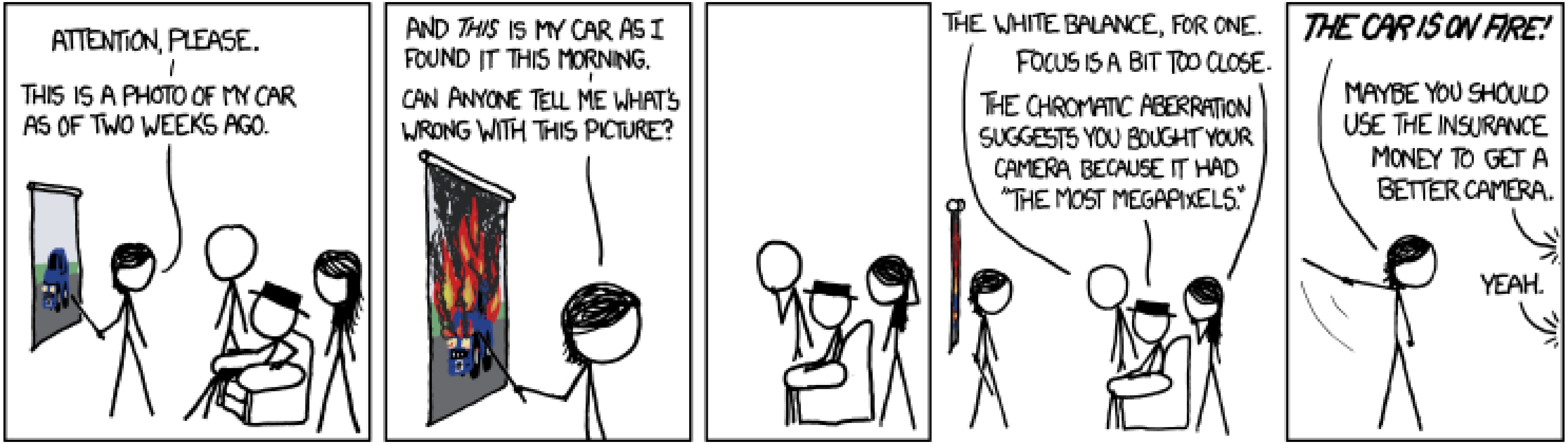
Digital photography



http://www.cs.cmu.edu/~16385/

16-385 Computer Vision
Spring 2019, Lecture 16

Course announcements

- Homework 4 is ongoing.
  - Due Wednesday March 27th.
  - Any questions about the homework?
  - How many of you have looked at/started/finished homework 4?
- Extra office hours this week by Yannis:
  - Tuesday, 3-5 pm.
  - Wednesday, 4-8 pm.
  - All on graphics lounge/my office.
- Talks last week:
  - How many of you attended Jun-Yan Zhu's talk?
  - How many of you attended Angjoo Kanazawa's talk?
- Talks this week:
  - Abe Davis, "Augmenting Imagination: Capturing, Modeling, and Exploring the World Through Video," Monday 2:00 PM GHC 6115.
  - Matthias Niessner, "AI-Driven Videos Synthesis and its Implications," Wednesday, 12:00 PM GHC 6115.
  - Angela Dai, "Understanding 3D Scans," Thursday 12:00 PM GHC 6115.

Overview of today’s lecture

- Imaging sensor primer.
- Color sensing in cameras.
- In-camera image processing pipeline.
- Some general thoughts on the image processing pipeline.
- Radiometric calibration (a.k.a. HDR imaging).

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

Slide credits

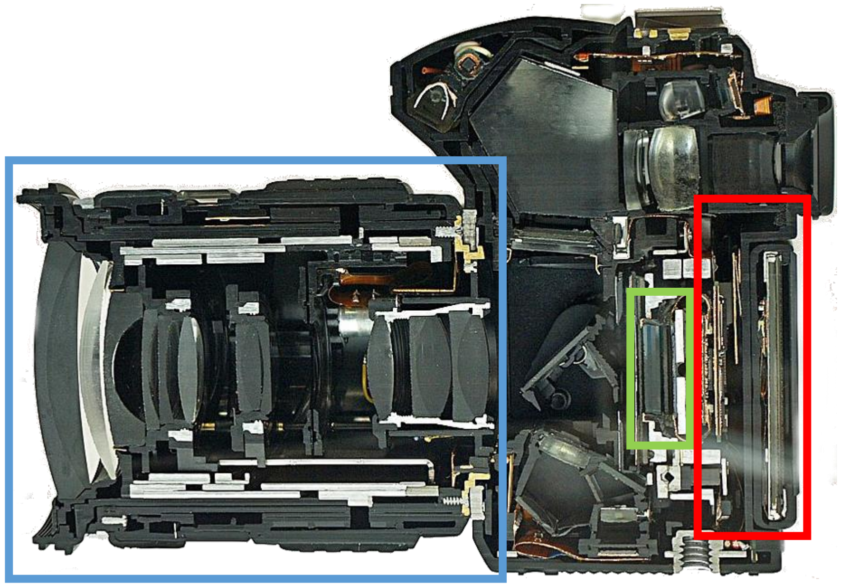
A lot of inspiration and quite a few examples for these slides were taken directly from:

- Kayvon Fatahalian (15-769, Fall 2016).
- Michael Brown (CVPR 2016 Tutorial on understanding the image processing pipeline).

The modern photography pipeline

The modern photography pipeline

post-capture processing
(16-385, 15-463)



optics and optical controls

(15-463)

sensor, analog front-end, and color filter array

(this lecture)

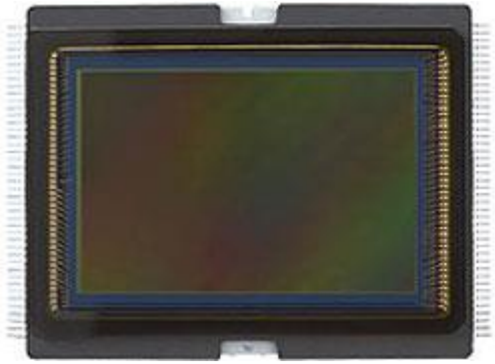
in-camera image processing pipeline

(this lecture)

Imaging sensor primer

Imaging sensors

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

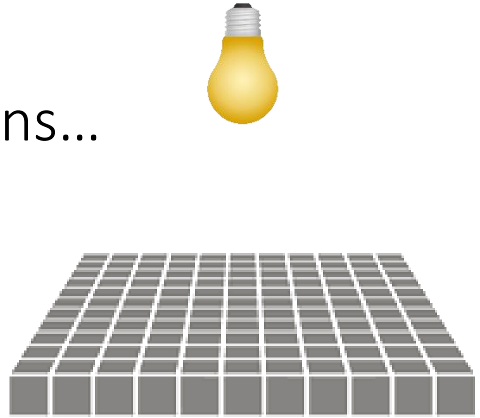


Canon 6D sensor
(20.20 MP, full-frame)

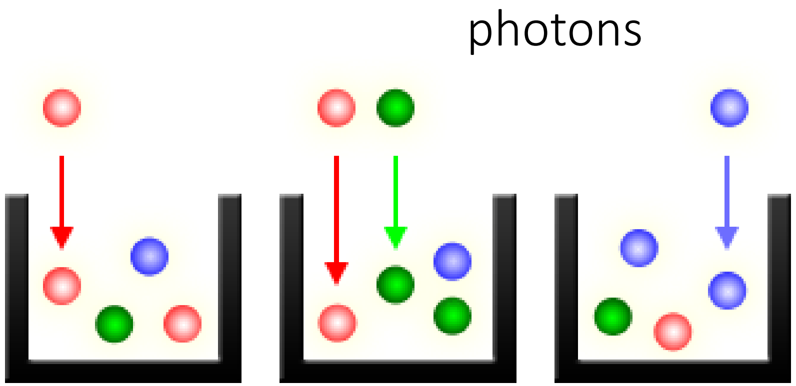
What does an imaging sensor do?

When the camera shutter opens…

… exposure begins…



array of photon buckets

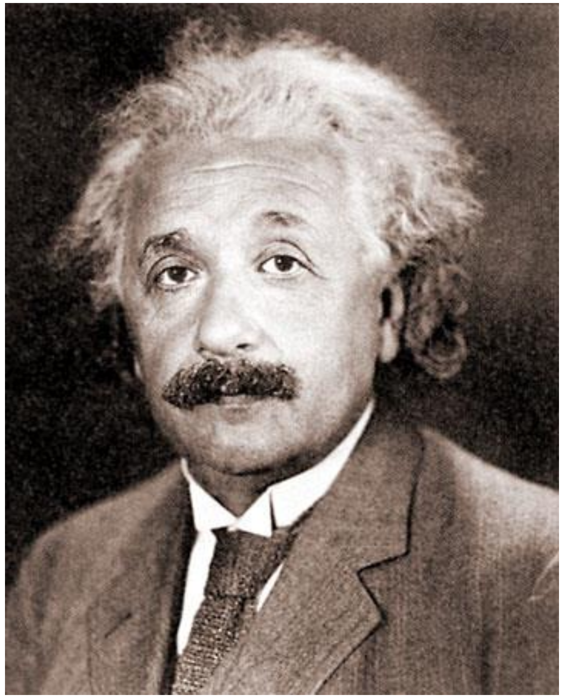


close-up view of photon buckets

… photon buckets begin to store photons...

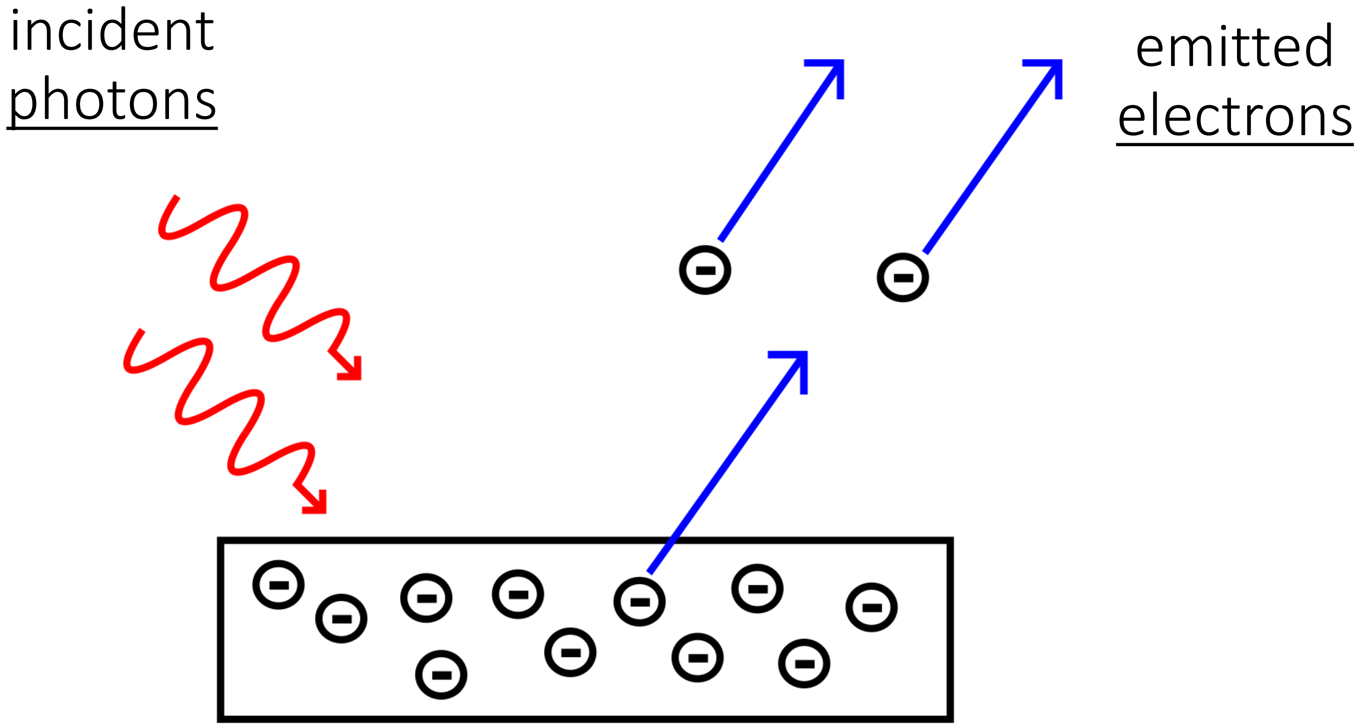
… until the camera shutter closes. Then, they convert stored photons to intensity values.

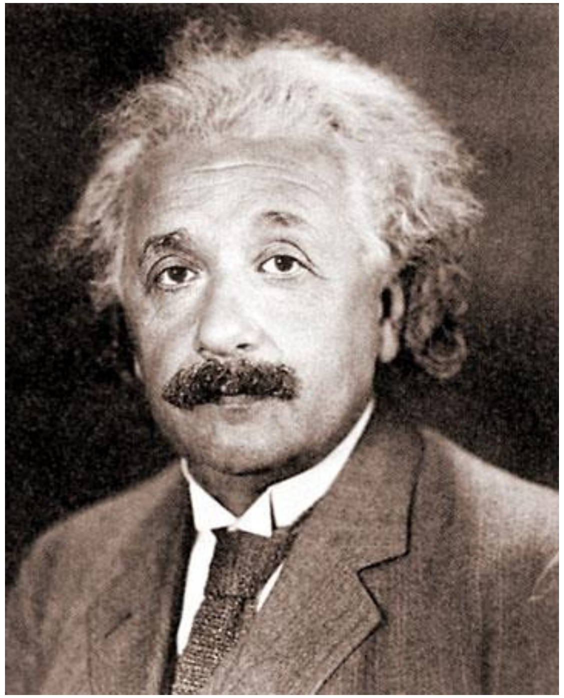
Nobel Prize in Physics



What is this guy known for?

Photoelectric effect

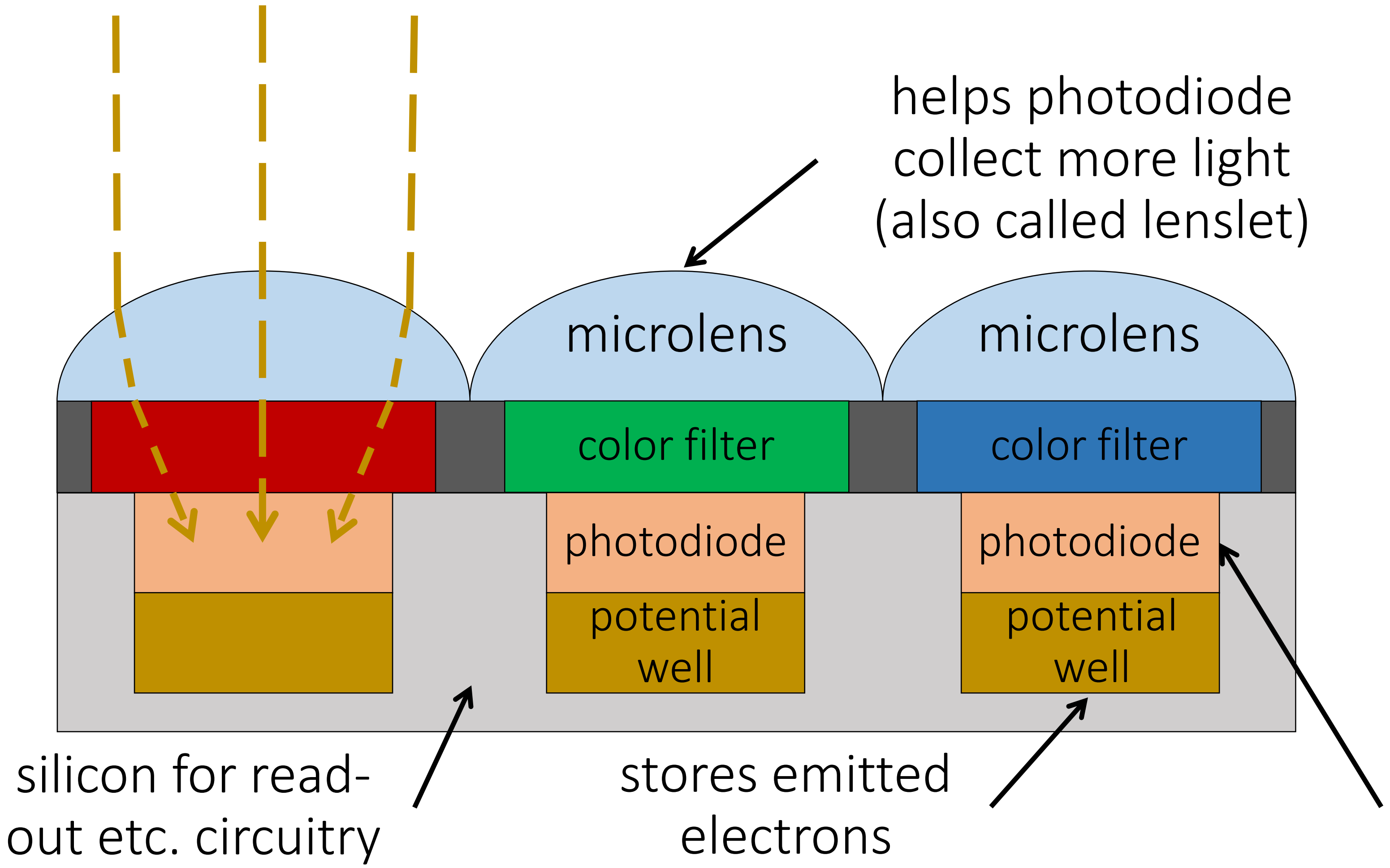




Albert Einstein

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



- 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.

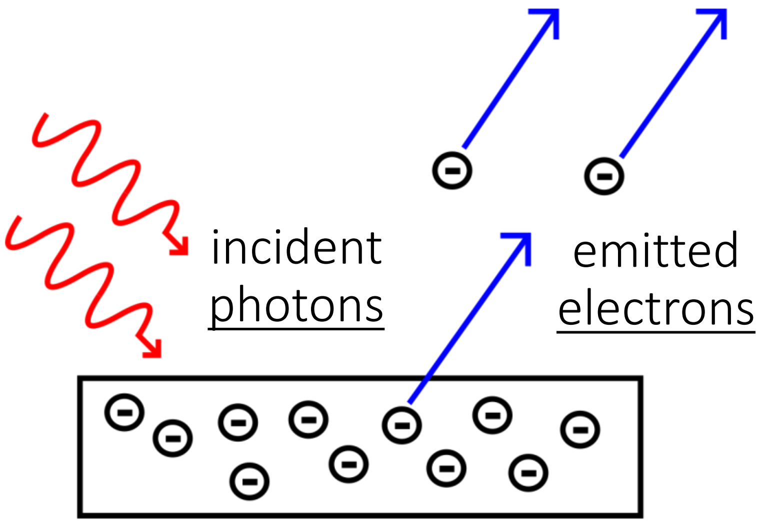
We will discuss these issues in more detail at a later lecture.

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

Photodiode quantum efficiency (QE)

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

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



- Fundamental optical performance metric of imaging sensors.
- Not the only important optical performance metric!
- We will see a few more later in the lecture.

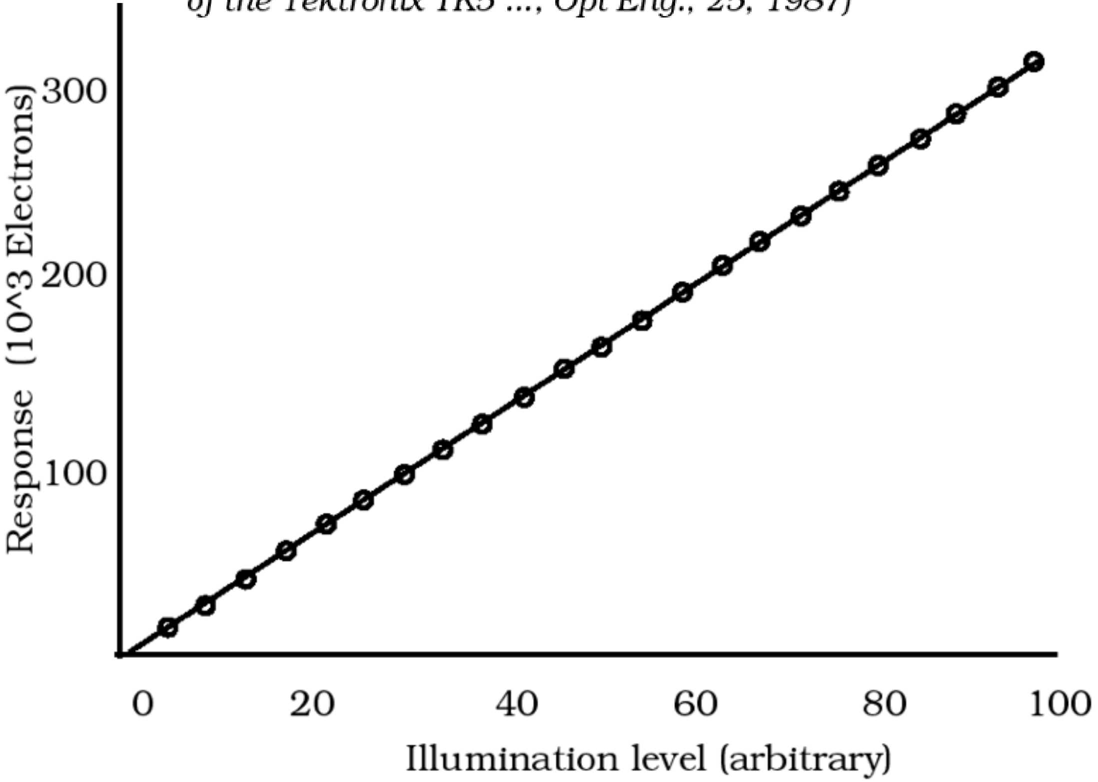
Photodiode response function

For silicon photodiodes, usually linear, but:

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

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

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

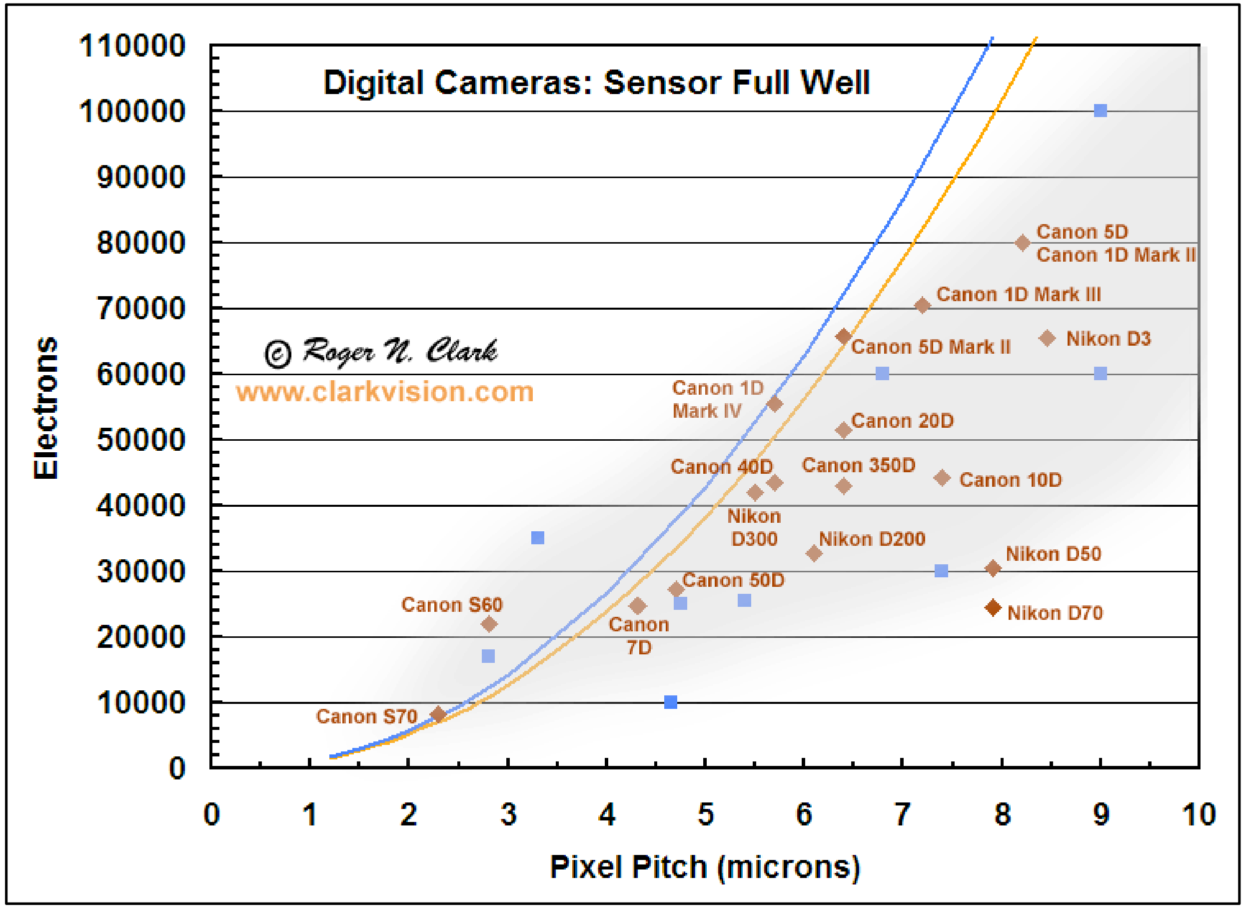


under-exposure (non-linearity due to sensor noise)

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

Photodiode full well capacity

How many electrons can photodiode store before saturation?

