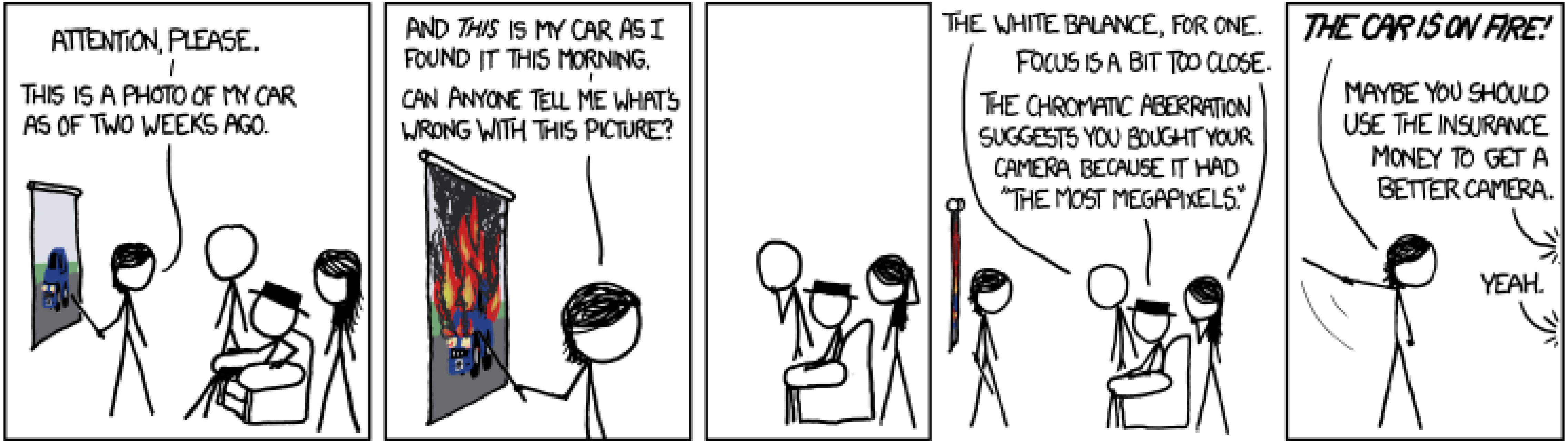
# Digital photography




046746 Algorithms and applications in computer vision
Spring 2020, Lecture 1

Slide credits: Ioannis Gkioulekas, Tali Treibitz, Kayvon Fatahalian, Michael Brown

# Overview of today’s lecture

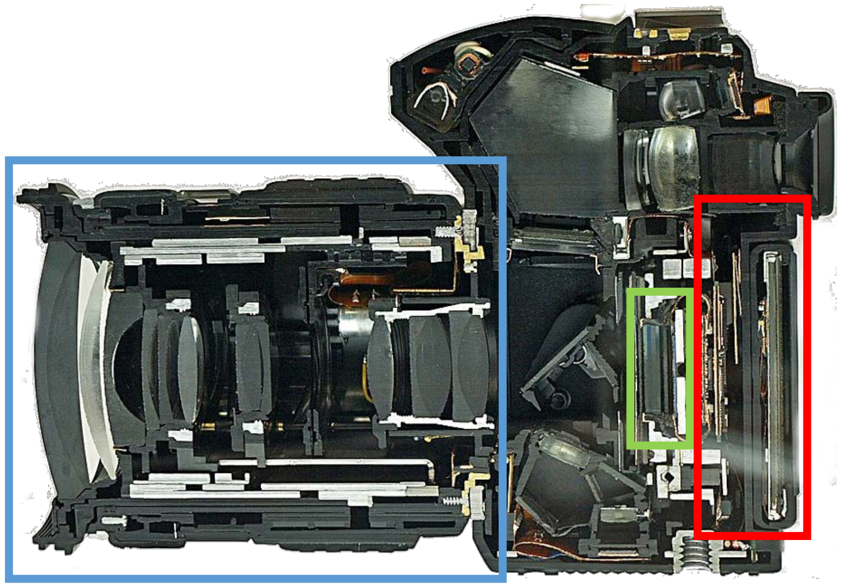
- Imaging sensor primer.
- Color primer.
- In-camera image processing pipeline.
- Some general thoughts on the image processing pipeline.

Take-home message: The values of pixels in a photograph and the values output by your camera’s sensor are two very different things.

# The modern photography pipeline

# 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

# Imaging sensors

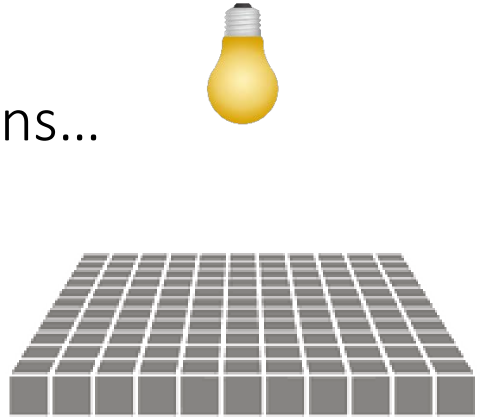
- Very high-level overview of digital imaging sensors.
- We could spend an entire course covering imaging sensors.

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

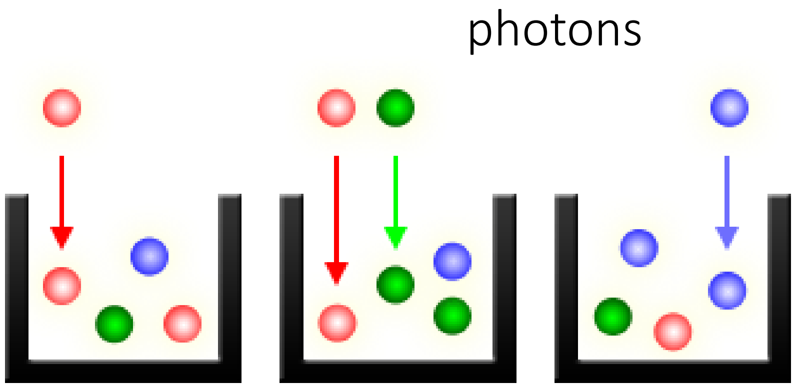
# What does an imaging sensor do?

When the camera shutter opens…

… exposure begins…



array of photon buckets

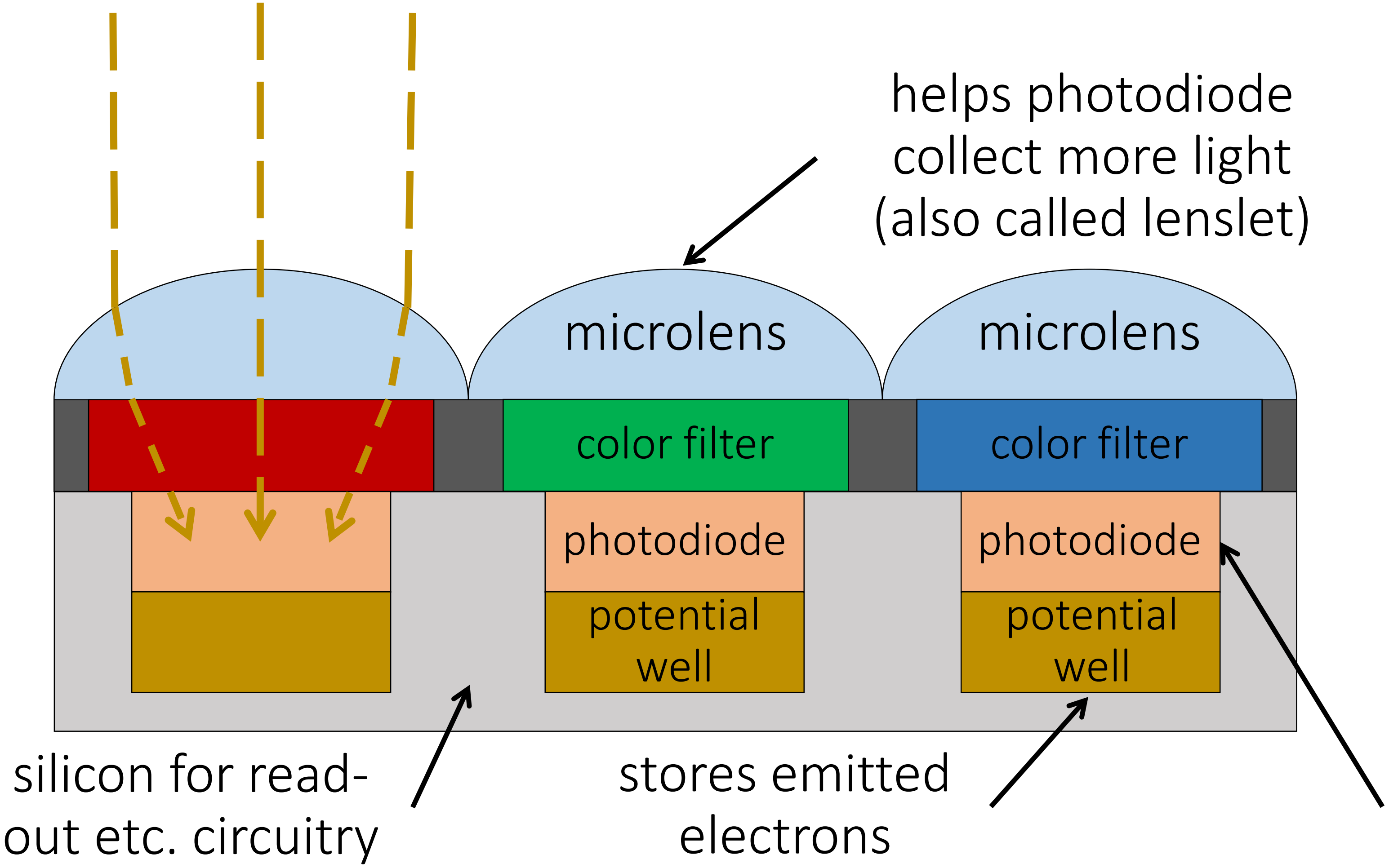



close-up view of photon buckets

… photon buckets begin to store photons...

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

# Basic imaging sensor design




- Lenslets also filter the image to avoid resolution artifacts.

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

# Photodiode response function

For silicon photodiodes, usually linear, but:

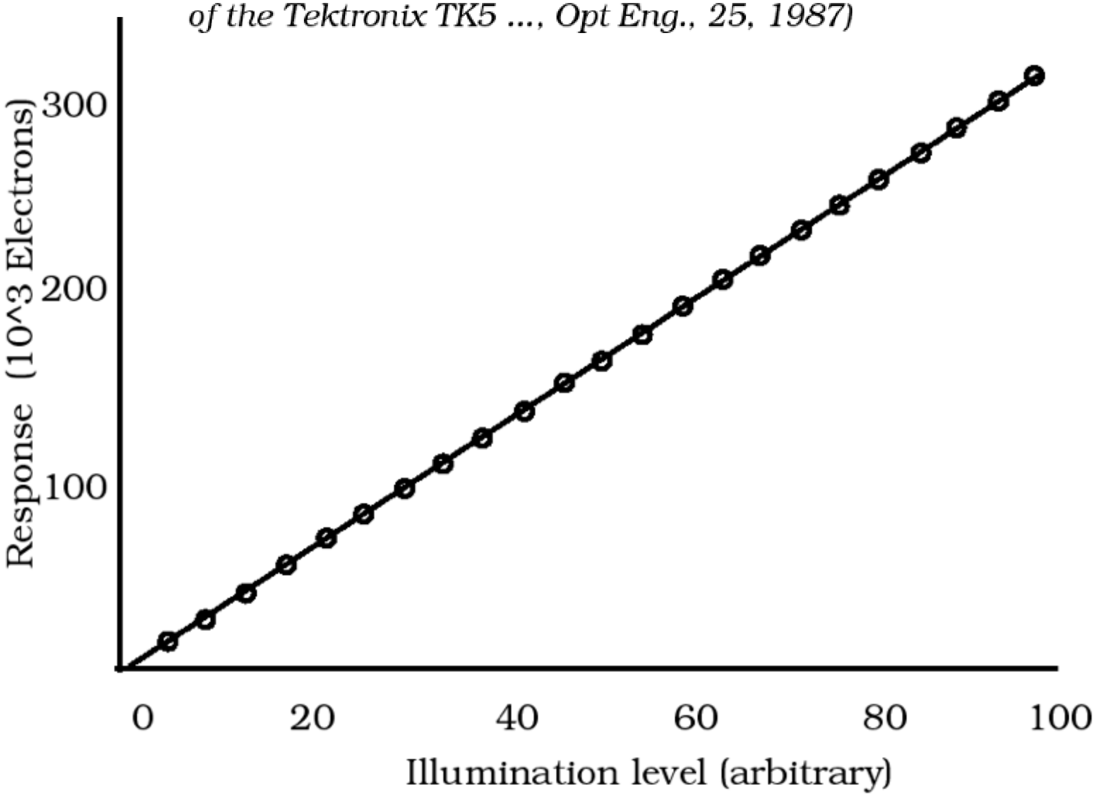
- non-linear when potential well is saturated (over-exposure)

- non-linear near zero (due to noise)

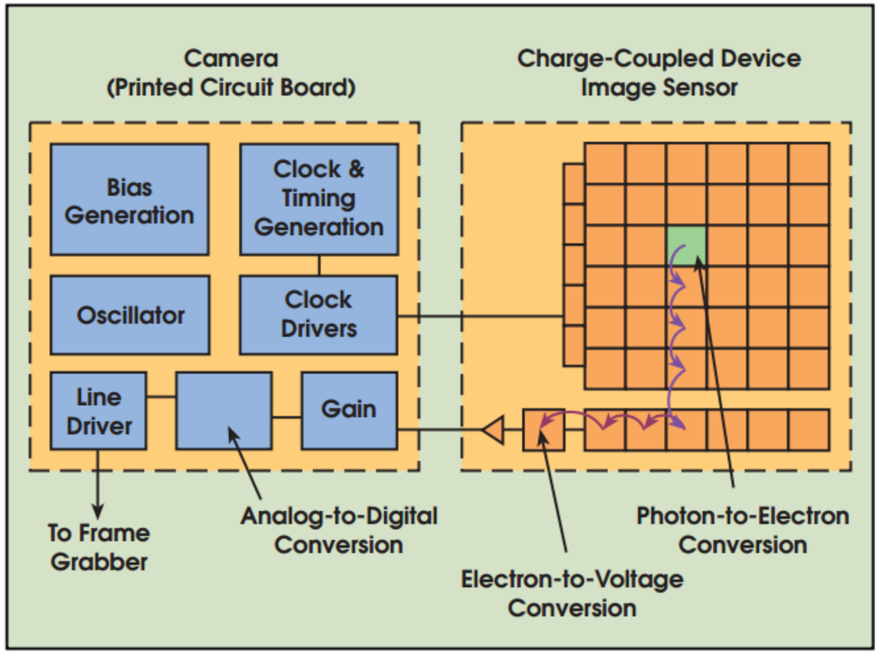
under-exposure (non-linearity due to sensor noise)

