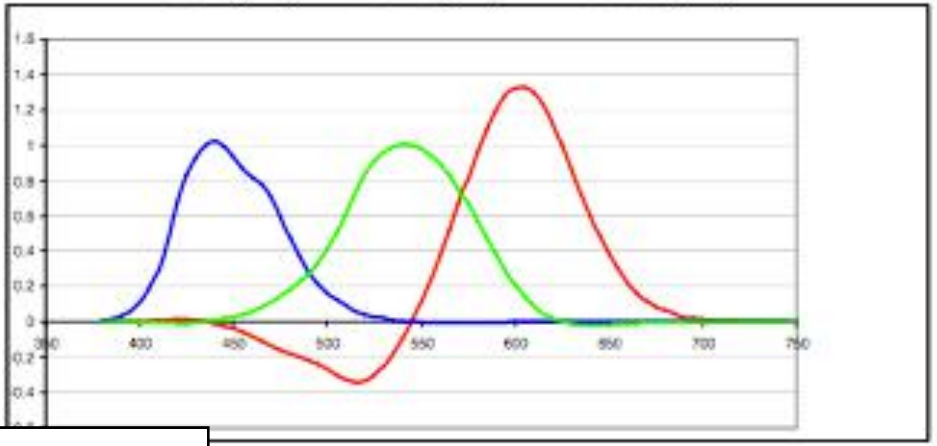
sRGB color space

A few important color spaces

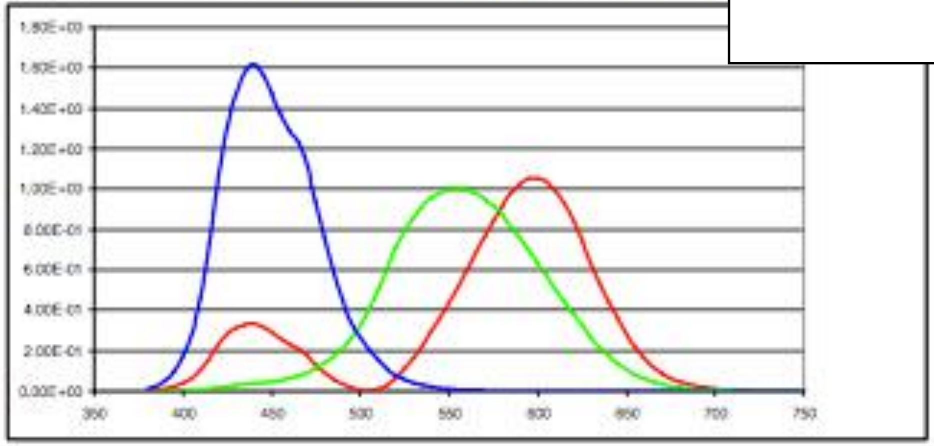


LMS color space

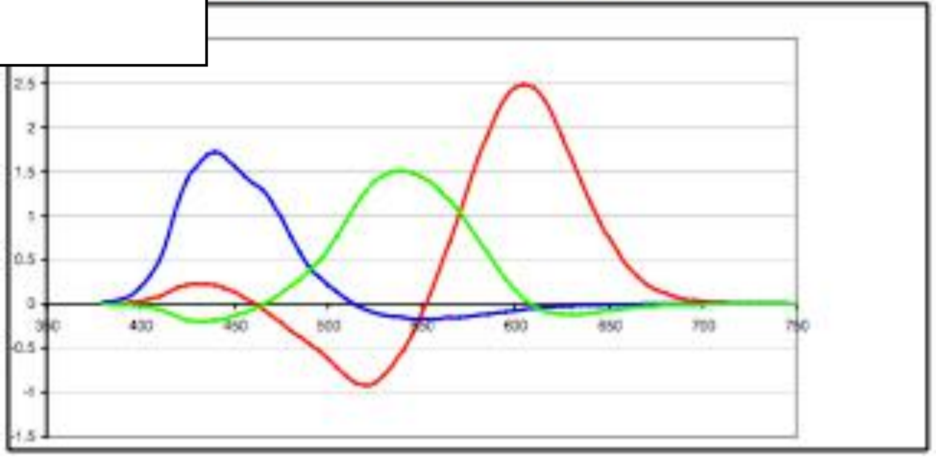
Is there a way to "compare" all these color spaces?



CIE RGB color space



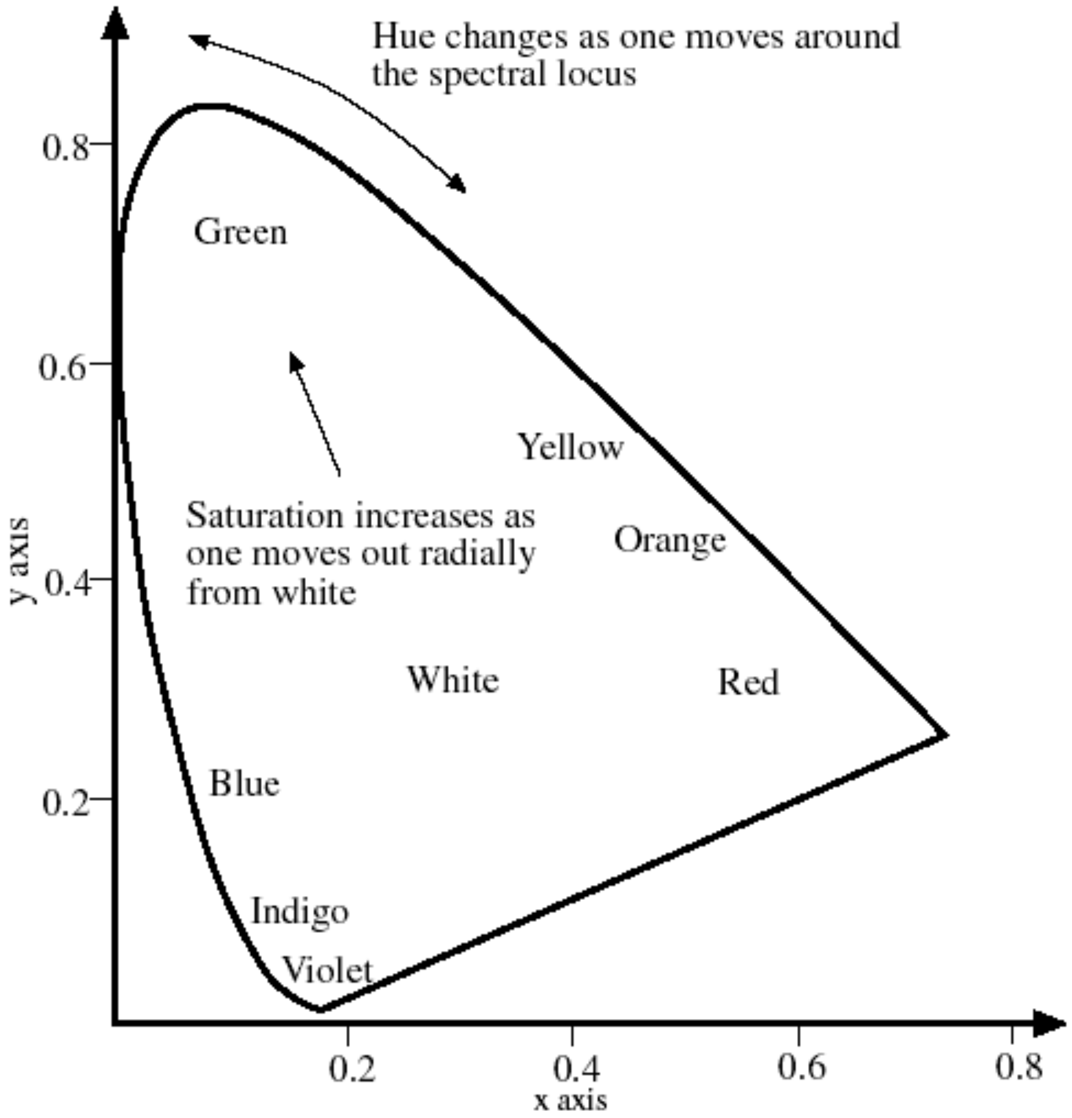
CIE XYZ color space



sRGB color space

Chromaticity

CIE xy (chromaticity)



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

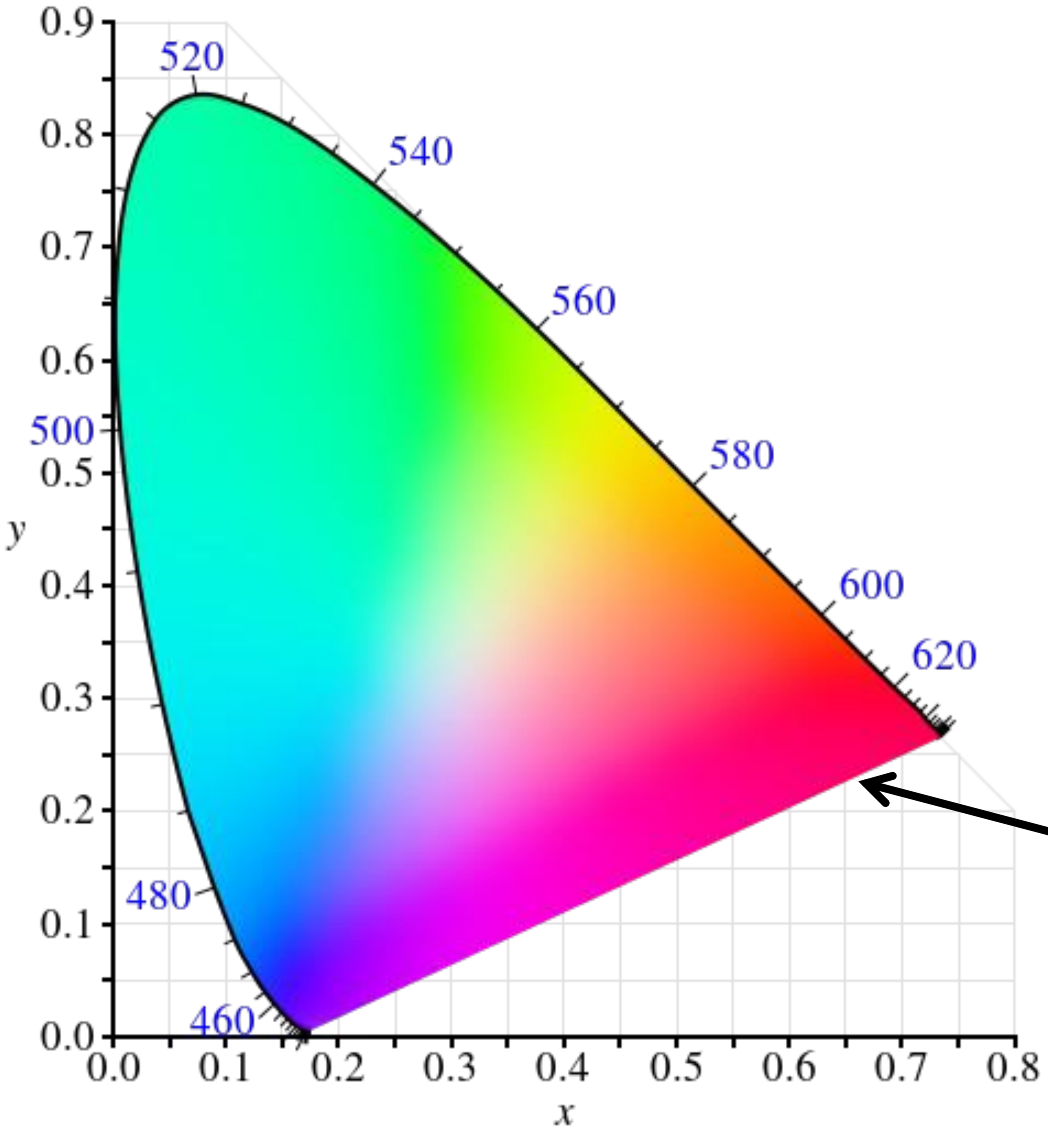
$$y = \frac{Y}{X+Y+Z}$$

$$(X, Y, Z) \longleftrightarrow (\underbrace{x, y}_{\text{chromaticity}}, Y)$$

luminance/brightness

Perspective projection of 3D retinal color space to two dimensions.

CIE xy (chromaticity)