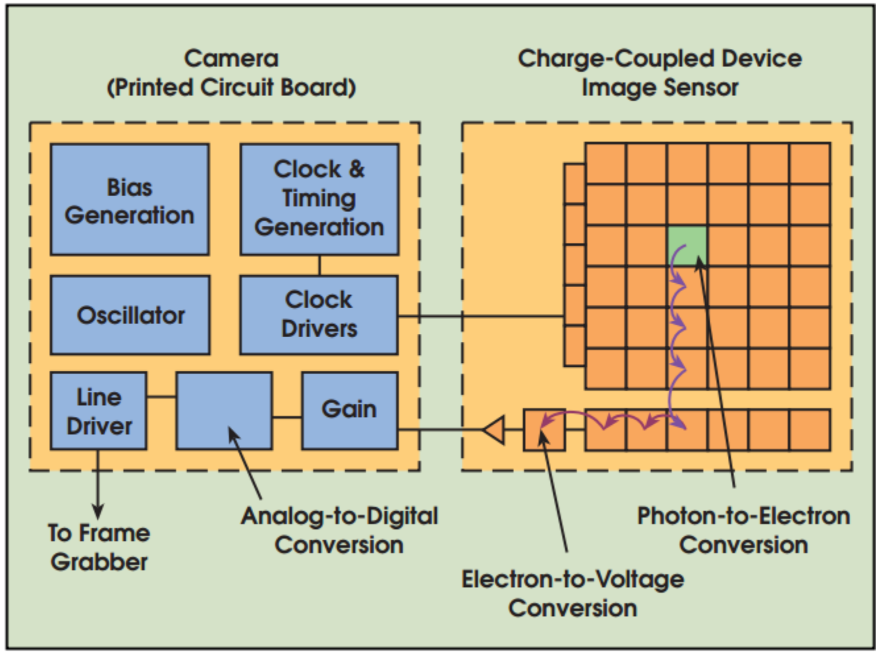


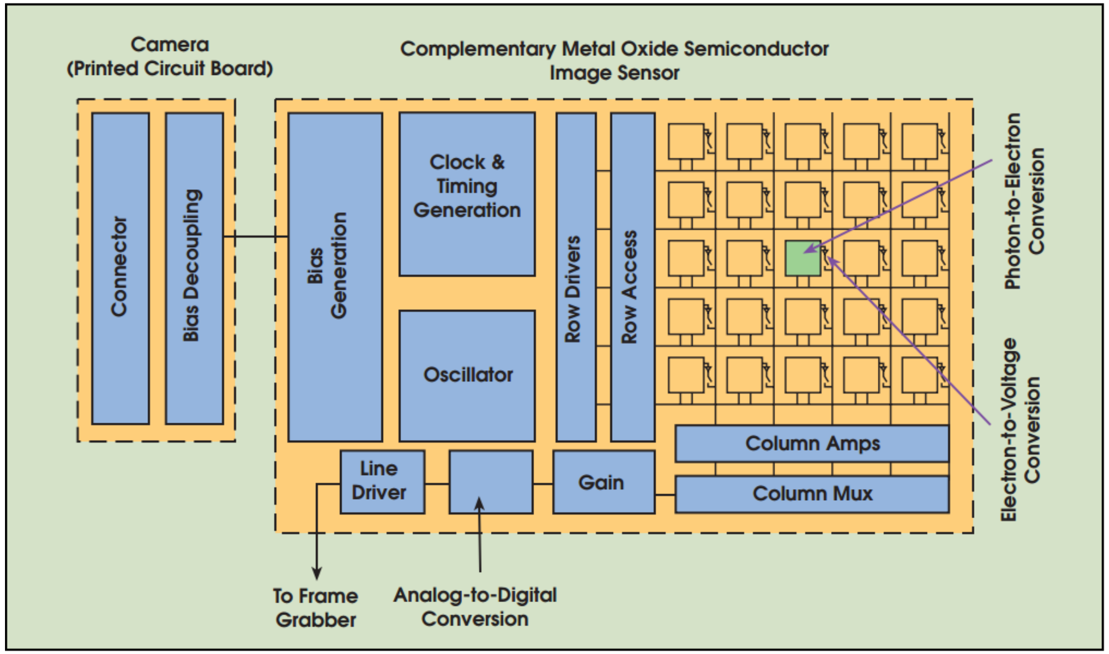
- Another important optical performance metric of imaging sensors.

Two main types of imaging sensors

Do you know them?

Two main types of imaging sensors



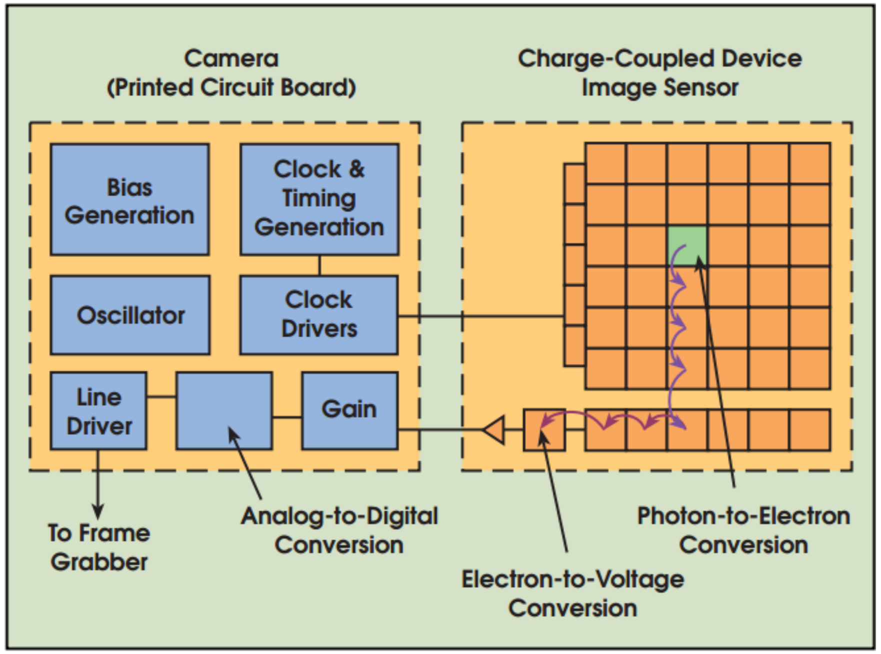


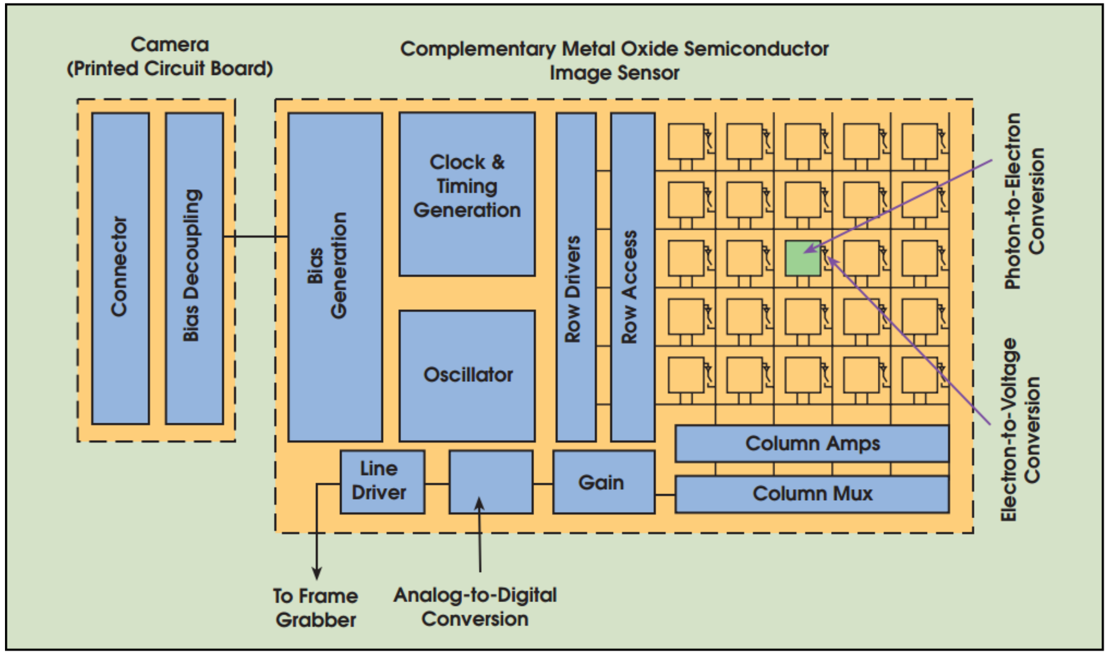
Charged Coupled Device (CCD):
converts electrons to voltage using
readout circuitry separate from pixel

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

Can you think of advantages and disadvantages of each type?

Two main types of imaging sensors





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

- ✓ higher sensitivity
- ✓ lower noise

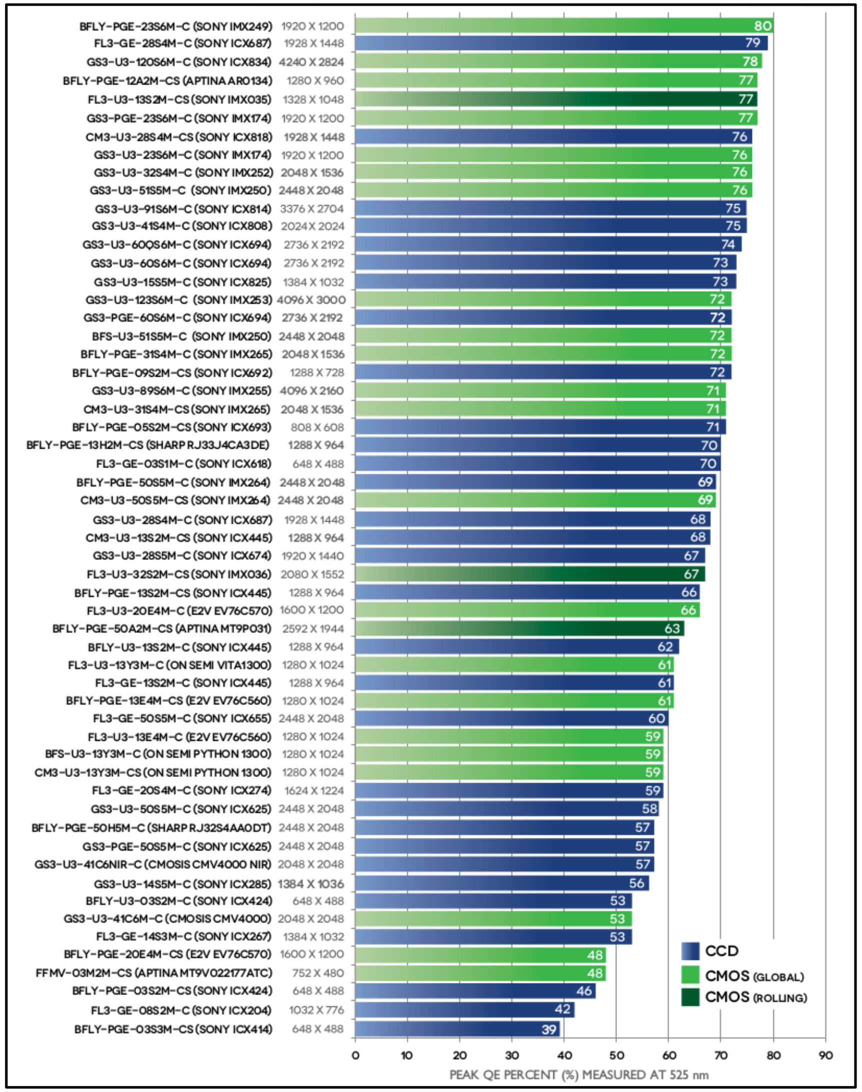
Complementary Metal Oxide Semiconductor (CMOS):
converts electrons to voltage using
per-pixel readout circuitry

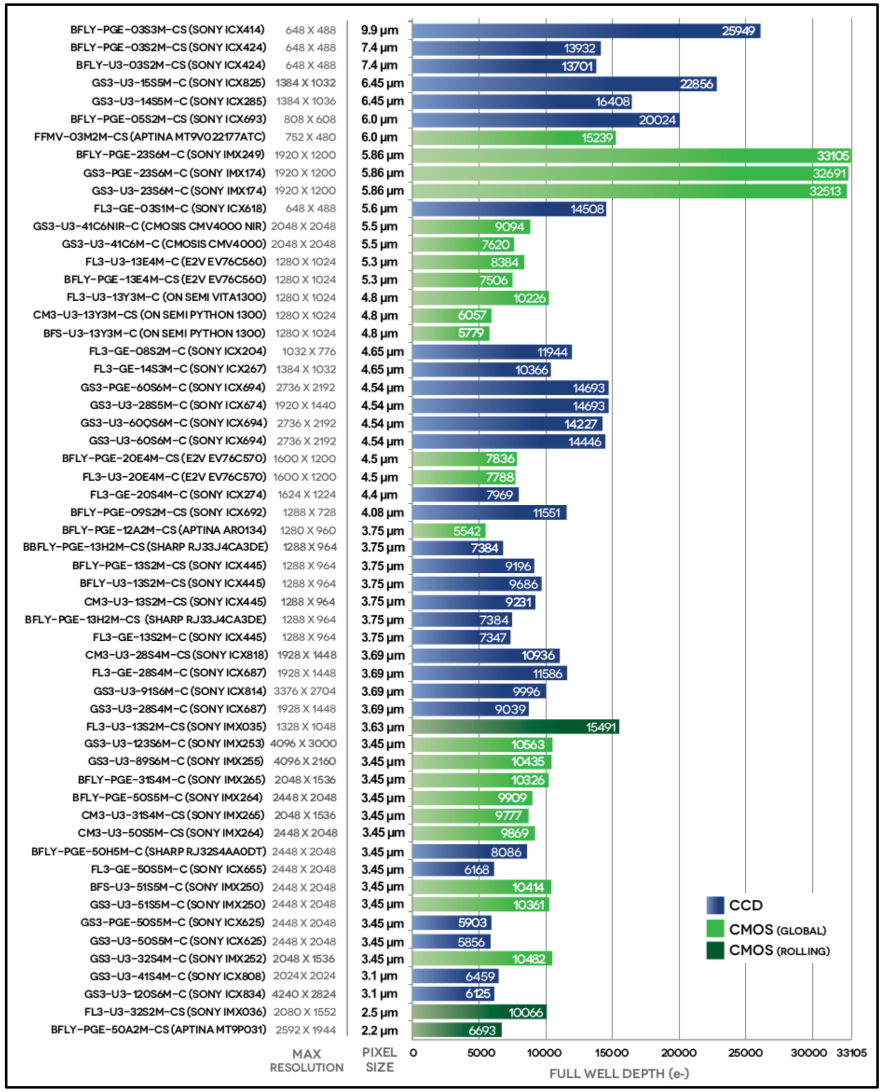
- ✓ faster read-out
- ✓ lower cost

CCD vs CMOS

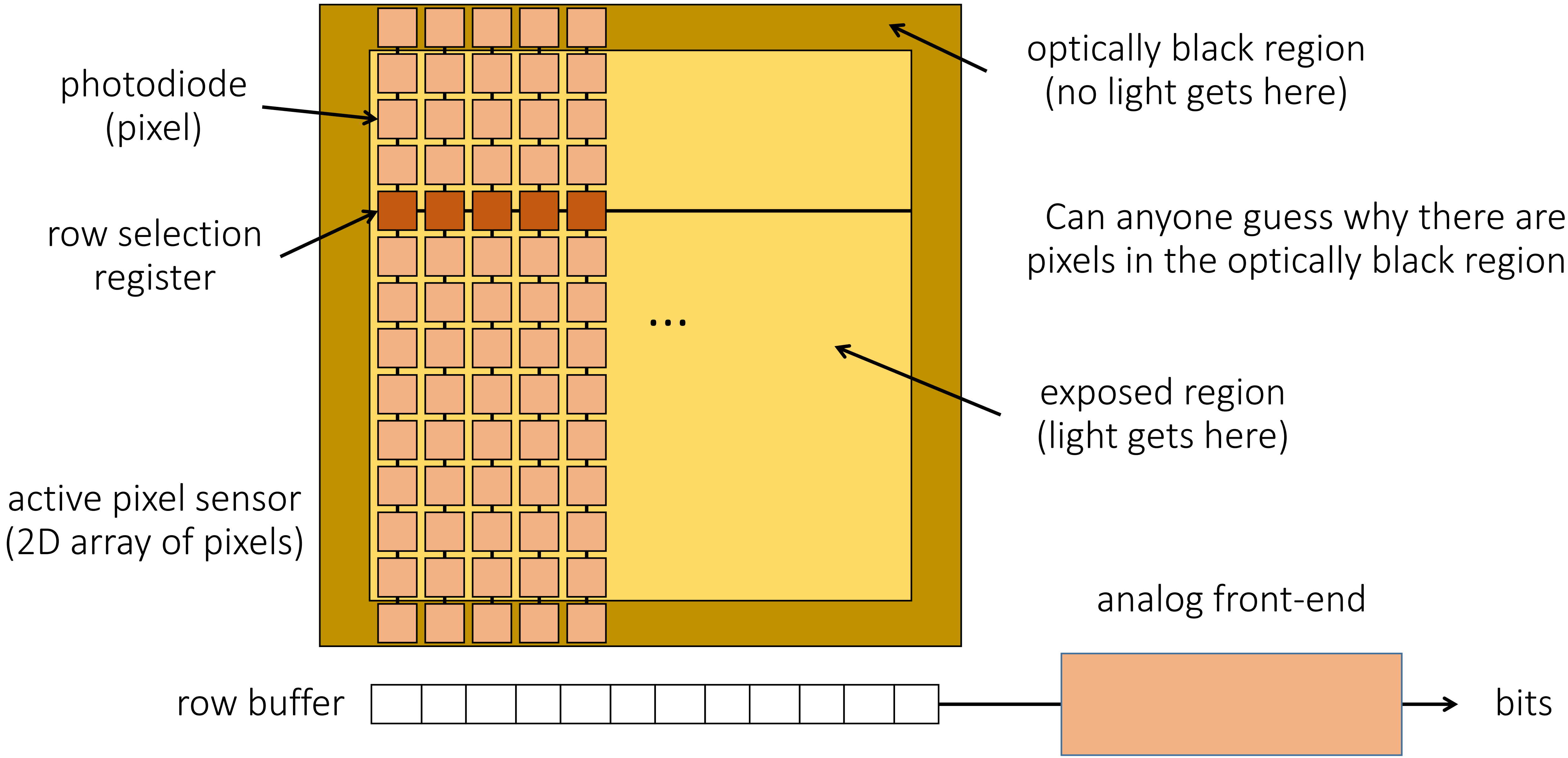
- Modern CMOS sensors have optical performance comparable to CCD sensors.
- Most modern commercial and industrial cameras use CMOS sensors.

Can you guess what the QE of the human eye is?