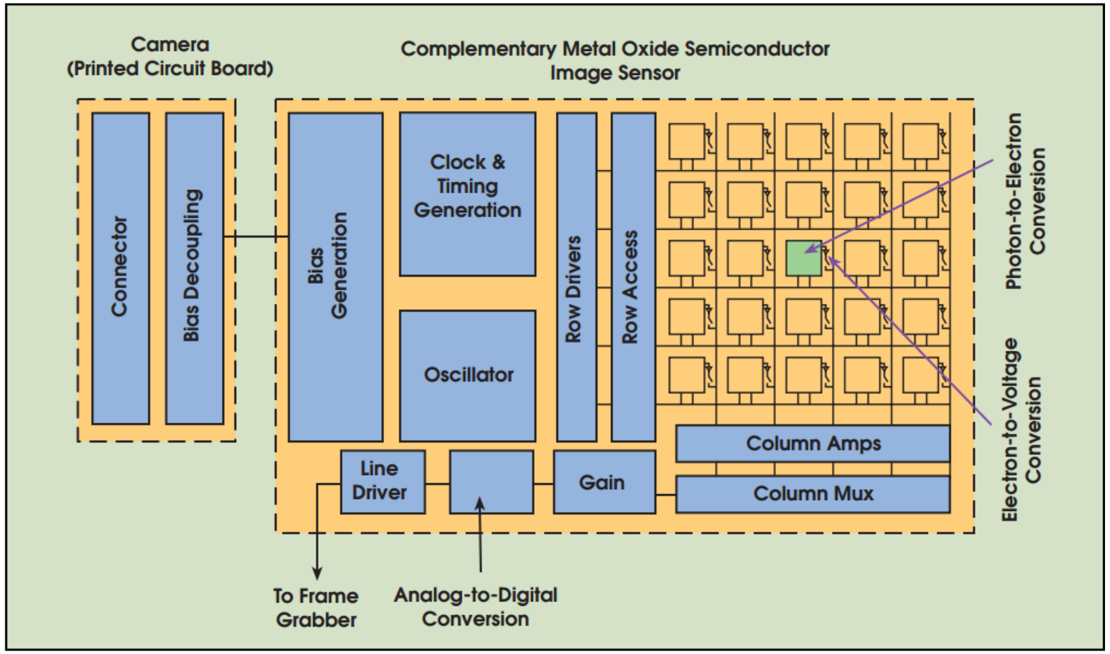
over-exposure (non-linearity due to sensor saturation)

(Epperson, P.M. et al. Electro-optical characterization of the Tektronix TK5 ..., Opt Eng., 25, 1987)



# Two main types of imaging sensors




Charged Coupled Device (CCD):
converts electrons to voltage using
readout circuitry separate from pixel

Complementary Metal Oxide Semiconductor (CMOS):
converts electrons to voltage using
per-pixel readout circuitry

- ✓ higher sensitivity
- ✓ lower noise

- ✓ faster read-out
- ✓ lower cost

# Vignetting

Fancy word for: pixels far off the center receive less light

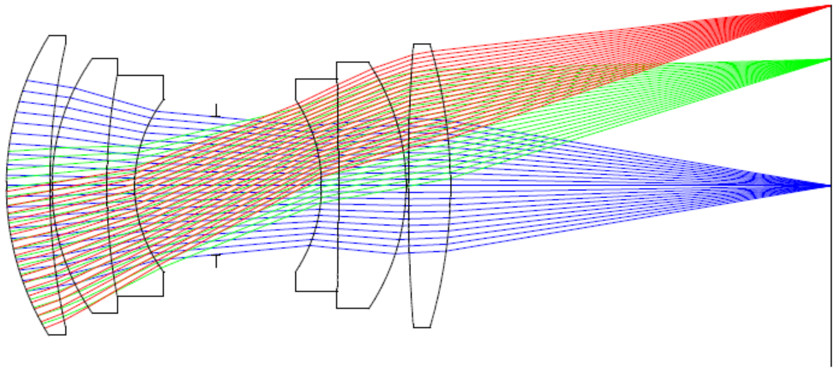
white wall under uniform light

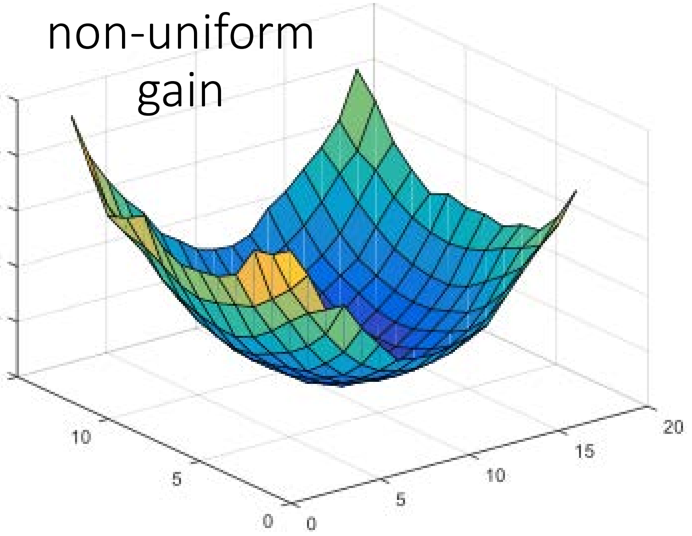
more interesting example of vignetting

# Vignetting

Four types of vignetting:

- Mechanical: light rays blocked by hoods, filters, and other objects.
- Lens: similar, but light rays blocked by lens elements.
- Natural: due to radiometric laws ("cosine fourth falloff").
- Pixel: angle-dependent sensitivity of photodiodes.






# What does an imaging sensor do?

When the camera shutter opens, the sensor:

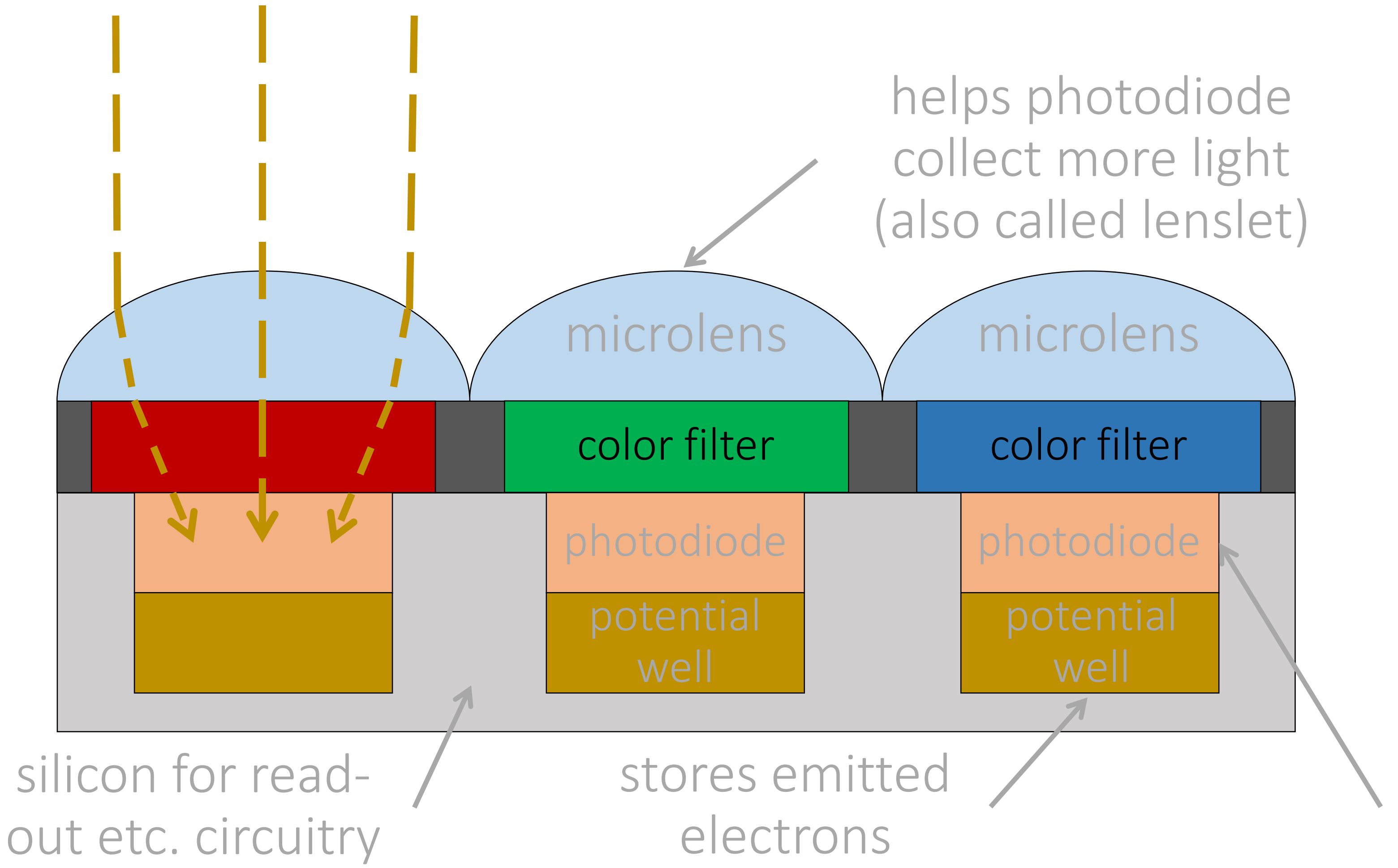
- at every photodiode, converts incident photons into electrons
- stores electrons into the photodiode’s potential well while it is not full

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

- reads out photodiodes’ 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?



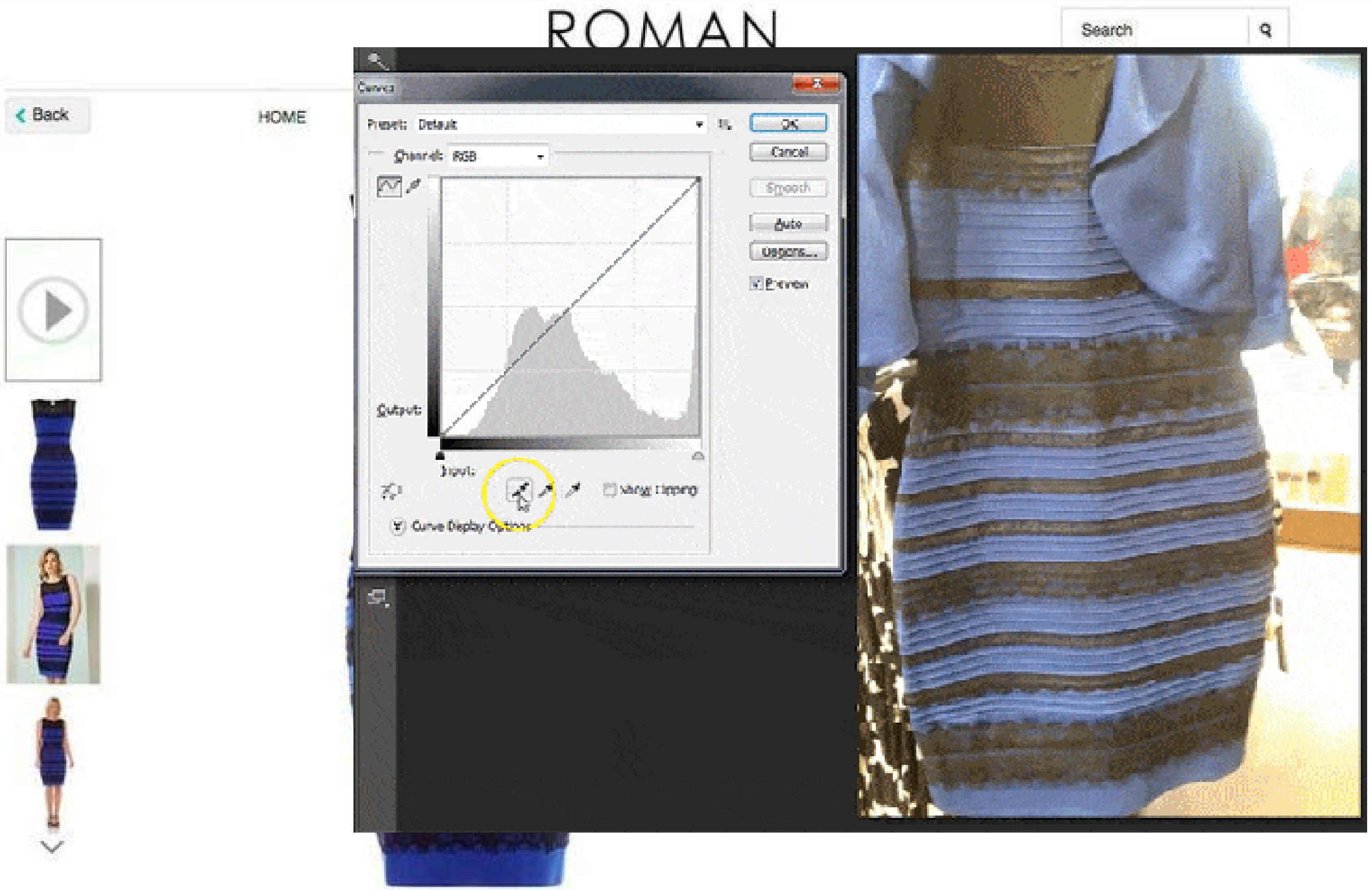

- Lenslets also filter the image to avoid resolution artifacts.

Color filters ?