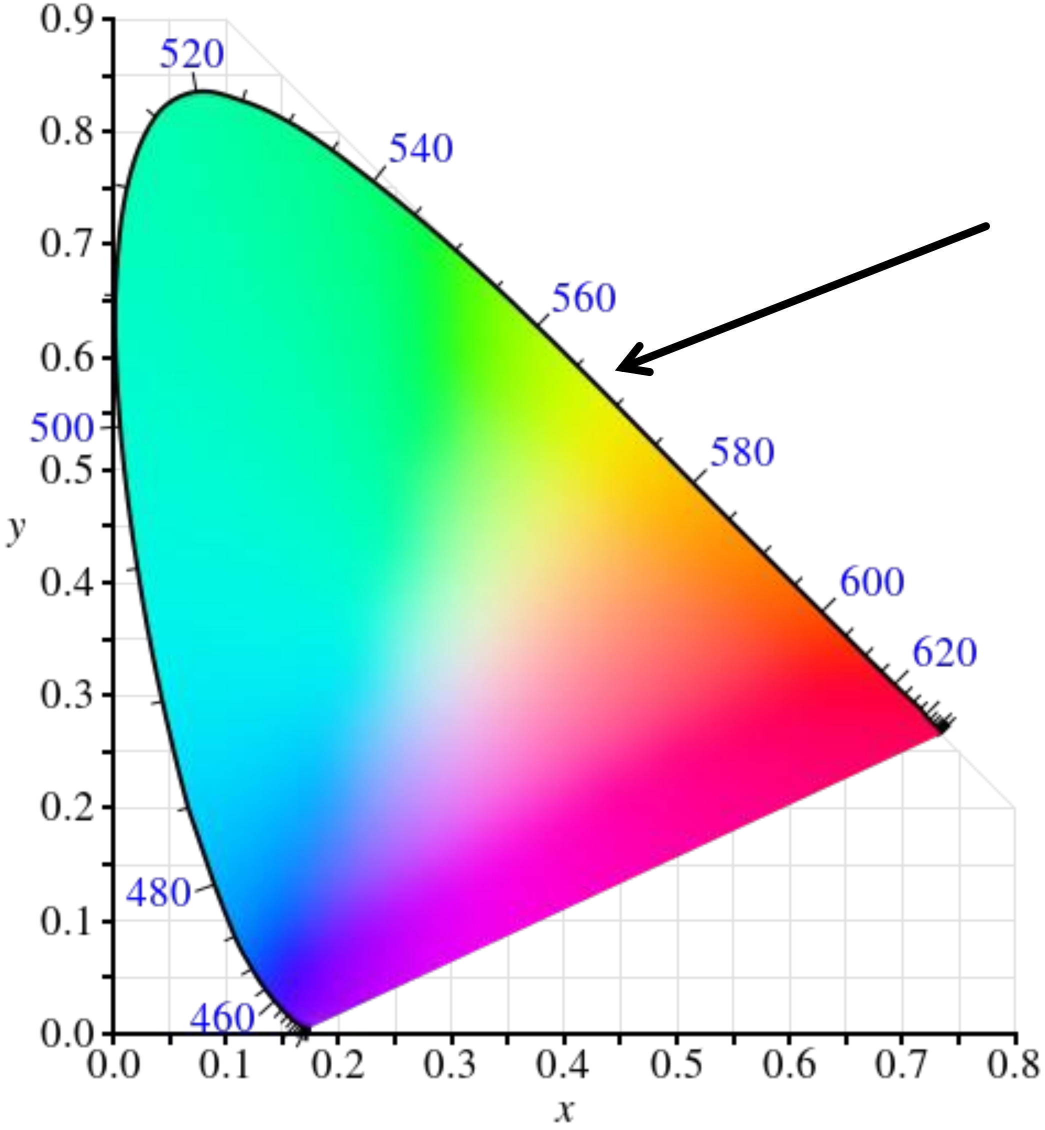
$$x = \frac{X}{X+Y+Z}$$

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

$$(X, Y, Z) \longleftrightarrow (x, y, Y)$$

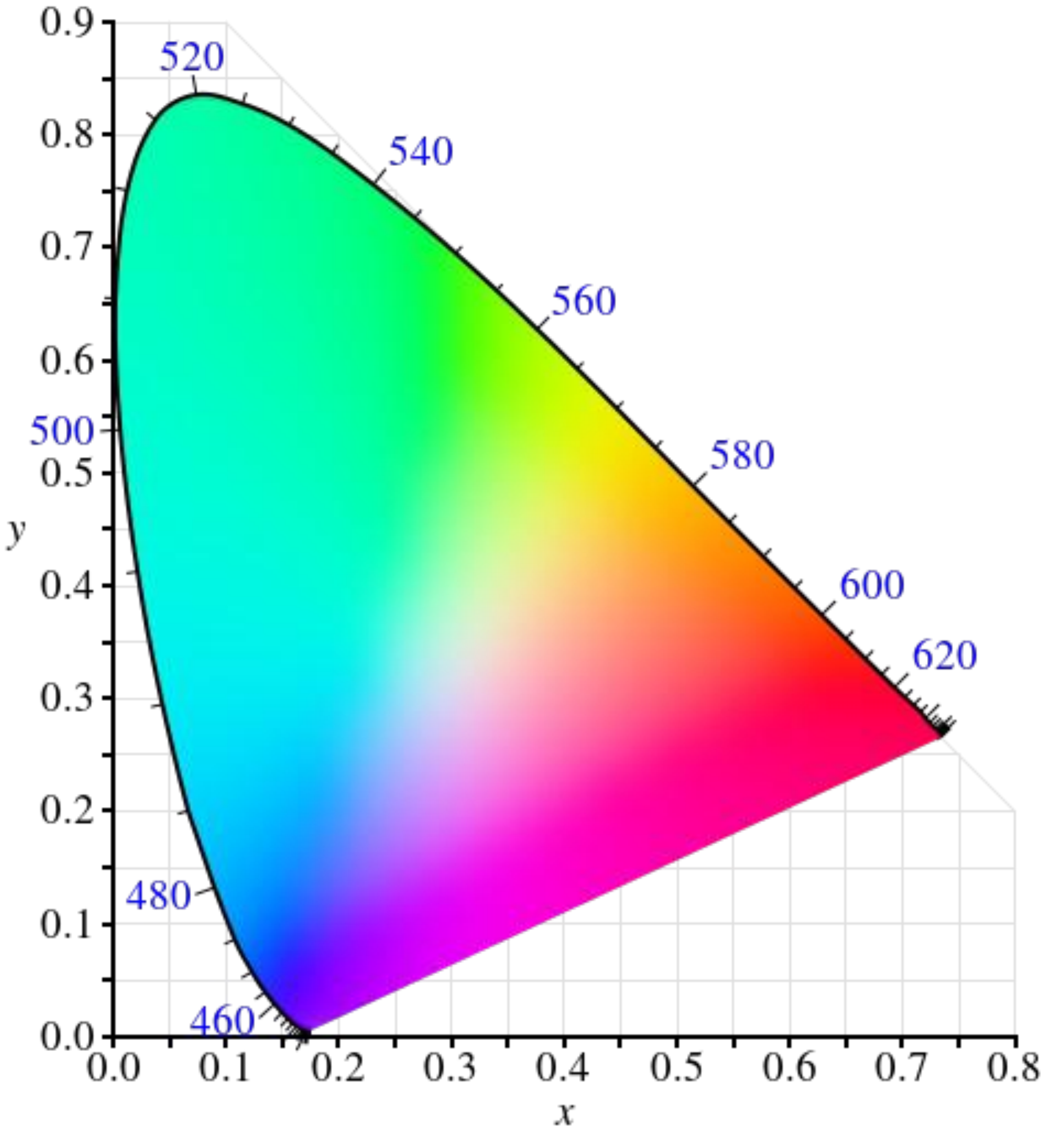
Note: These colors can be extremely misleading depending on the file origin and the display you are using

CIE xy (chromaticity)



What does the boundary of the chromaticity diagram correspond to?

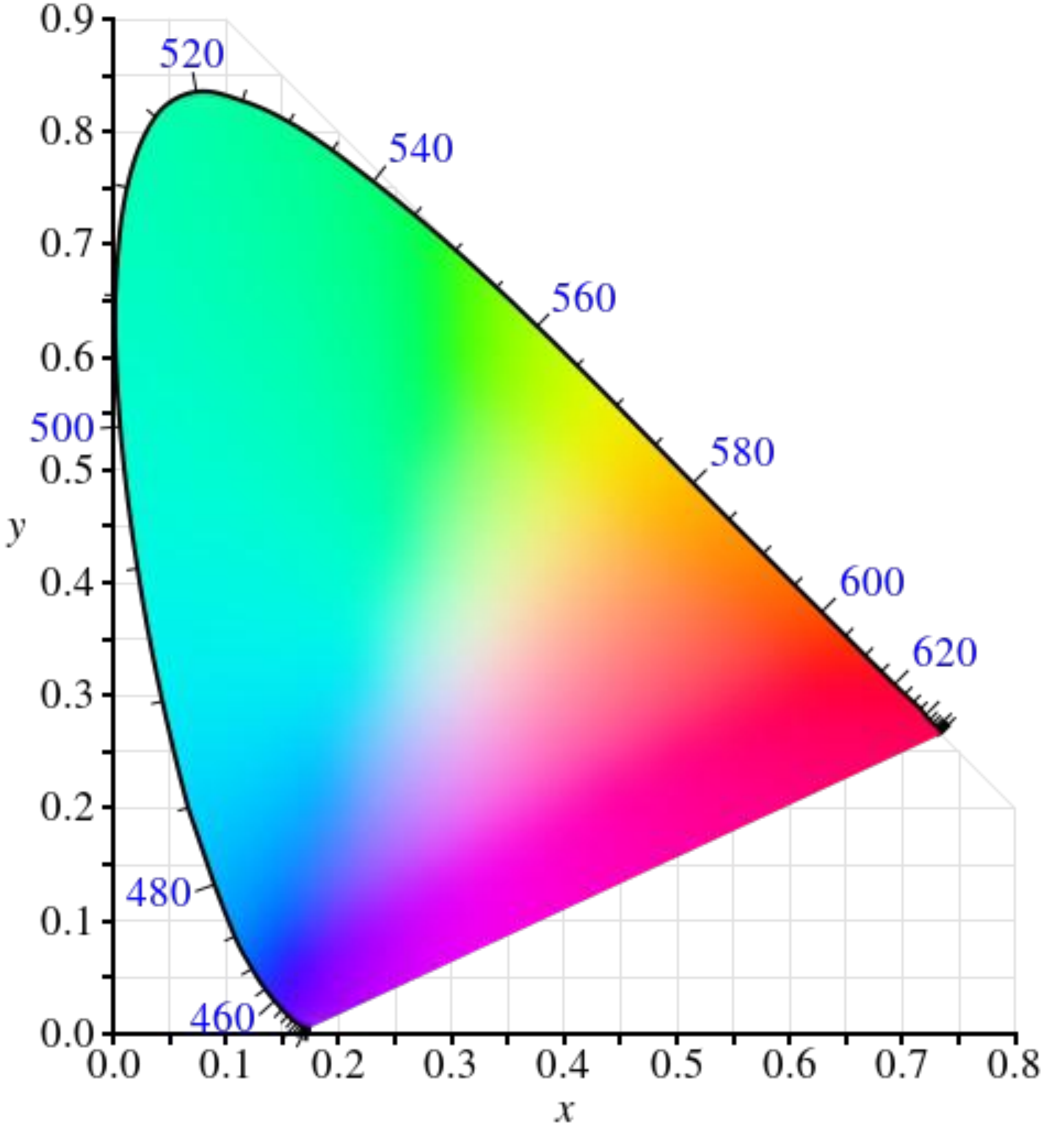
Color gamuts



We can compare color spaces by looking at what parts of the chromaticity space they can reproduce with their primaries.

But why would a color space not be able to reproduce all of the chromaticity space?

Color gamuts

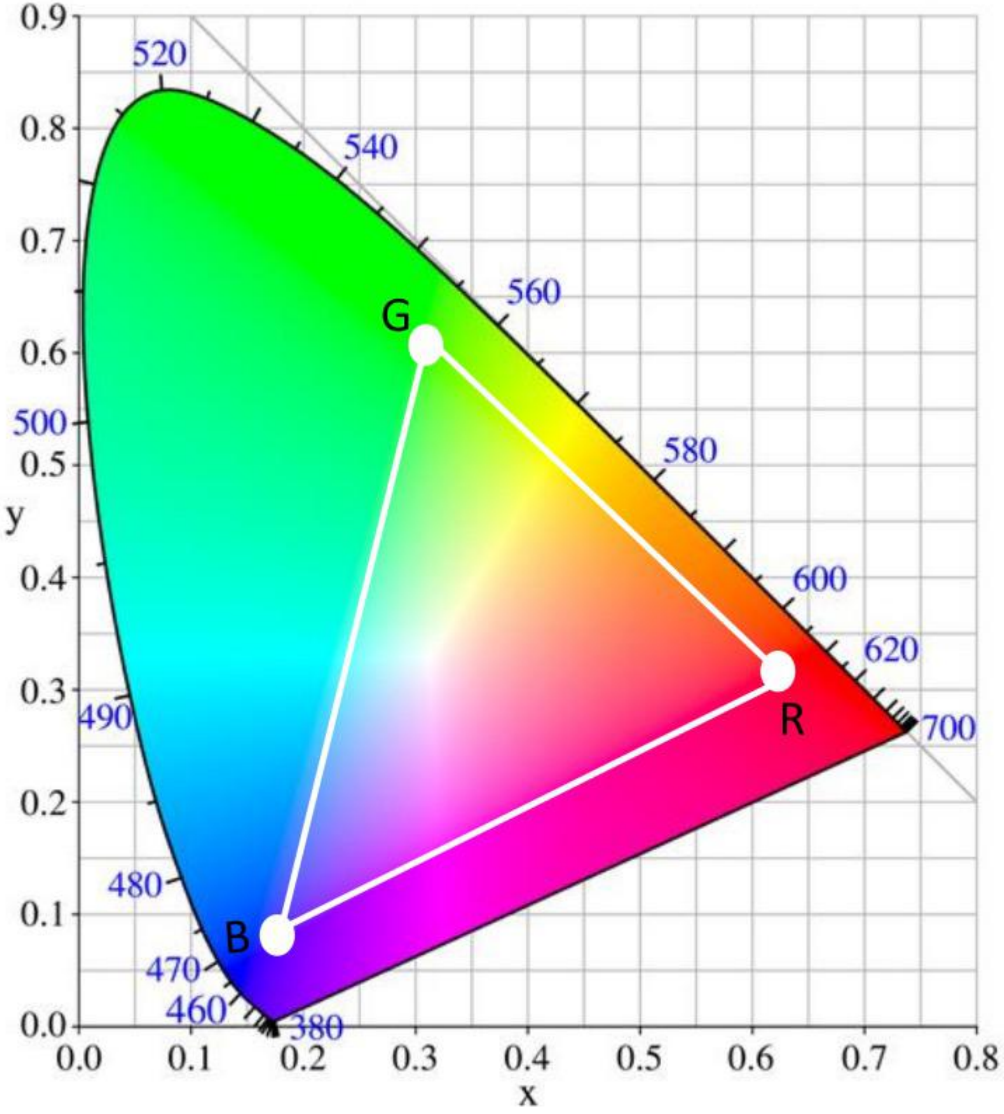


We can compare color spaces by looking at what parts of the chromaticity space they can reproduce with their primaries.

But why would a color space not be able to reproduce all of the chromaticity space?

- Many colors require negative weights to be reproduced, which are not realizable.

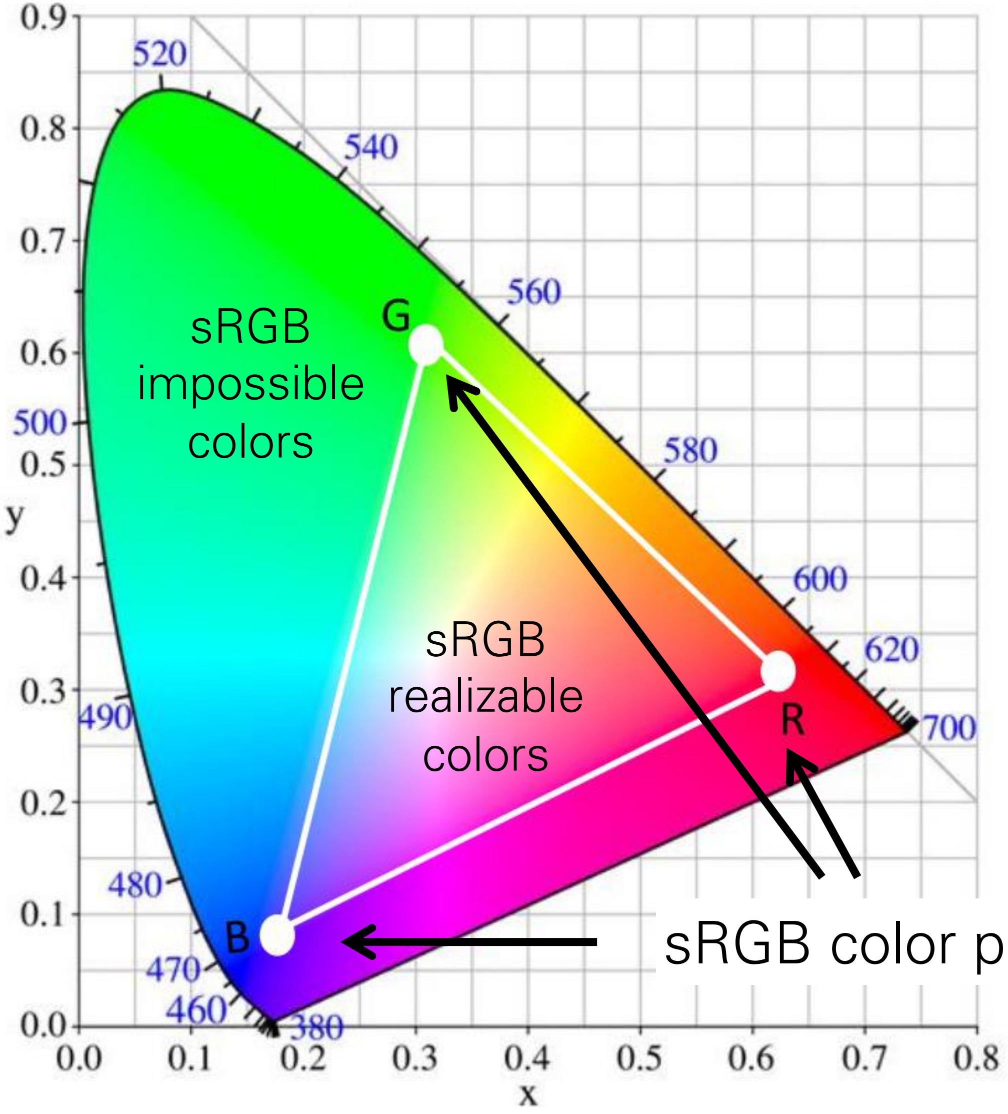
Color gamuts



sRGB color gamut:

- What are the three triangle corners?
- What is the interior of the triangle?
- What is the exterior of the triangle?