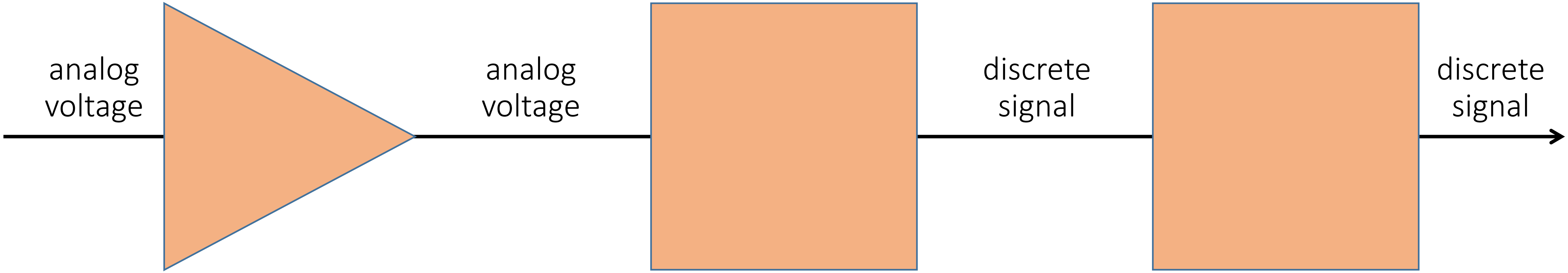


CMOS sensor (very) simplified layout



Can anyone guess why there are pixels in the optically black region?

Analog front-end



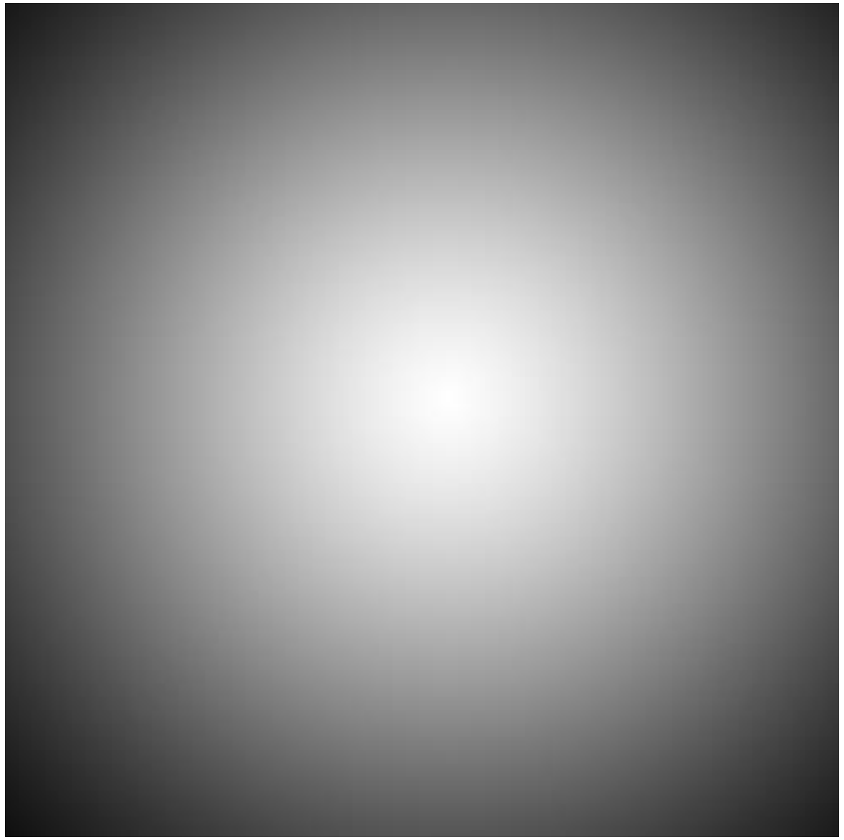
analog amplifier (gain):
- gets voltage in range needed by A/D converter.
- accommodates ISO settings.
- accounts for vignetting.

analog-to-digital converter (ADC):
- depending on sensor, output has 10-16 bits.
- most often (?) 12 bits.

look-up table (LUT):
- corrects non-linearities in sensor's response function (within proper exposure).
- corrects defective pixels.

Vignetting

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



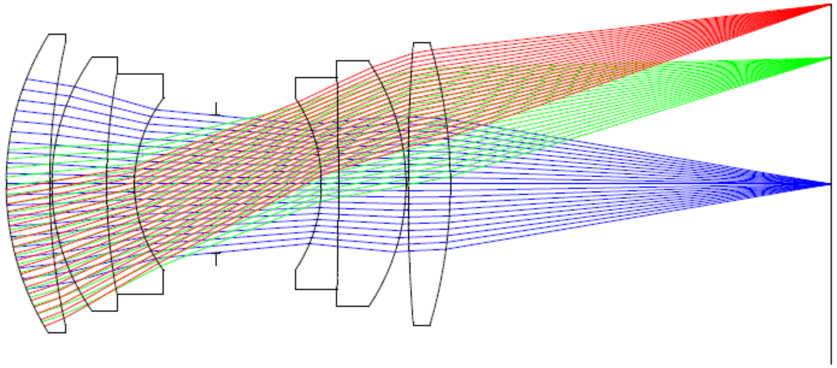
white wall under uniform light

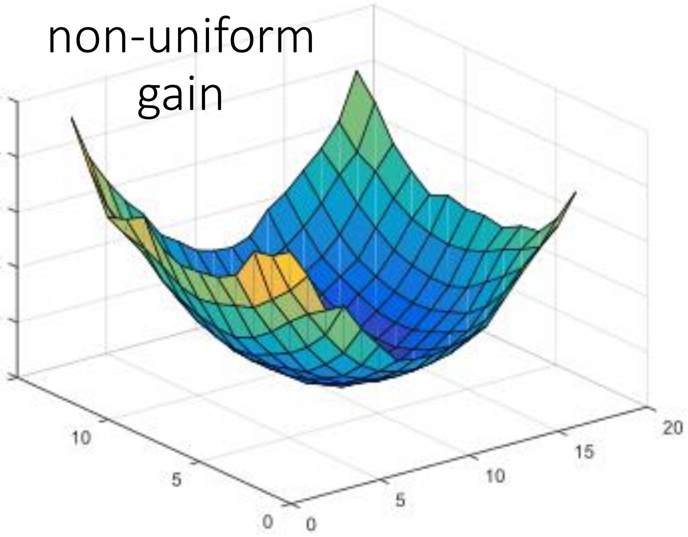
more interesting example of vignetting

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:

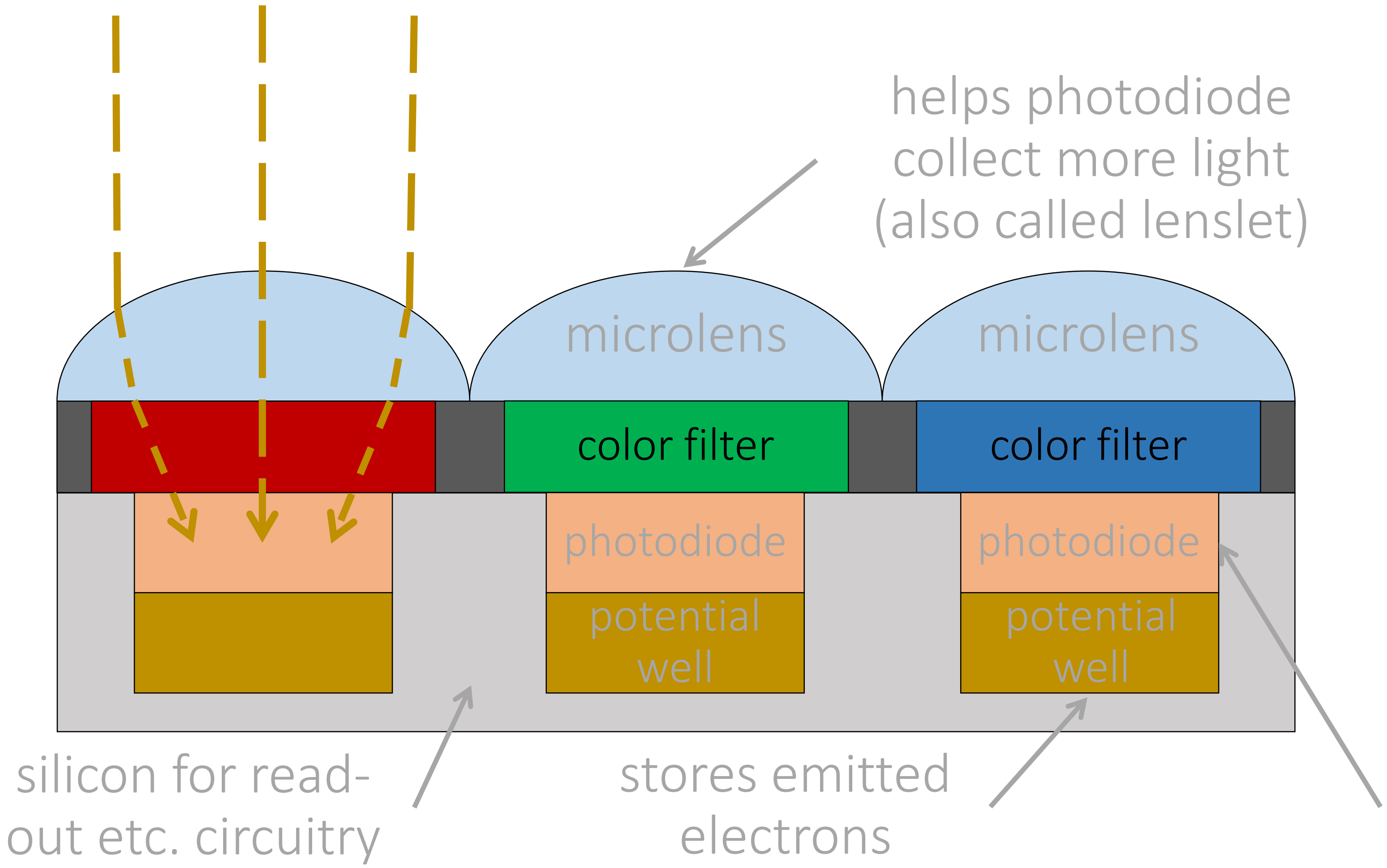
- at every photodiode, converts incident photons into electrons
- stores electrons into the photodiode’s potential well until it is full

… 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.
- Lenslets are problematic when working with coherent light.
- Many modern cameras do not have lenslet arrays.

We will discuss these issues in more detail at a later lecture.

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

Color sensing in cameras

Color

- Very high-level discussion of color as it relates to digital photography.
- We could spend an entire course covering color.
- See 15-463/663/862 for more on color.

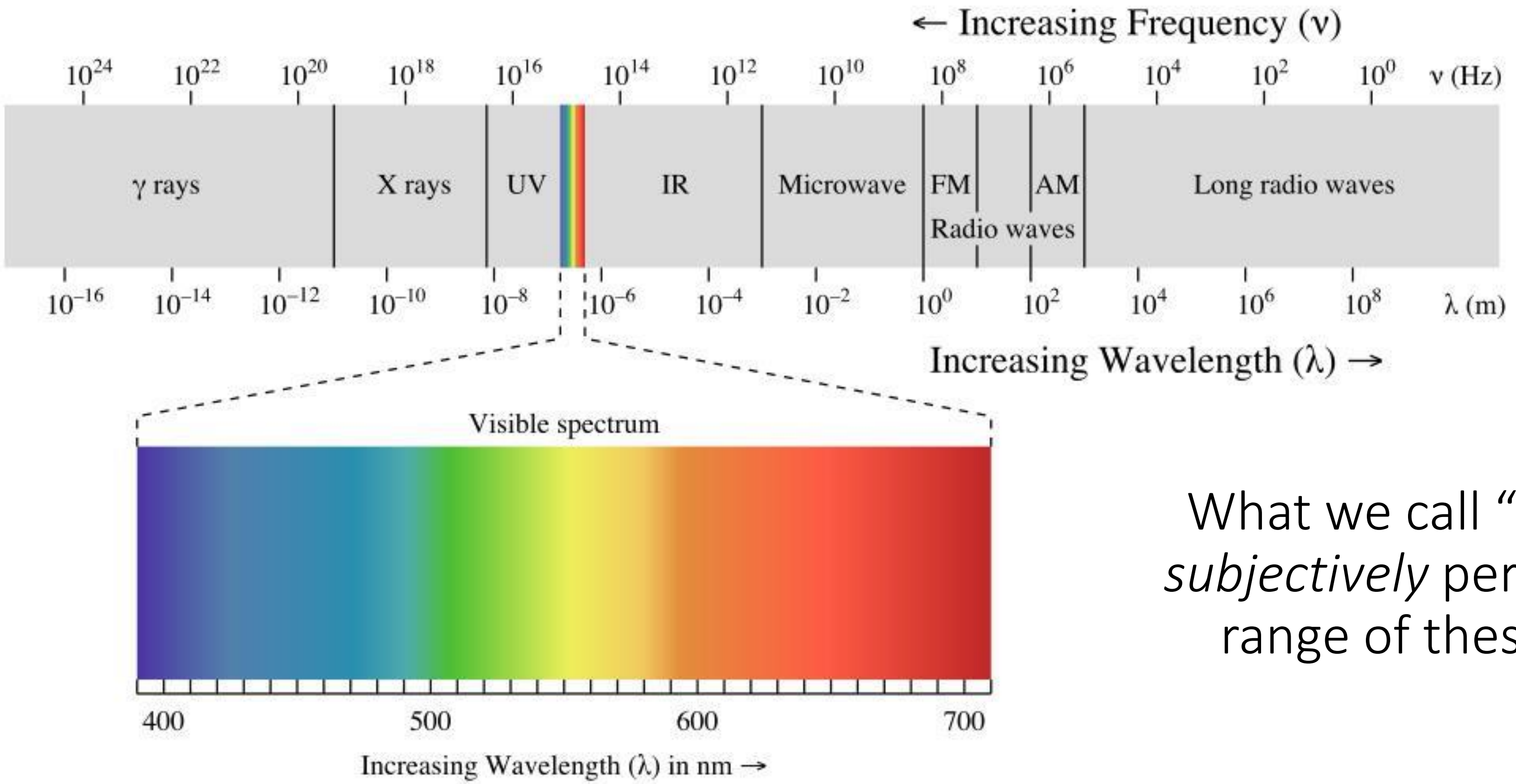
color is complicated

Retinal vs perceived color

Retinal vs perceived 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.



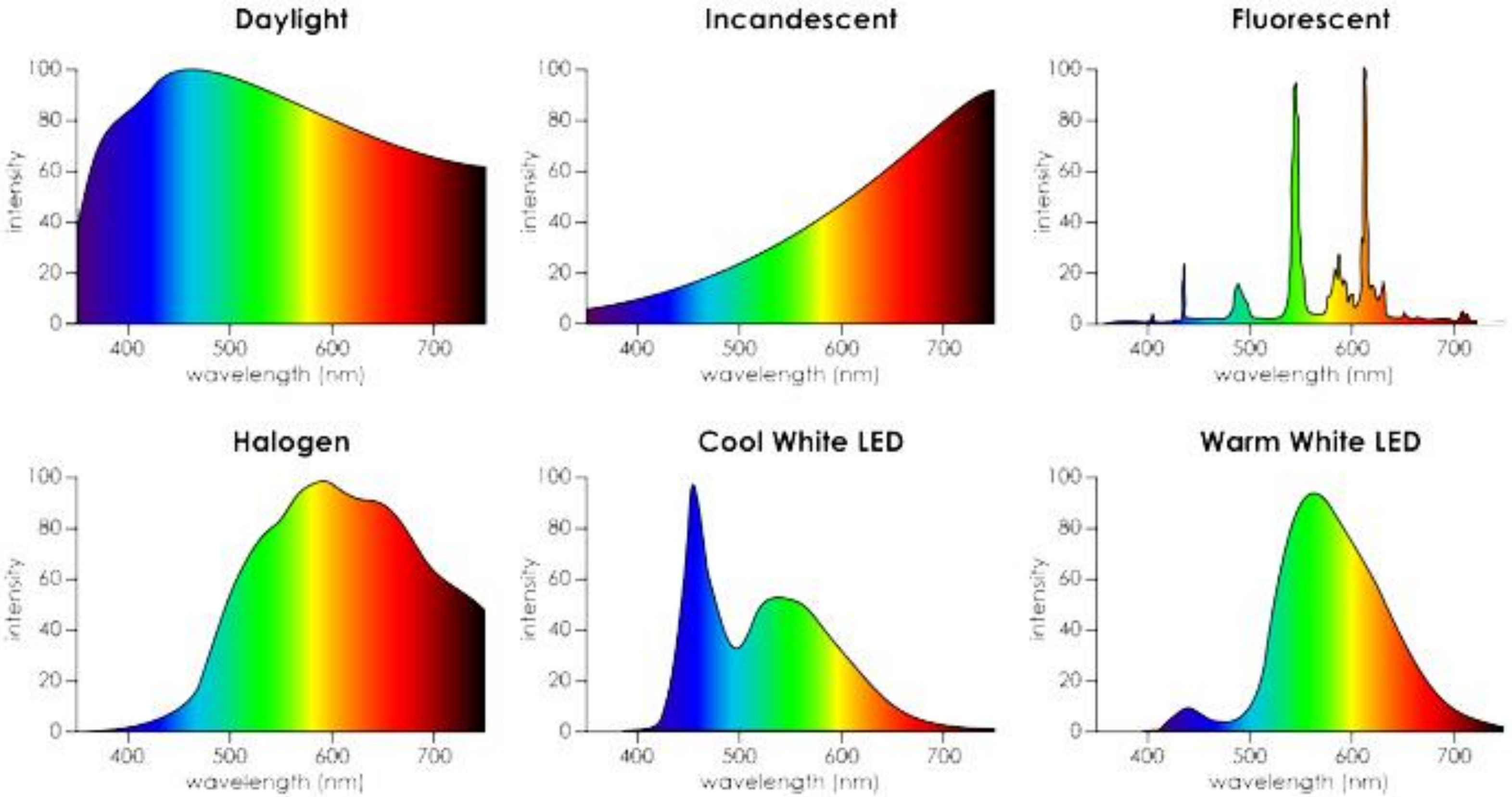
electromagnetic spectrum

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

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:

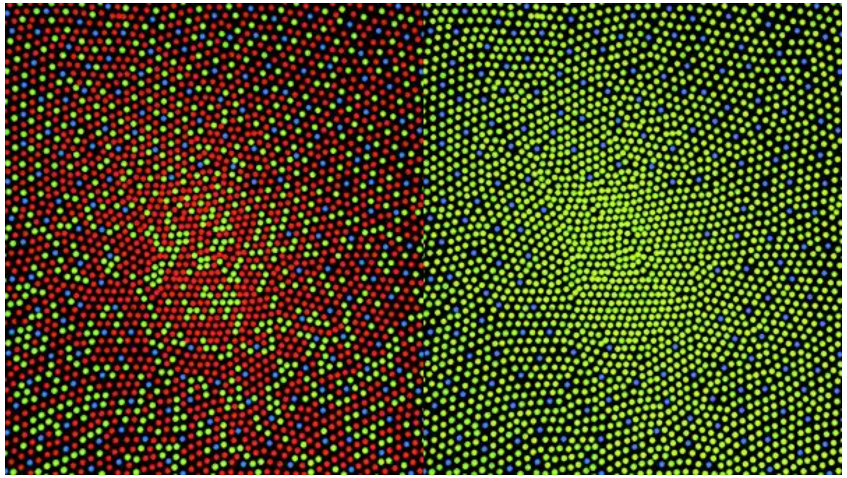
sensor response $\longrightarrow$

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

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

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*).

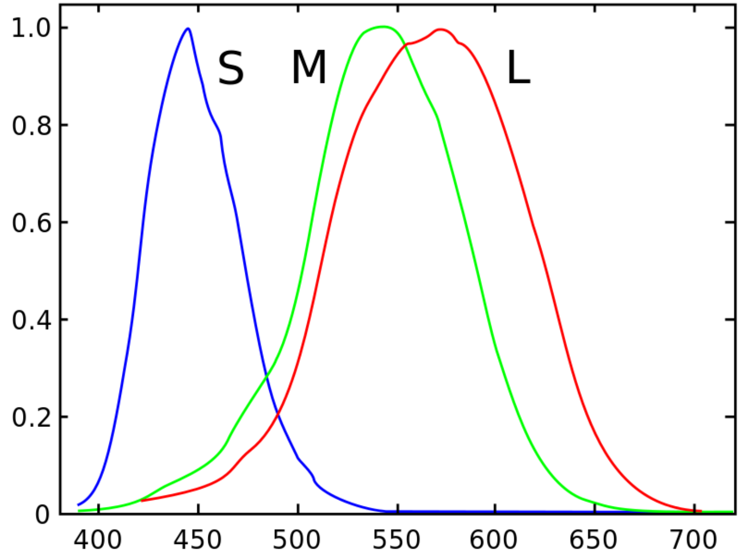


cone distribution
for normal vision
(64% L, 32% M)

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

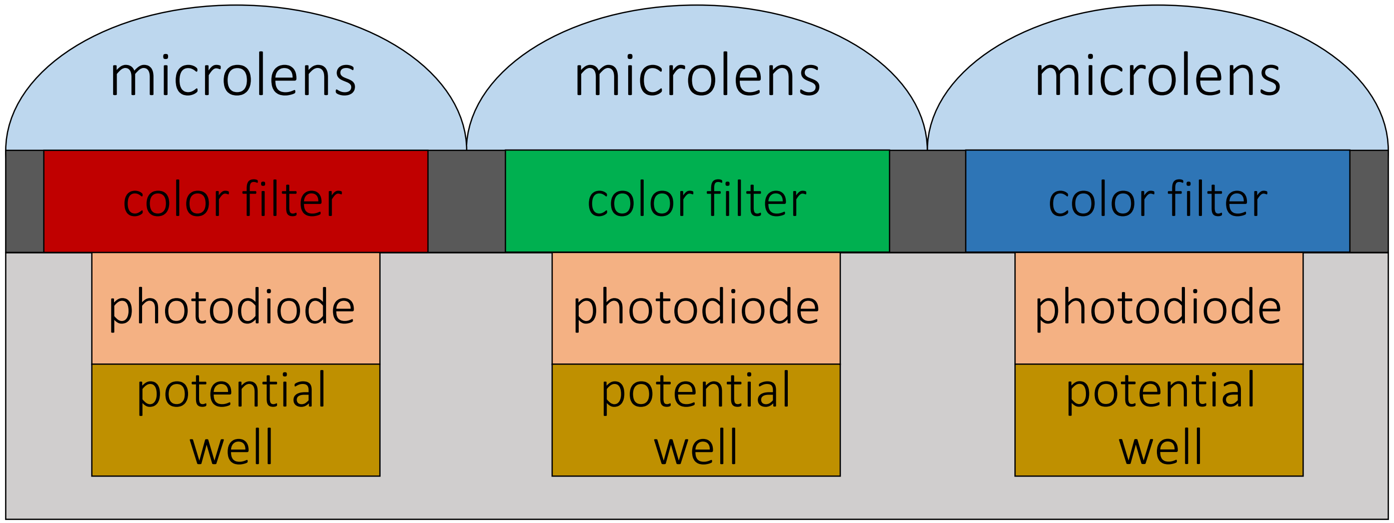
"medium" $$M = \int_\lambda \Phi(\lambda) M(\lambda) d\lambda$$

"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.