# Color primer

# Color

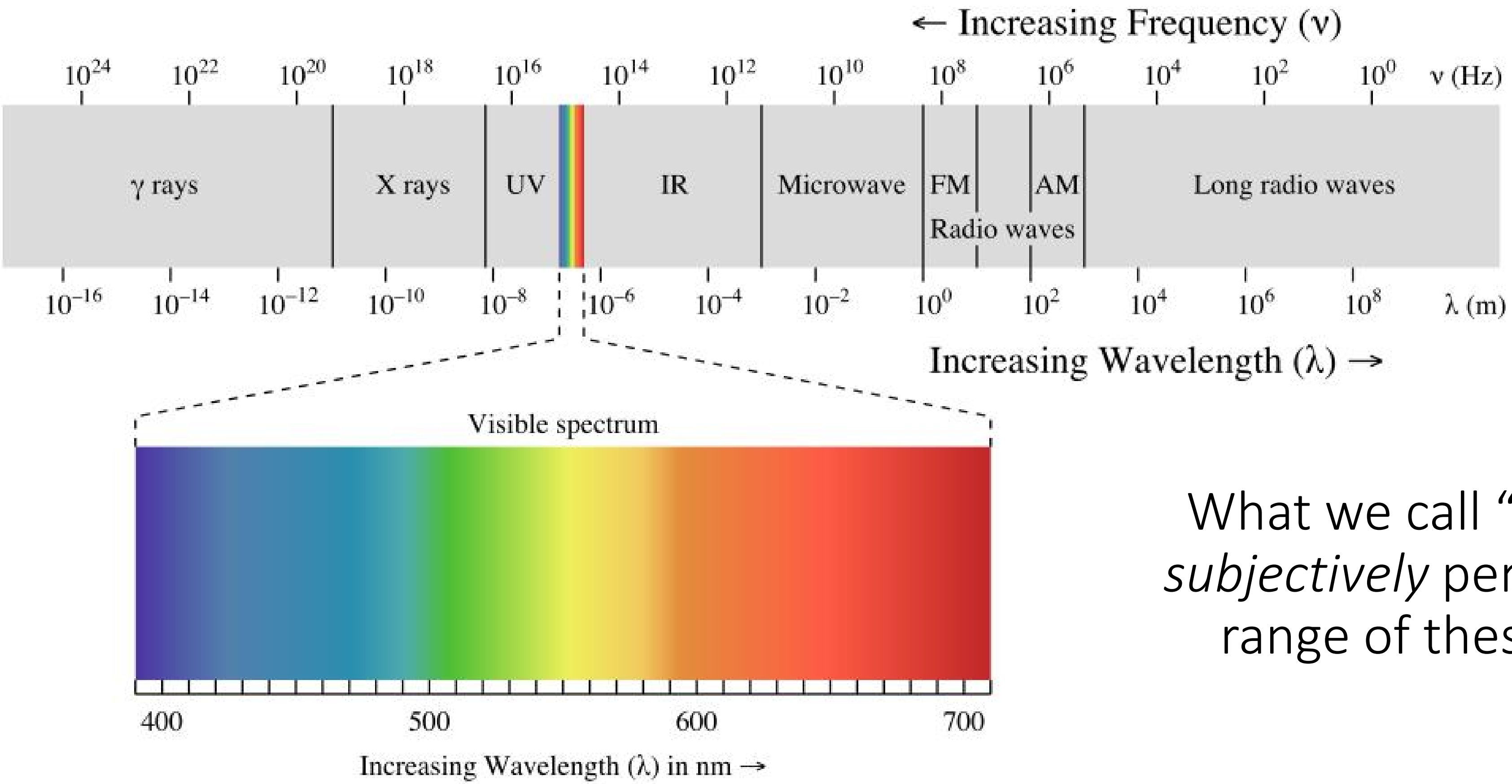
- Very high-level of color as it relat
- We could spend an entire course




color is complicated

# Color is an artifact of human perception

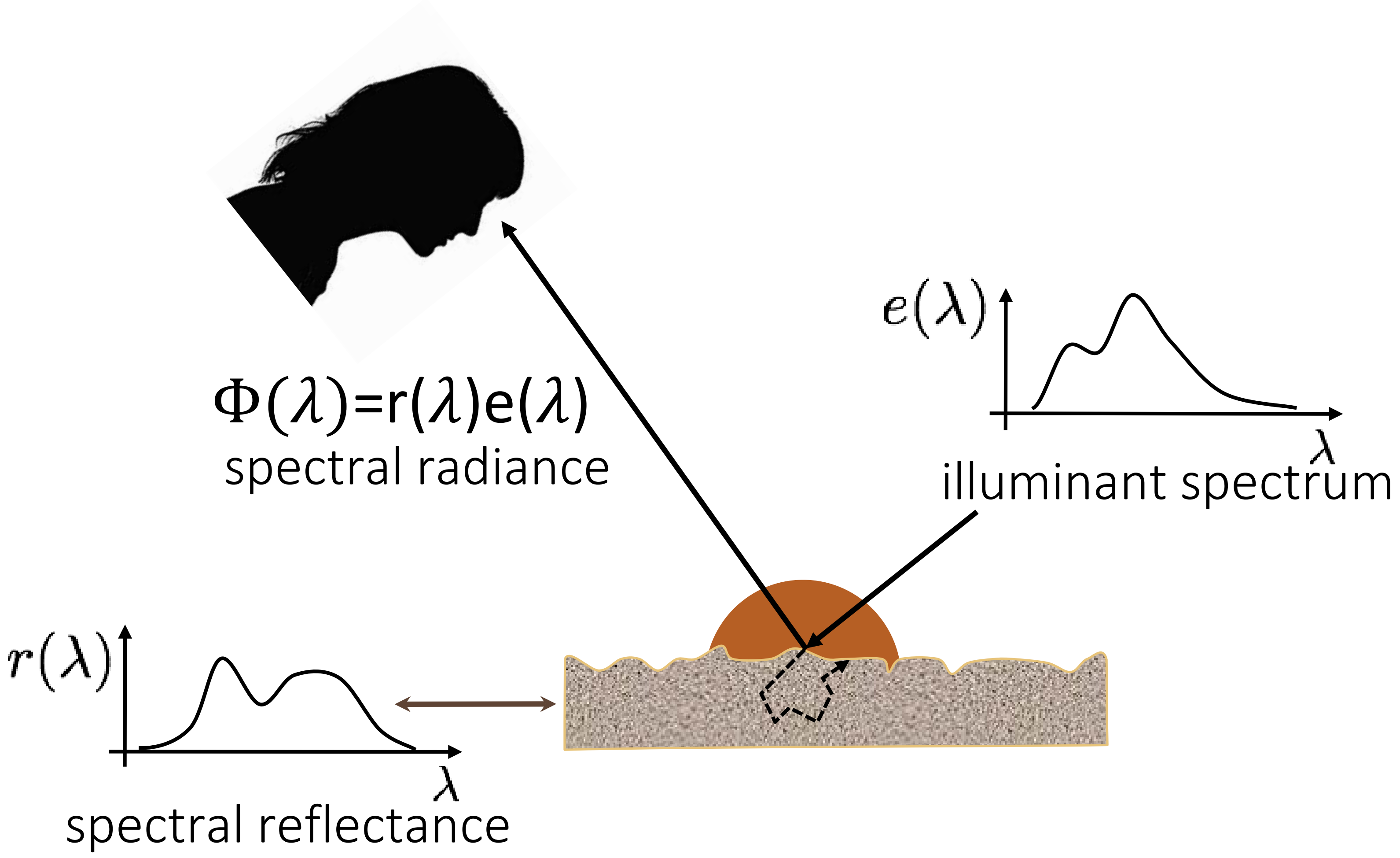
- "Color" is not an *objective* physical property of light (electromagnetic radiation).
- Instead, light is characterized by its wavelength.




electromagnetic spectrum

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

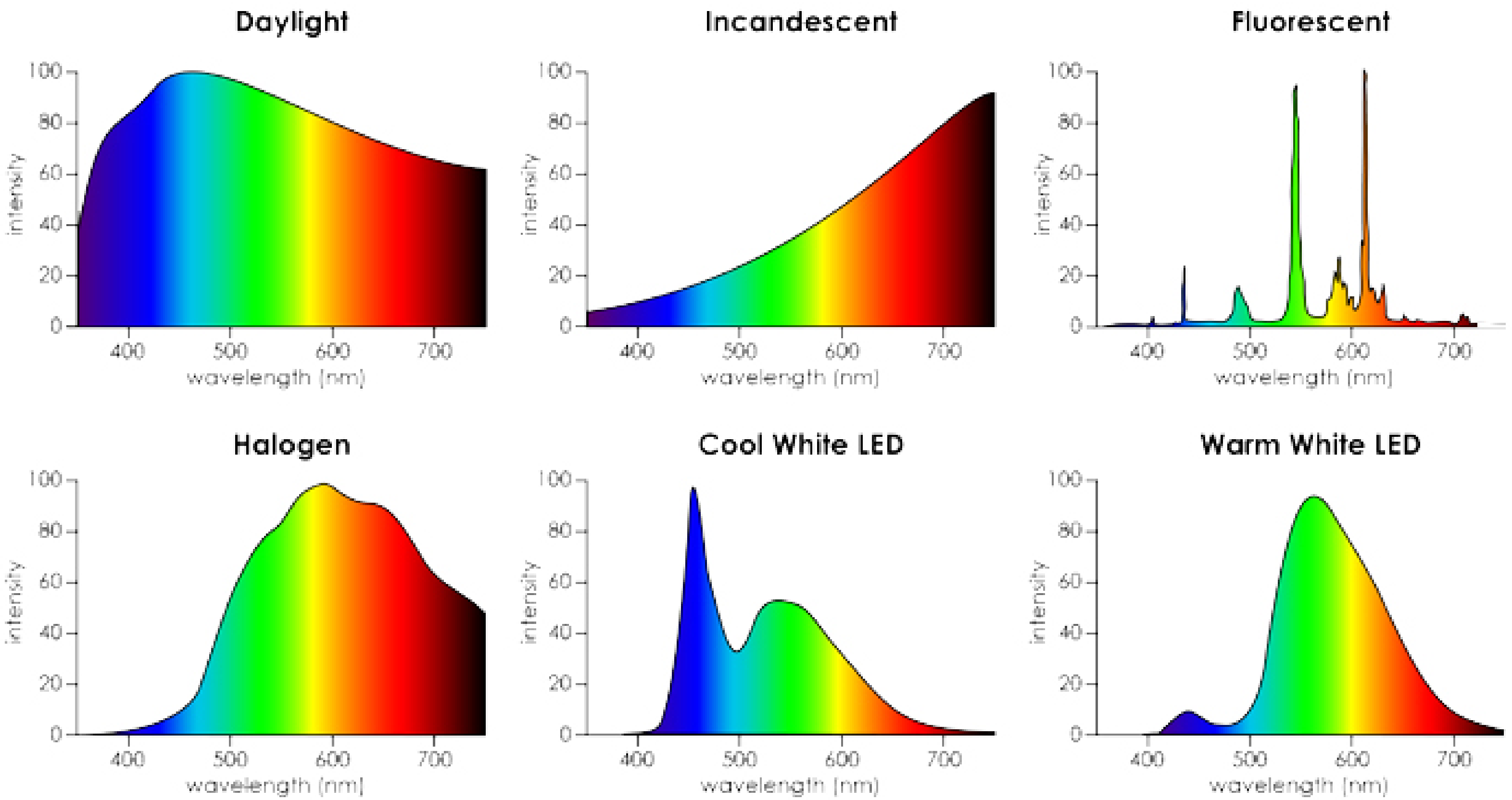
# Light-material interaction



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

$\Phi(\lambda) = r(\lambda) e(\lambda)$

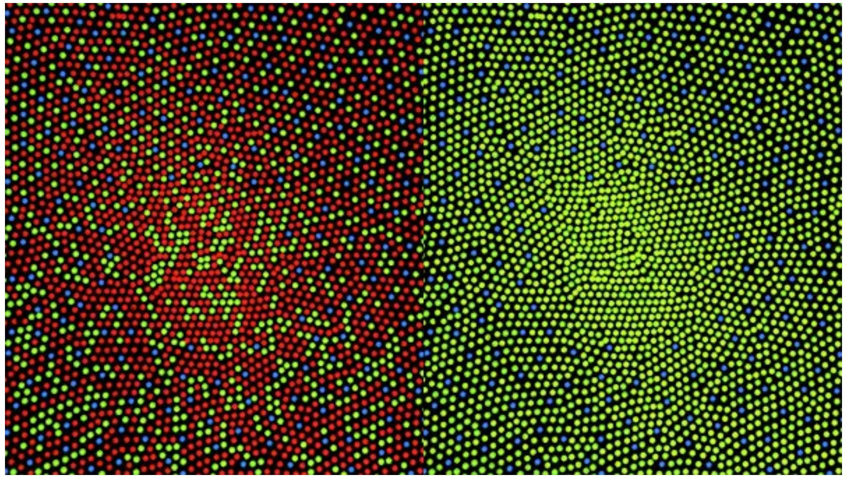
Material reflectance

illuminant spectrum

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*).

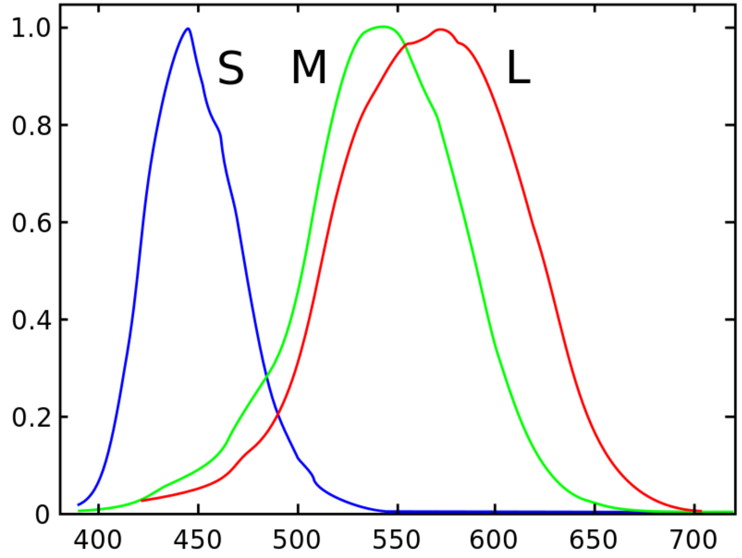


cone distribution
for normal vision
(64% L, 32% M)

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

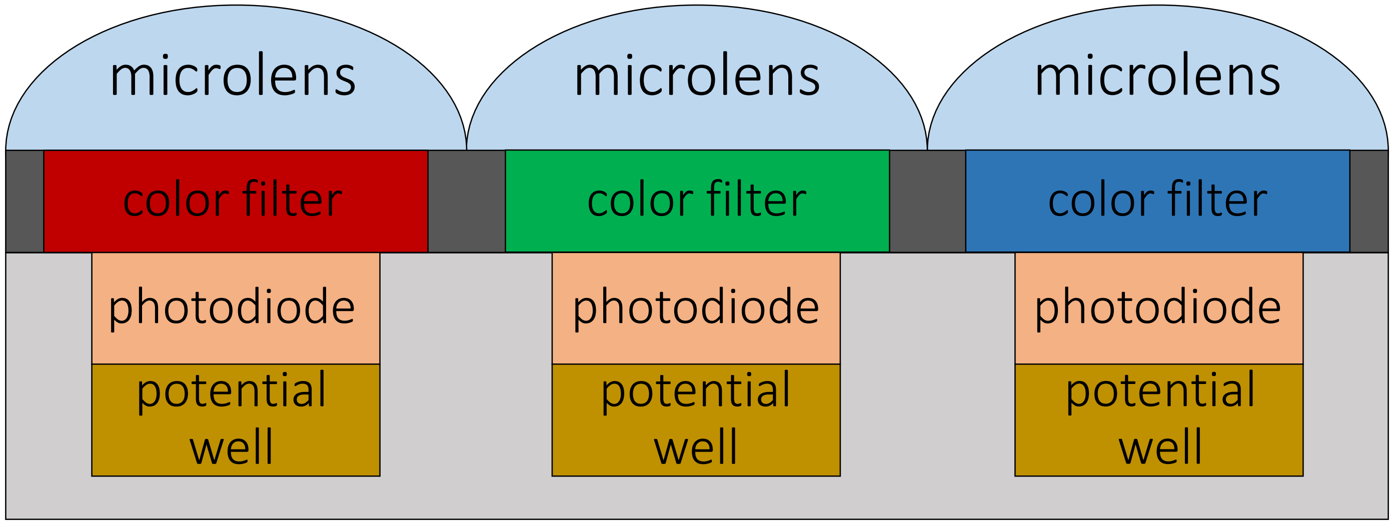
"medium" $$M = \int_\lambda \Phi(\lambda) M(\lambda) d\lambda$$

"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.