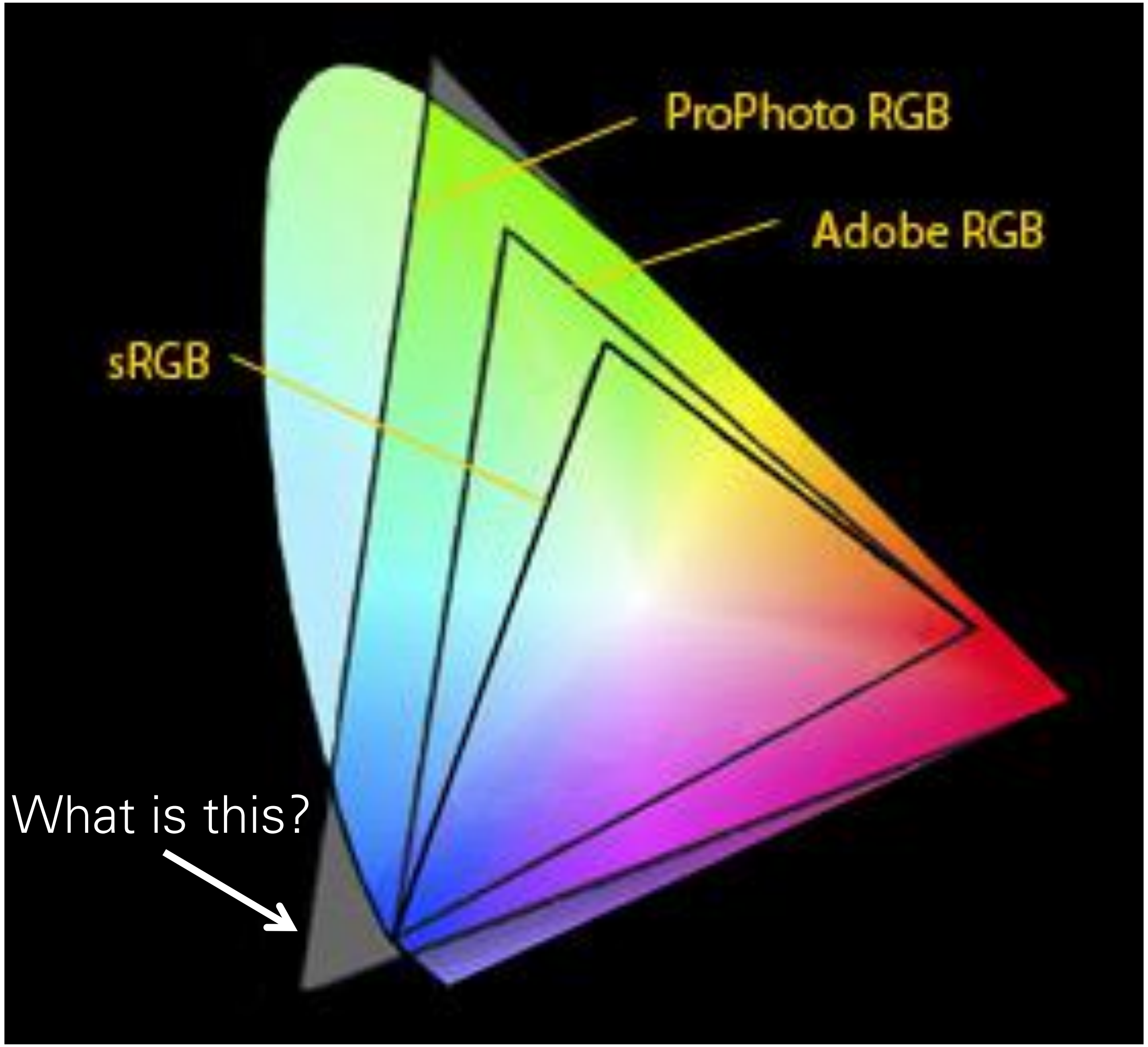
Color gamuts



sRGB color gamut

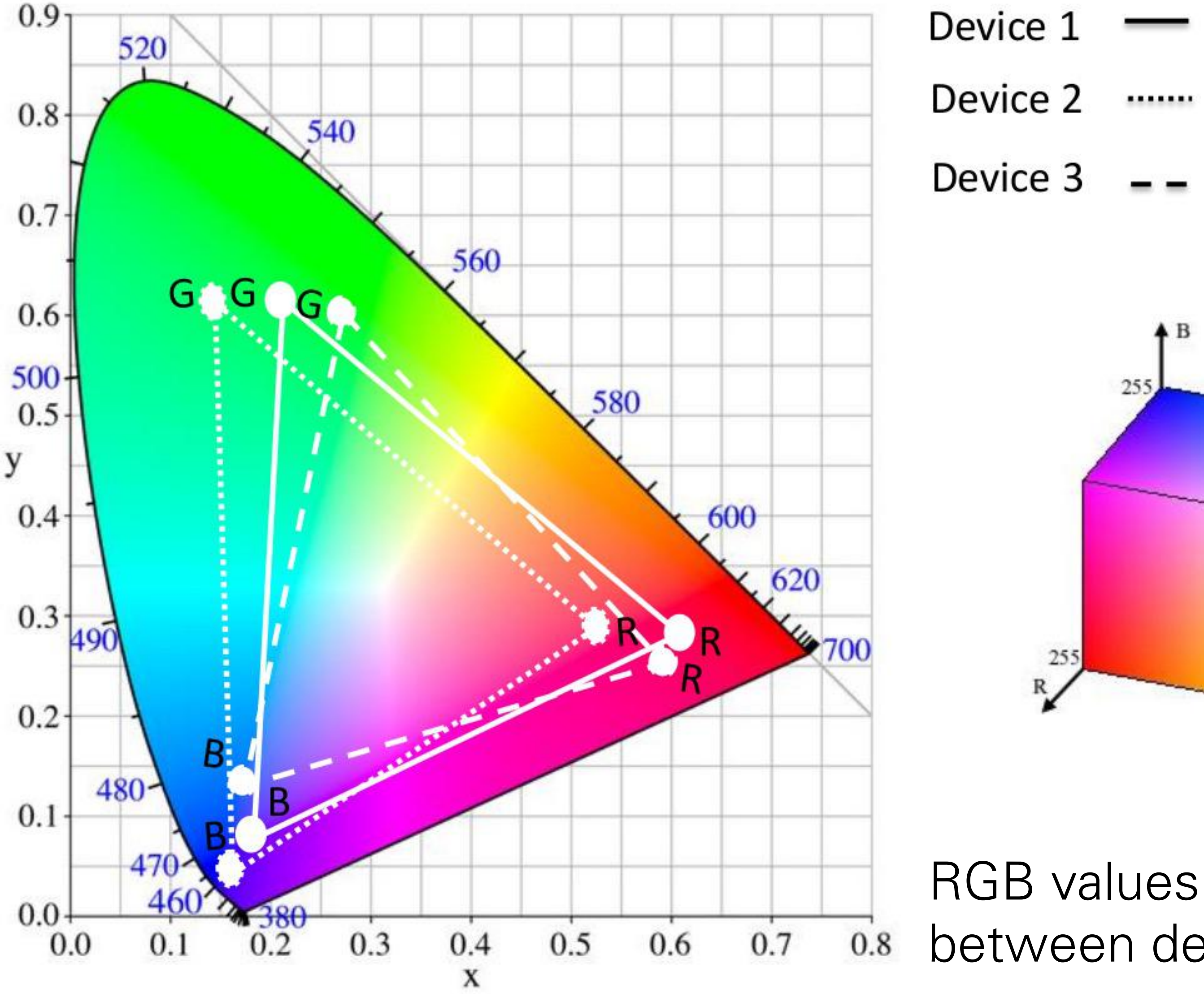
sRGB color primaries

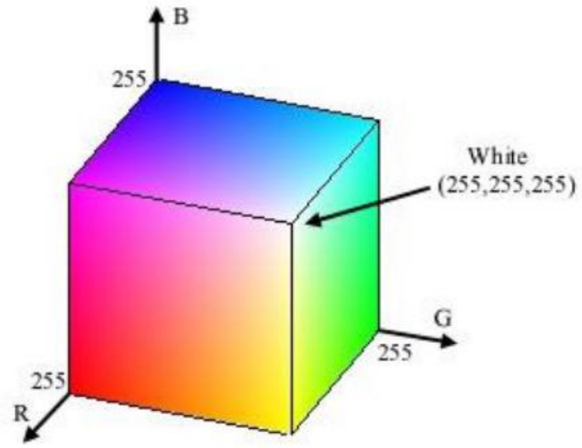
Color gamuts



Gamuts of various common industrial RGB spaces

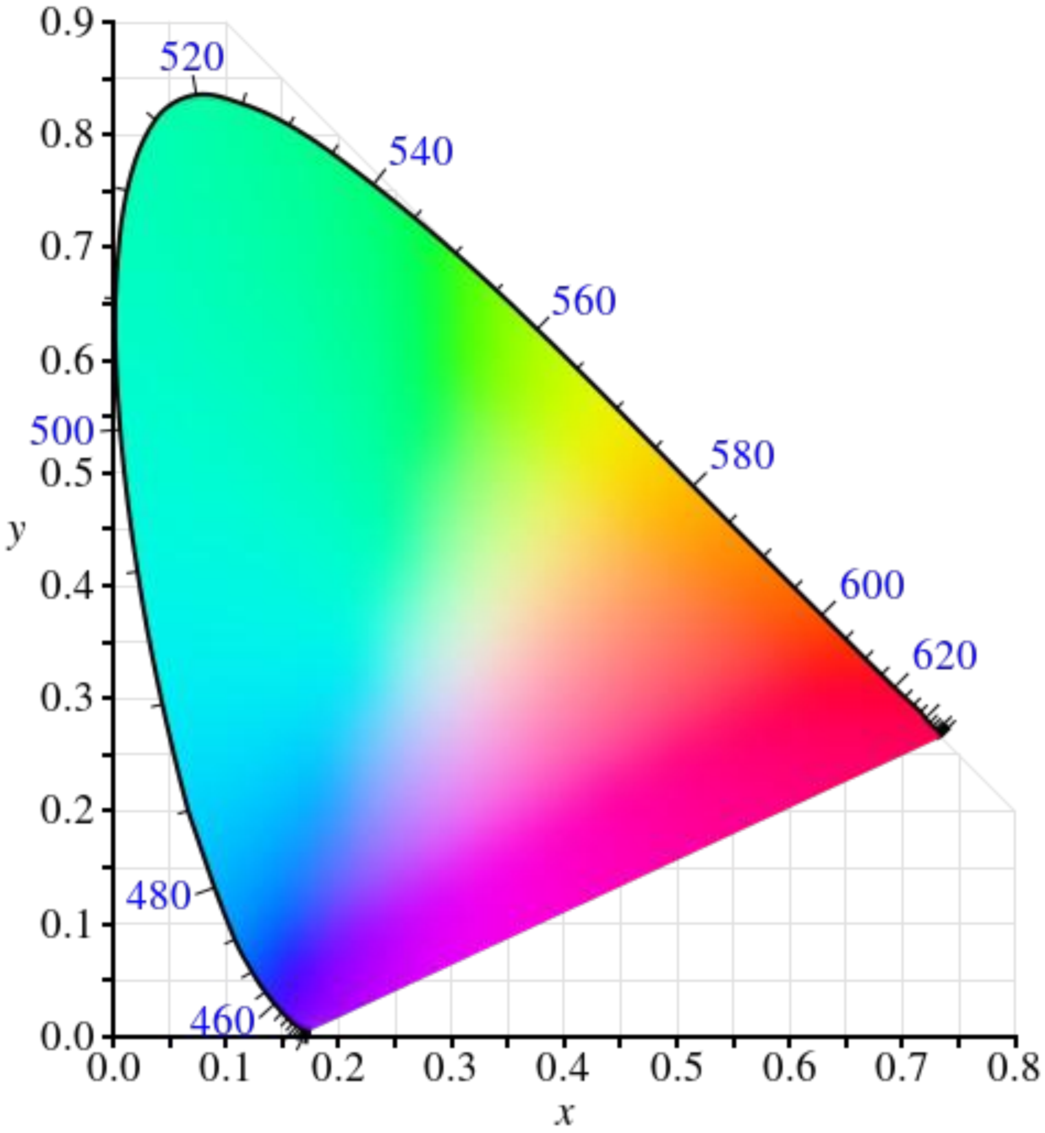
The problem with RGBs visualized in chromaticity space





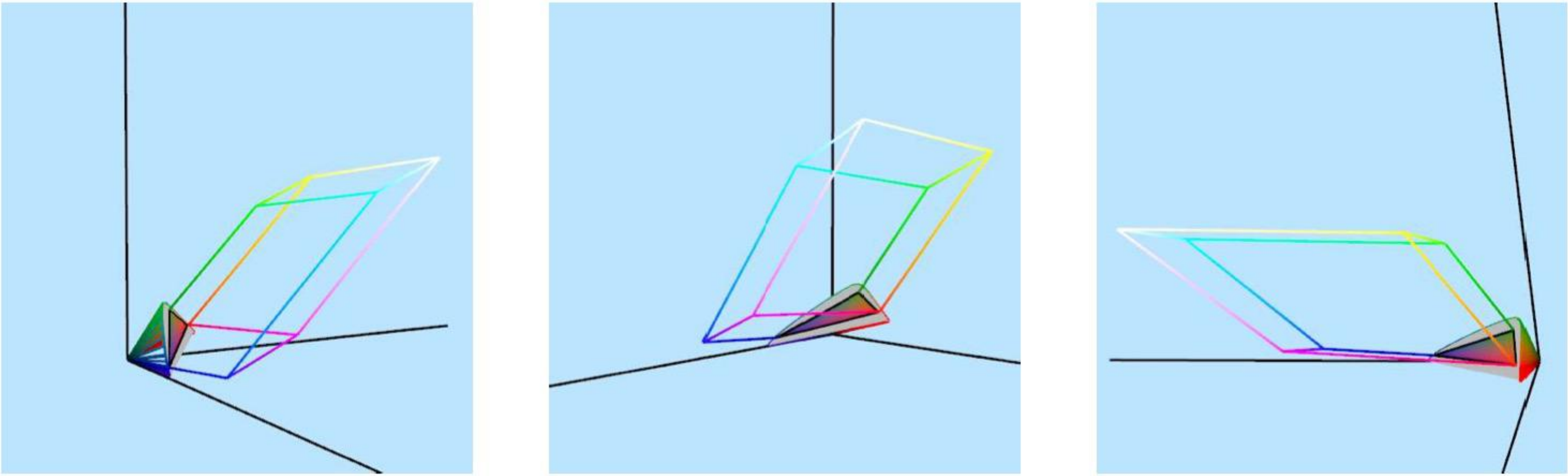
RGB values have no meaning if the primaries between devices are not the same!

Color gamuts



- Can we create an RGB color space that reproduces the entire chromaticity diagram?
- What would be the pros and cons of such a color space?
- What devices would you use it for?

Chromaticity diagrams can be misleading



Different gamuts may compare very differently when seen in full 3D retinal color space.

Some take-home messages about color spaces

Analytic: Retinal color is three numbers formed by taking the dot product of a power spectral distribution with three color matching/sensitivity functions.

Synthetic: Retinal color is three numbers formed by assigning weights to three color primaries to match the perception of a power spectral distribution.