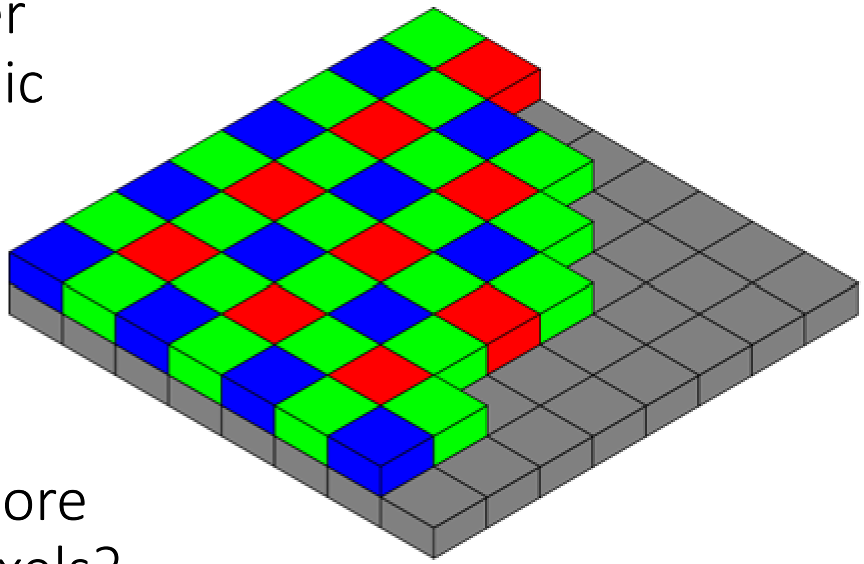


What color filters to use?

Two design choices:

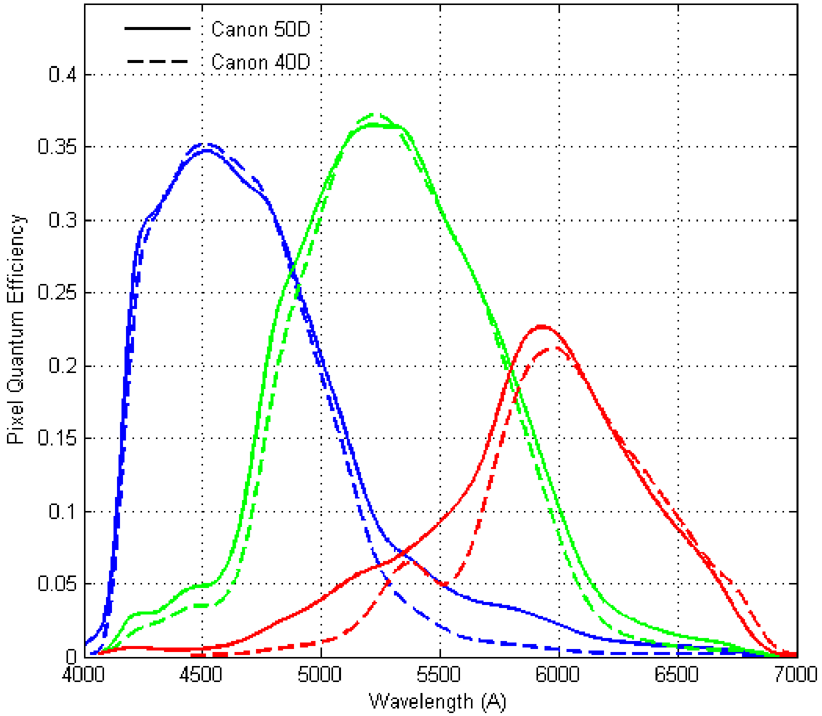
- What spectral sensitivity functions $f(\lambda)$ to use for each color filter?
- How to spatially arrange ("mosaic") different color filters?

Bayer mosaic



Why more green pixels?

SSF for Canon 50D

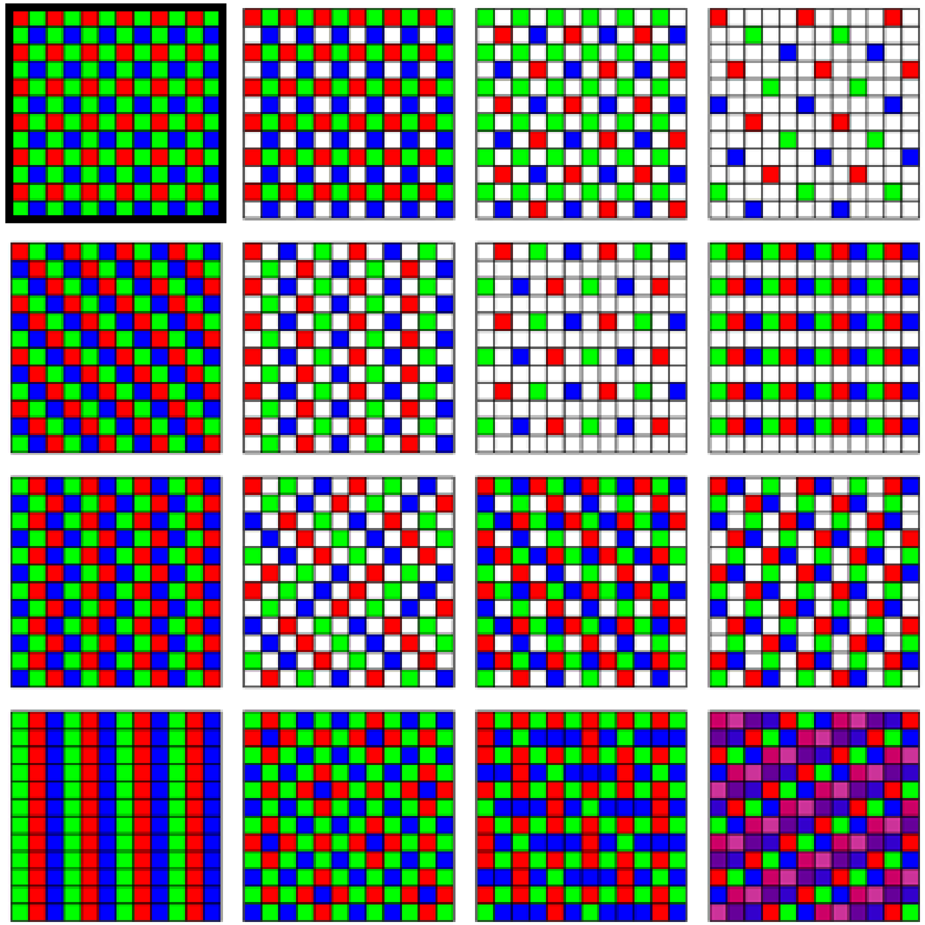


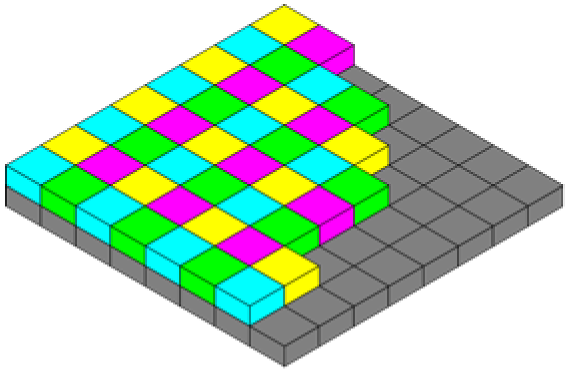
$f(\lambda)$

Generally do not match human LMS.

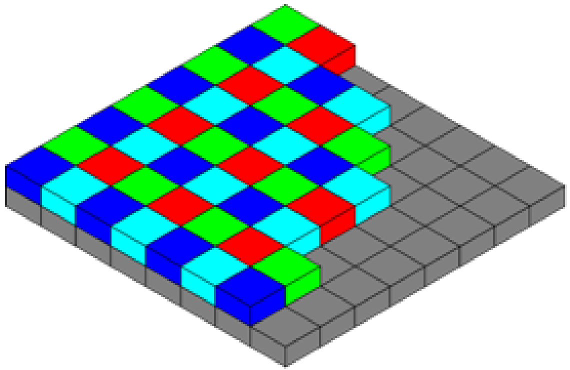
Many different CFAs

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





CYGM
Canon IXUS, Powershot



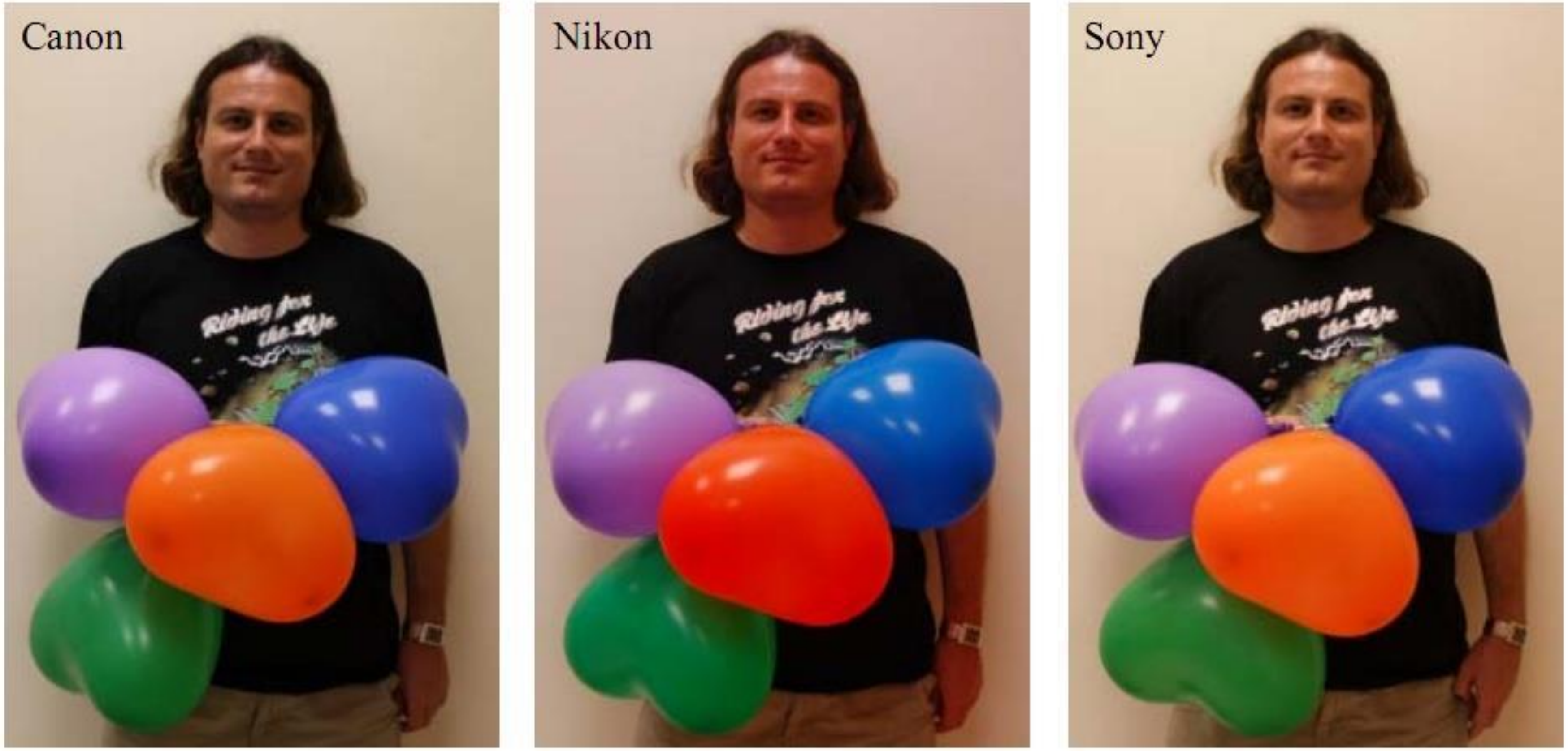
RGBE
Sony Cyber-shot

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.



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

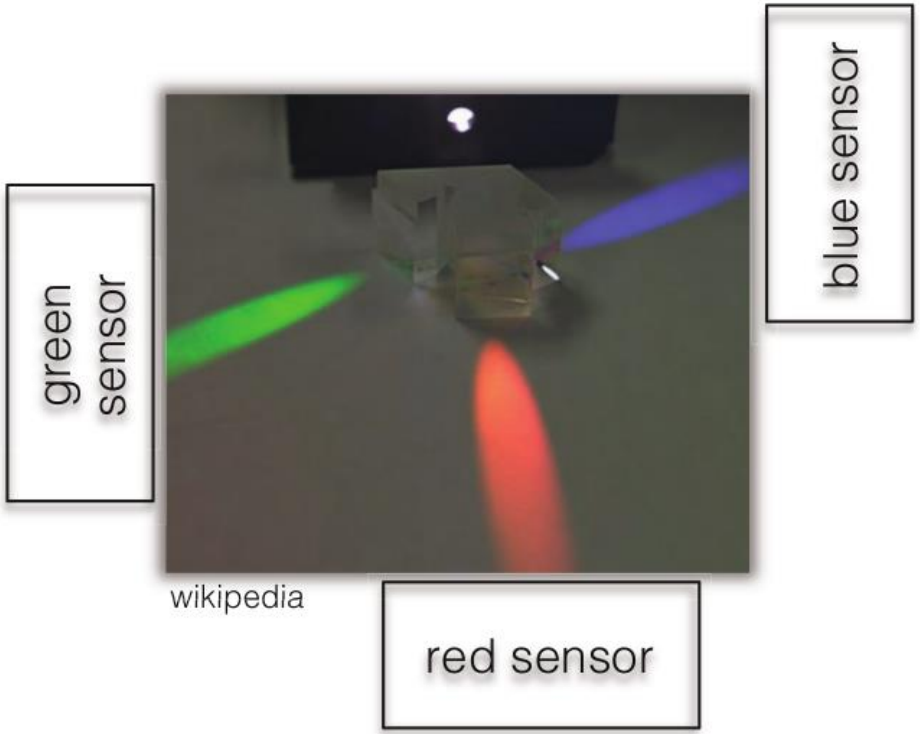
Aside: can you think of other ways to capture color?

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

field sequential

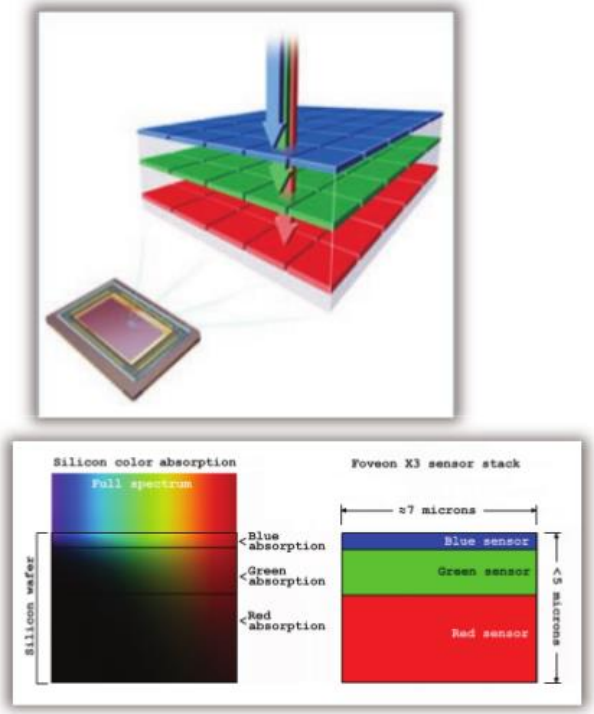
Prokudin-Gorsky

multiple sensors



wikipedia

vertically stacked



Foveon X3

[Slide credit: Gordon Wetzstein]

What does an imaging sensor do?

When the camera shutter opens, the sensor:

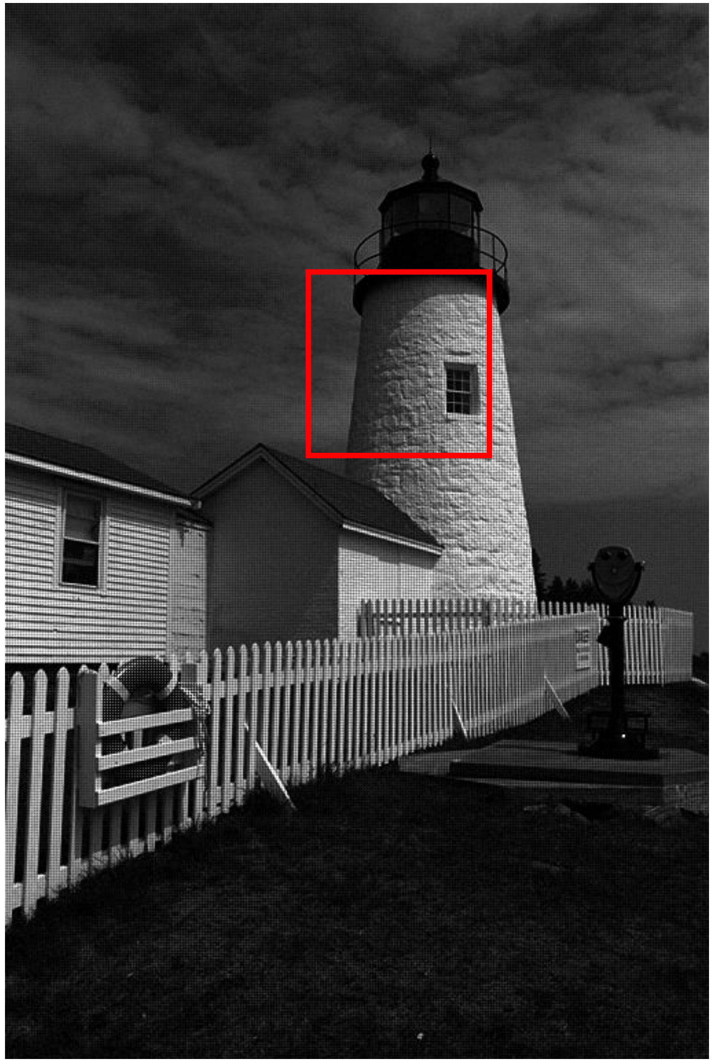
- at every photodiode, converts incident photons into electrons using mosaic's SSF
- stores electrons into the photodiode's potential well until it is full

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

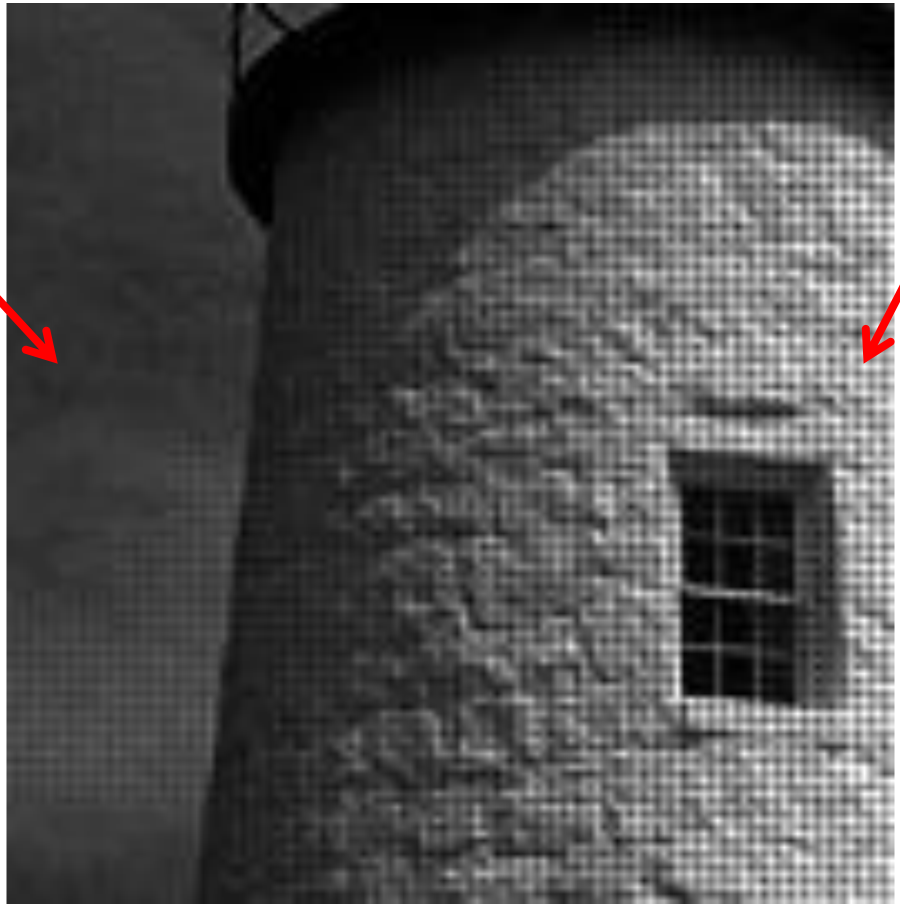
- reads out photodiodes' wells, row-by-row, and converts them to analog signals
- applies a (possibly non-uniform) gain to these analog signals
- converts them to digital signals
- corrects non-linearities

… and finally returns an image.