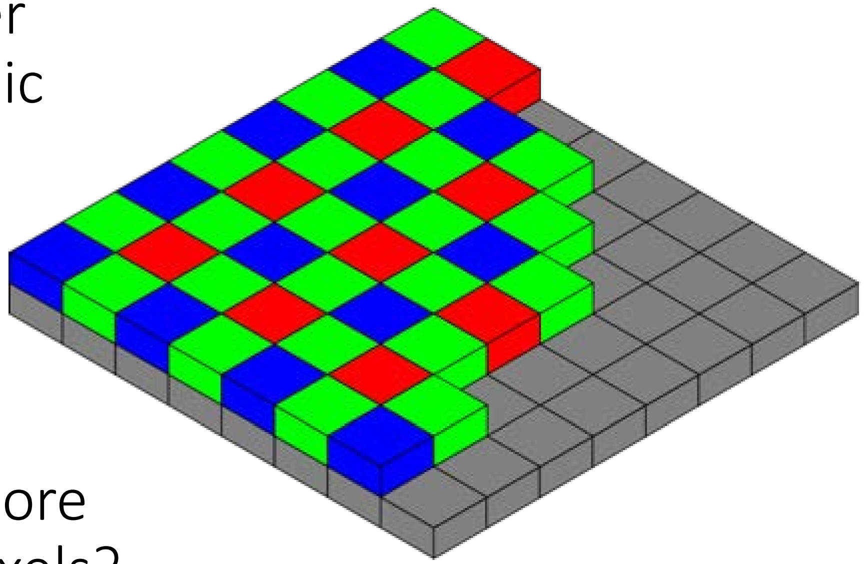


# What color filters to use?

Two design choices:

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

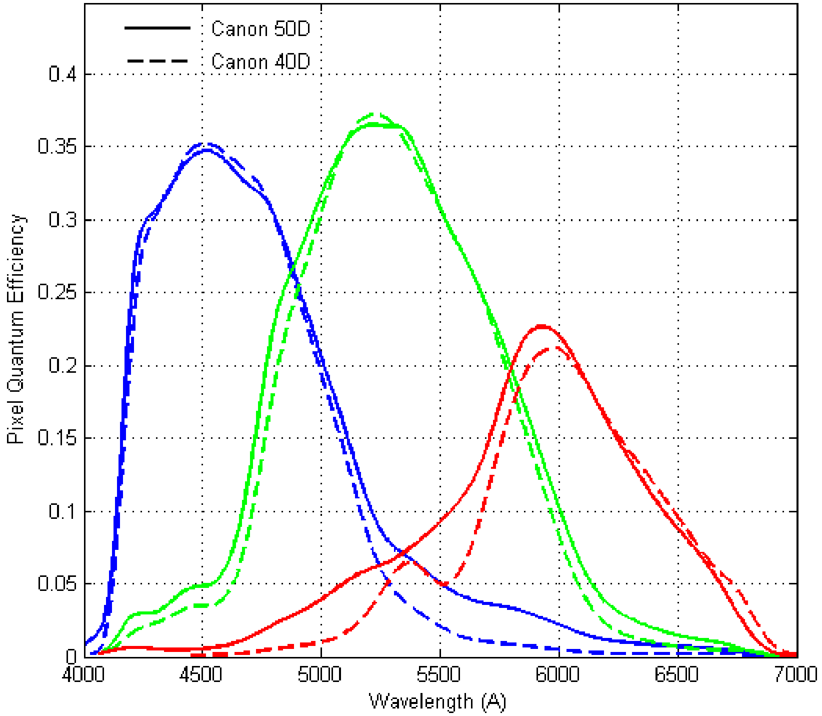
Bayer mosaic



Why more green pixels?

SSF for Canon 50D

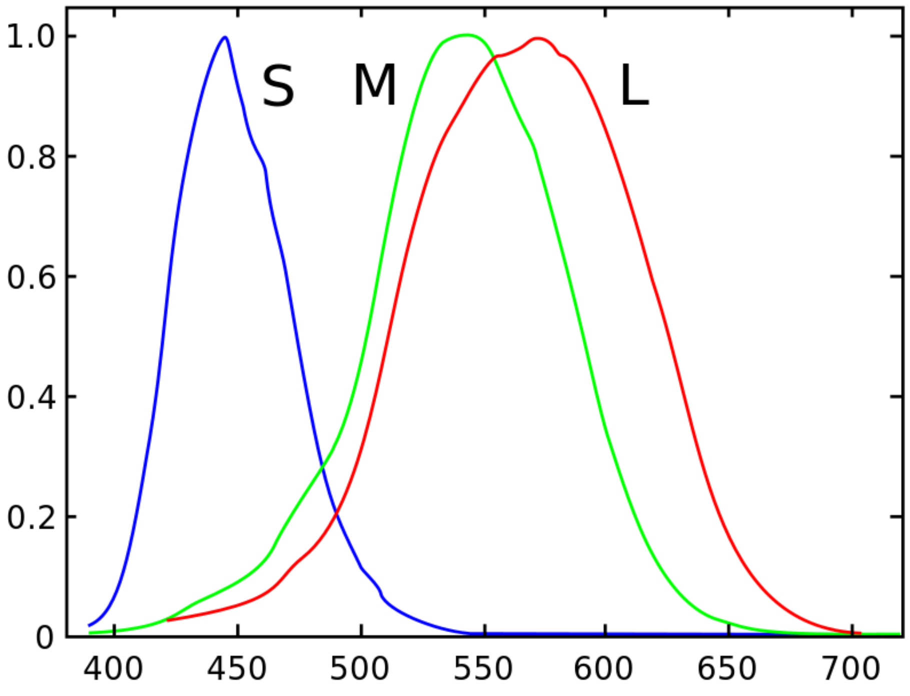
Generally do not match human LMS.




$f(\lambda)$

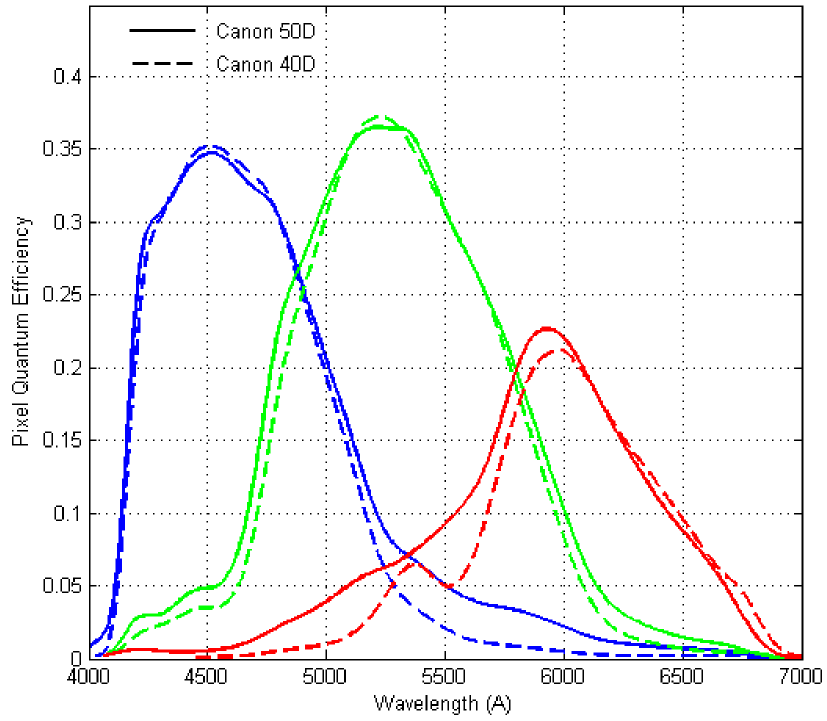
# What color filters to use?

Human




Generally do not match human LMS.

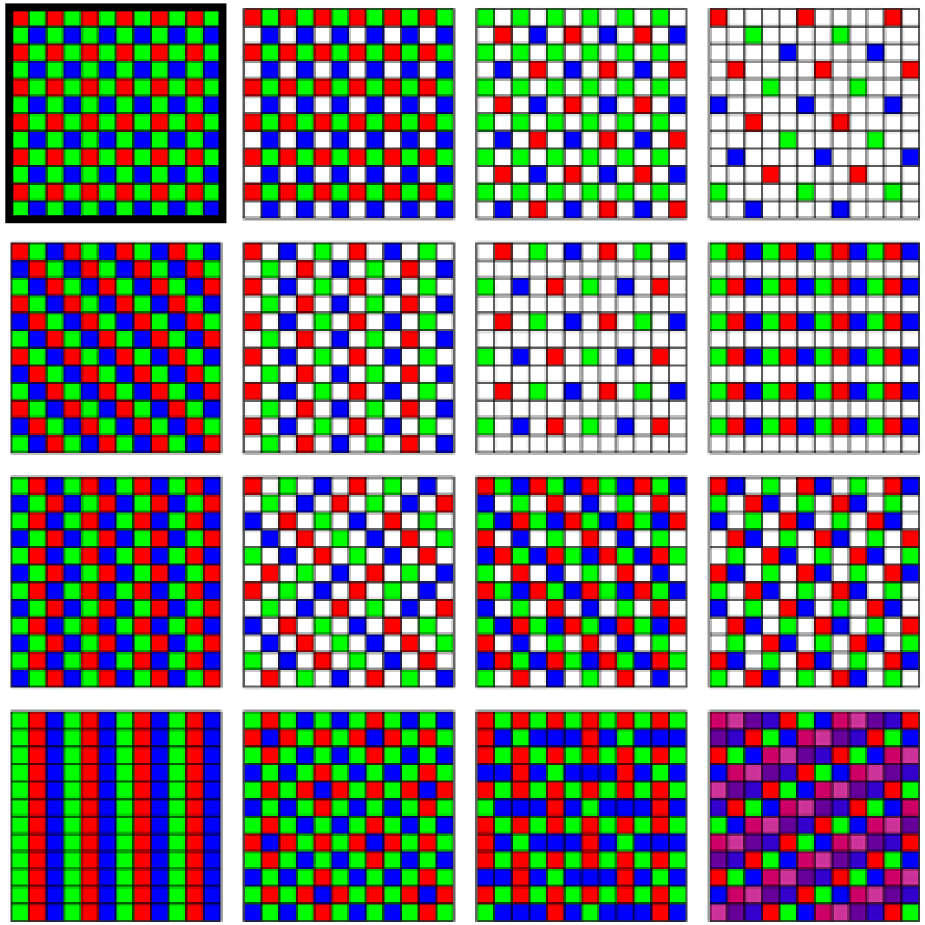
SSF for Canon 50D/ 40D

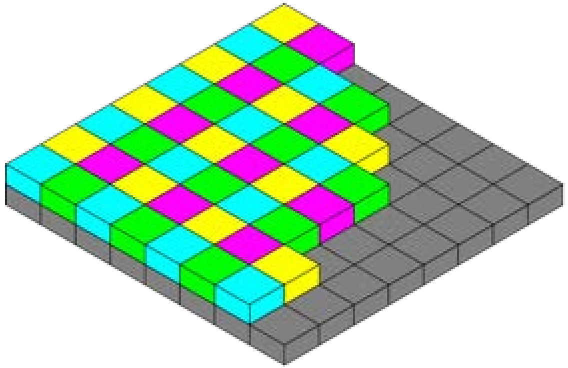



$f(\lambda)$

# Many different CFAs

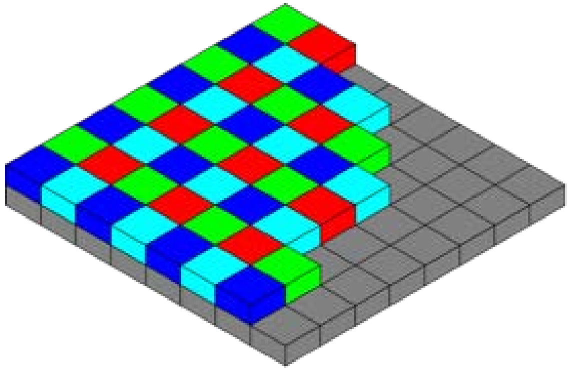
Finding the "best" CFA mosaic is an active research area.





CYGM

Canon IXUS, Powershot



RGBE

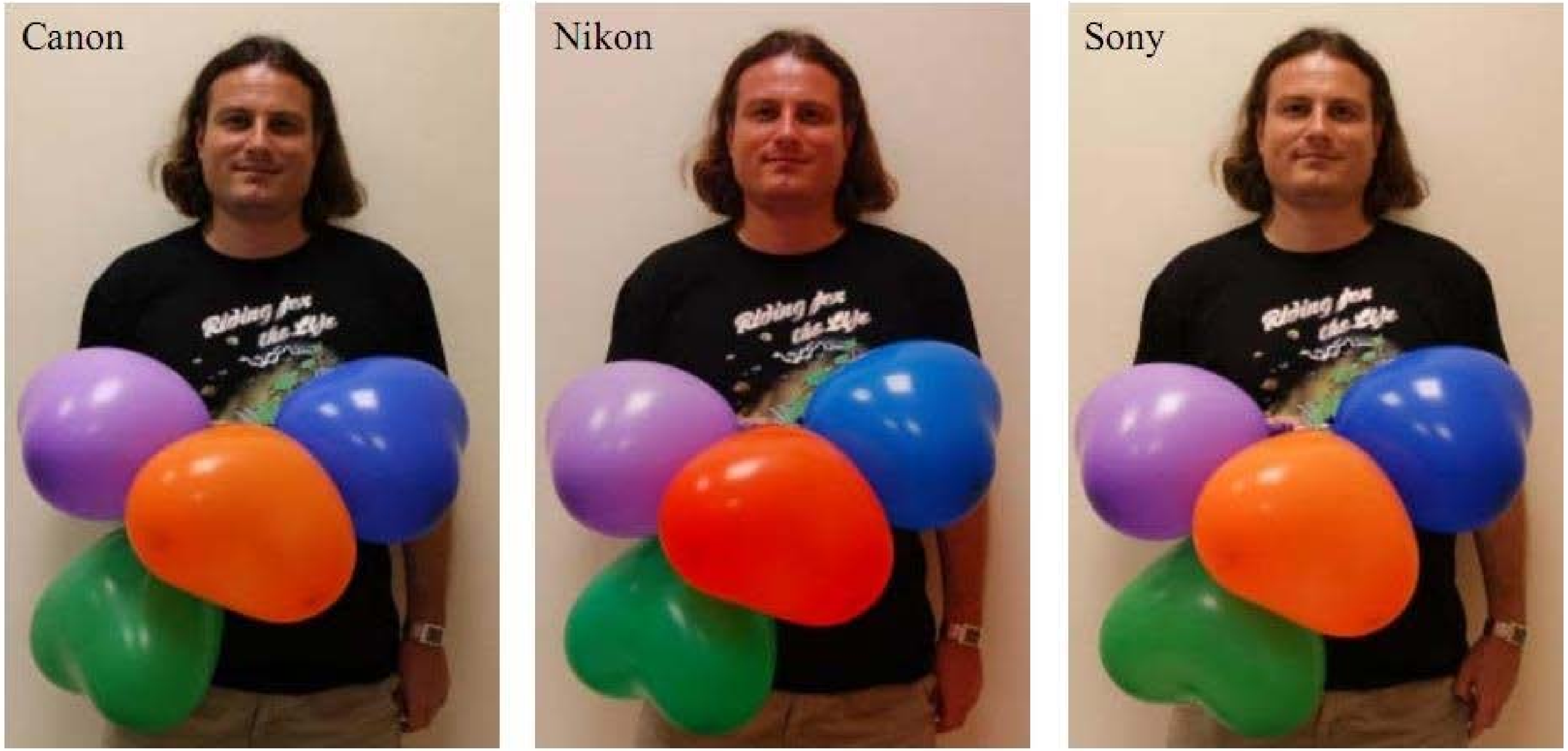
Sony Cyber-shot

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.