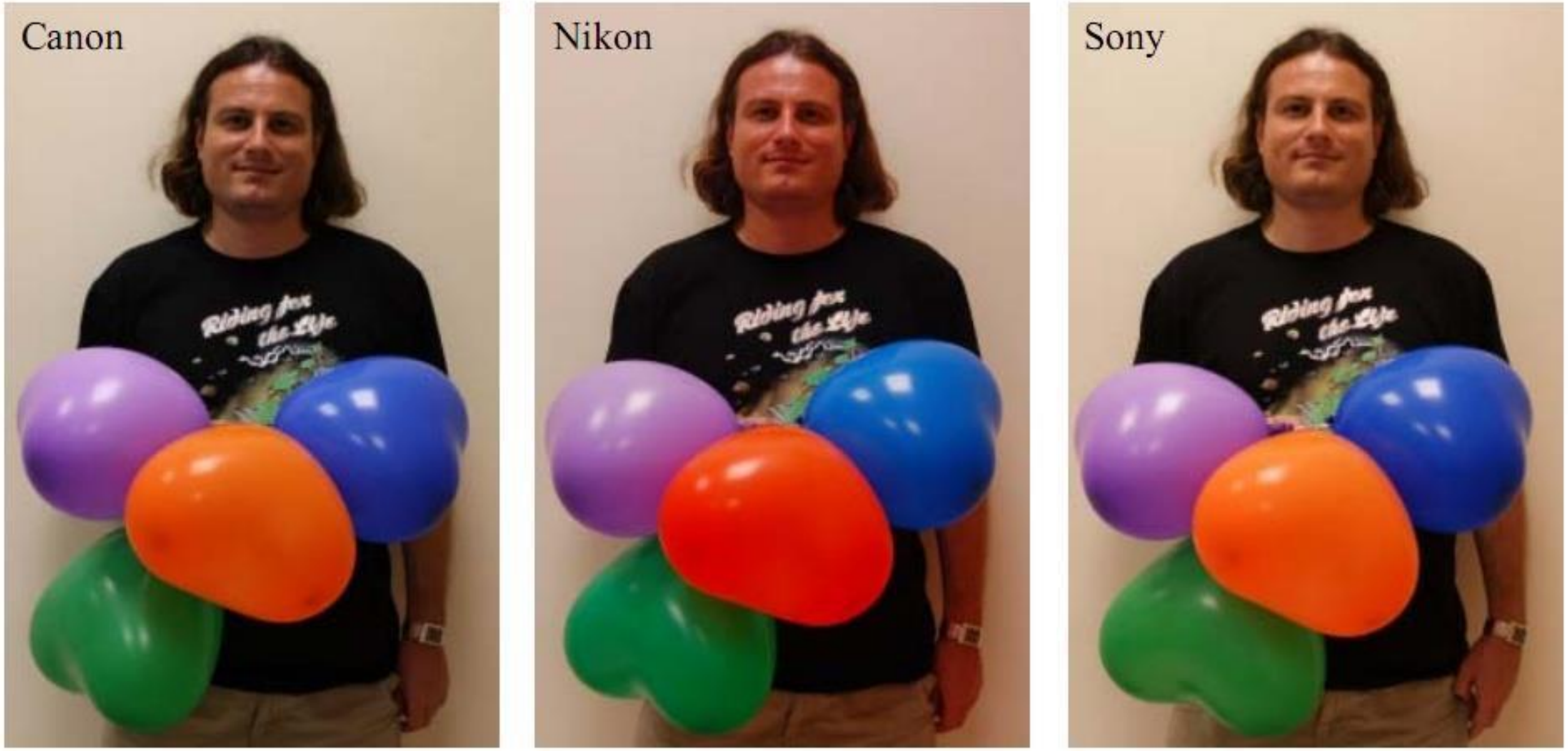
Fundamental problem: Analysis spectrum (camera, eyes) cannot be the same as synthesis one (display) - impossible to encode all possible colors without something becoming negative

- CIE XYZ only needs positive coordinates, but need primaries with negative light.
- RGB must use physical (non-negative) primaries, but needs negative coordinates for some colors.

Problem with current practice: Many different RGB color spaces used by different devices, without clarity of what exactly space a set of RGB color values are in.

- Huge problem for color reproduction from one device to another.

See for yourself



Images of the same scene captured using 3 different cameras with identical settings, supposedly in sRGB space.

Color calibration and affine transform estimation

Color calibration

Apply linear scaling and translation to RGB vectors in the image:

$$c' = M \cdot c + t$$

transformed RGB vector

original RGB vector

What are the dimensions of each quantity in this equation?

Color calibration

Apply linear scaling and translation to RGB vectors in the image:

$$c' = M \cdot c + t$$

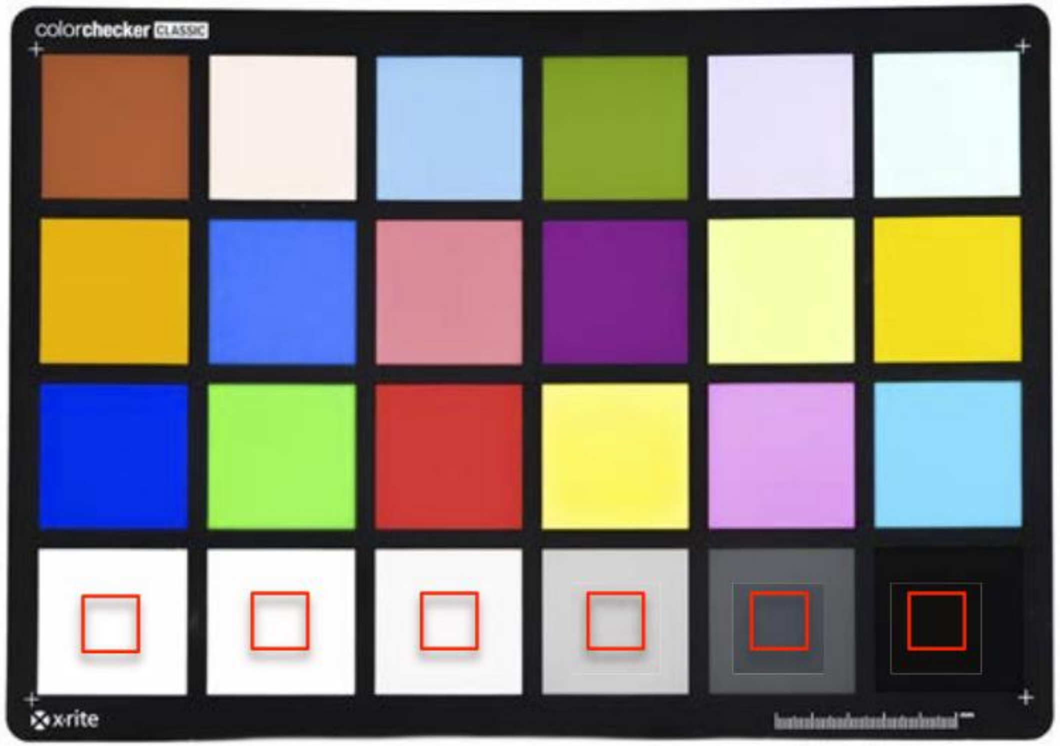
transformed RGB vector

original RGB vector

What are the dimensions of each quantity in this equation?

How do we decide what transformed vectors to map to?

Using (again) a colorchecker

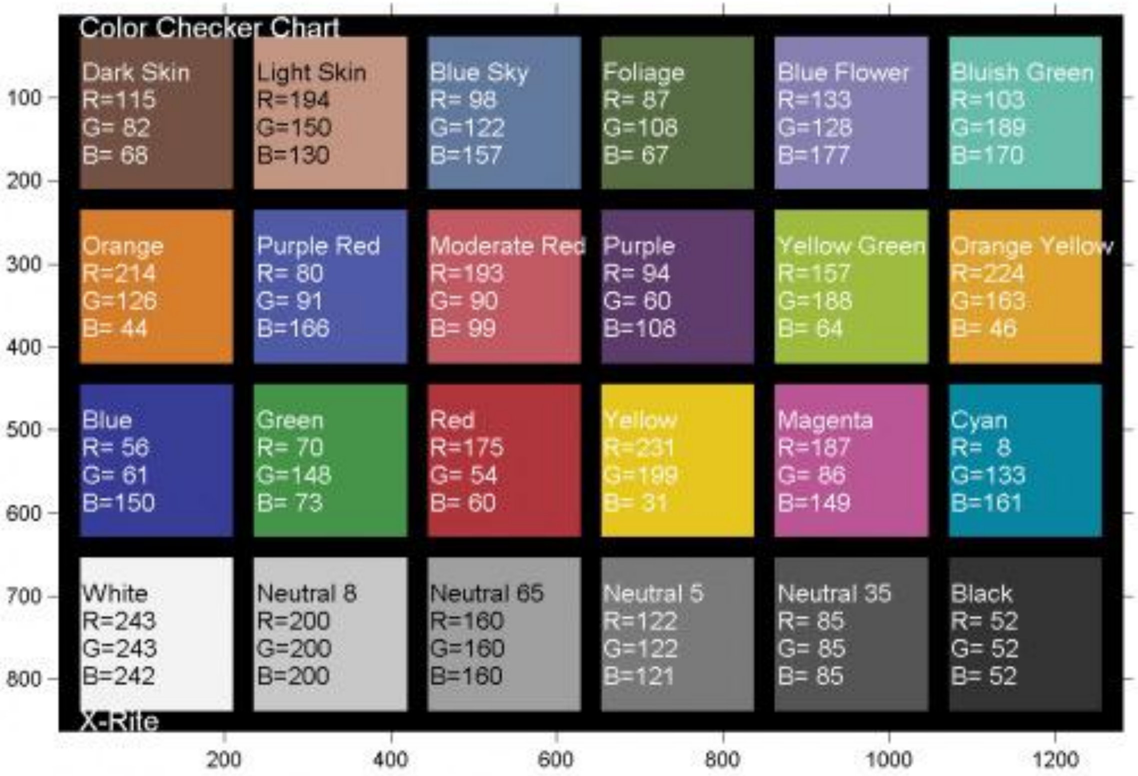


Color patches manufactured to have pre-calibrated XYZ coordinates.

Calibration chart can be used for:

1. color calibration
2. radiometric calibration (i.e., response curve) using the bottom row

Using (again) a colorchecker



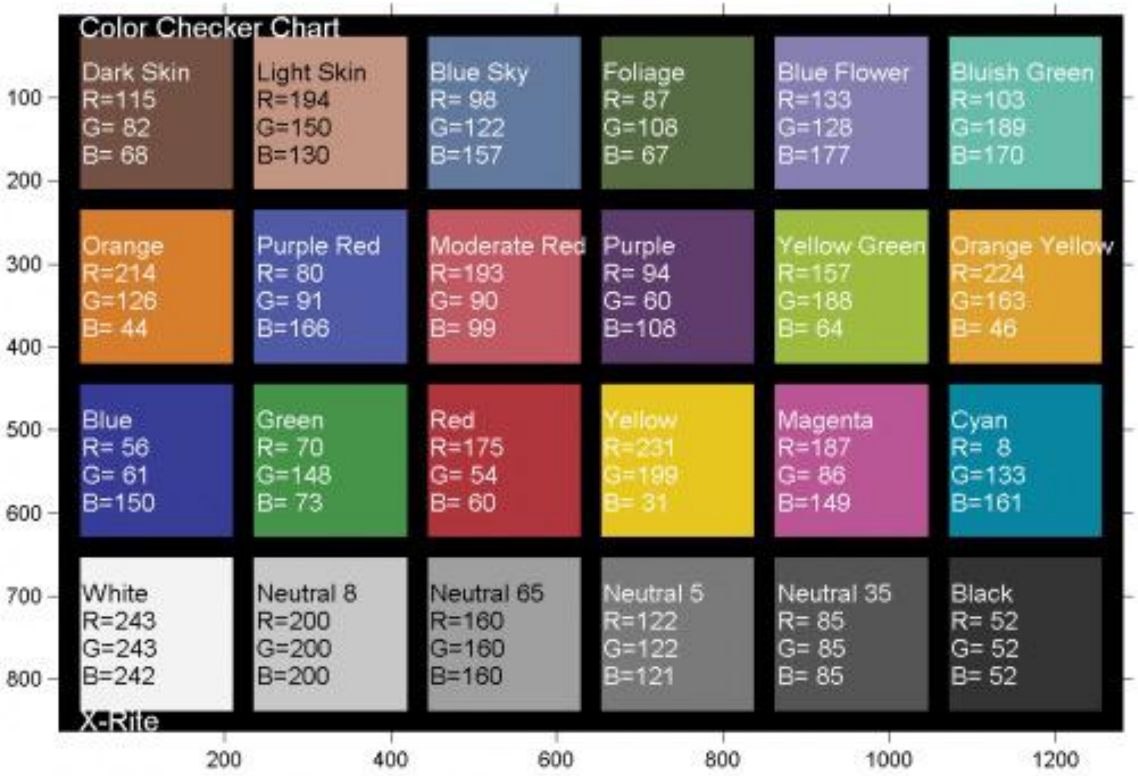
Color patches manufactured to have pre-calibrated XYZ coordinates.

Can we use any colorchecker image for color calibration?

Calibration chart can be used for:

1. color calibration
2. radiometric calibration (i.e., response curve) using the bottom row

Using (again) a colorchecker



Color patches manufactured to have pre-calibrated XYZ coordinates.

Can we use any colorchecker image for color calibration?

- It needs to be a linear image!
- Do radiometric calibration first.

Calibration chart can be used for:

1. color calibration
2. radiometric calibration (i.e., response curve) using the bottom row

Color calibration

Apply linear scaling and translation to RGB vectors in the image:

$$c' = M \cdot c + t$$

transformed RGB vector

original RGB vector

What are the dimensions of each quantity in this equation?

How do we decide what transformed vectors to map to?

How do we solve for matrix M and vector t?

Color calibration

Apply linear scaling and translation to RGB vectors in the image:

$$c' = \begin{bmatrix} M & t \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix}$$

Color calibration

Apply linear scaling and translation to RGB vectors in the image:

$$c' = \underbrace{[M \quad t]}_{T} \underbrace{\begin{bmatrix} c \\ 1 \end{bmatrix}}_{C}$$

Color calibration

Apply an affine transform to homogeneous RGB vectors in the image:

$$c' = T \cdot C$$

heterogeneous
transformed RGB vector

homogeneous
original RGB vector

How do we solve for an affine transformation?

Determining the affine transform matrix

Write out linear equation for each color vector correspondence:

$$c' = T \cdot C \qquad \text{or} \qquad \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \\ 1 \end{bmatrix}$$

Determining the affine transform matrix

Rearrange into an equation involving a vectorized form of T:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} r & g & b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r & g & b & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r & g & b & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{12} \end{bmatrix}$$

Determining the affine transform matrix

Stack equations from multiple color vector correspondences:

$$\underbrace{\begin{bmatrix} r' \\ g' \\ b' \\ \vdots \\ r' \\ g' \\ b' \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} r & g & b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r & g & b & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r & g & b & 1 \\ & & & & & & \vdots & & & & & \\ r & g & b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r & g & b & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r & g & b & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{12} \end{bmatrix}}_{x}$$

Solving the linear system

Convert the system to a linear least-squares problem:

$$E_{\mathrm{LLS}} = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$$

Expand the error:

$$E_{\mathrm{LLS}} = \boldsymbol{x}^\top(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} - 2\boldsymbol{x}^\top(\mathbf{A}^\top\boldsymbol{b}) + \|\boldsymbol{b}\|^2$$

Minimize the error:

Set derivative to 0 $(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} = \mathbf{A}^\top\boldsymbol{b}$

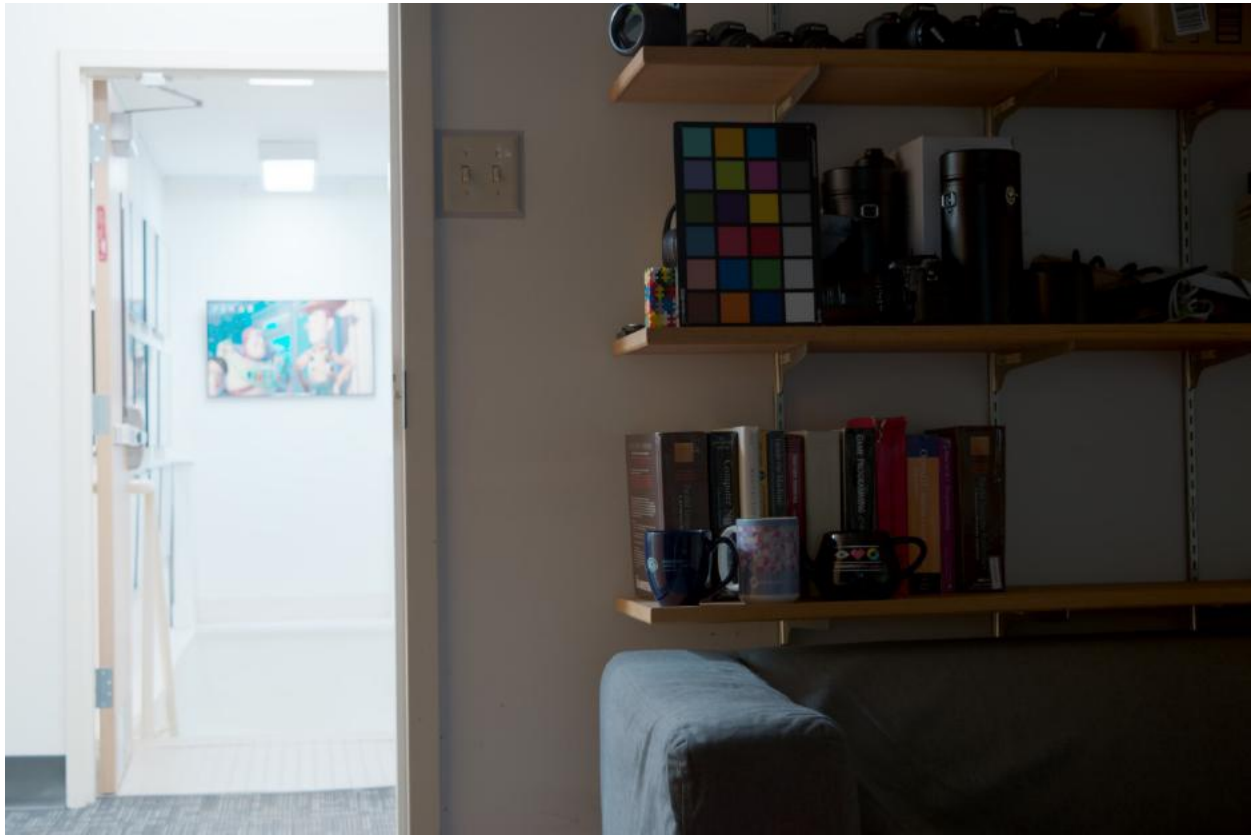
Solve for x $\boldsymbol{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{b}$ ⟵ Note: You almost <u>never</u> want to compute the inverse of a matrix.