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



lots of noise

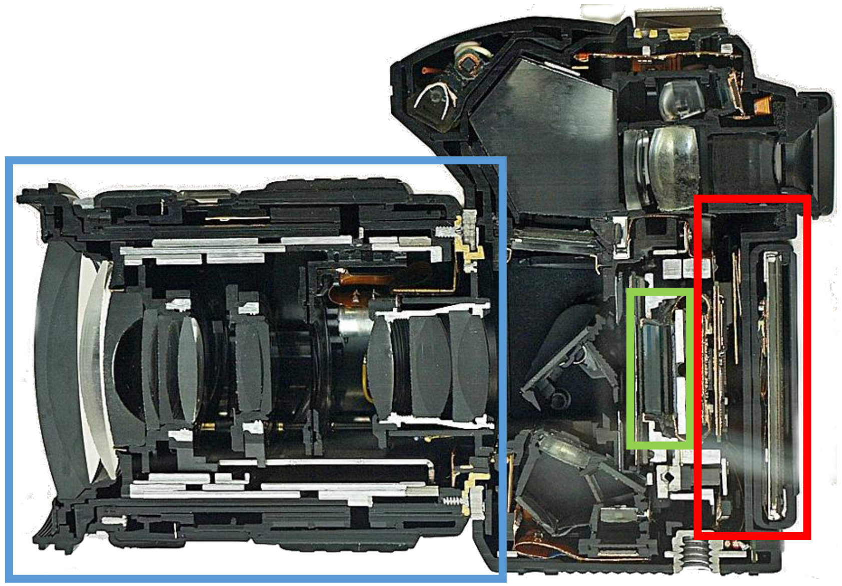


mosaicking artifacts

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

The modern photography pipeline

post-capture processing
(lectures 3-12)

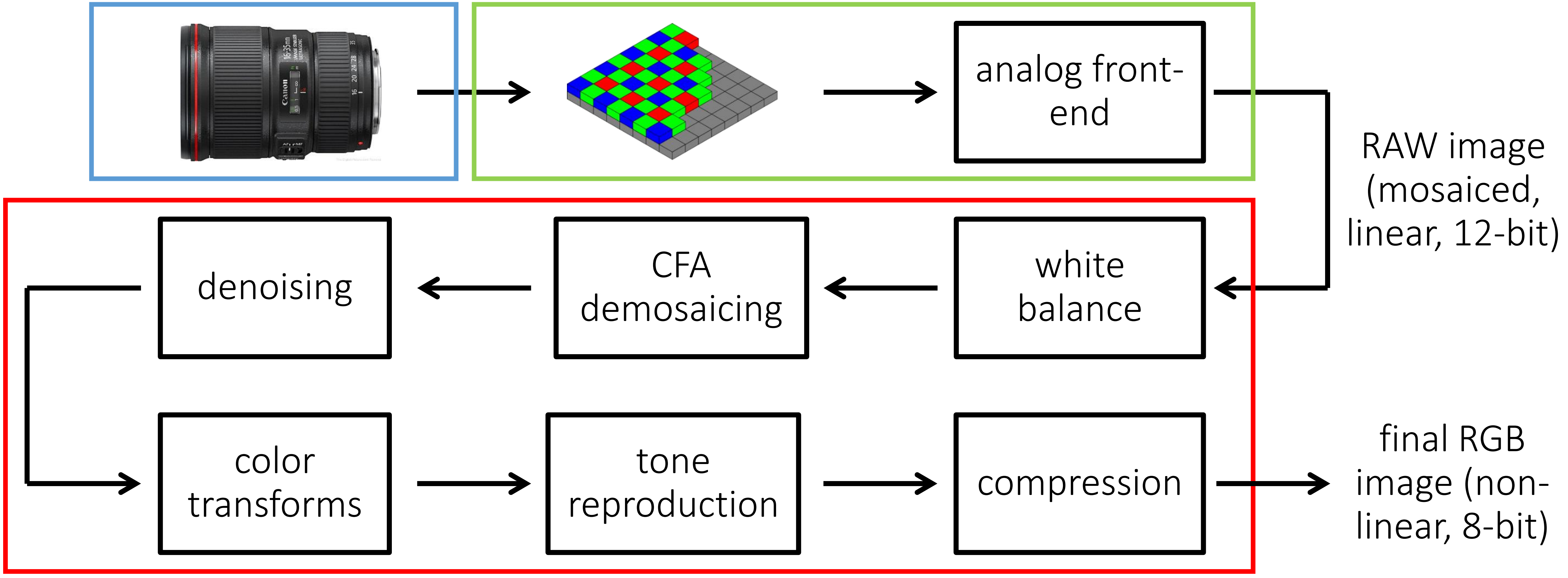


optics and optical controls (lectures 13-16) → sensor, analog front-end, and color filter array (this lecture) → in-camera image processing pipeline (this lecture)

The in-camera image processing pipeline

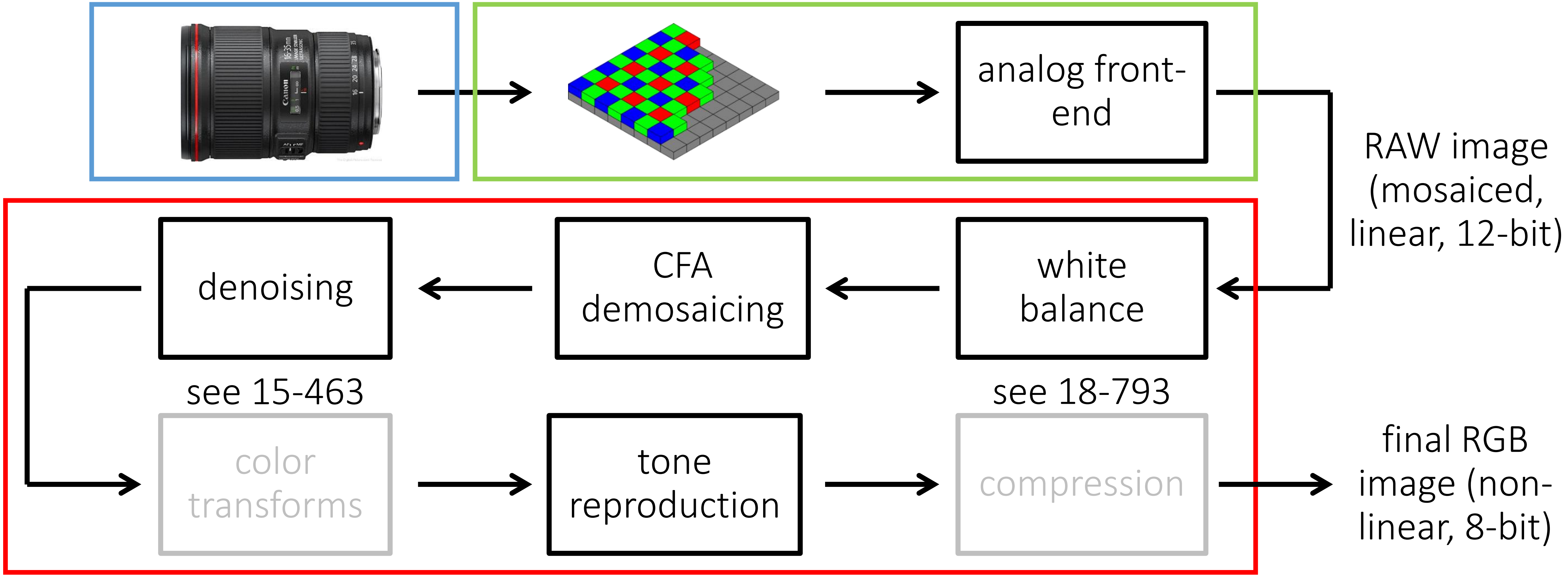
The (in-camera) image processing pipeline

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



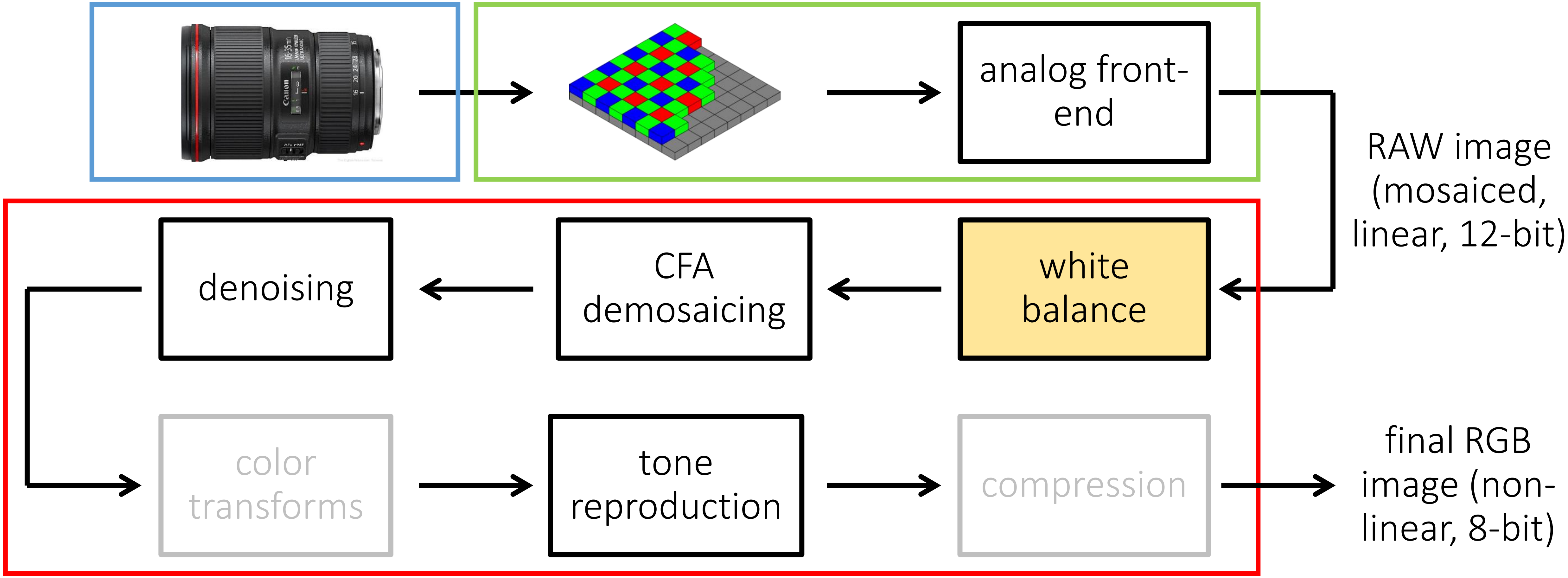
The (in-camera) image processing pipeline

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.

Retinal vs perceived color.

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:

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

How can we do automatic white balancing?

Manual vs automatic white balancing

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

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

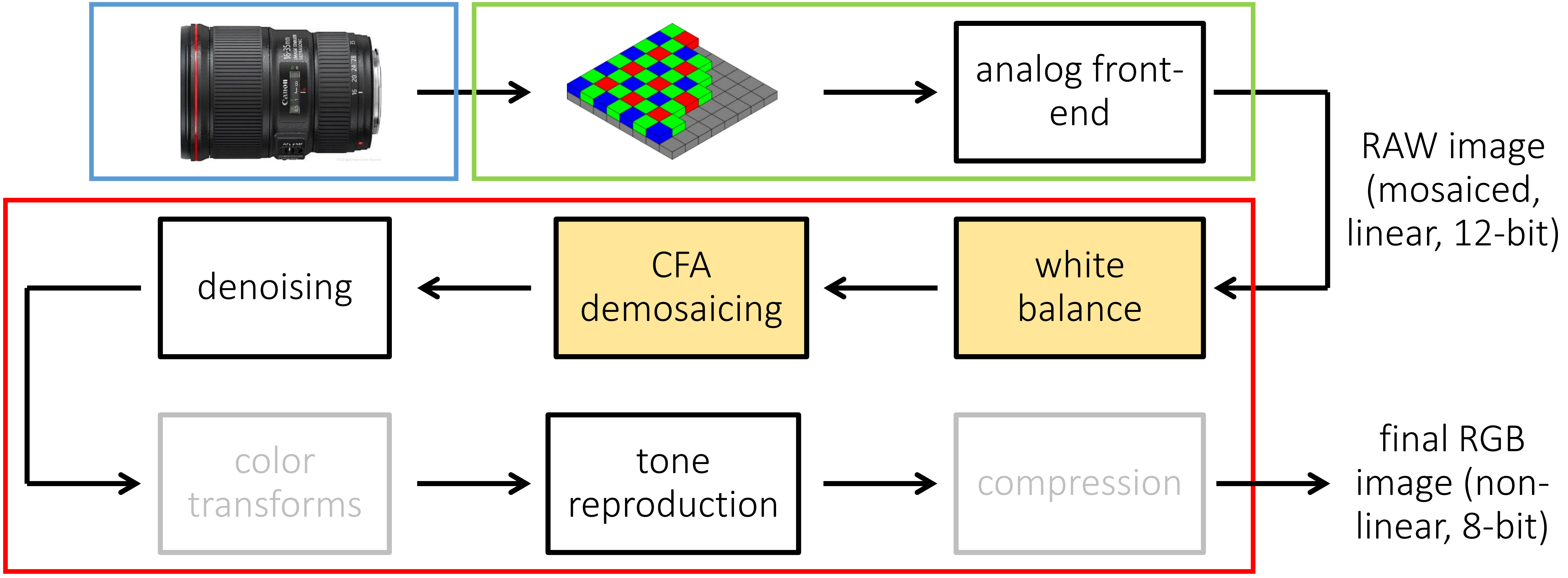
input image

grey world

white world

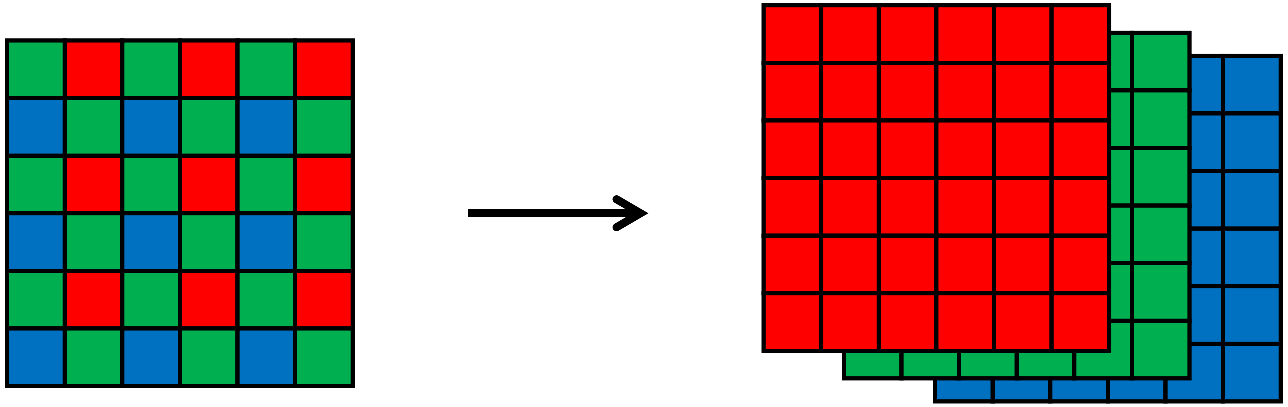
The (in-camera) image processing pipeline

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

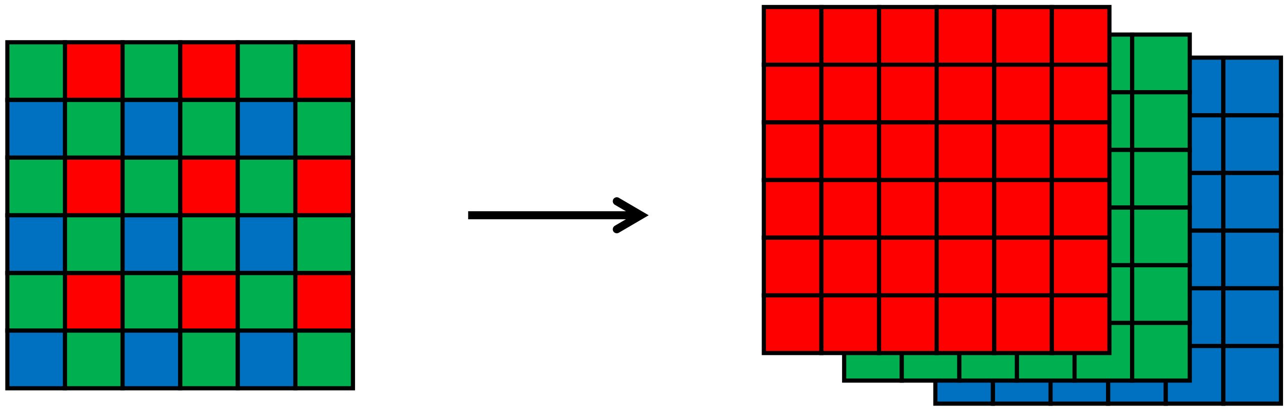
Produce full RGB image from mosaiced sensor output.



Any ideas on how to do this?

CFA demosaicing

Produce full RGB image from mosaiced sensor output.



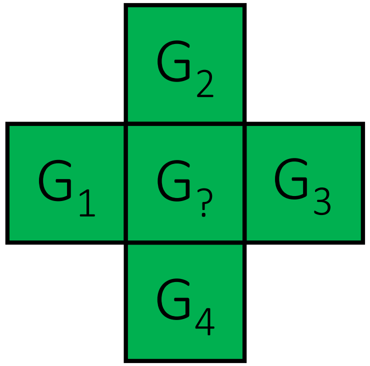
Interpolate from neighbors:

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

Large area of research.

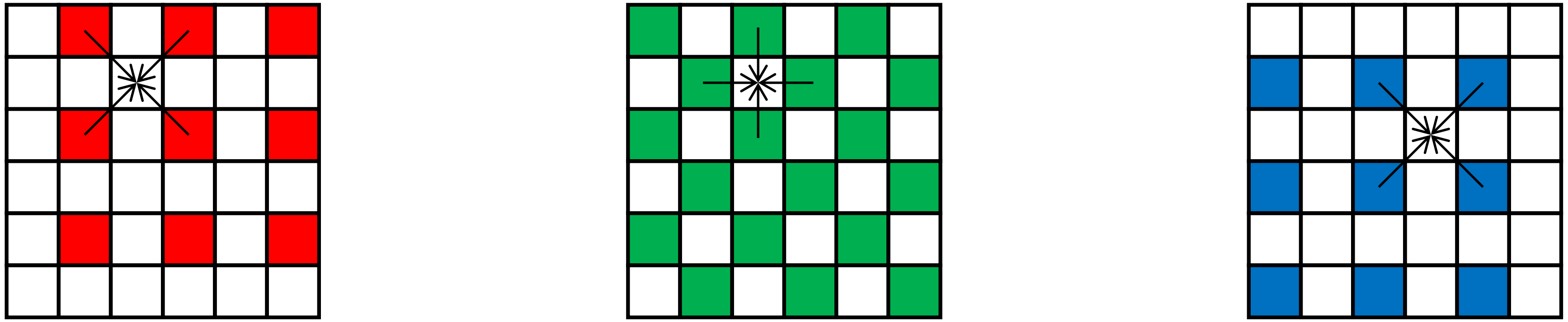
Demosaicing by bilinear interpolation

Bilinear interpolation: Simply average your 4 neighbors.



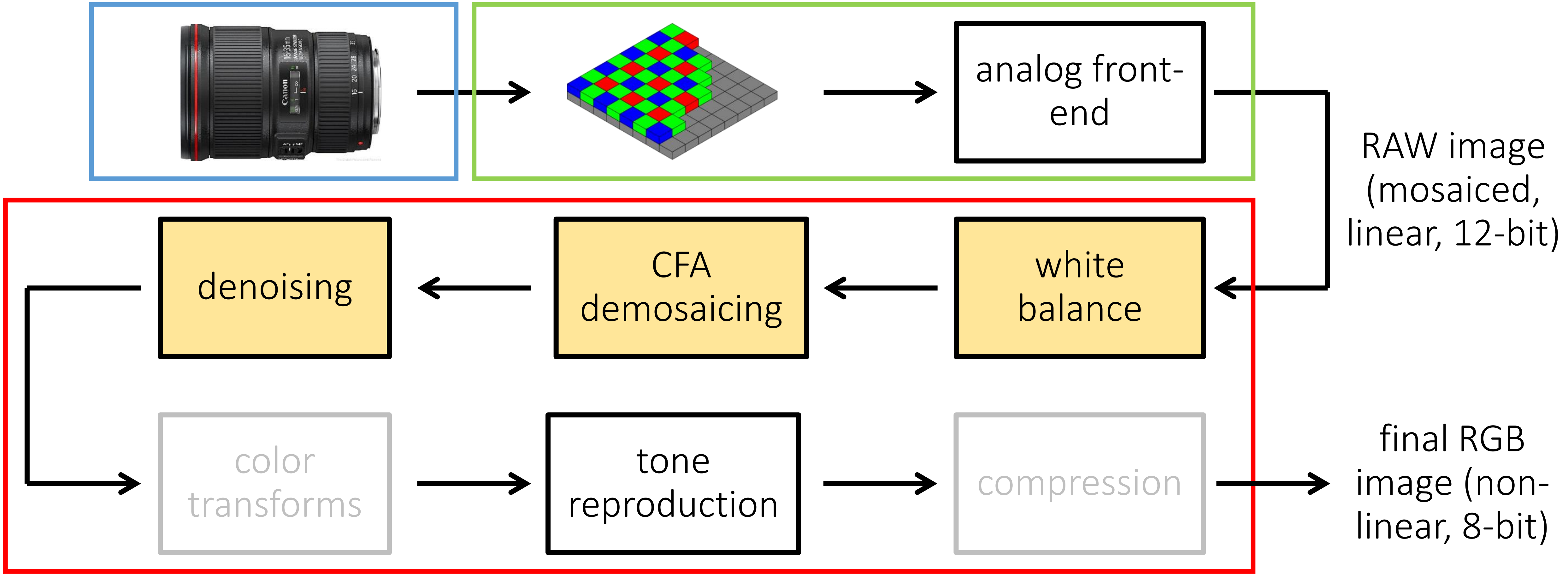
$$G_? = \frac{G_1 + G_2 + G_3 + G_4}{4}$$

Neighborhood changes for different channels:



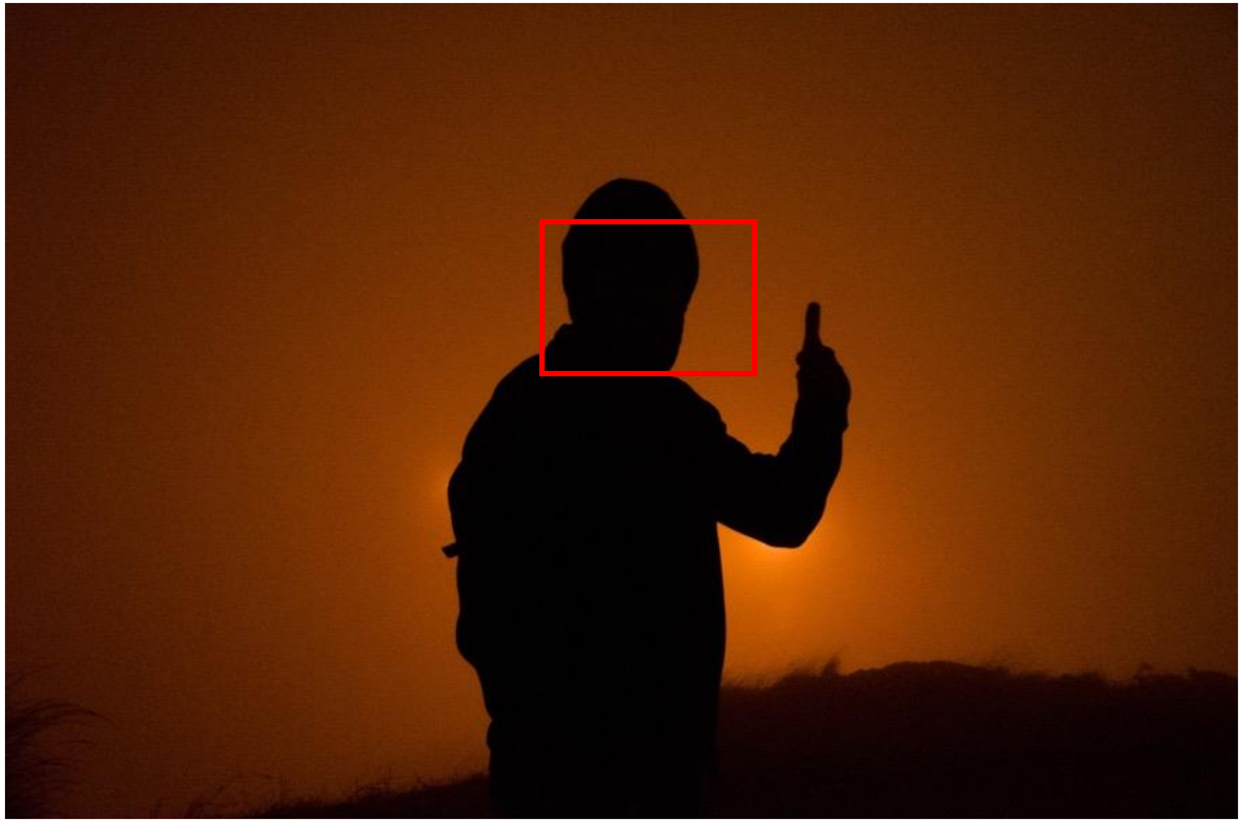
The (in-camera) image processing pipeline

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 larger the variance of the distribution.