# Aside: can you think of other ways to capture color?

# What does an imaging sensor do?

When the camera shutter opens, the sensor:

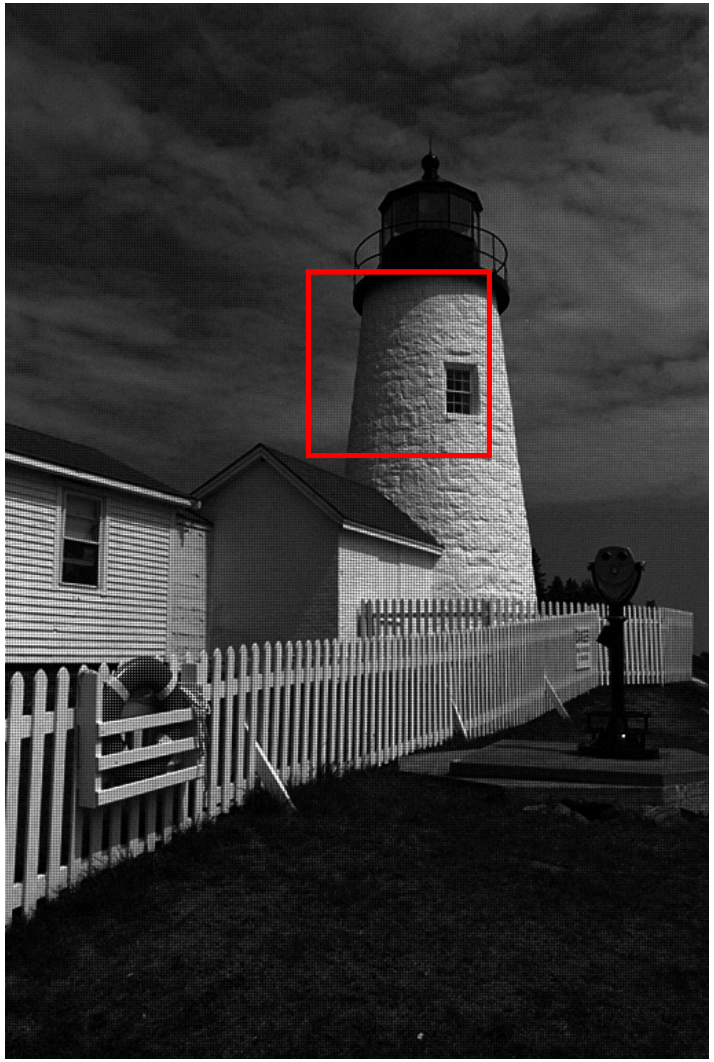
- at every photodiode, converts incident photons into electrons using mosaic's SSF
- stores electrons into the photodiode's potential well while it is not full

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

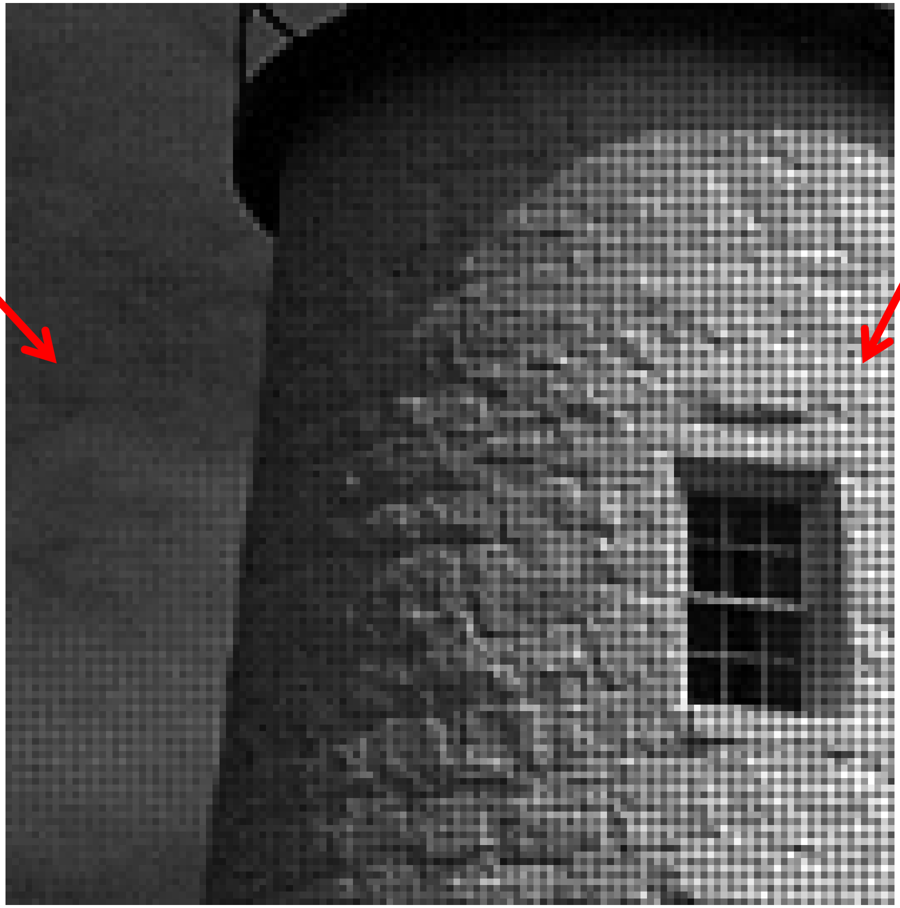
- reads out photodiodes' wells, row-by-row, and converts them to analog signals
- applies a (possibly non-uniform) gain to these analog signals
- converts them to digital signals
- corrects non-linearities

… and finally returns an image.

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



lots of noise

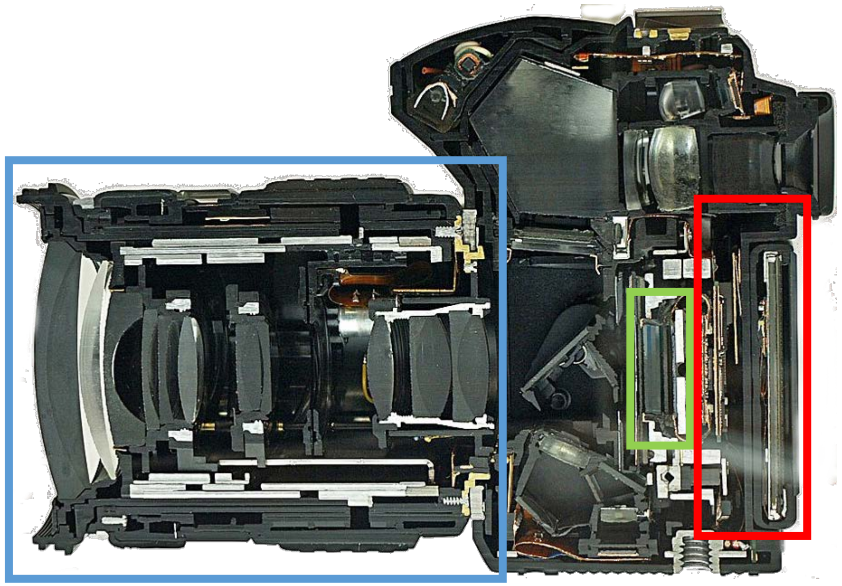


mosaicking artifacts

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

# The modern photography pipeline

post-capture processing

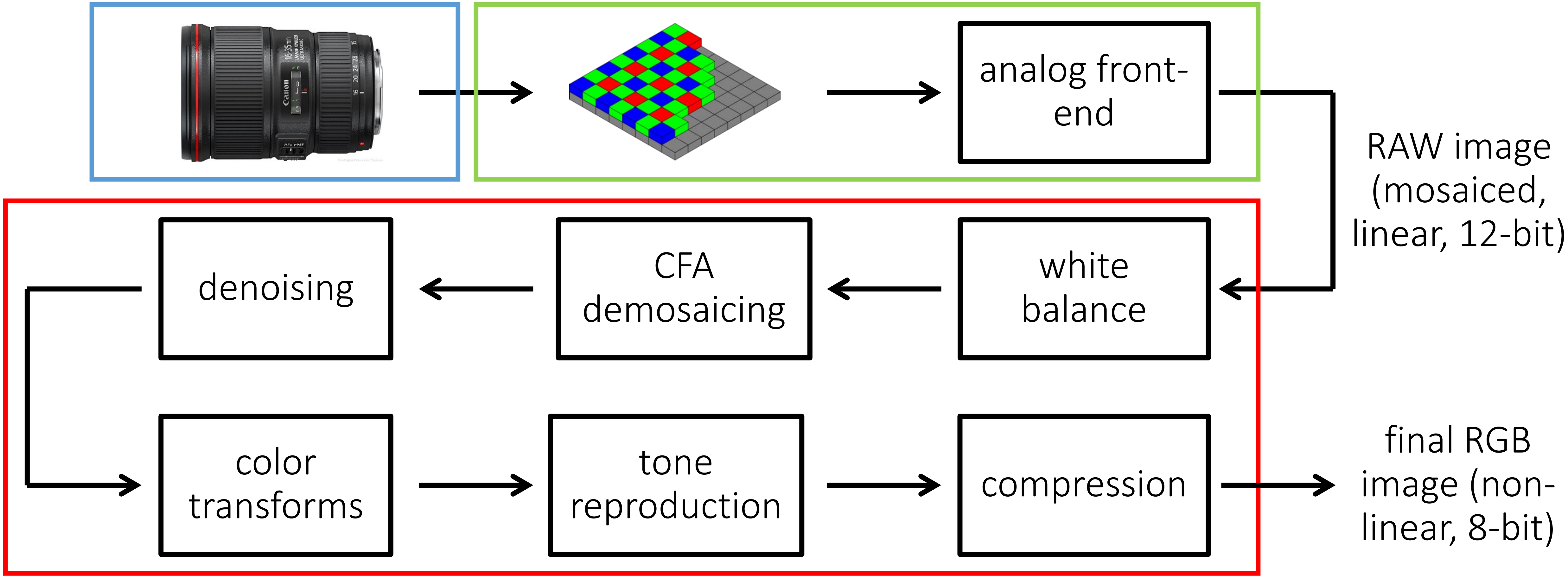


optics and optical controls → sensor, analog front-end, and color filter array → in-camera image processing pipeline

# 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

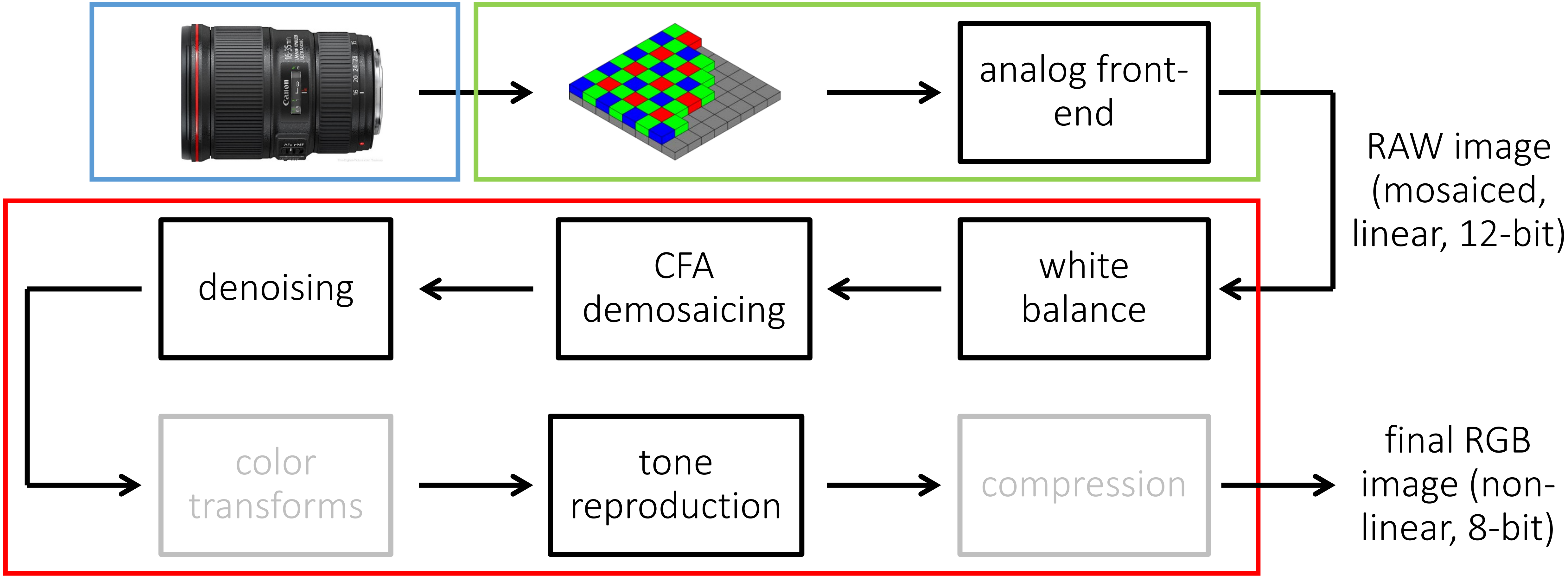
- Sometimes the term *image signal processor* (ISP) is used to refer to the image processing pipeline itself.

- 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*.

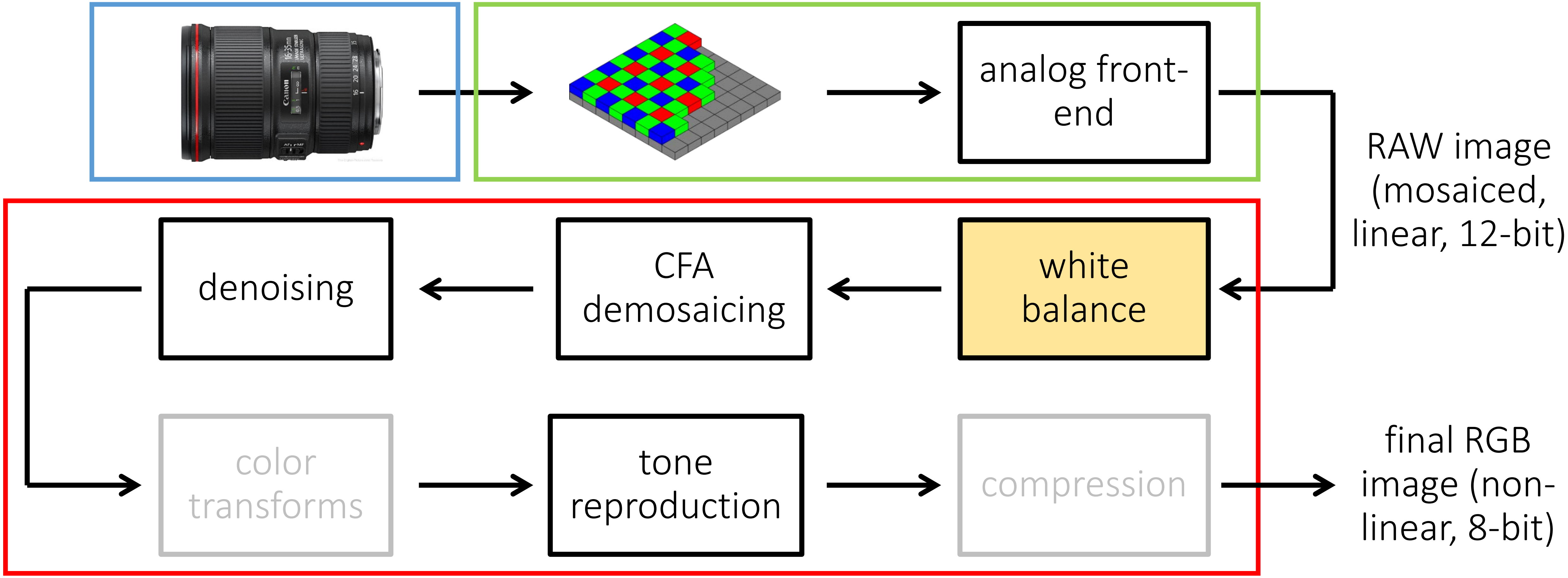
# The (in-camera) image processing pipeline

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.