In Matlab:

```
x = A \ b
```

An example

original

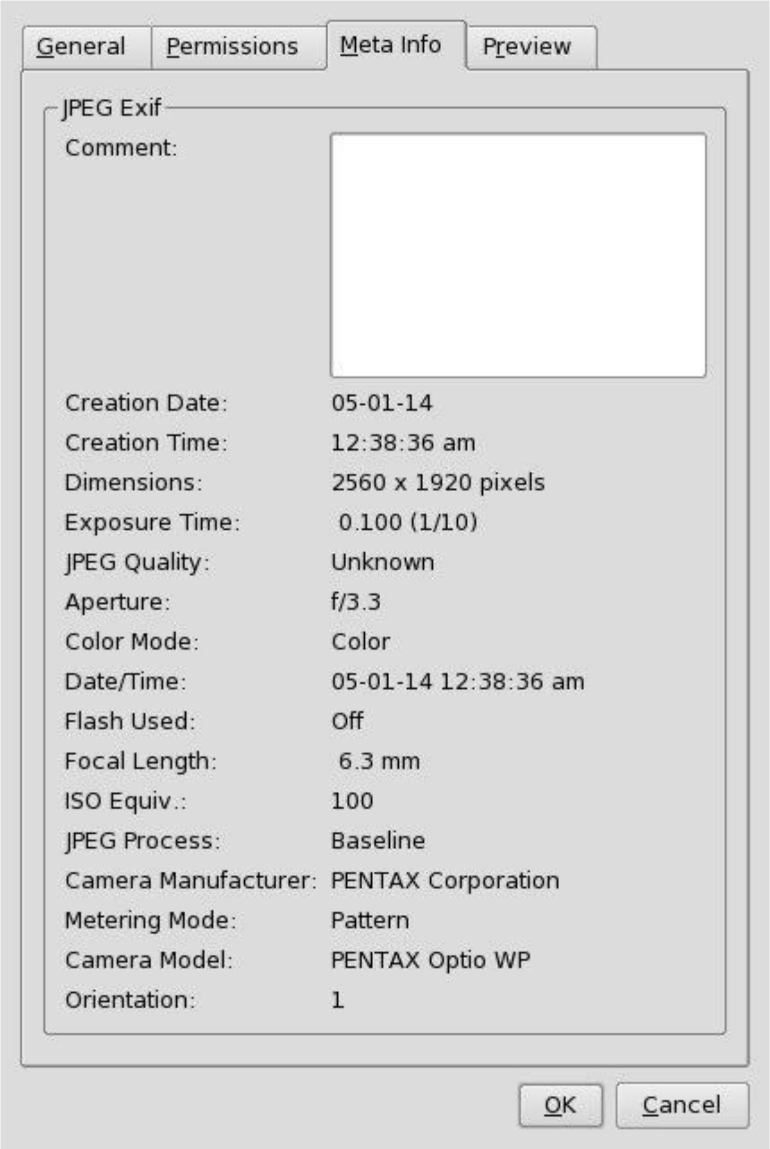


color-corrected

Quick note

If you cannot do calibration, take a look at the image’s EXIF data (if available).

Often contains information about tone reproduction curve and color space.



Color profiling for displays

colorimeter: device calibrated to measure displayed radiance in some reference color space (usually CIE XYZ)

program displaying multiple color patches with known coordinates in the same color space as the colorimeter

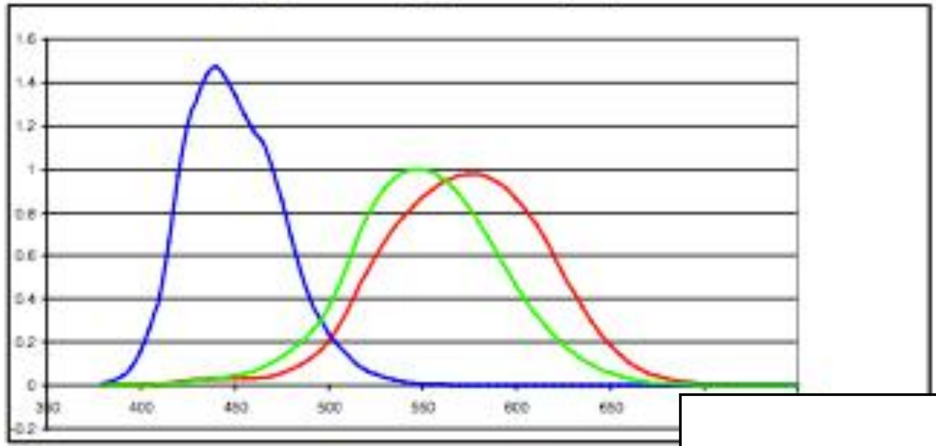
Exactly analogous procedure for figuring out the color space of a display.

Note: In displays, color calibration refers to changing the display’s primaries so that colors are shown differently. This is a completely separate procedure from color profiling.

Note also the discrepancy in terminology between cameras and displays.

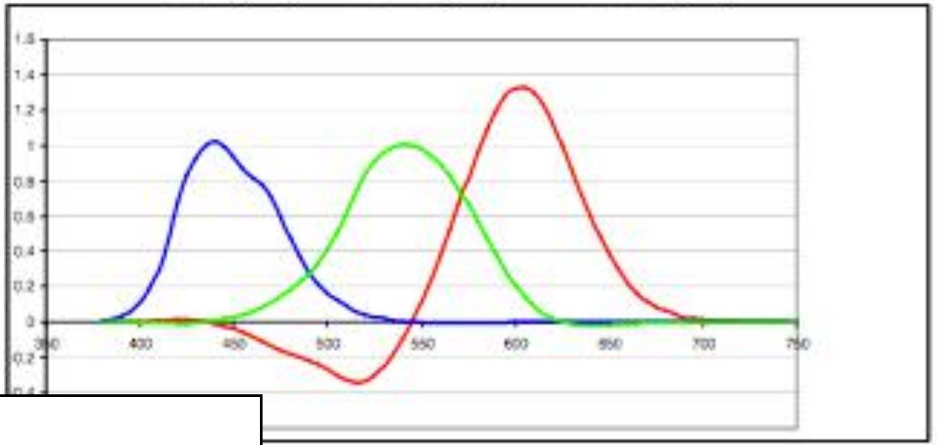
Non-linear color spaces

A few important linear color spaces

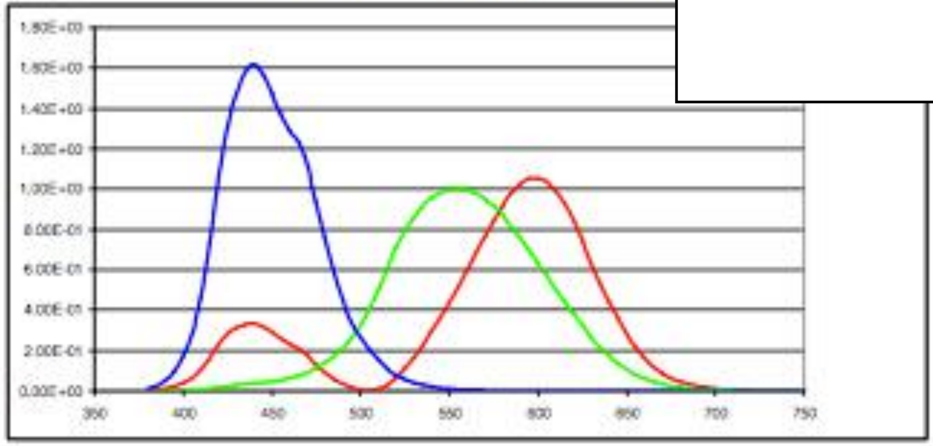


LMS color space

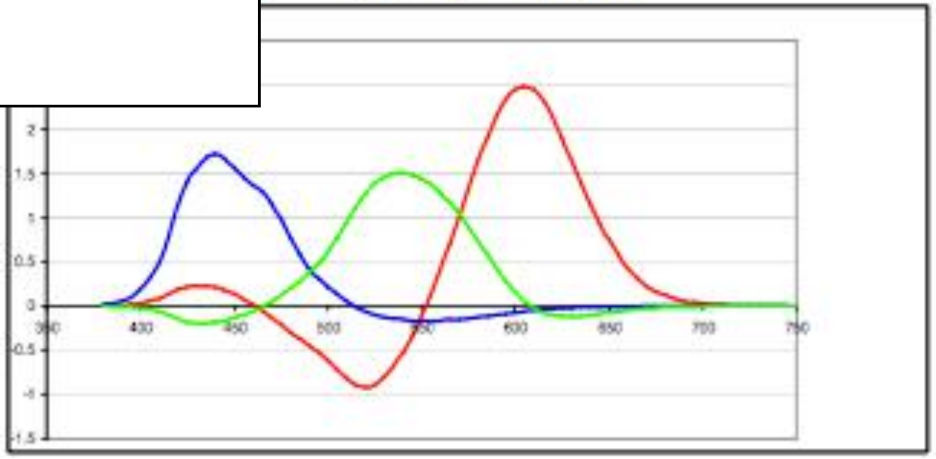
What about non-linear color spaces?



E RGB color space

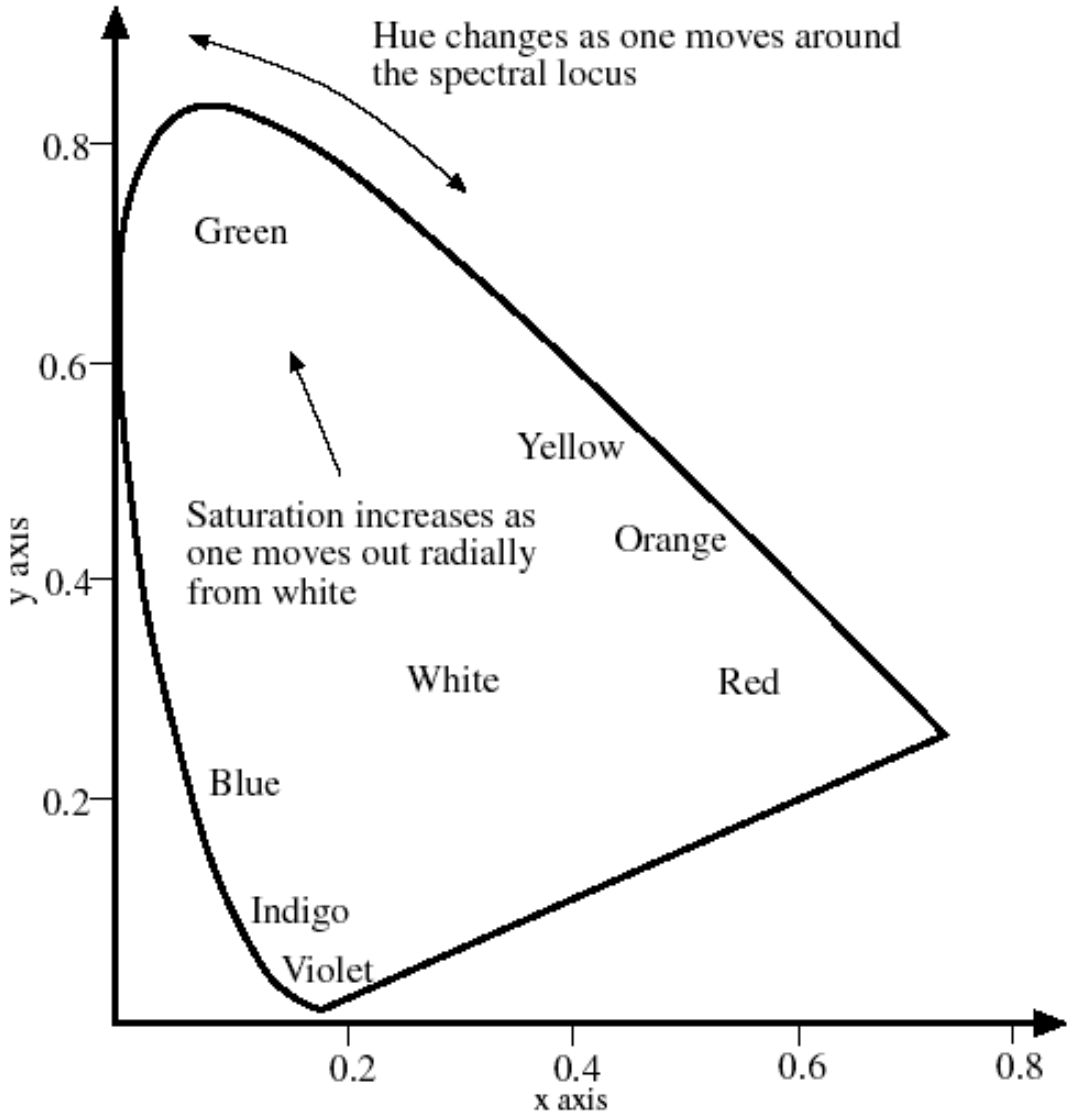


CIE XYZ color space



sRGB color space

CIE xy (chromaticity)