2) Dark-shot noise:

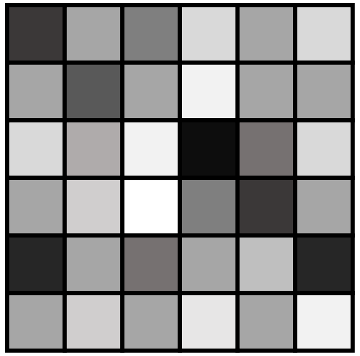
- Emitted electrons due to thermal activity (becomes worse as sensor gets hotter.)

3) Read noise:

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

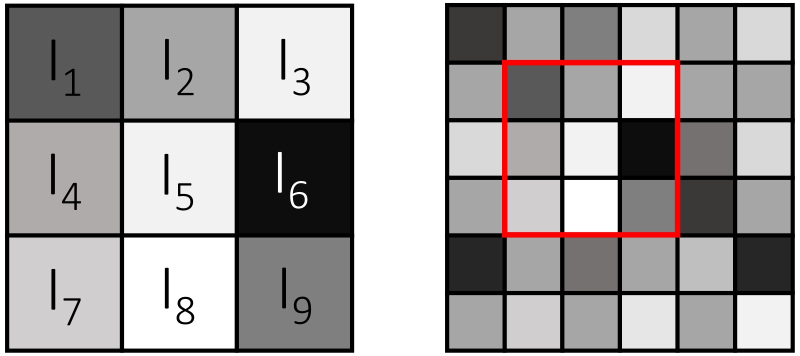
Bright scene and large pixels: photon shot noise is the main noise source.

How to denoise?



How to denoise?

Simple denoising: look at the neighborhood around you.



- Mean filtering (take average):

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

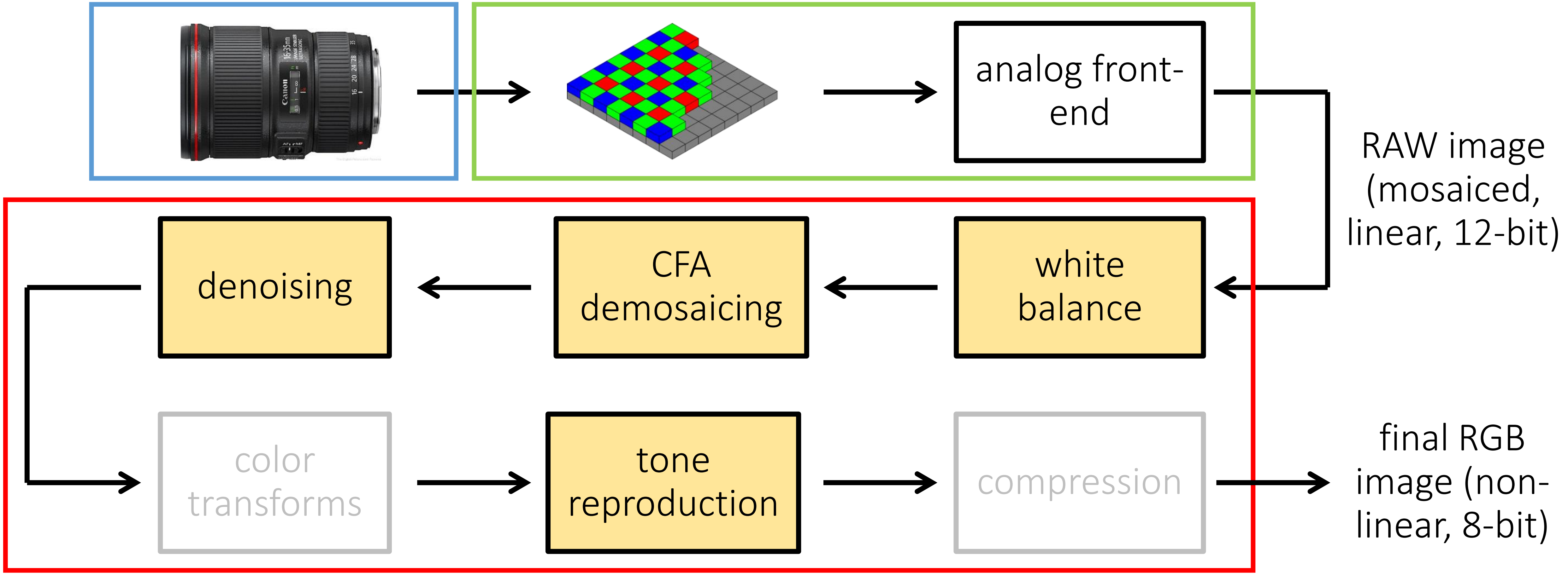
- Median filtering (take median):

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

Large area of research.

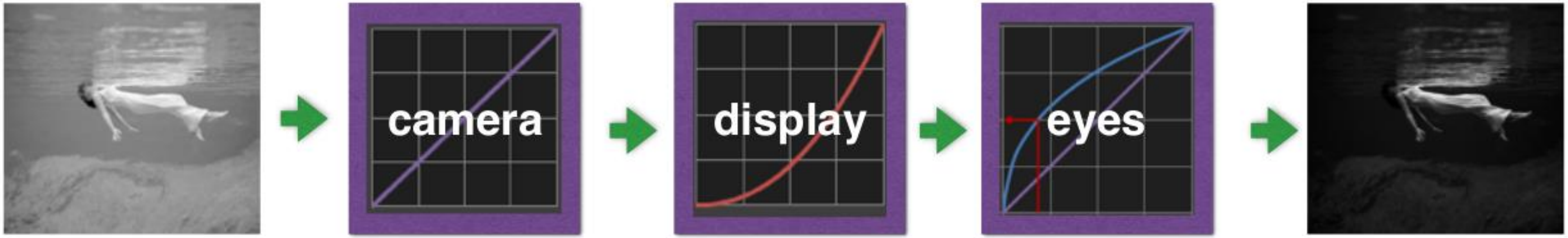
The (in-camera) image processing pipeline

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



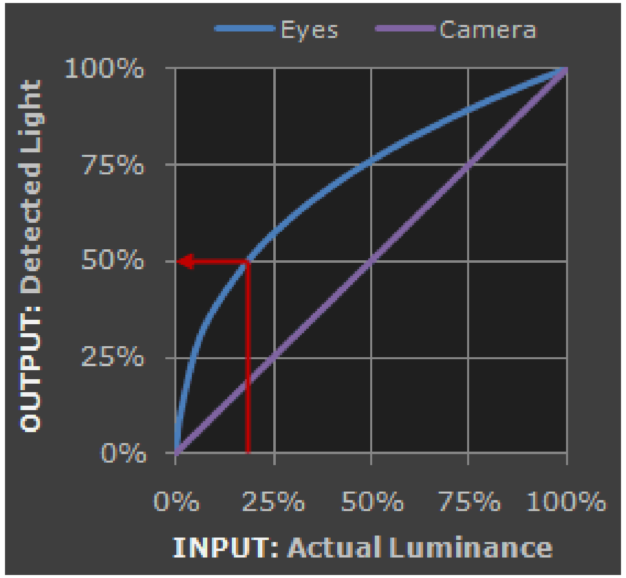
Tone reproduction

- Also known as gamma correction.
- Without tone reproduction, images look very dark.



Why does this happen?

Perceived vs measured brightness by human eye



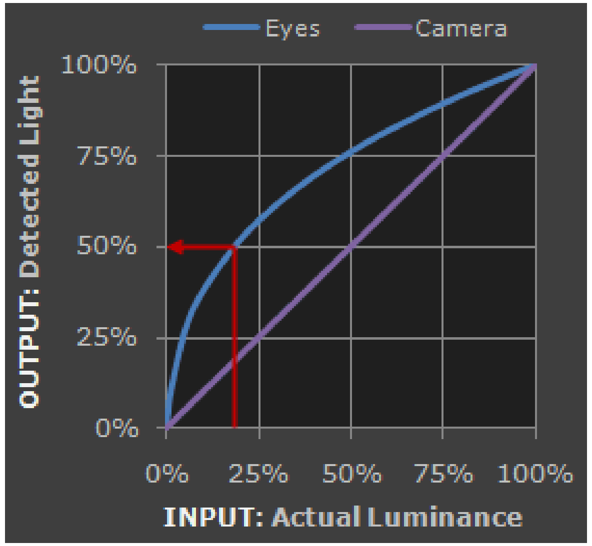
We have already seen that sensor response is linear.

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?

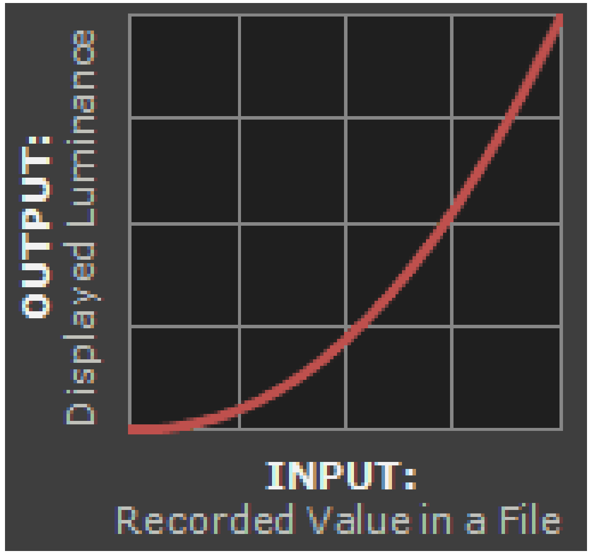


We have already seen that sensor response is linear.

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

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

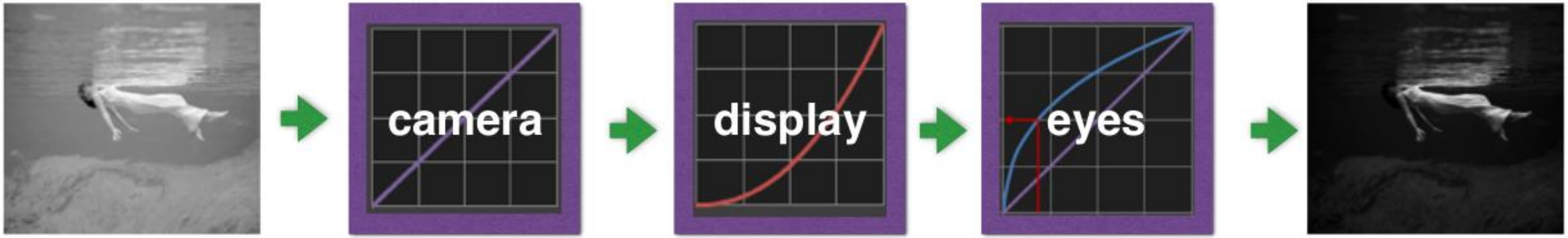
- More sensitive to dark tones.
- Approximately a Gamma function.



Displays have a response opposite to that of human perception.

Tone reproduction

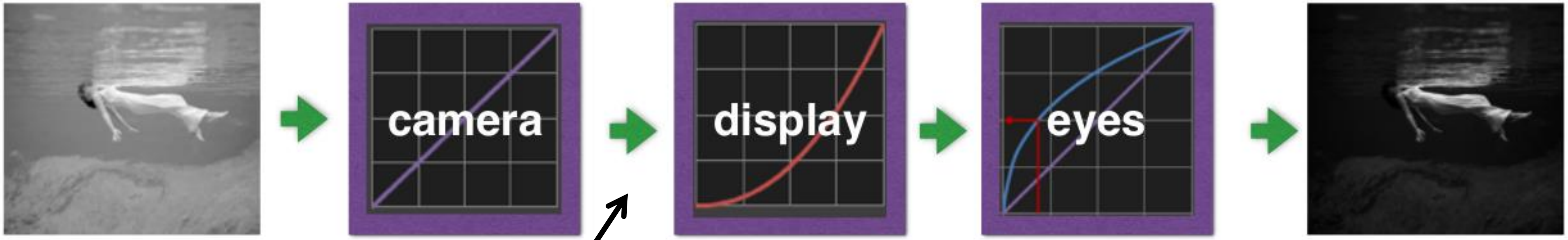
- Because of mismatch in displays and human eye perception, images look very dark.

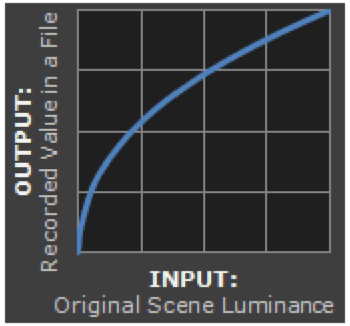


How do we fix this?

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.

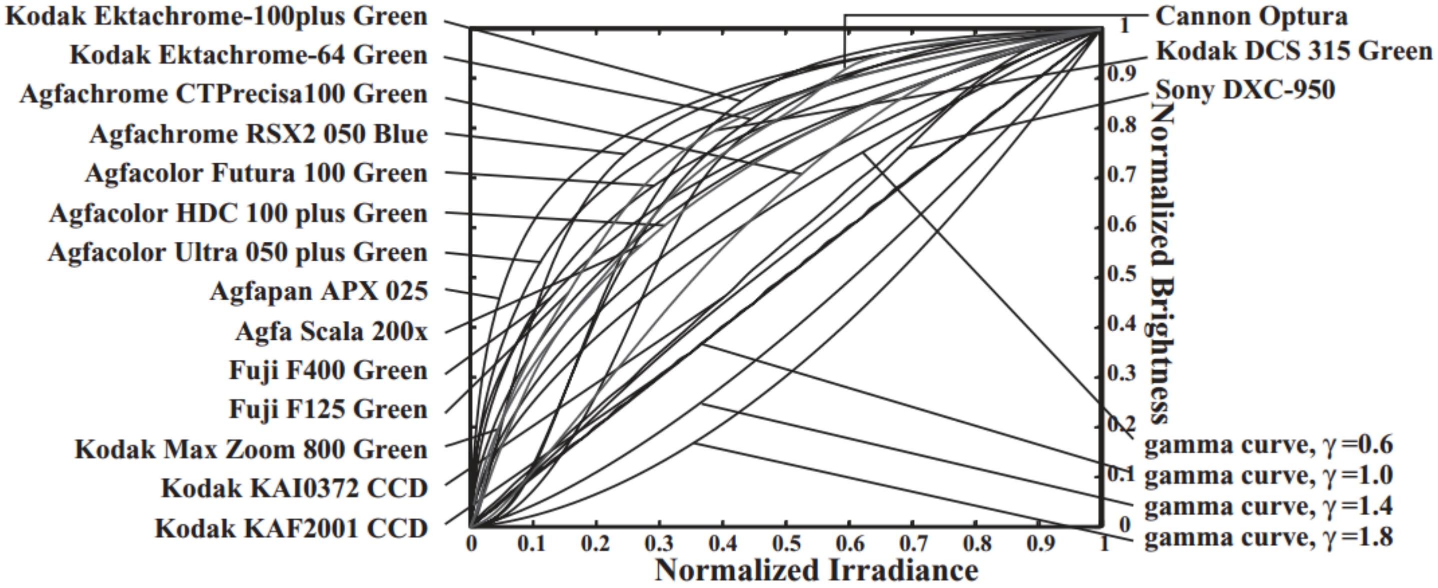
Tone reproduction curves

The exact tone reproduction curve depends on the camera.

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

before gamma

after gamma



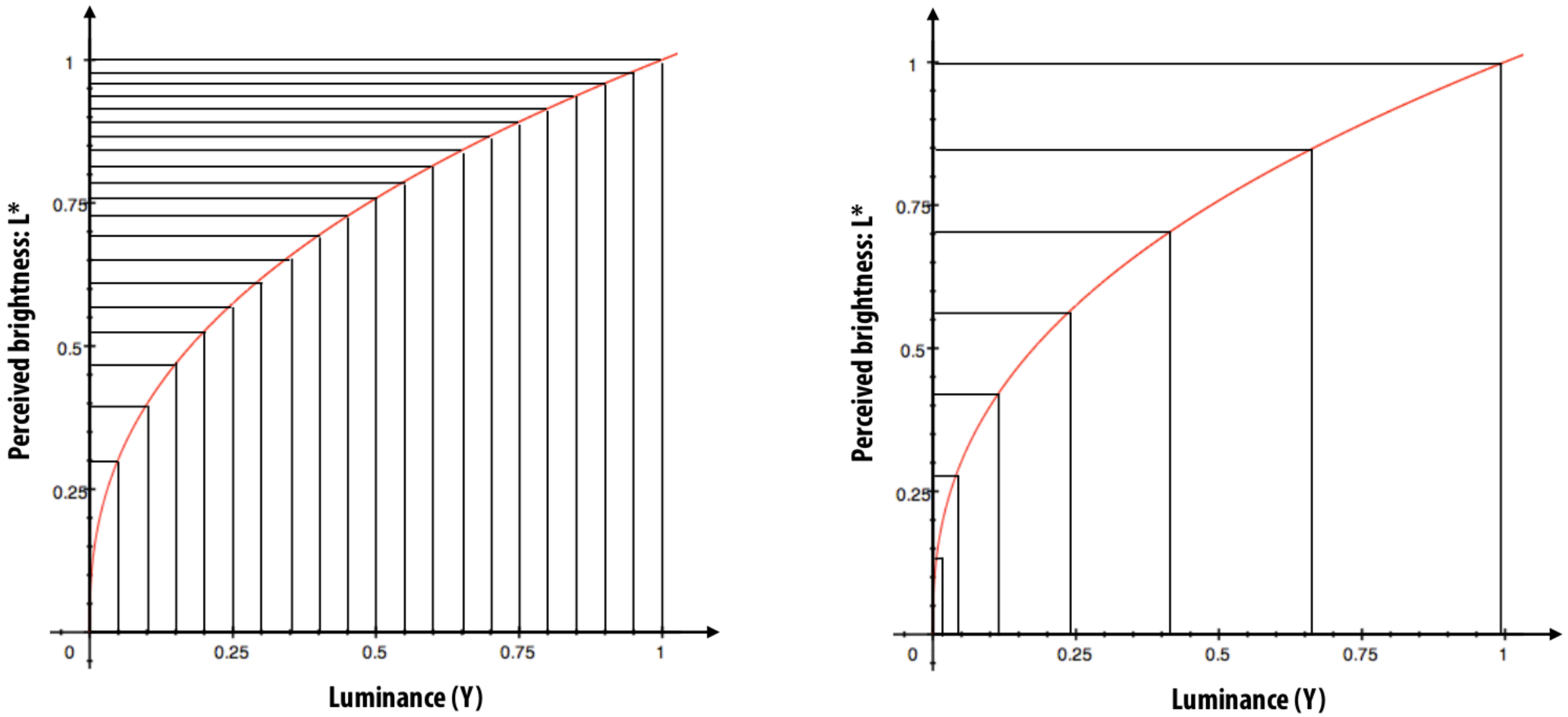
Warning: Our values are no longer linear relative to scene radiance!

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?

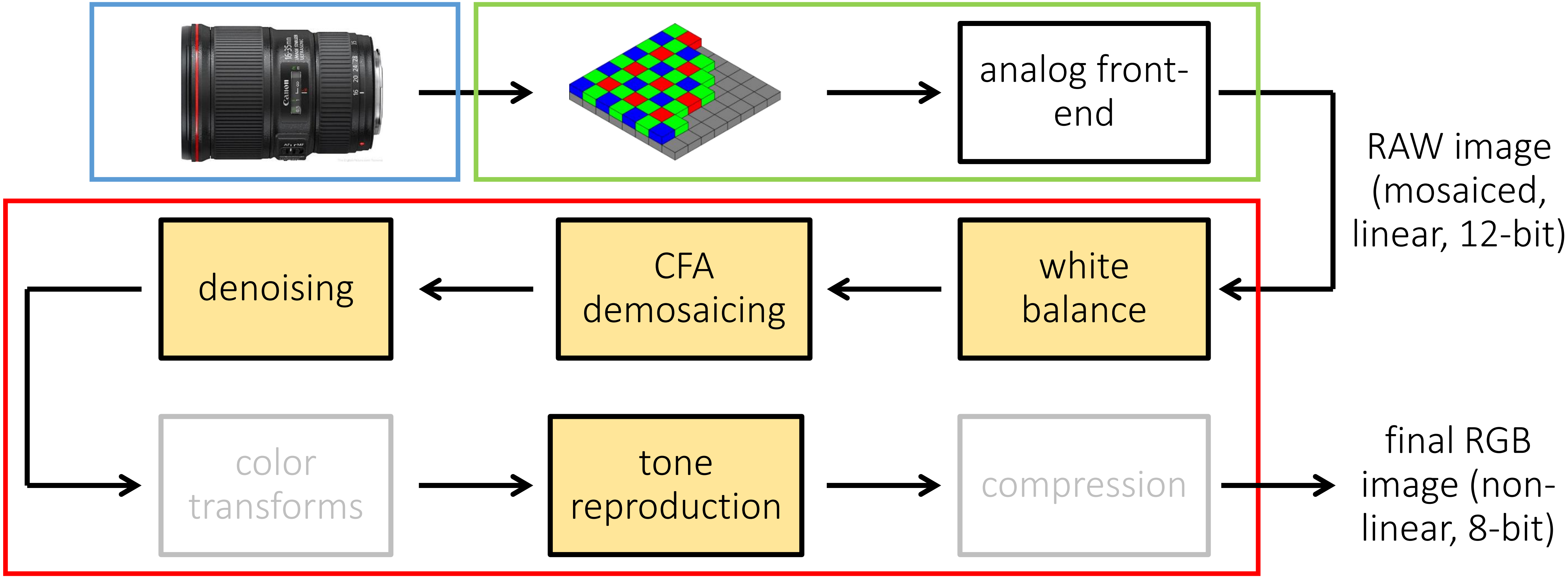


Answer: After this stage, we perform compression, which includes change from 12 to 8 bits.