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

# Sensor integration

$$I_c^{camera} = \int_\lambda \Phi(\lambda) f_c^{camera}(\lambda)$$

light SPD

sensor SSF at color channel c

$$I_c^{camera} = \int_\lambda r(\lambda) e(\lambda) f_c^{camera}(\lambda)$$

Material reflectance

illuminant spectrum

$$I_c^{eye} = \int_\lambda \Phi(\lambda) f_c^{eye}(\lambda)$$

**Can we match?**

$$I_c^{eye} = ?s\, I_c\, I_c^{camera}$$

# White balancing

Find a diagonal matrix satisfying

white-balanced RGB $\longrightarrow$ 

$$\begin{bmatrix} R_p^{world} \\ G_p^{world} \\ B_p^{world} \end{bmatrix} = \begin{bmatrix} S_R & 0 & 0 \\ 0 & S_G & 0 \\ 0 & 0 & S_B \end{bmatrix} \begin{bmatrix} R_p^{camera} \\ G_p^{camera} \\ B_p^{camera} \end{bmatrix}$$

$\longleftarrow$ sensor RGB

For all image pixels p

Question: can we indeed explain all image pixels by the same 3 numbers?

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

White world assumption:

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

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

# Automatic white balancing example

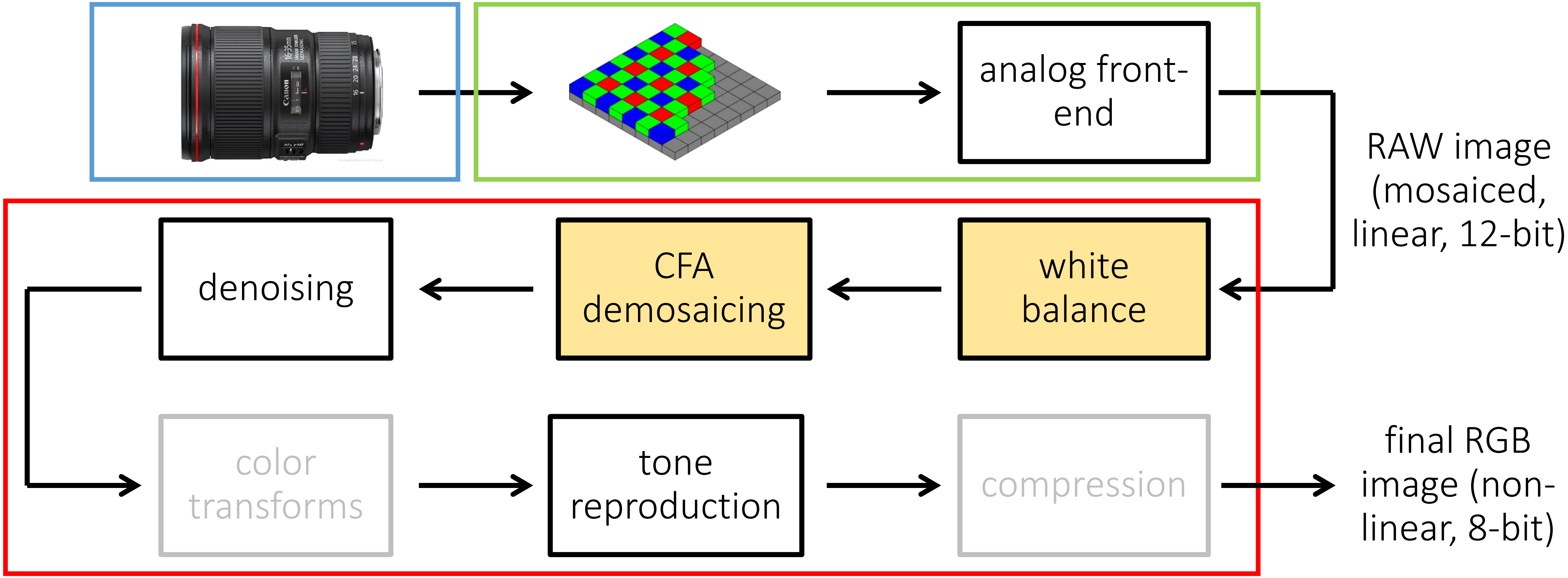
input image

grey world

white world

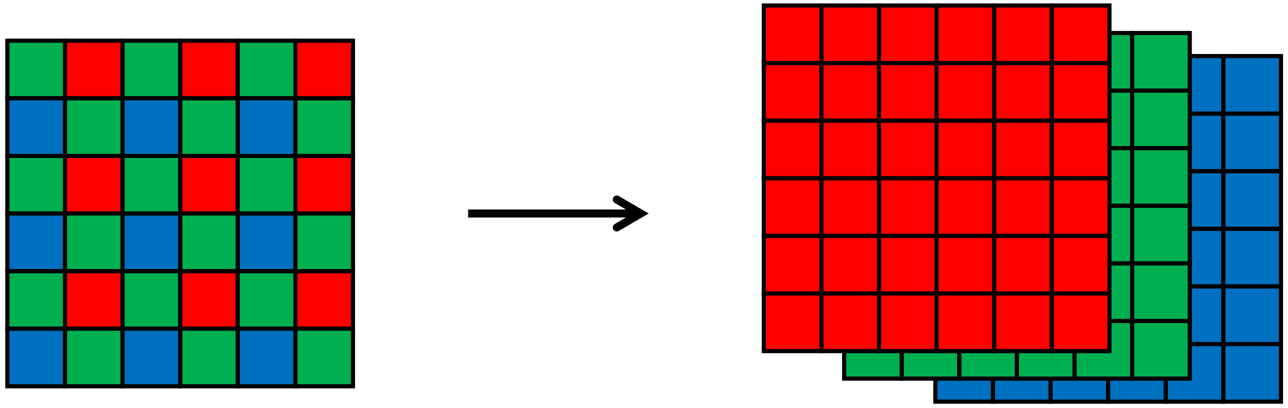
# The (in-camera) image processing pipeline

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

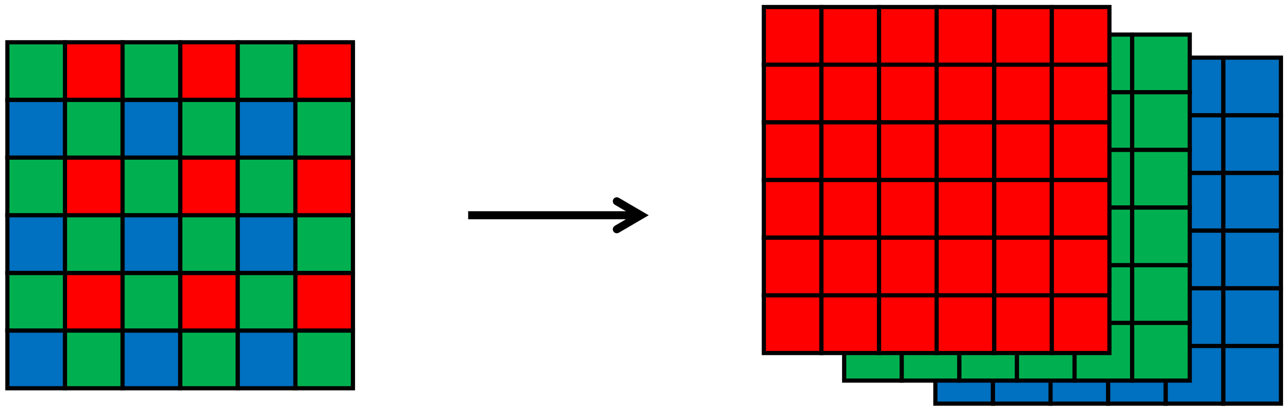
Produce full RGB image from mosaiced sensor output.



Any ideas on how to do this?

# CFA demosaicing

Produce full RGB image from mosaiced sensor output.

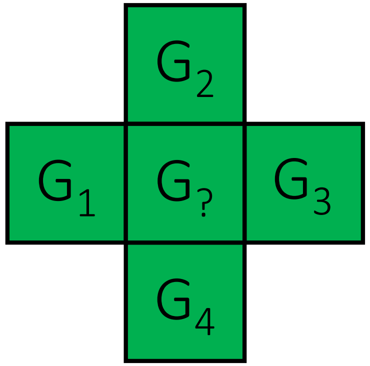


Interpolate from neighbors:

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

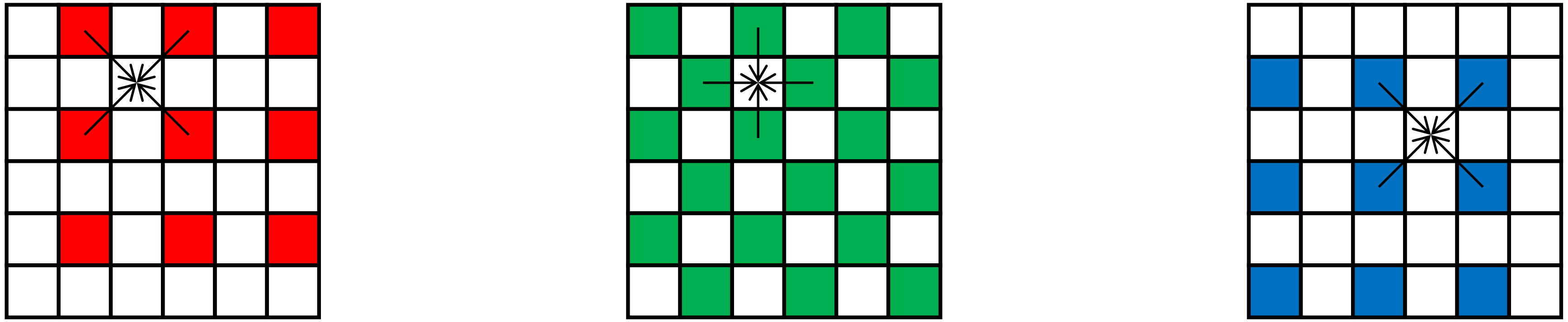
# Demosaicing by bilinear interpolation

Bilinear interpolation: Simply average your 4 neighbors.



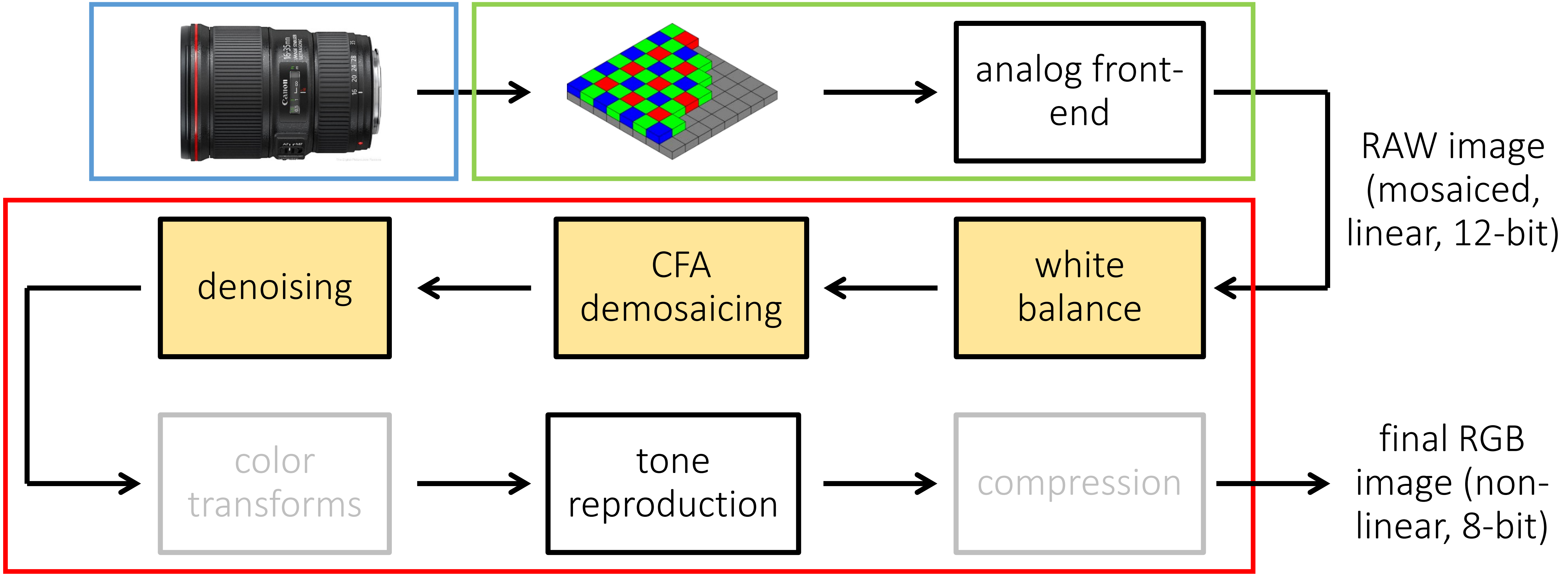

$$G_? = \frac{G_1 + G_2 + G_3 + G_4}{4}$$

Neighborhood changes for different channels:



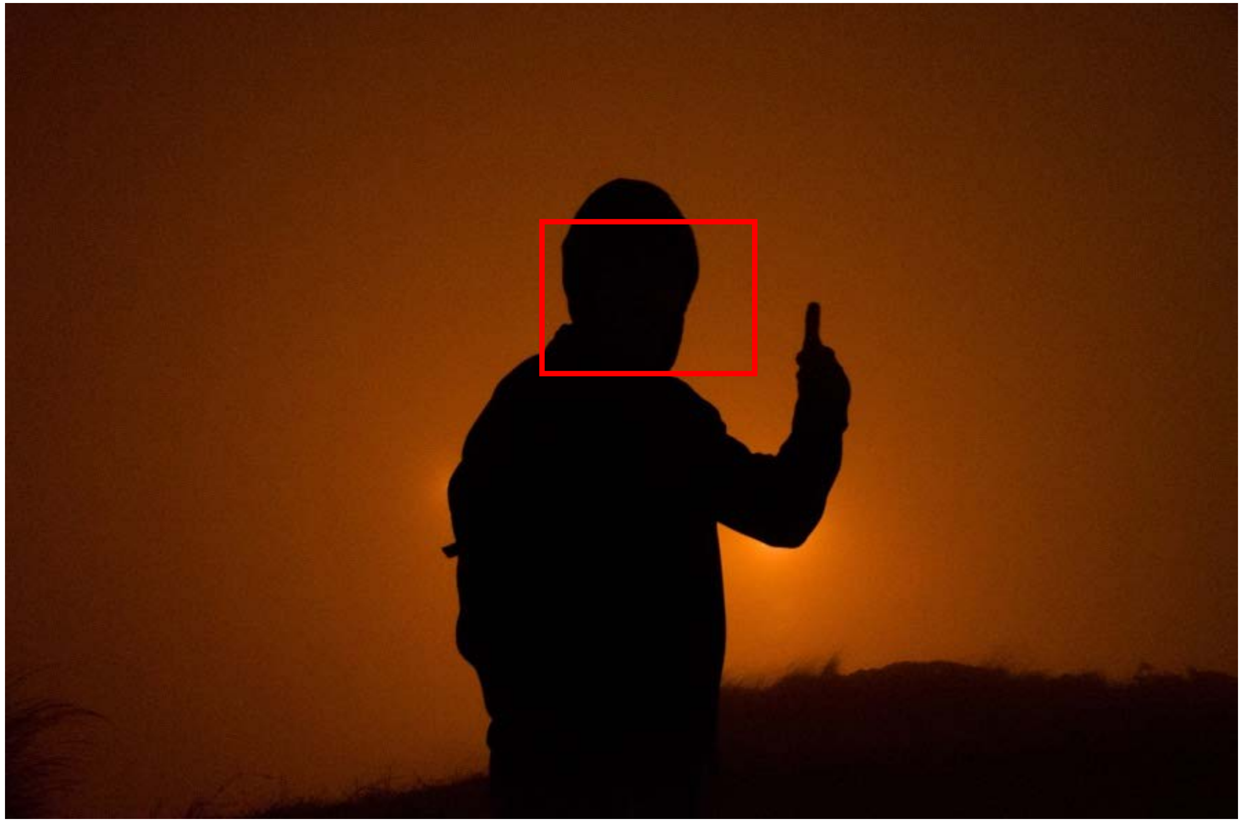
# The (in-camera) image processing pipeline

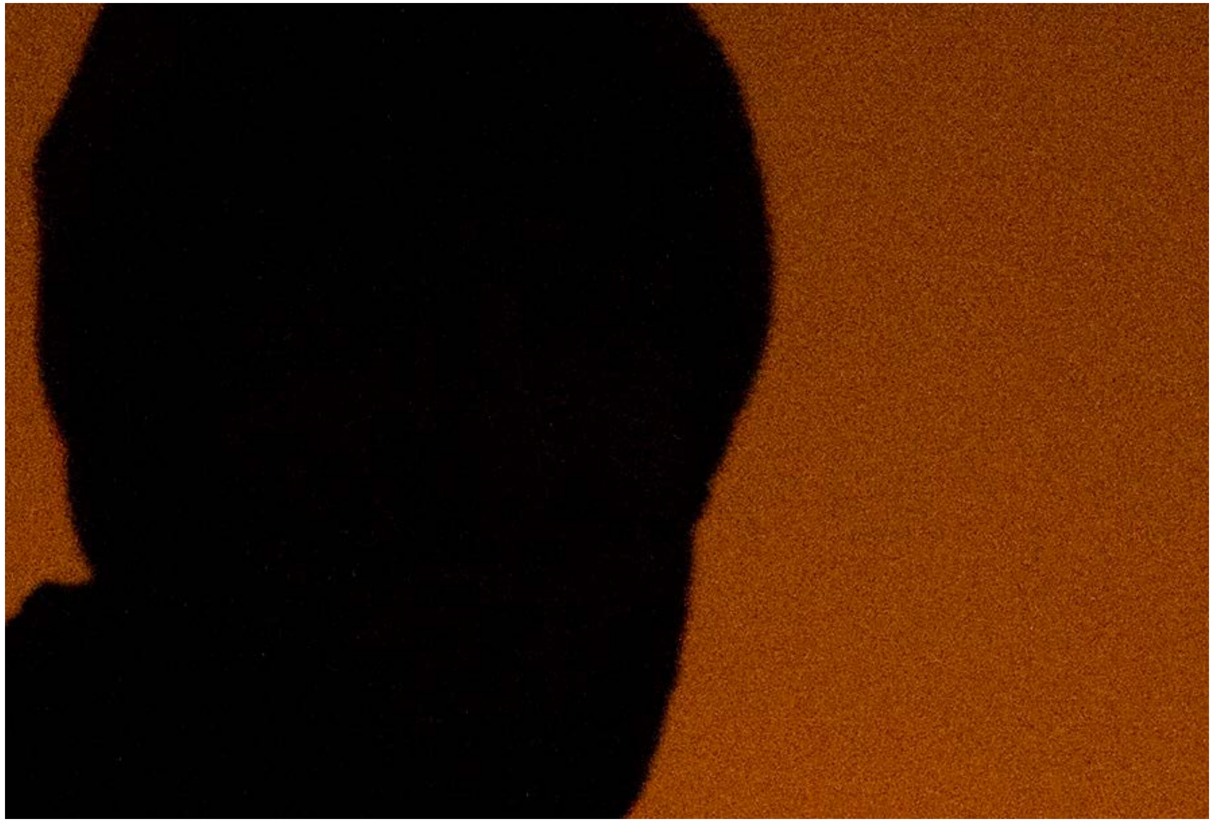
The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



# 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 smaller the variance of the distribution.

2) Dark-shot noise:

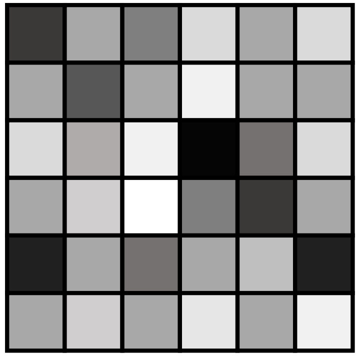
- Emitted electrons due to thermal activity (becomes worse as sensor gets hotter.)

3) Read noise:

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

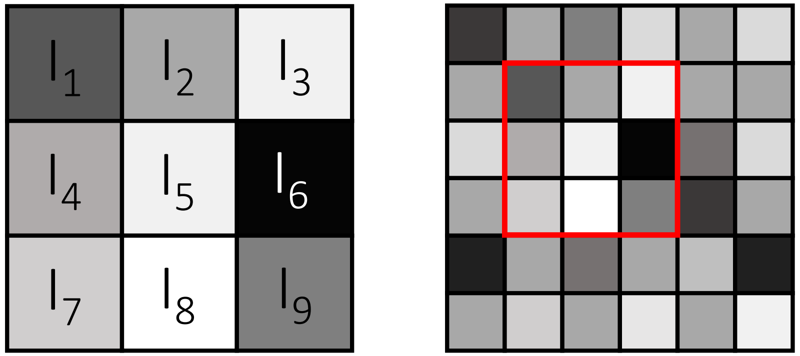
Bright scene and large pixels: photon shot noise is the main noise source.

# How to denoise?



# How to denoise?

Look at the neighborhood around you.




- Mean filtering (take average):

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

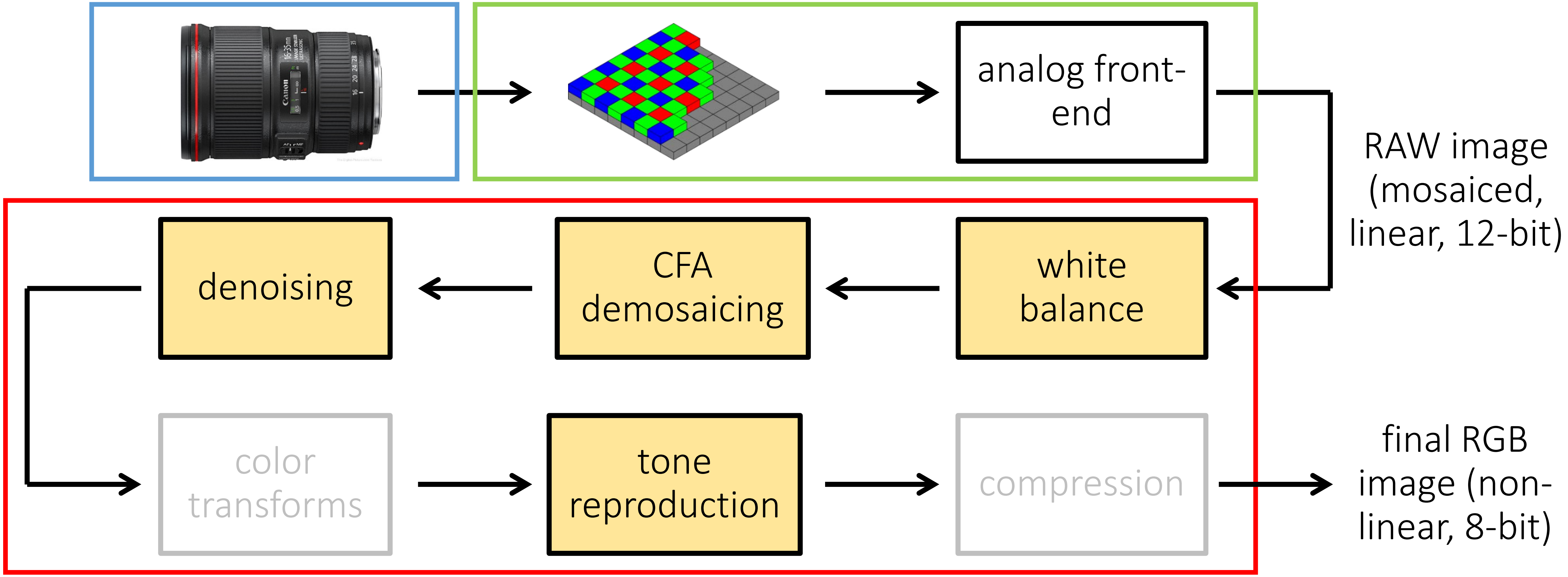
- Median filtering (take median):

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

Large area of research. Covered in may other classes.

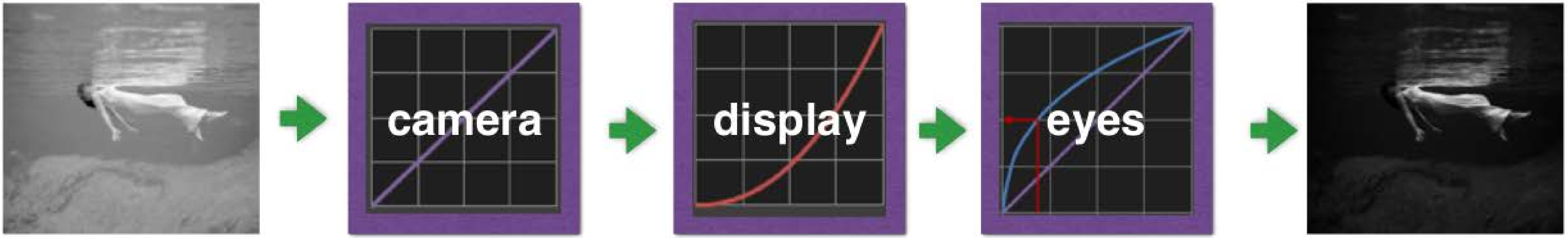
# The (in-camera) image processing pipeline

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



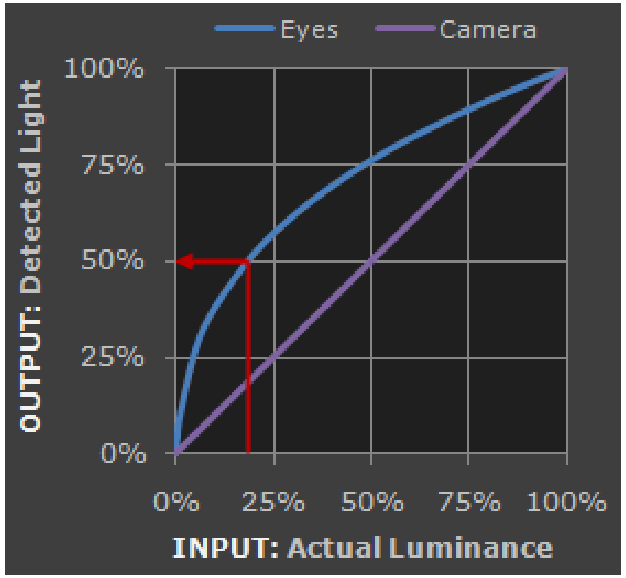
# Tone reproduction

- Also known as gamma encoding (and erroneously as gamma correction).
- Without tone reproduction, images look very dark.




Why does this happen?

# Perceived vs measured brightness by human eye



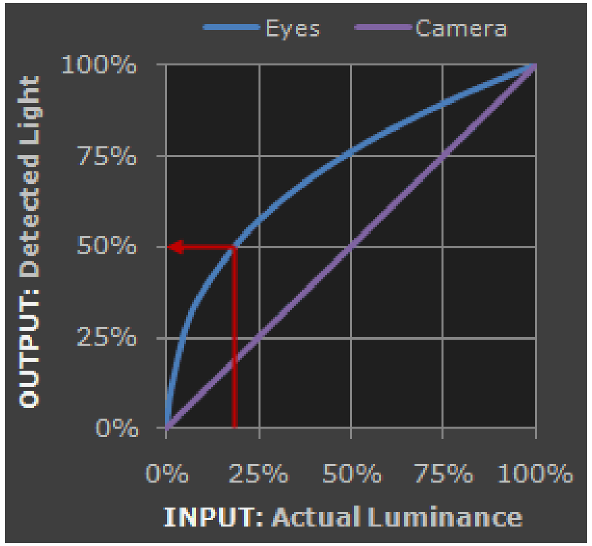

We have already seen that sensor response is linear.

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

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

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

# What about displays?

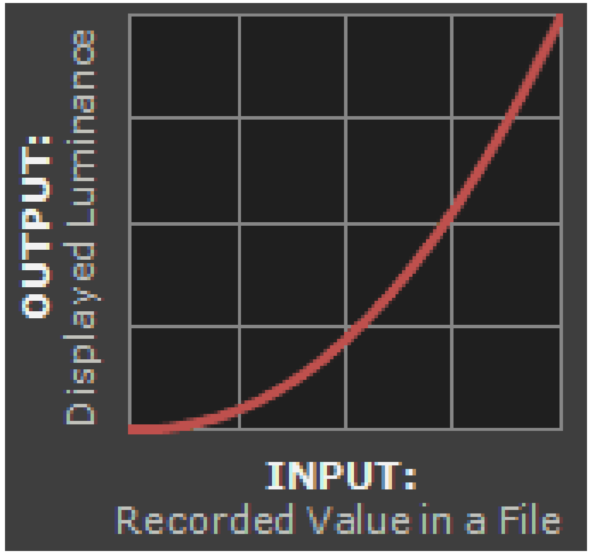



We have already seen that sensor response is linear.