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

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

$$(X, Y, Z) \longleftrightarrow (\underbrace{x, y}_{\text{chromaticity}}, Y)$$

luminance/brightness

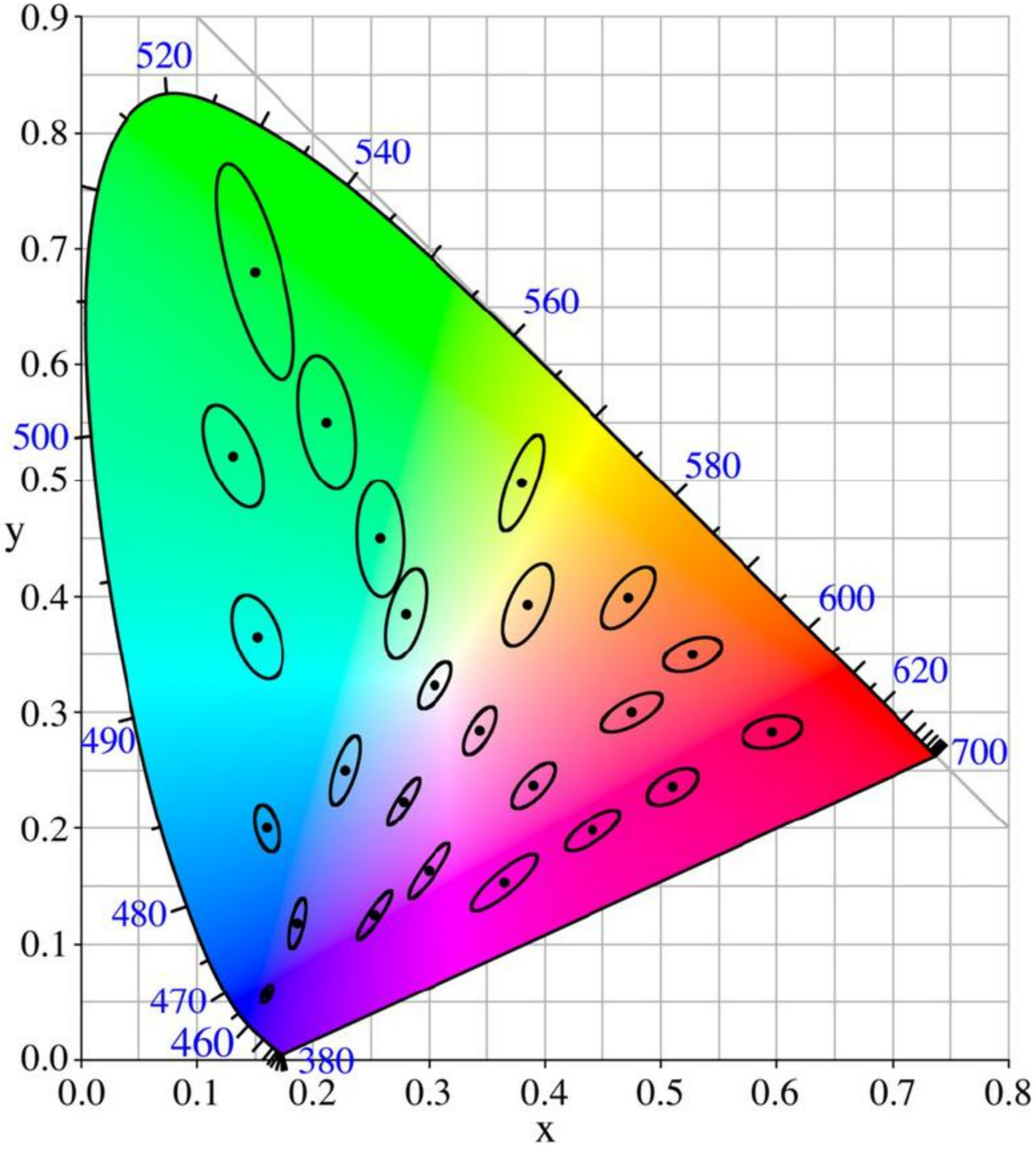
CIE xyY is a non-linear color space.

Uniform color spaces

Find map $F : \mathbb{R}^3 \to \mathbb{R}^3$ such that perceptual distance can be well approximated using Euclidean distance:

$$d(\vec{c}, \vec{c}') \approx ||F(\vec{c}) - F(\vec{c}')||_2$$

MacAdam ellipses

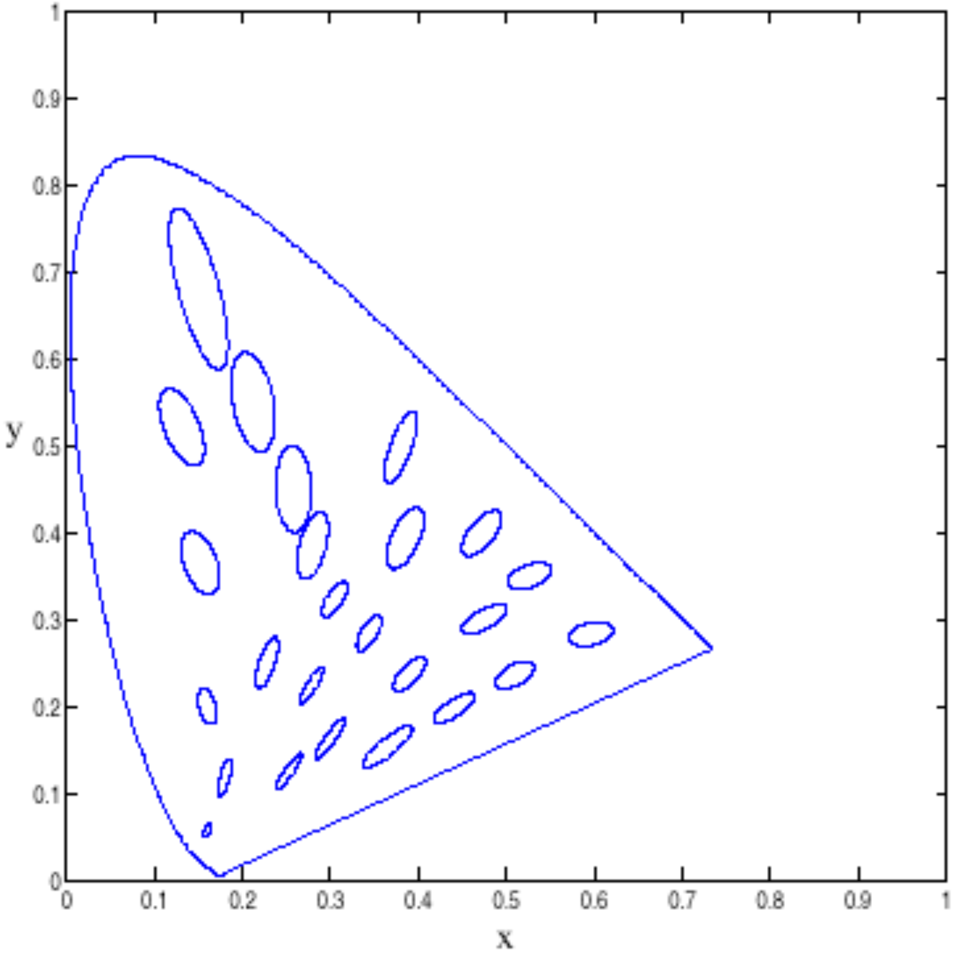


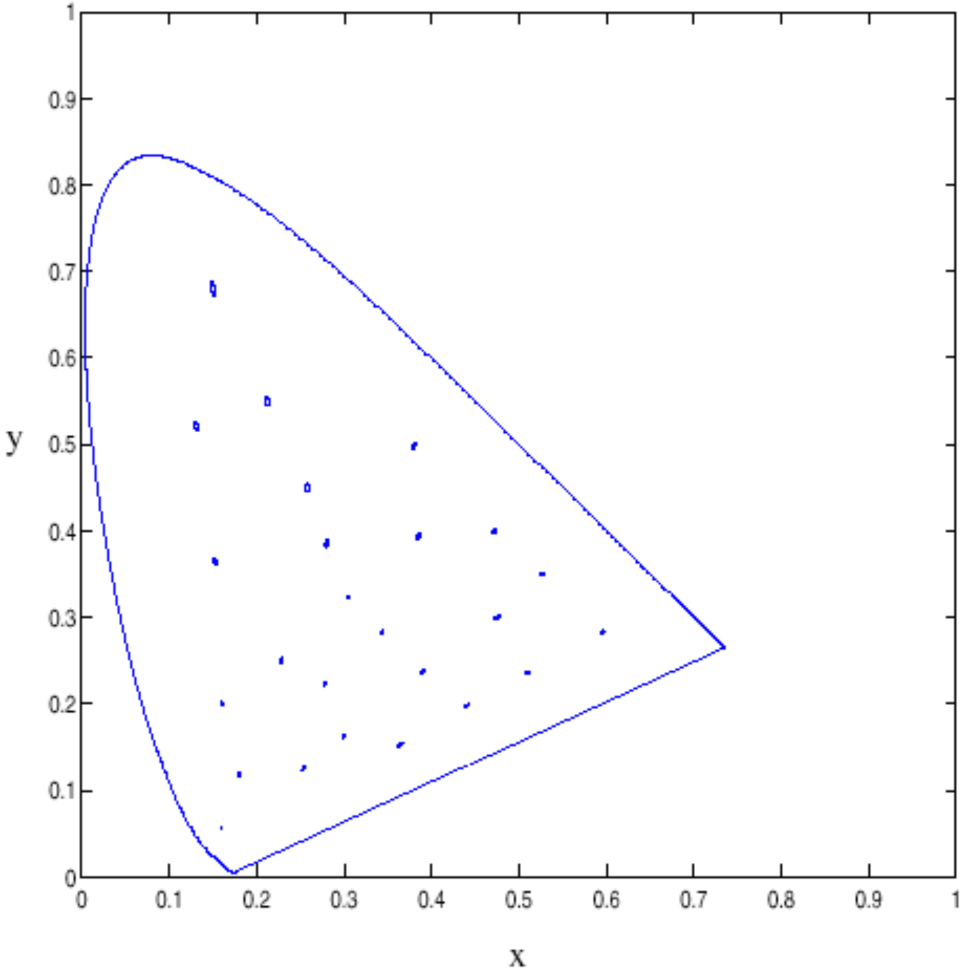
Areas in chromaticity space of imperceptible change:

- They are ellipses instead of circles.
- They change scale and direction in different parts of the chromaticity space.

MacAdam ellipses

Note: MacAdam ellipses are almost always shown at 10x scale for visualization. In reality, the areas of imperceptible difference are much smaller.





The Lab (aka L*ab, aka L*a*b*) color space

The L* component of *lightness* is defined as

$$L^* = 116 f\left(\frac{Y}{Y_n}\right), \tag{2.105}$$

where Y_n is the luminance value for nominal white (Fairchild 2005) and

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases} \tag{2.106}$$

is a finite-slope approximation to the cube root with $\delta = 6/29$. The resulting $0 \ldots 100$ scale roughly measures equal amounts of lightness perceptibility.

In a similar fashion, the a* and b* components are defined as

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \text{ and } b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right], \tag{2.107}$$

where again, (X_n, Y_n, Z_n) is the measured white point. Figure 2.32i–k show the L*a*b* representation for a sample color image.

The Lab (aka L*ab, aka L*a*b*) color space

The L* component of *lightness* is defined as

$$L^* = 116 f\left(\frac{Y}{Y_n}\right), \tag{2.105}$$

where Y_n is the luminance value for nominal white (Fairchild 2005) and

What is this?

$$\boxed{f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases}} \tag{2.106}$$

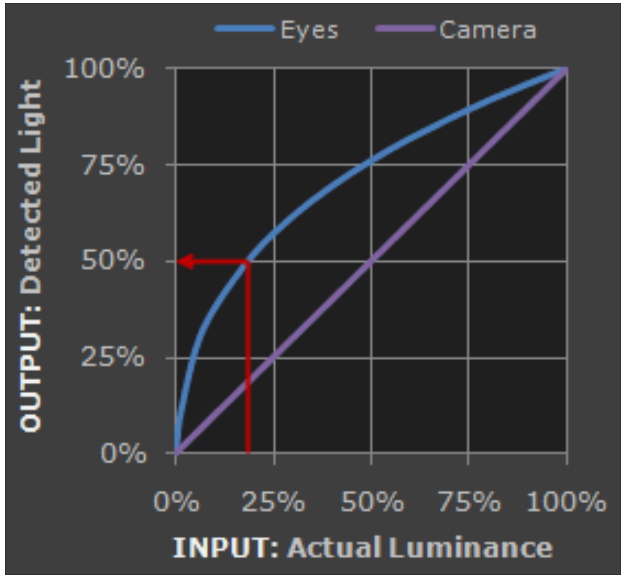
is a finite-slope approximation to the cube root with $\delta = 6/29$. The resulting $0 \ldots 100$ scale roughly measures equal amounts of lightness perceptibility.

In a similar fashion, the a* and b* components are defined as

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \text{ and } b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right], \tag{2.107}$$

where again, (X_n, Y_n, Z_n) is the measured white point. Figure 2.32i–k show the L*a*b* representation for a sample color image.