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

The (in-camera) image processing pipeline

The sequence of 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 DSLR 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.

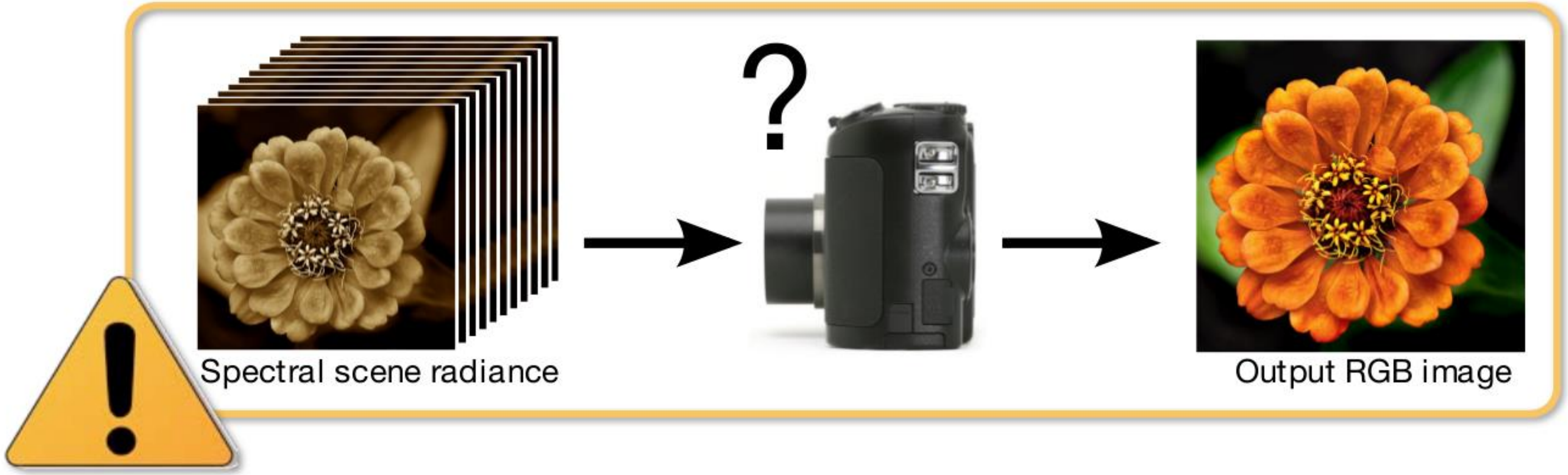
I forgot to set my camera to RAW, can I still get the RAW file?

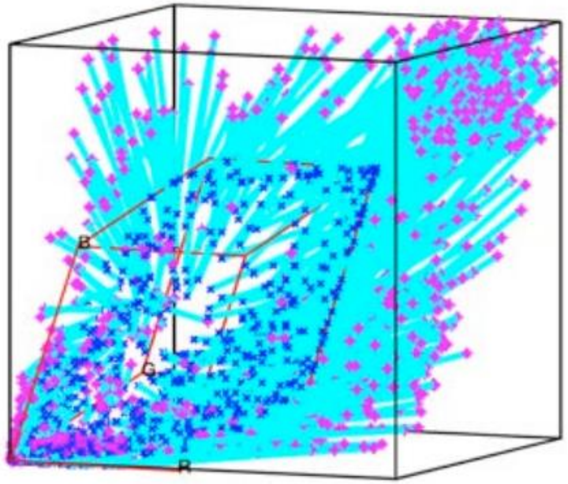
I forgot to set my camera to RAW, can I still get the RAW file?

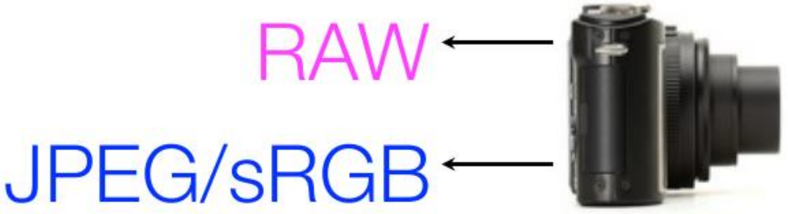
Nope, tough luck.

- 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 know as “de-rendering” and is an active research area.

Derendering







Panasonic DMC-LX3

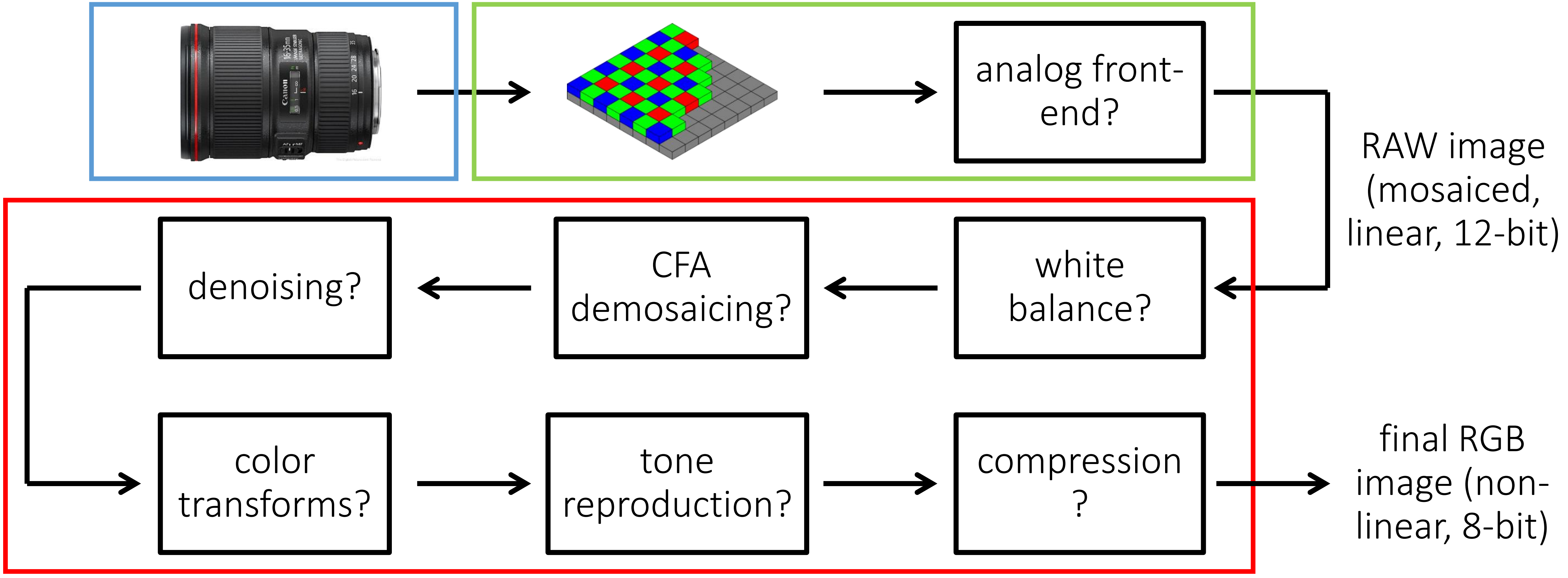
Why did you use italics in the previous slide?

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

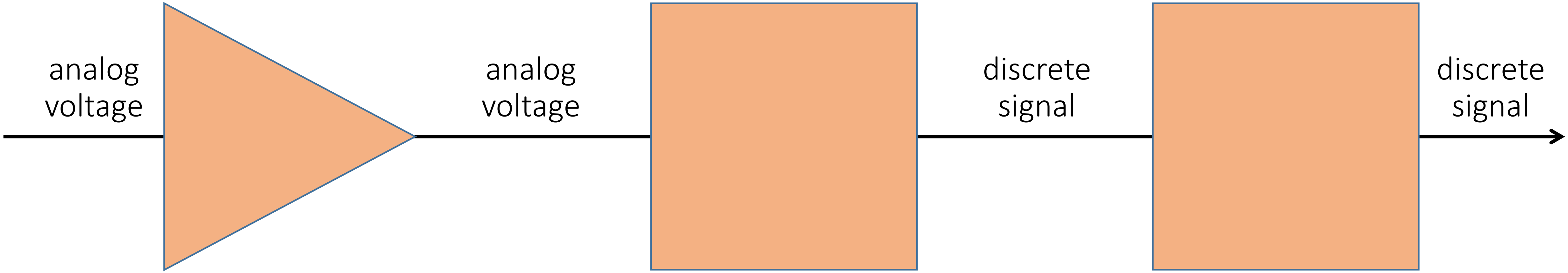
- Almost all of the steps in both the sensor and image processing pipeline I described earlier are camera-dependent.
- Even if we know the basic steps, the implementation details are proprietary information that companies actively try to keep secret.
- I will go back to a few of my slides to show you examples of the above.

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.



The hypothetical analog front-end



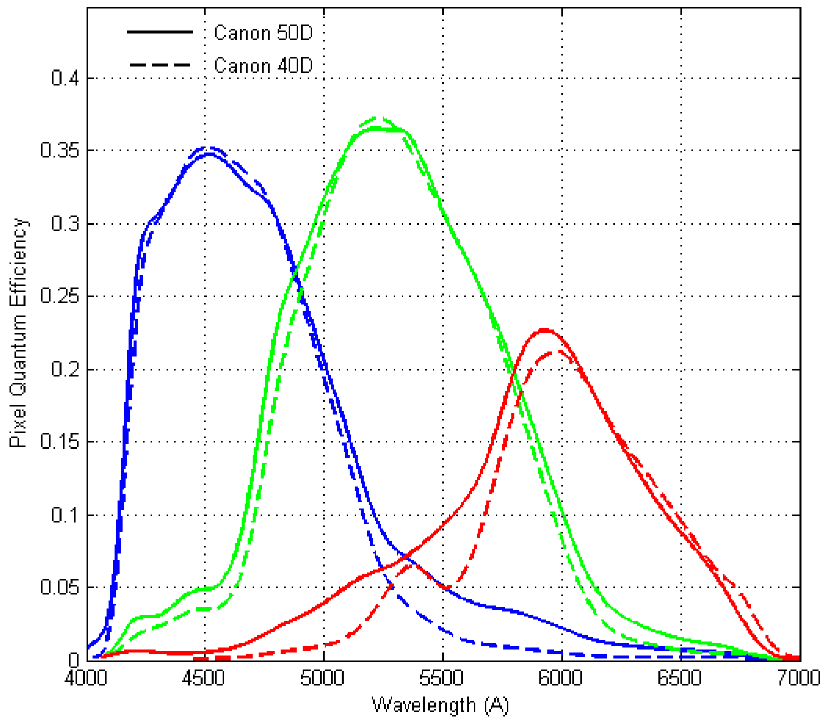
<u>analog amplifier (gain)</u>:
- gets voltage in range needed by A/D converter?
- accommodates ISO settings?
- accounts for vignetting?

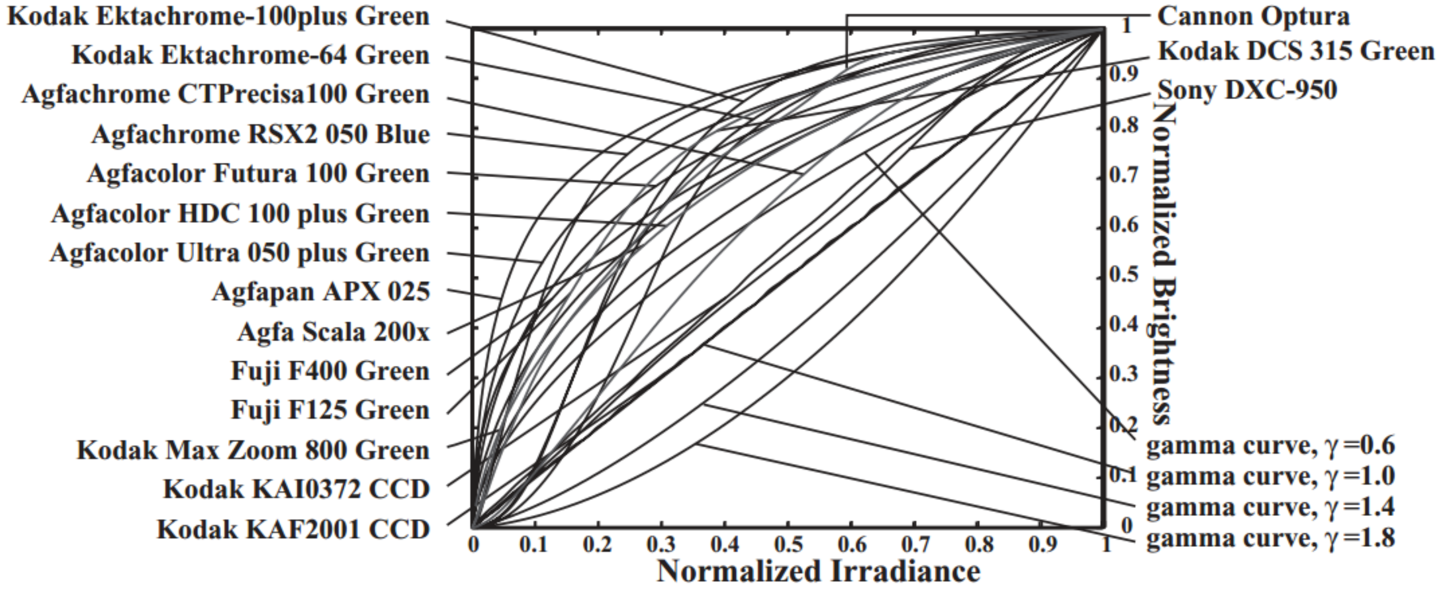
<u>analog-to-digital converter (ADC)</u>:
- depending on sensor, output has 10-16 bits.
- most often (?) 12 bits.

<u>look-up table (LUT)</u>:
- corrects non-linearities in sensor's response function (within proper exposure)?
- corrects defective pixels?

Various curves

All of these sensitivity curves are different from camera to camera and kept secret.





Serious inhibition for research

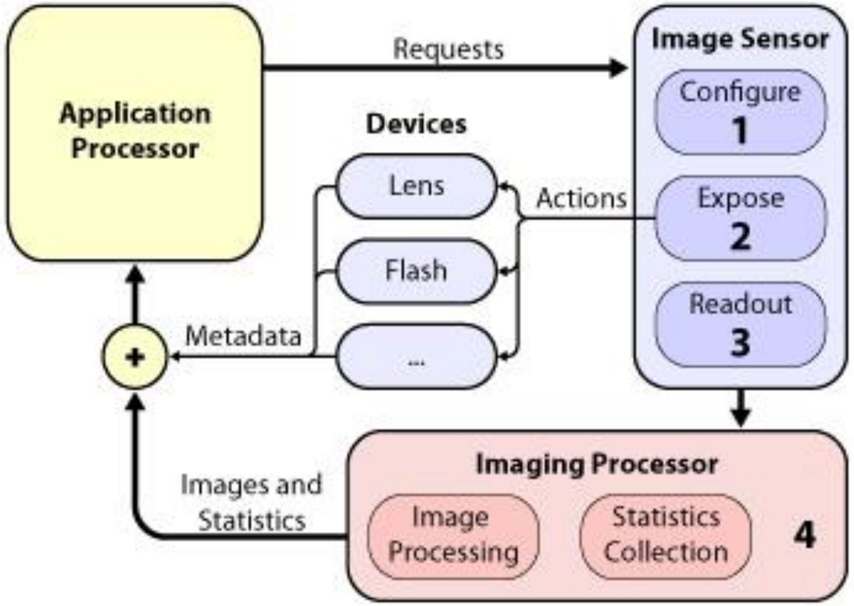
- Very difficult to get access to ground-truth data at intermediate stages of the pipeline.
- Very difficult to evaluate effect of new algorithms for specific pipeline stages.

…but things are getting better

The Frankencamera: An Experimental Platform for Computational Photography

Andrew Adams, Eino-Ville Talvala, Sung Hee Park, David E. Jacobs, Boris Ajdin, Natasha Gelfand, Jennifer Dolson, Daniel Vaquero, Jongmin Baek, Marius Tico, Hendrik P. A. Lensch, Wojciech Matusik, Kari Pulli, Mark Horowitz, Marc Levoy

Presented at SIGGRAPH 2010



…but things are getting better

Camera 2 API Overview

- ● Android.hardware.camera2 API to facilitate fine-grain photo capture and image processing.
- ● The android.hardware.camera2 package provides an interface to individual camera devices connected to an Android device. It replaces the deprecated Camera class.

How do I open a RAW file in Matlab?

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 0 for more details.

Radiometric calibration
(a.k.a. high dynamic range imaging)
(a.k.a. capturing linear images)

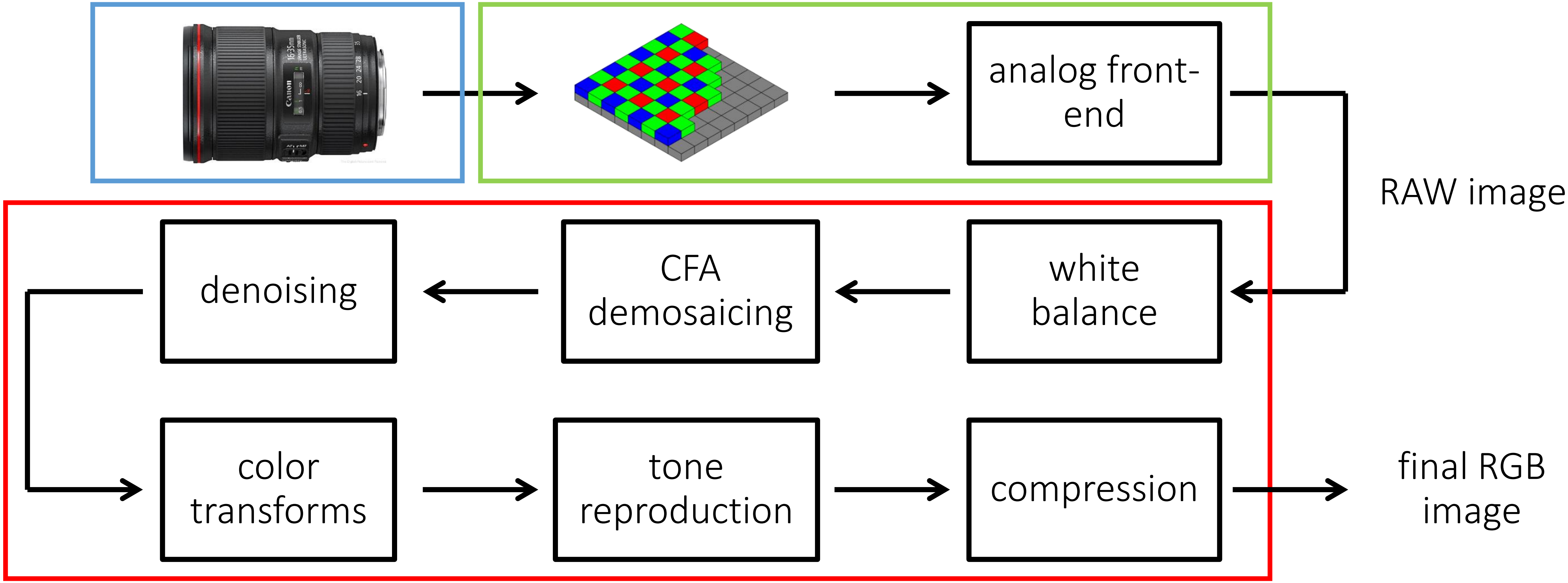
What does it mean to “calibrate a camera”?

Many different ways to calibrate a camera:

- Radiometric calibration.
- Color calibration.
- Geometric calibration.
- Noise calibration.
- Lens (or aberration) calibration.

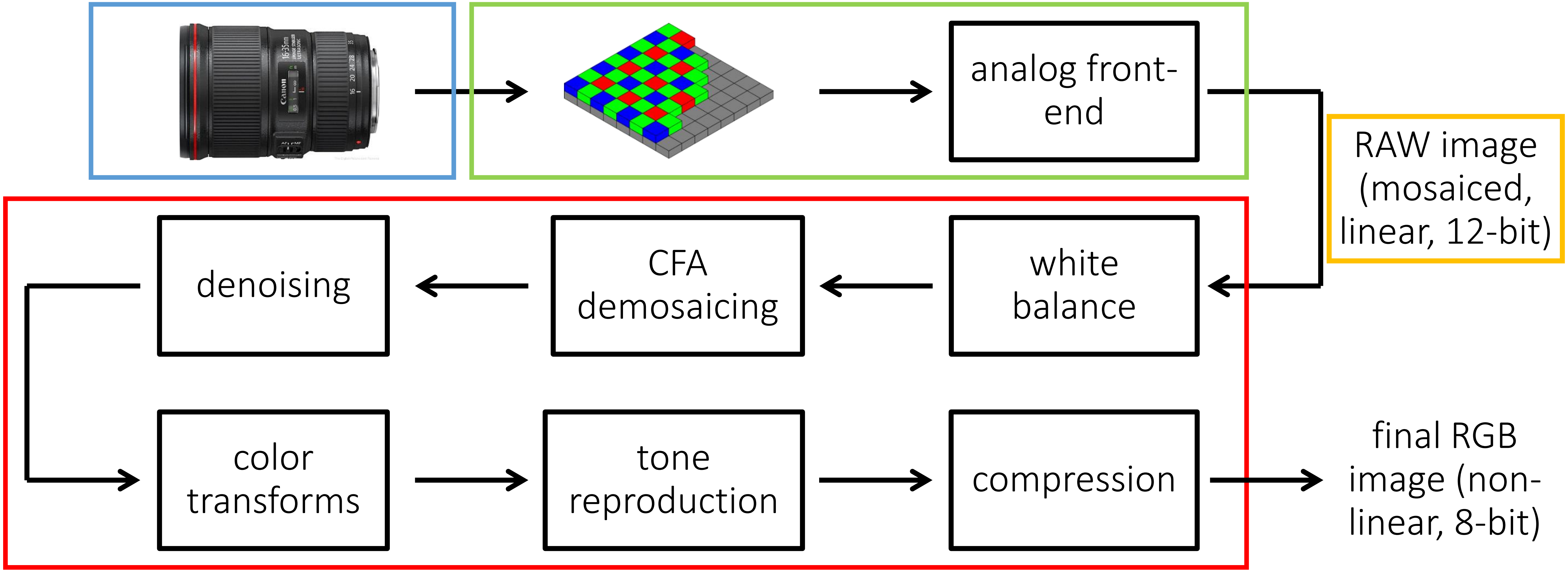
The image processing pipeline

Which parts of the image processing pipeline introduce non-linearities?