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

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

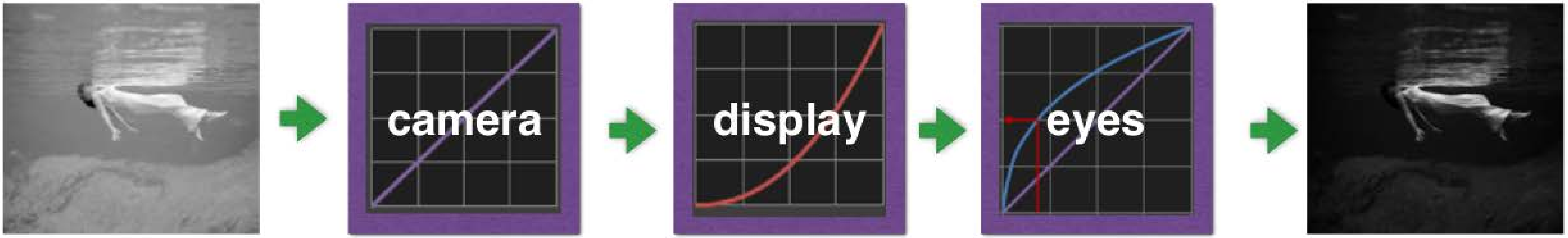
- More sensitive to dark tones.
- Approximately a Gamma function.




Displays have a response opposite to that of human perception.

# Tone reproduction

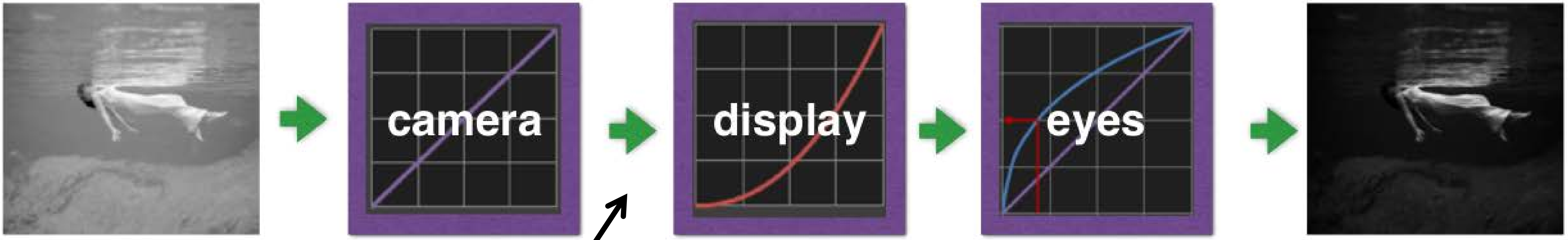
- Because of mismatch in displays and human eye perception, images look very dark.

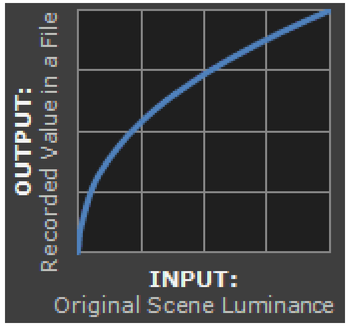



How do we fix this?

# Tone reproduction

- Because of mismatch in displays and human eye perception, images look very dark.




- Pre-emptively cancel-out the display response curve.
- Add inverse display transform here.
- This transform is the tone reproduction or gamma correction.

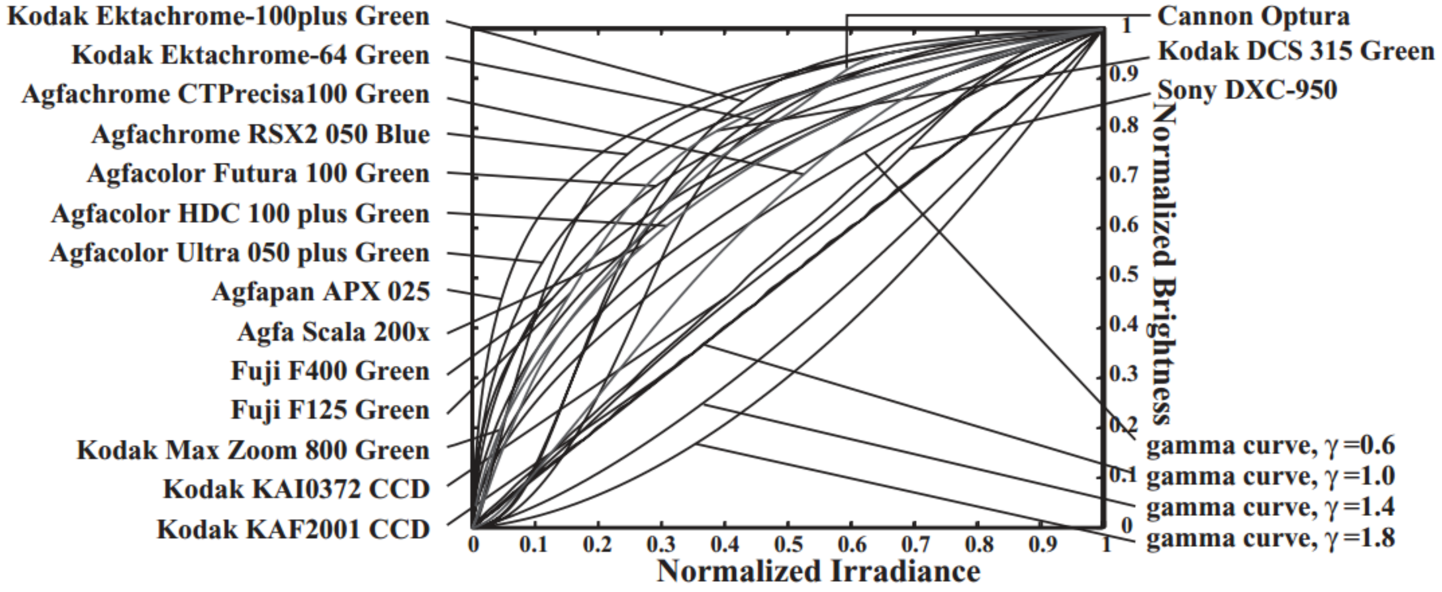
# Tone reproduction curves

The exact tone reproduction curve depends on the camera.

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

before gamma

after gamma



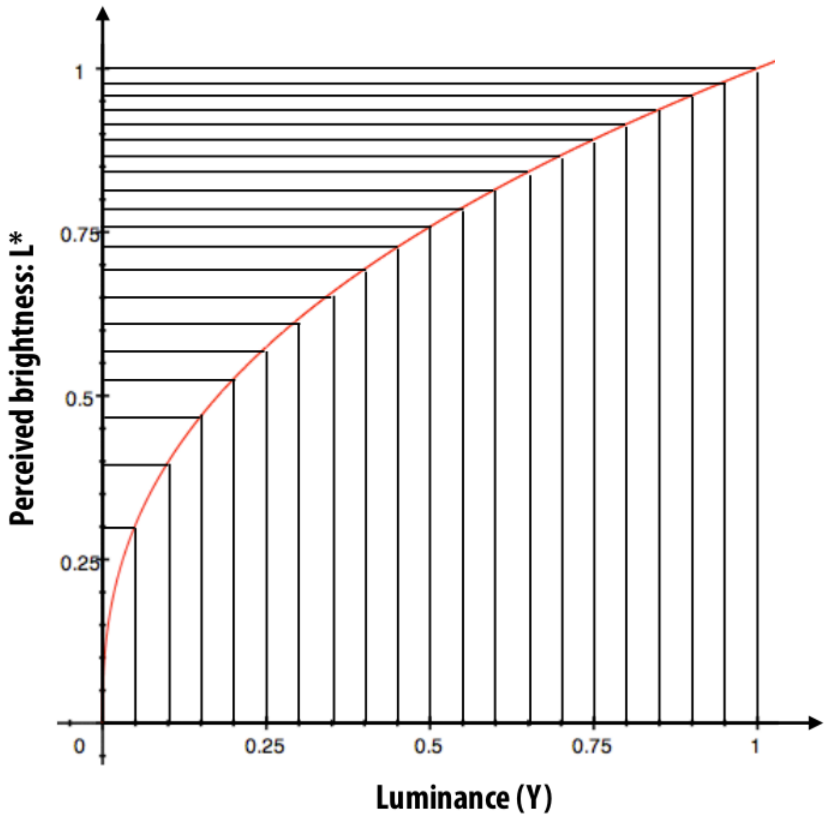

Warning: Our values are no longer linear relative to scene radiance!

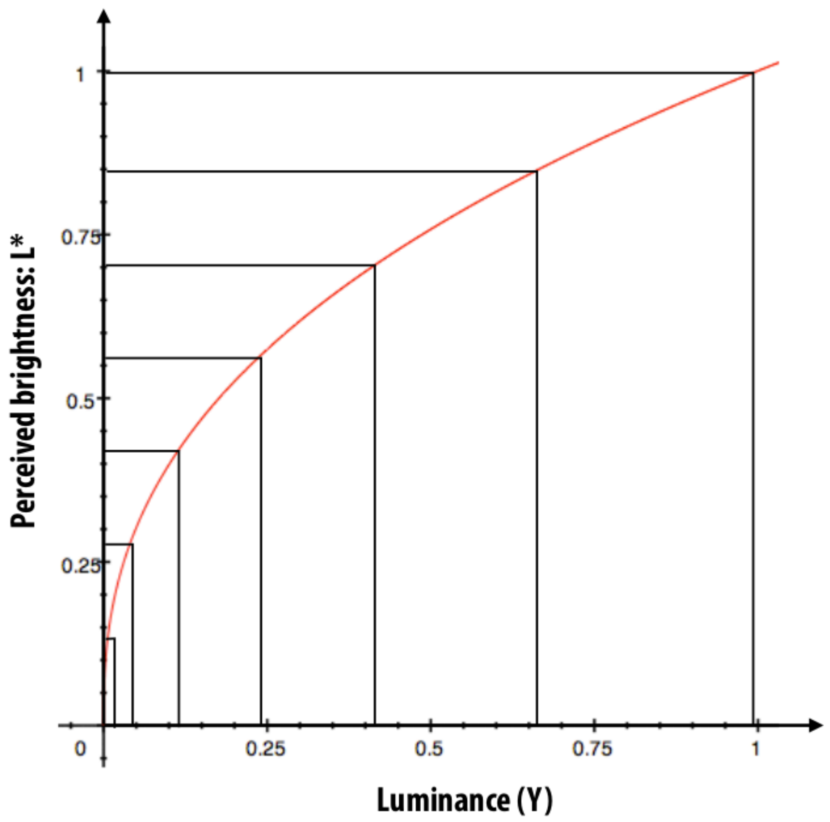
# Tone reproduction

Question: Why not just keep measurements linear and do gamma correction right before we display the image?

# Tone reproduction

Question: Why not just keep measurements linear and do gamma correction right before we display the image?



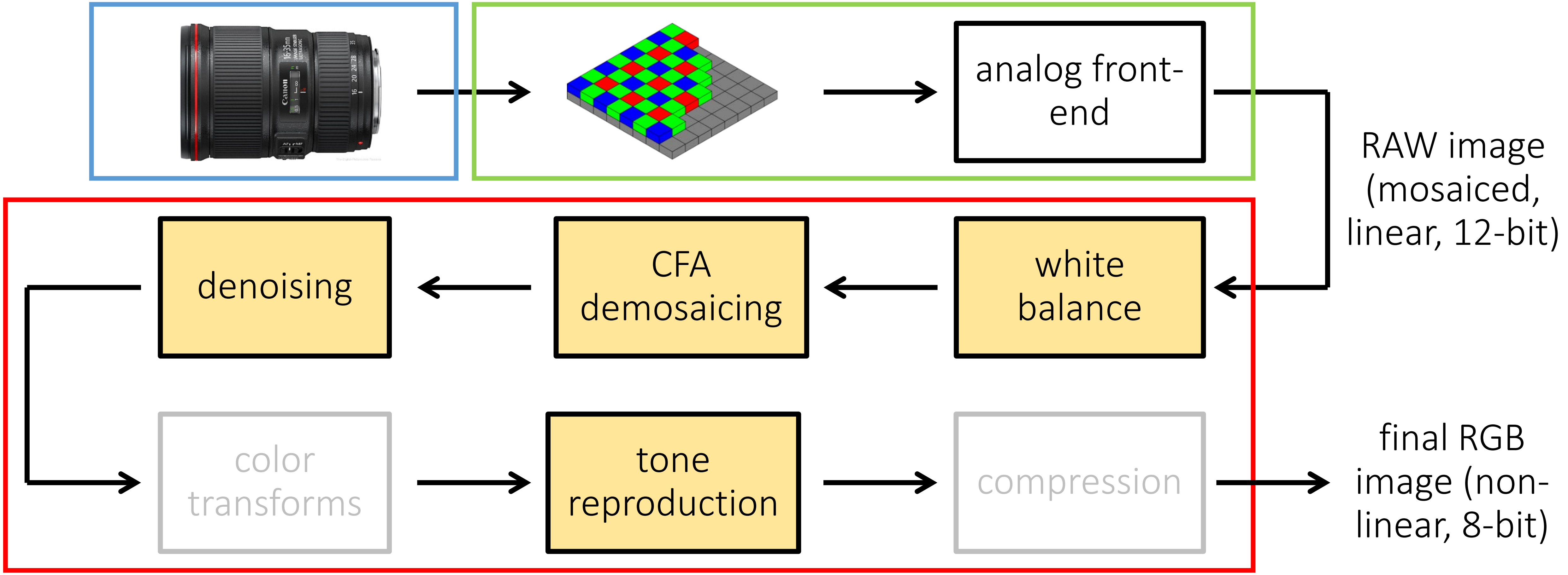




Answer: After this stage, we perform compression, which includes change from 12 to 8 bits.

- Better to use our available bits to encode the information we are going to need.

# The (in-camera) image processing pipeline

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



# Take-home messages

The values of pixels in a photograph and the values output by your camera’s sensor are two very different things.

The relationship between the two is complicated and unknown.

# References

Basic reading:

- Szeliski textbook, Section 2.3.
- Michael Brown, "Understanding the In-Camera Image Processing Pipeline for Computer Vision," CVPR 2016,
  slides available at: http://www.comp.nus.edu.sg/~brown/CVPR2016_Brown.html

Additional reading:

- Adams et al., "The Frankencamera: An Experimental Platform for Computational Photography," SIGGRAPH 2010.
  The first open architecture for the image processing pipeline, and precursor to the Android Camera API.
- Heide et al., "FlexISP: A Flexible Camera Image Processing Framework," SIGGRAPH Asia 2014.
  Discusses how to implement a single-stage image processing pipeline.
- Buckler et al., "Reconfiguring the Imaging Pipeline for Computer Vision," ICCV 2017.
- Diamond et al., "Dirty Pixels: Optimizing Image Classification Architectures for Raw Sensor Data," arXiv 2017.
  Both papers discuss how to adaptively change the conventional image processing pipeline so that it is better suited to various computer vision problems.
- Chakrabarti et al., "Rethinking Color Cameras," ICCP 2014.
  Discusses different CFAs, including ones that have white filters, and how to do demosaicing for them.
- Gunturk et al., "Demosaicking: Color Filter Array Interpolation," IEEE Signal Processing Magazine 2005
  A nice review of demosaicing algorithms.
- Chakrabarti et al., "Probabilistic Derendering of Camera Tone-mapped Images," PAMI 2014.
  Discusses how to (attempt to) derender an image that has already gone through the image processing pipeline of some (partially calibrated) camera.