Perceived vs measured brightness by human eye

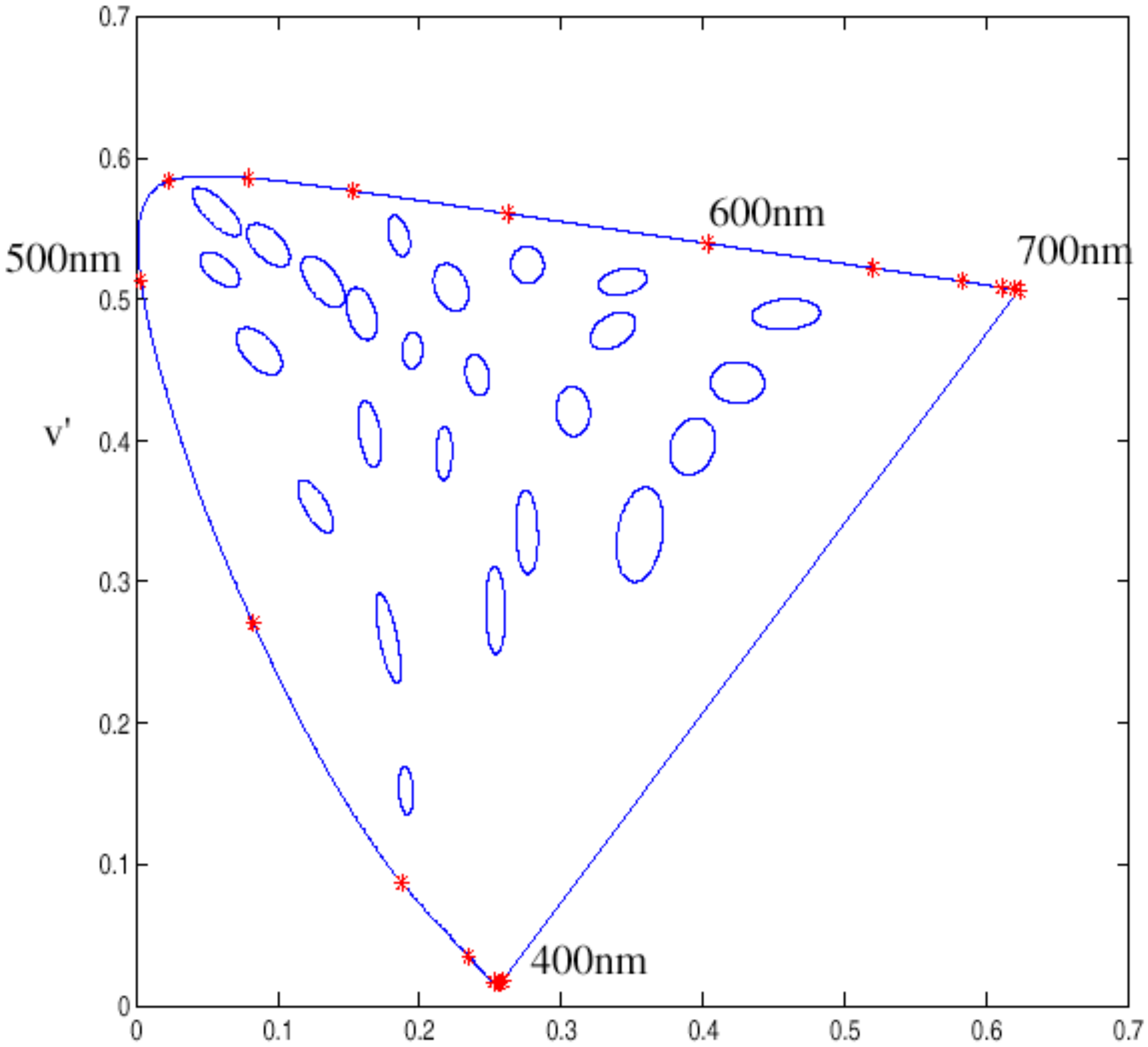


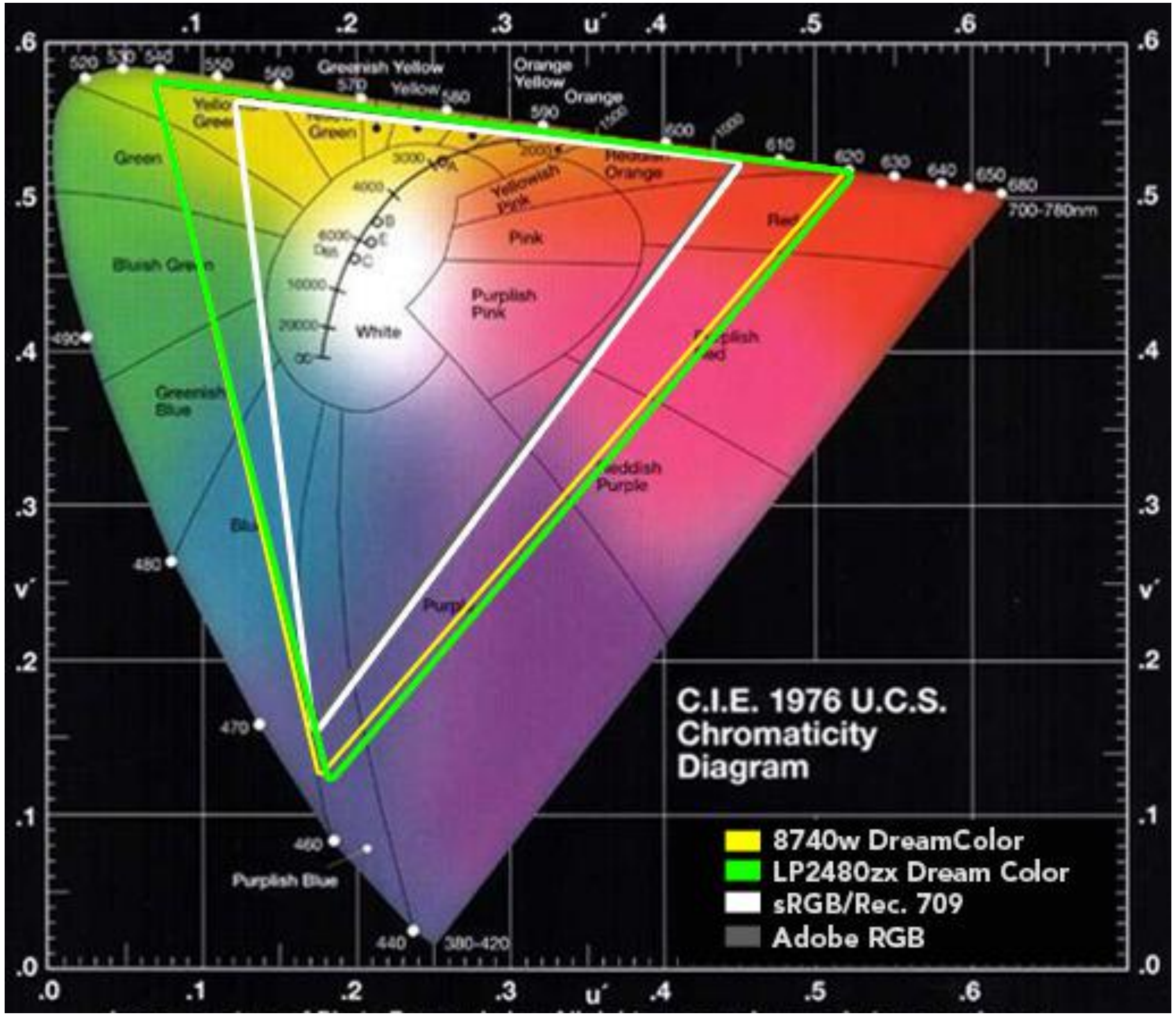
Human-eye response (measured brightness) is linear.

However, human-eye perception (perceived brightness) is non-linear:

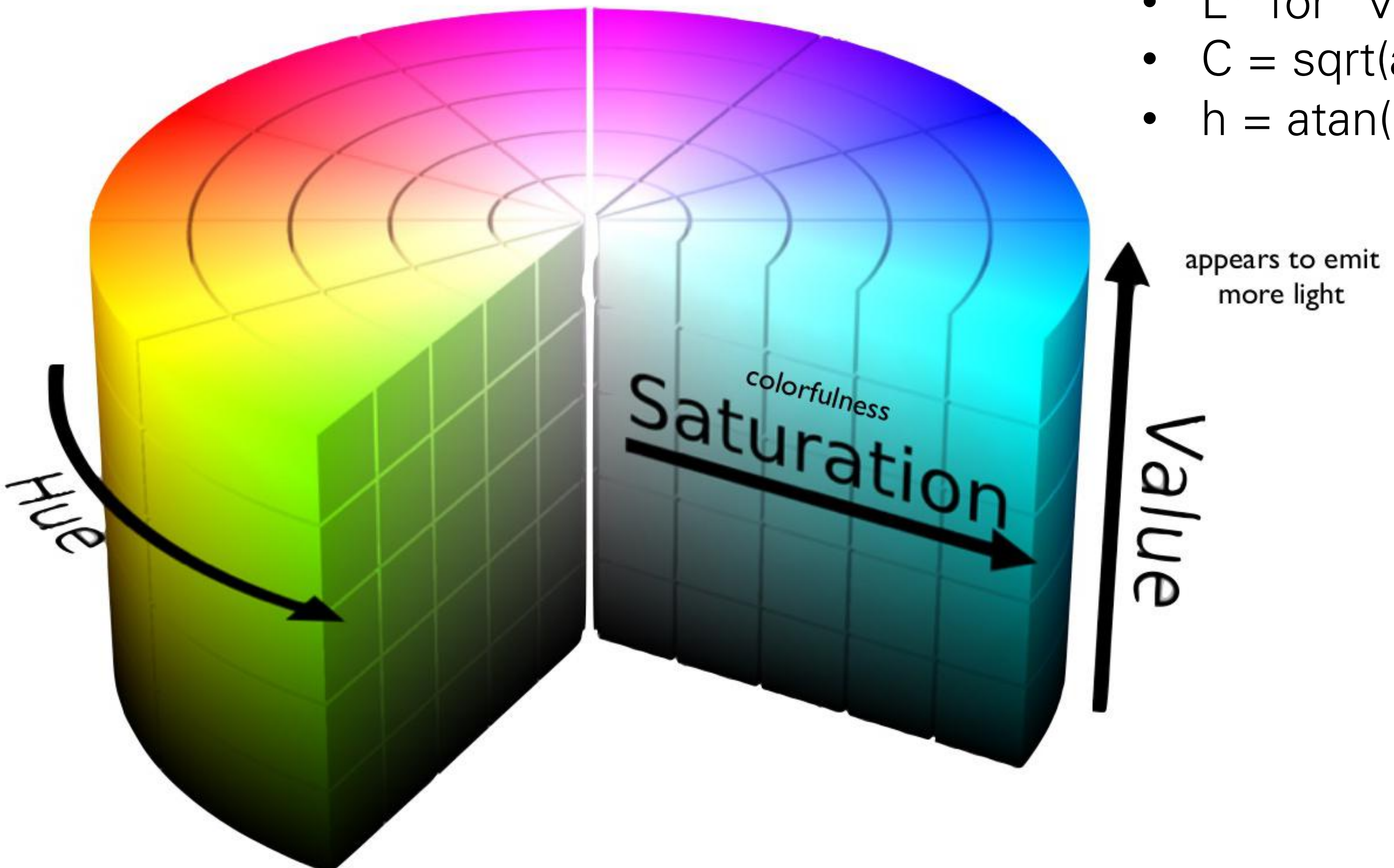
- More sensitive to dark tones.
- Approximately a Gamma function.

The Lab (aka L*ab, aka L*a*b*) color space





Hue, saturation, and value



Do not use color space HSV! Use LCh:

- L* for "value".
- C = sqrt($a^2 + b^2$) for "saturation" (chroma)
- h = atan(b / a) for "hue".

How could you make an image like this from a color image?

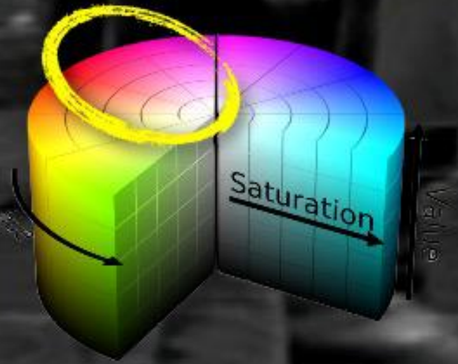
http://en.wikipedia.org/wiki/Schindler's_List

How could you make an image like this from a color image?

Zero saturation

Higher saturation

Control saturation with red-pass filter



LCh

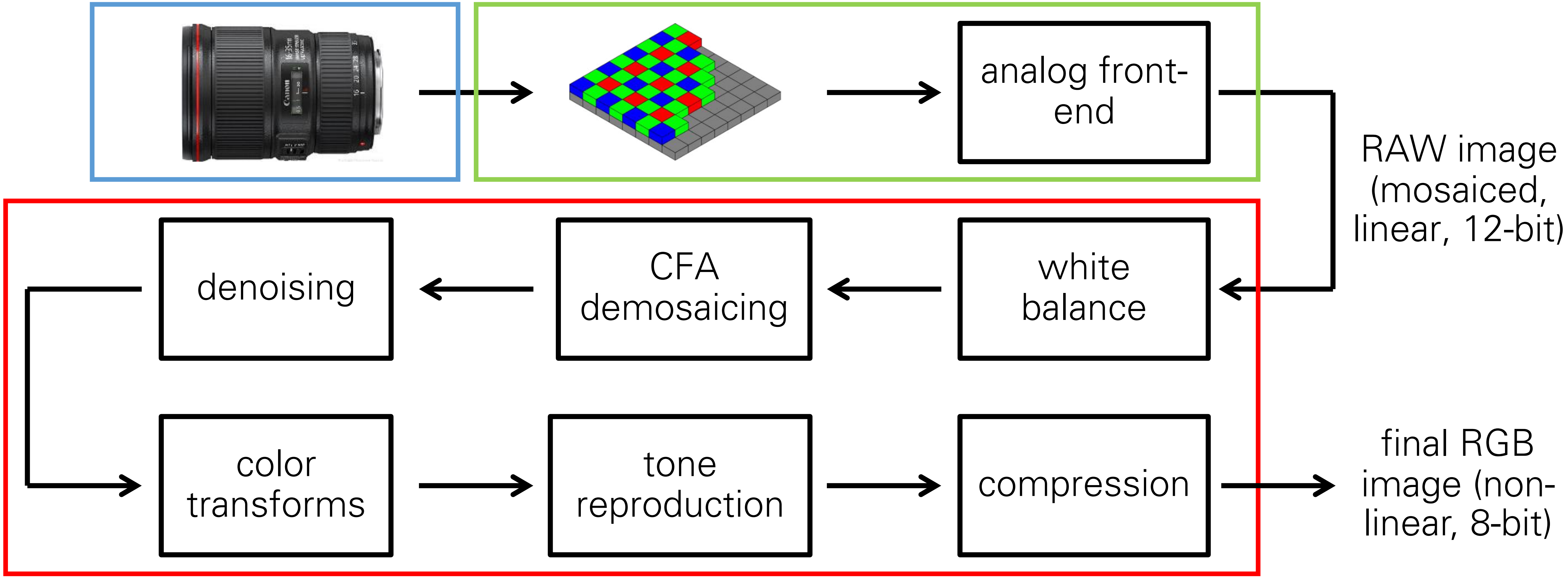
Easier to do color processing in ~~HSV~~

http://en.wikipedia.org/wiki/Schindler's_List

Some thoughts about color reproduction

The image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Putting it together: color processing

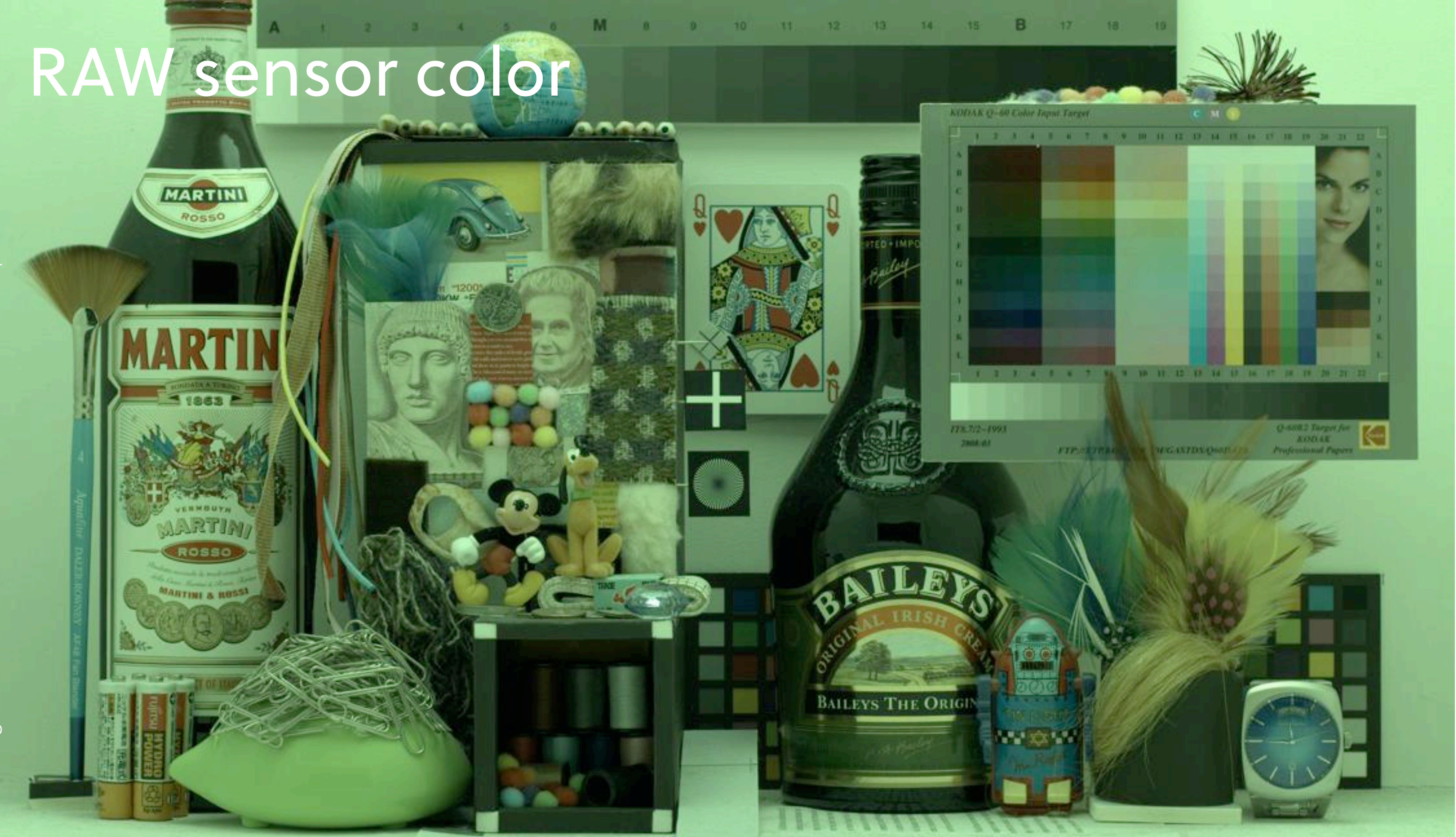
- Calibrate your color matrix using a carefully white-balanced image

  when solving for M, constrain to ensure rows sum to 1
  (then M will leave neutral colors exactly alone)

- For each photograph:

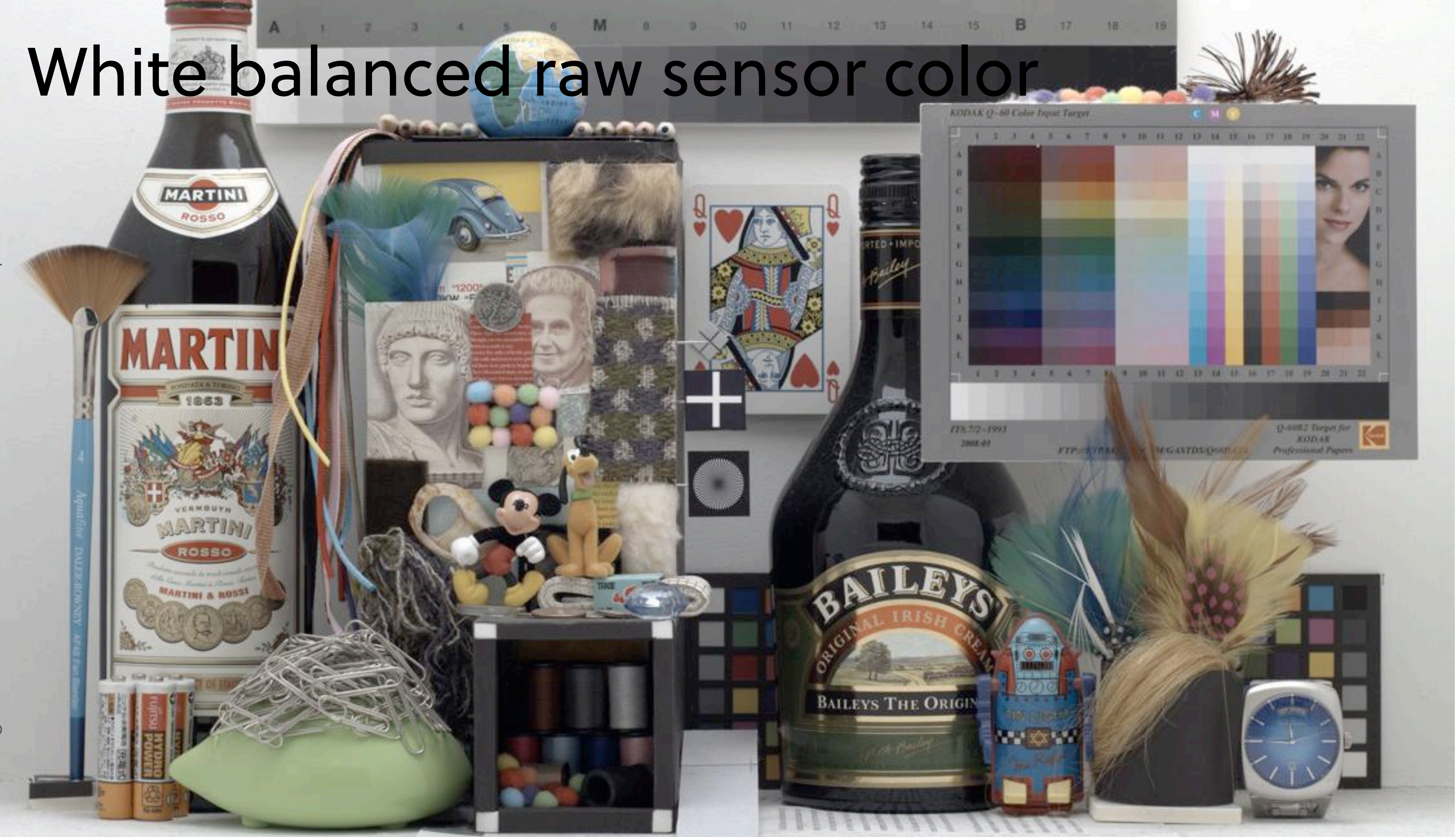
1. determine illuminant
2. apply von Kries
3. apply color matrix
4. apply any desired nonlinearity
5. display the image!

RAW sensor color



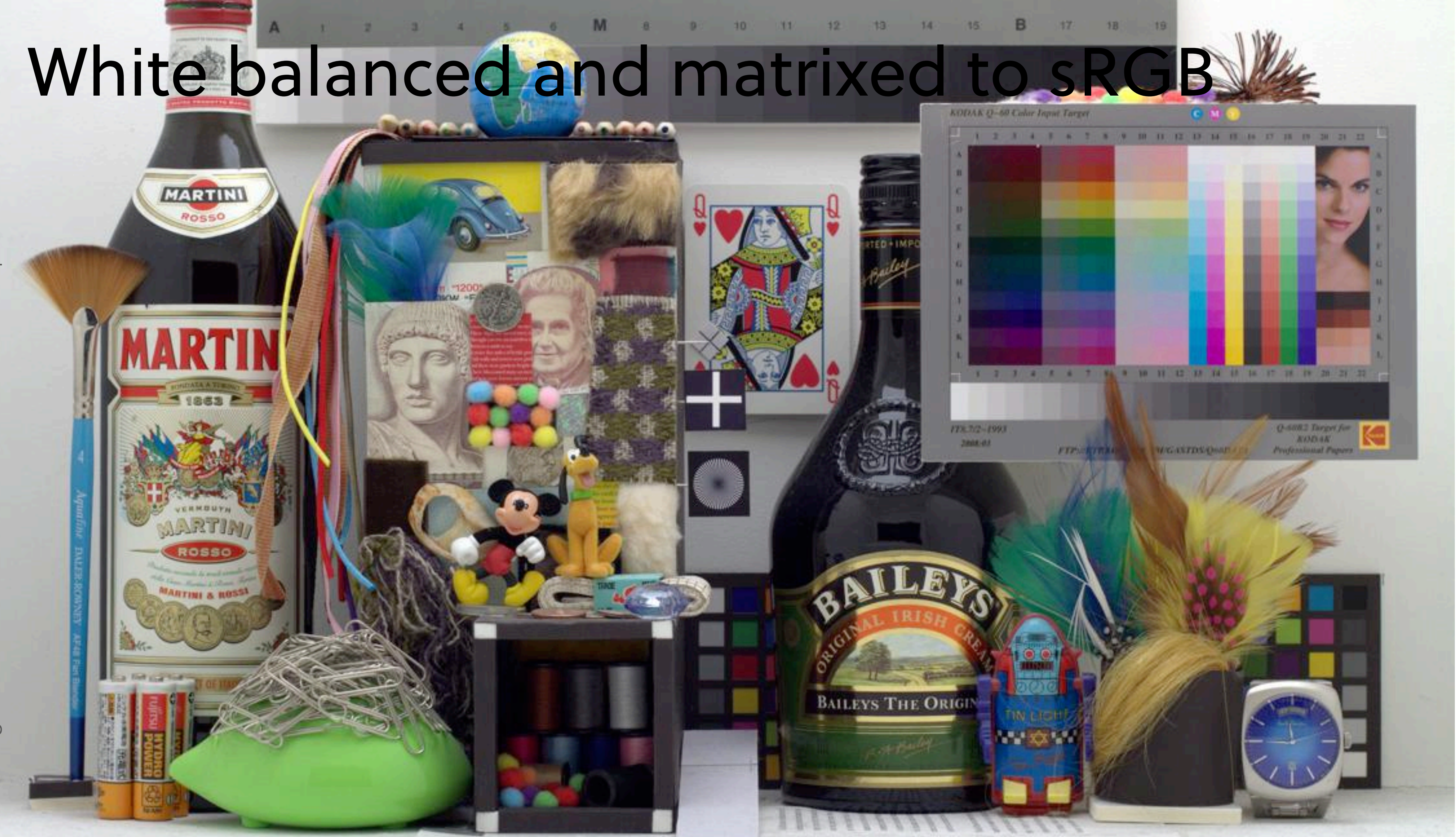
test image for Canon EOS 650D (Rebel T4i) from dpreview.com

White balanced raw sensor color



test image for Canon EOS 650D (Rebel T4i) from dpreview.com

White balanced and matrixed to sRGB



test image for Canon EOS 650D (Rebel T4i) from dpreview.com

Color reproduction notes

To properly reproduce the color of an image file, you need to?

Color reproduction notes

To properly reproduce the color of an image file, you need to convert it from the color space it was stored in, to a reference color space, and then to the color space of your display.

On the camera side:

- If the file is RAW, it often has EXIF tags with information about the RGB color space corresponding to the camera's color sensitivity functions.
- If the file is not RAW, you may be lucky and still find accurate information in the EXIF tags about what color space the image was converted in during processing.
- If there is no such information and you own the camera that shot the image, then you can do color calibration for the camera.
- If all of the above fails, assume sRGB.

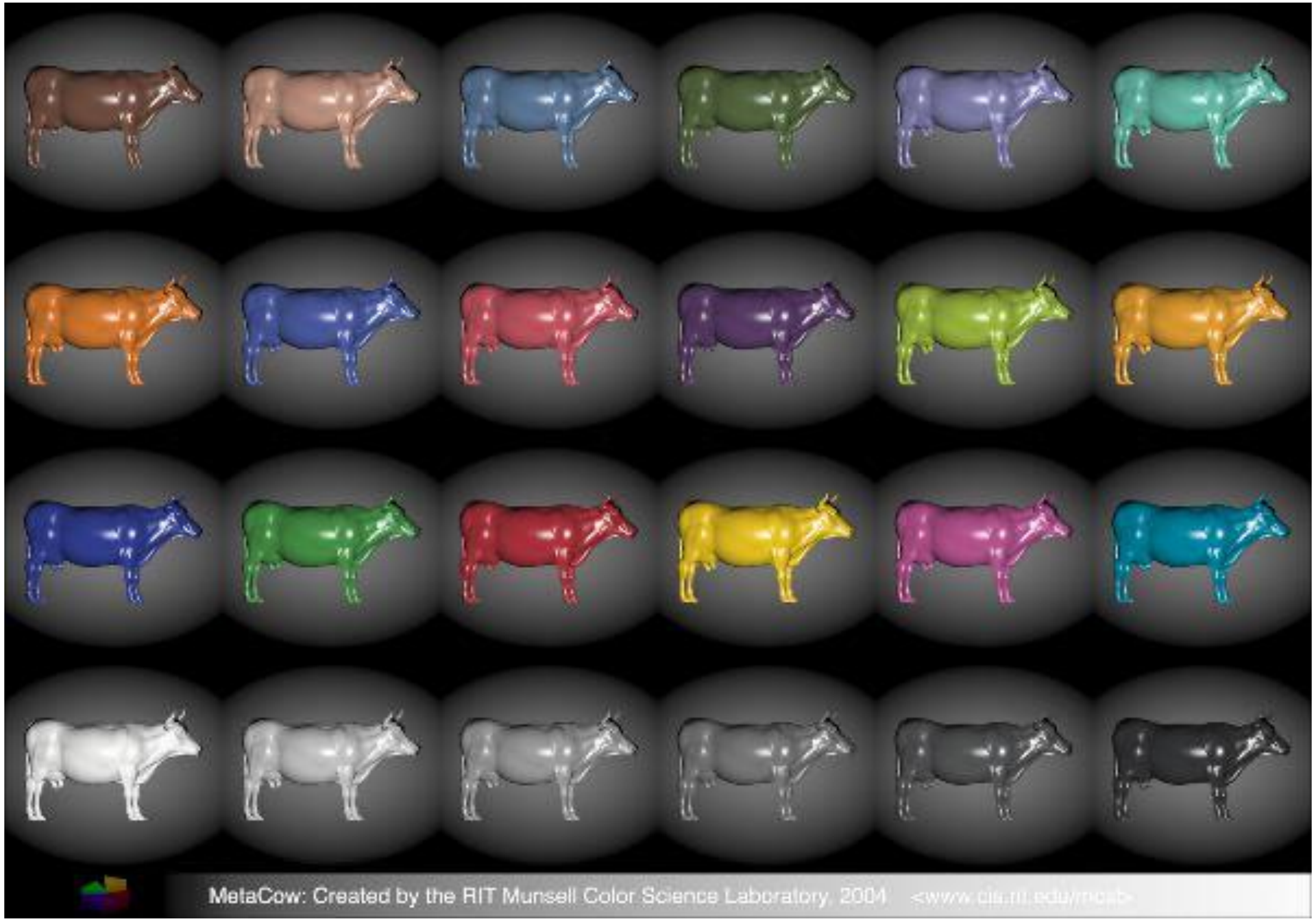
On the display side:

- If you own a high-end display, it likely has accurate color profiles provided by the manufacturer.
- If not, you can use a spectrometer to do color profiling (not color calibration).
- Make sure your viewer does not automatically do color transformations.

Be careful to account for any gamma correction!

Amazing resource for color management and photography: https://ninedegreesbelow.com/

The METACOW spectral image database



Amazing dataset for color management and photography: https://www.rit.edu/cos/colorscience/rc_db_metacow.php

How do you convert an image to grayscale?

How do you convert an image to grayscale?

First, you need to answer two questions:

1) Is your image linear or non-linear?

- If the image is linear (RAW, HDR, or otherwise radiometrically calibrated), skip this step.
- If the image is nonlinear (PNG, JPEG, etc.), you must undo the tone reproduction curve.
  - i. If you can afford to do radiometric calibration, do that.
  - ii. If your image has EXIF tags, check there about the tone reproduction curve.
  - iii. If your image is tagged as non-linear sRGB, use the inverse of the sRGB tone reproduction curve.
  - iv. If none of the above, assume sRGB and do as in (iii).

2) What is the color space of your image?

- If it came from an original RAW file, read the color transform matrix from there (e.g., dcraw).
- If not, you need to figure out the color space.
  - i. If you can afford to do color calibration, use that.
  - ii. If your image has EXIF tags, check there about the color space.
  - iii. If your image is tagged as non-linear sRGB, use the color transform matrix for linear sRGB.
  - iv. If none of the above, assume sRGB and do (iii).

With this information in hand:

- Transform your image into the XYZ color space. (If it is in sRGB, you may need to do whitepoint adaptation!!)
- Extract the Y channel.
- If you want brightness instead of luminance, apply the Lab brightness non-linearity.

How do you convert an image to grayscale?

Why You Should Forget Luminance Conversion and Do Something Better

Rang M. H. Nguyen
National University of Singapore
nguyenho@comp.nus.edu.sg

Michael S. Brown
York University
mbrown@eecs.yorku.ca

Abstract

One of the most frequently applied low-level operations in computer vision is the conversion of an RGB camera image into its luminance representation. This is also one of the most incorrectly applied operations. Even our most trusted softwares, Matlab and OpenCV, do not perform luminance conversion correctly. In this paper, we examine the main factors that make proper RGB to luminance conversion difficult, in particular: 1) incorrect white-balance, 2) incorrect gamma/tone-curve correction, and 3) incorrect equations. Our analysis shows errors up to 50% for various colors are not uncommon. As a result, we argue that for most computer vision problems there is no need to attempt luminance conversion; instead, there are better alternatives depending on the task.

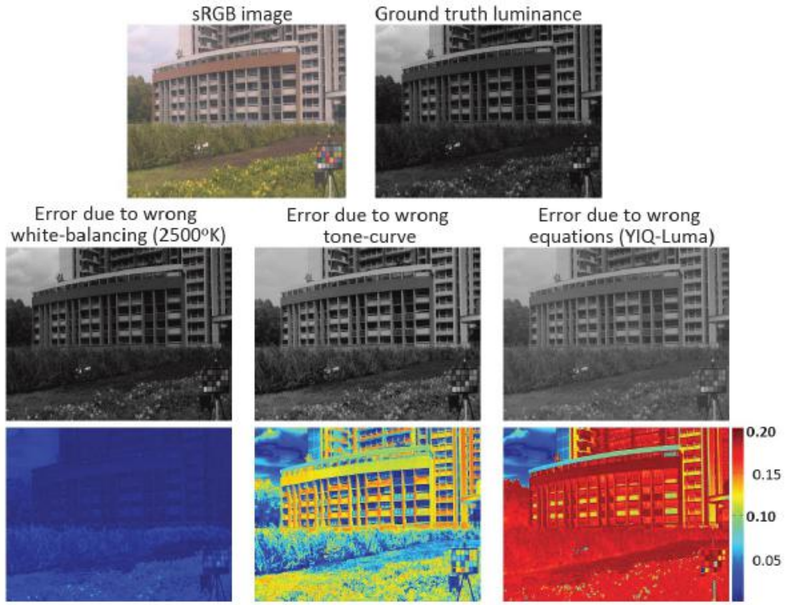


Figure 1. This figure shows examples of errors that arise due to improper luminance conversion. The ground truth luminance for this experiment is captured from a hyperspectral camera.

Next Lecture:
Exposure and
high-dynamic-range imaging