The image processing pipeline

Is using RAW images sufficient to get linear images?



Photodiode response function

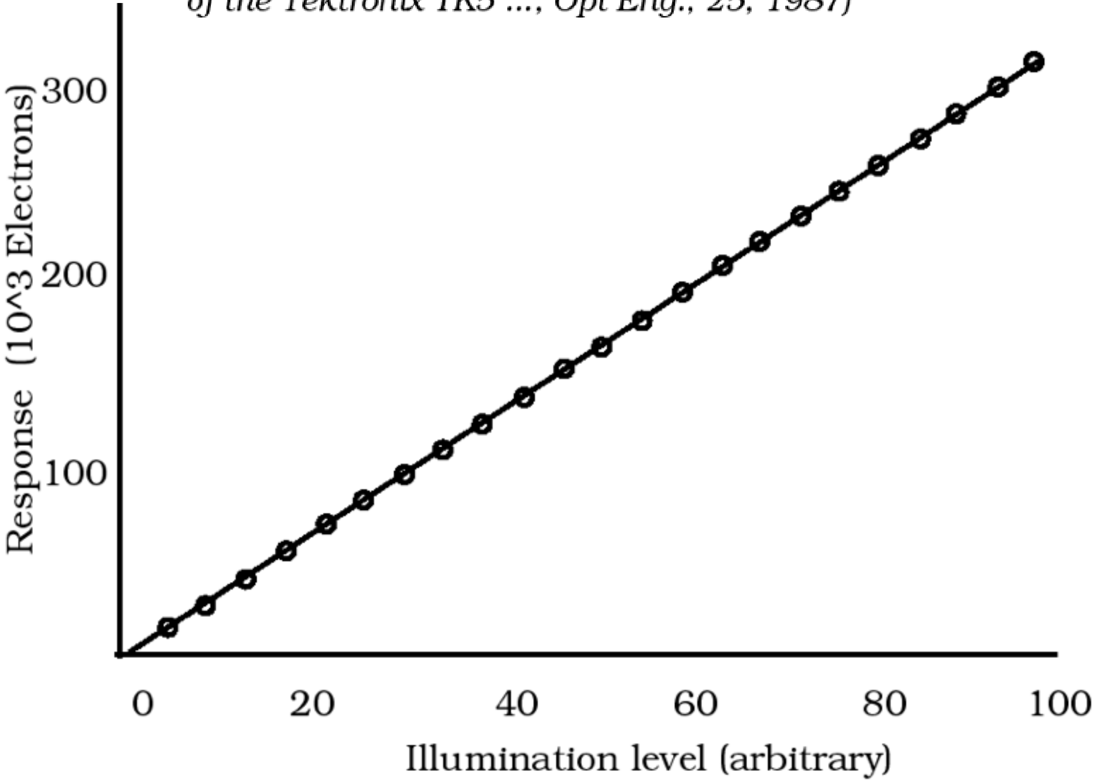
For silicon photodiodes, <u>usually</u> linear, but:

- <u>non-linear when potential well is saturated (over-exposure)</u>

- <u>non-linear near zero (due to noise)</u>

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

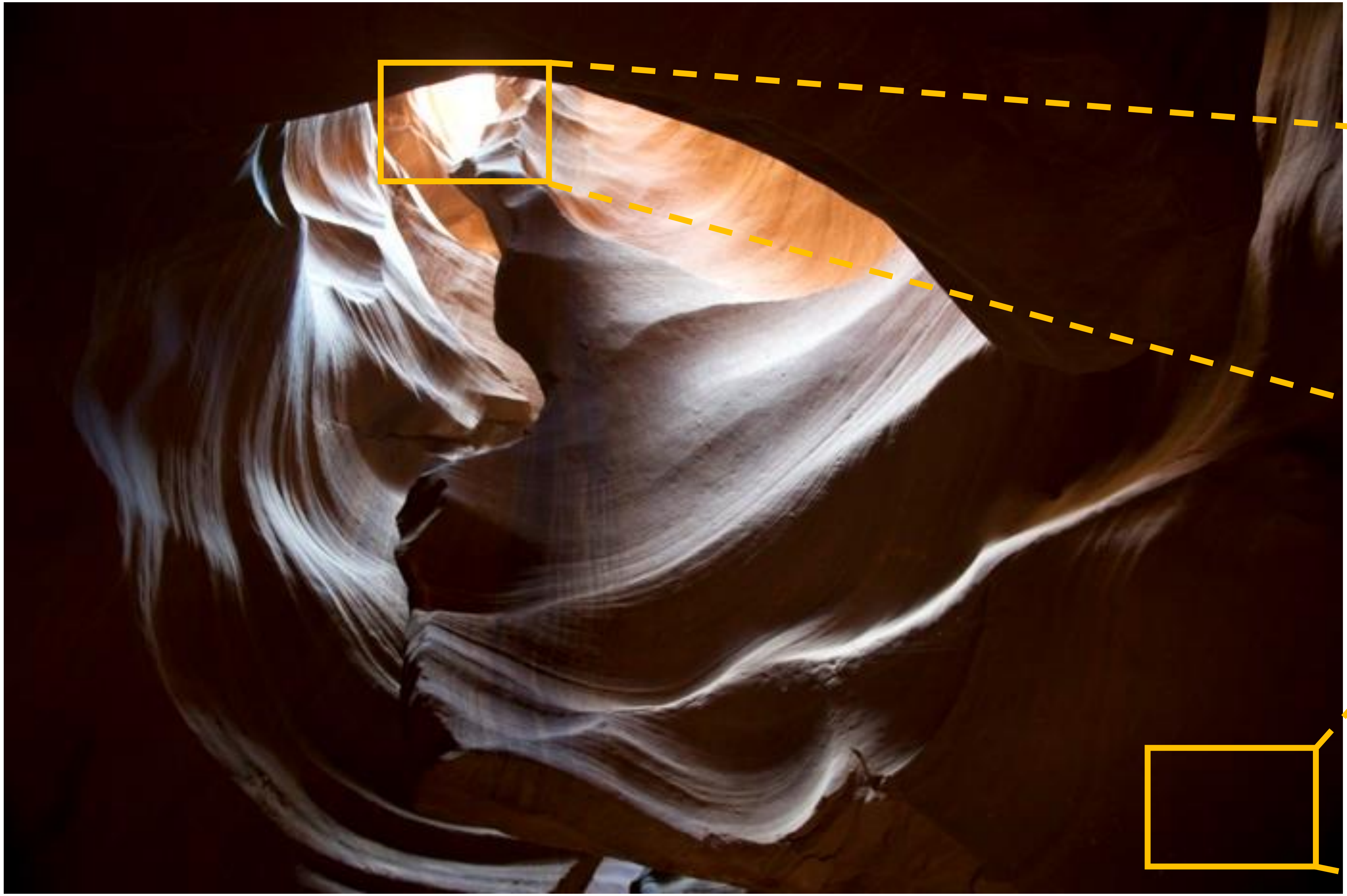
(Epperson, P.M. et al. Electro-optical characterization of the Tektronix TK5 …, Opt Eng., 25, 1987)



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

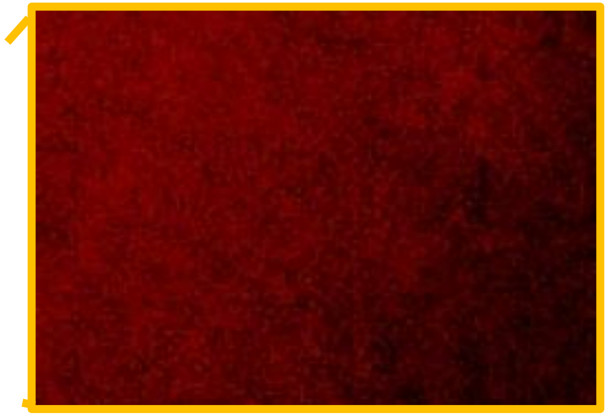
over-exposure (non-linearity due to sensor saturation)

Over/under exposure



in highlights we are limited by clipping

in shadows we are limited by noise




HYATT


HYATT


HYATT


THE INTERNATIONAL FUR EXCHANGE


HYATT

Our devices do not match the world

The world has a high dynamic range

1

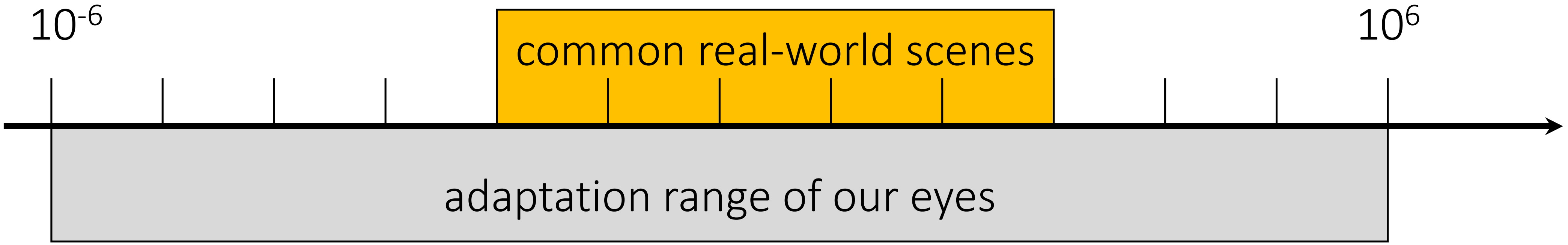
1500

25,000

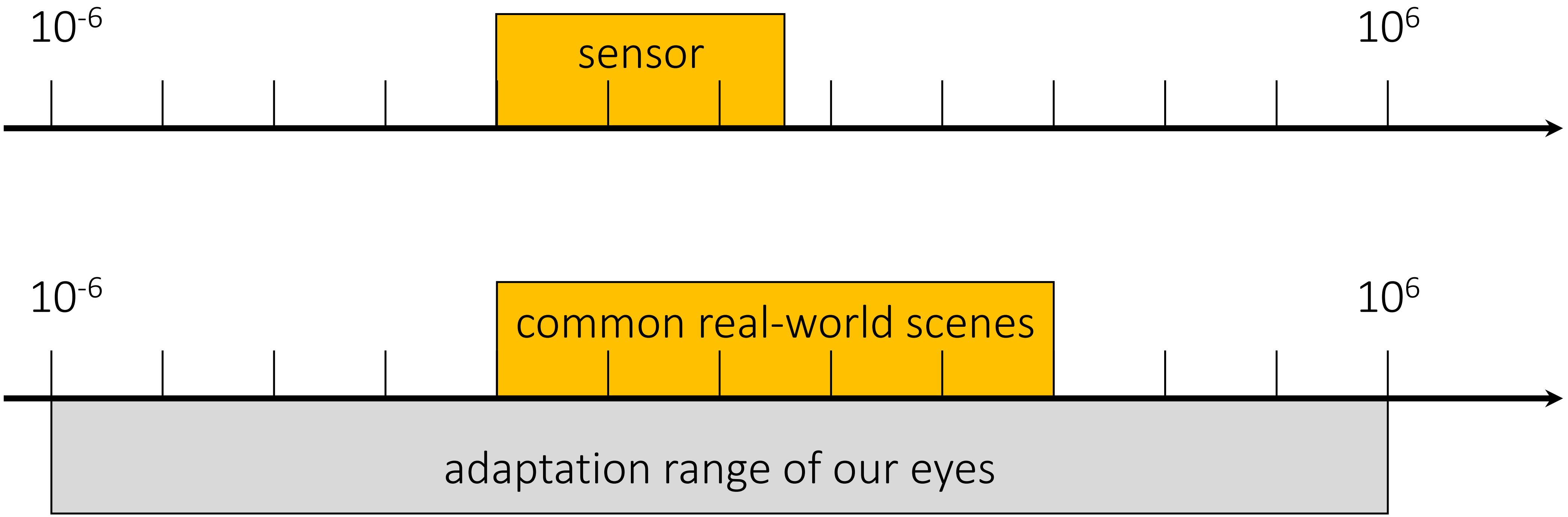
400,000

2,000,000,000

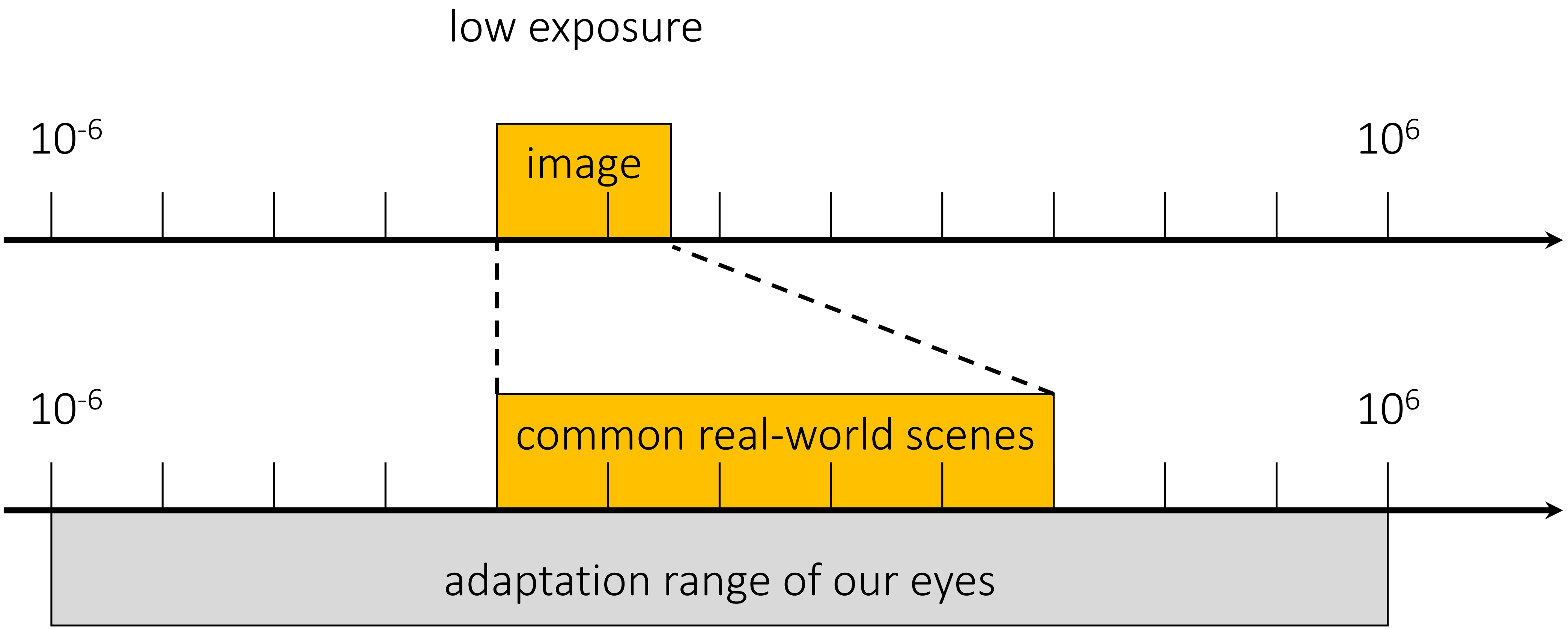
The world has a high dynamic range



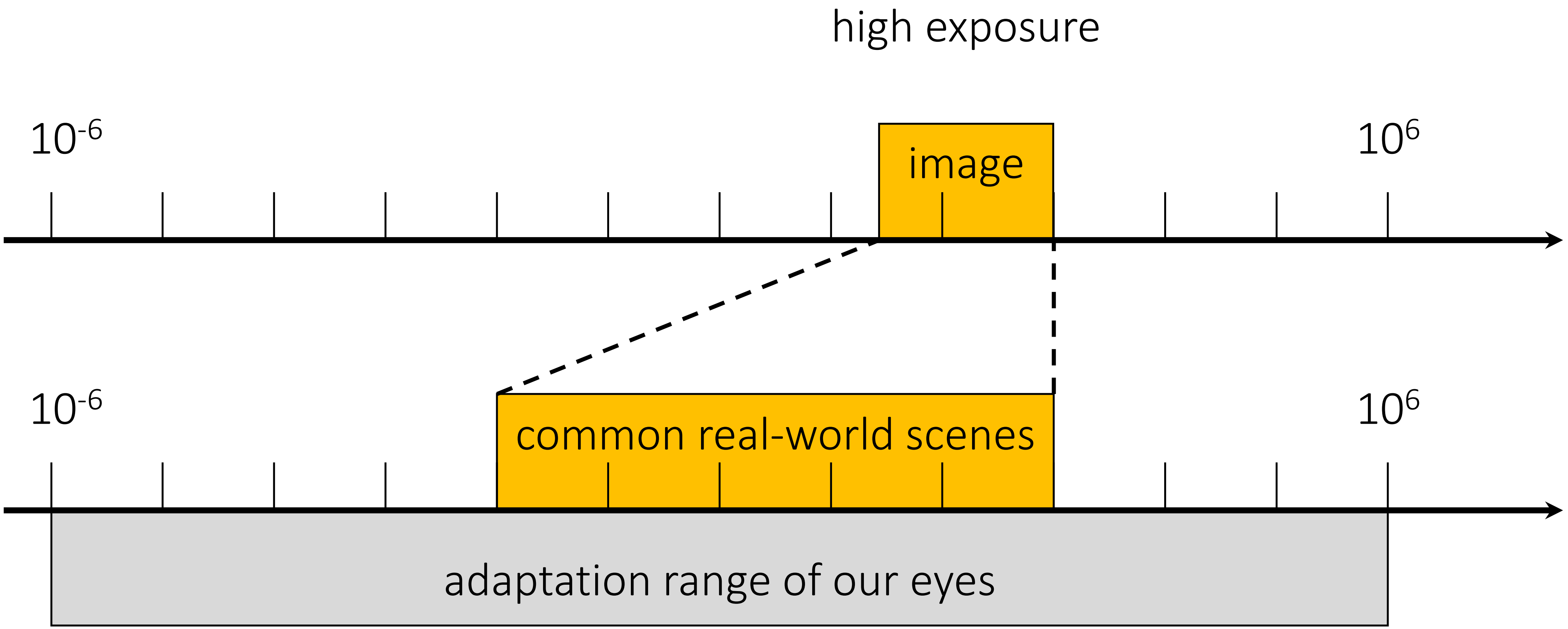
(Digital) sensors also have a low dynamic range



(Digital) images have an even lower dynamic range



(Digital) images have an even lower dynamic range



Our devices do not match the real world

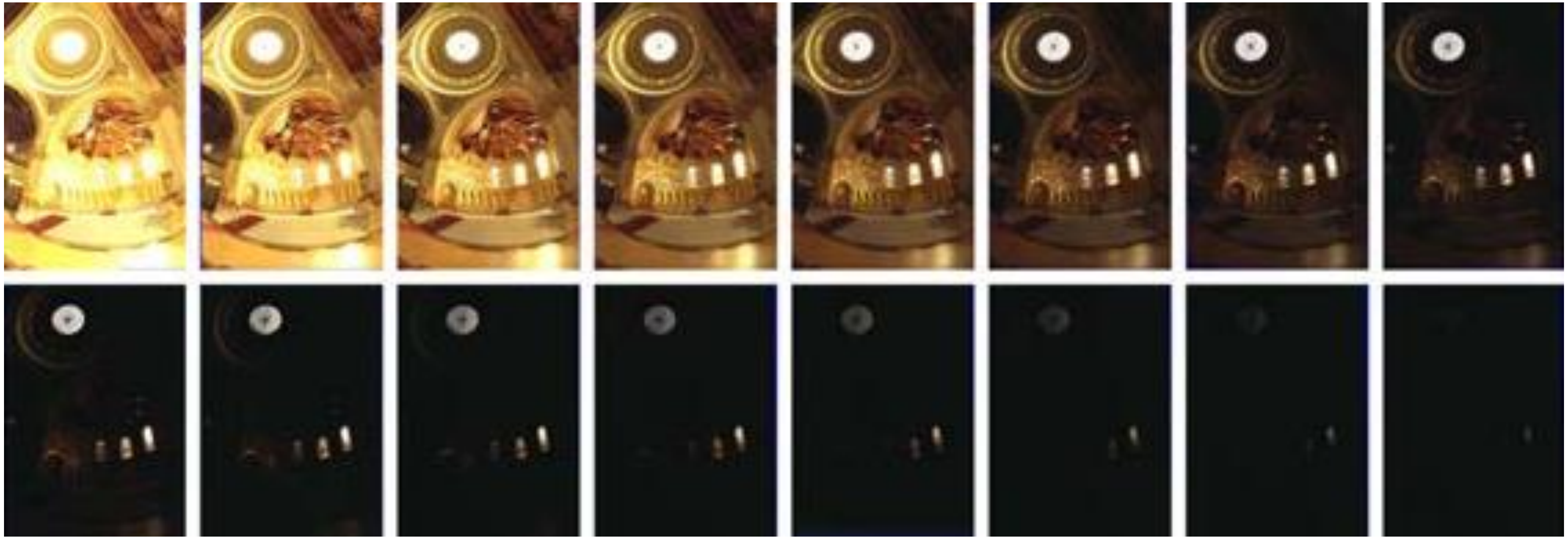
10:1	photographic print (higher for glossy paper)
20:1	artist's paints
200:1	slide film
500:1	negative film
1000:1	LCD display
2000:1	digital SLR (at 12 bits)
100000:1	real world

Two challenges:

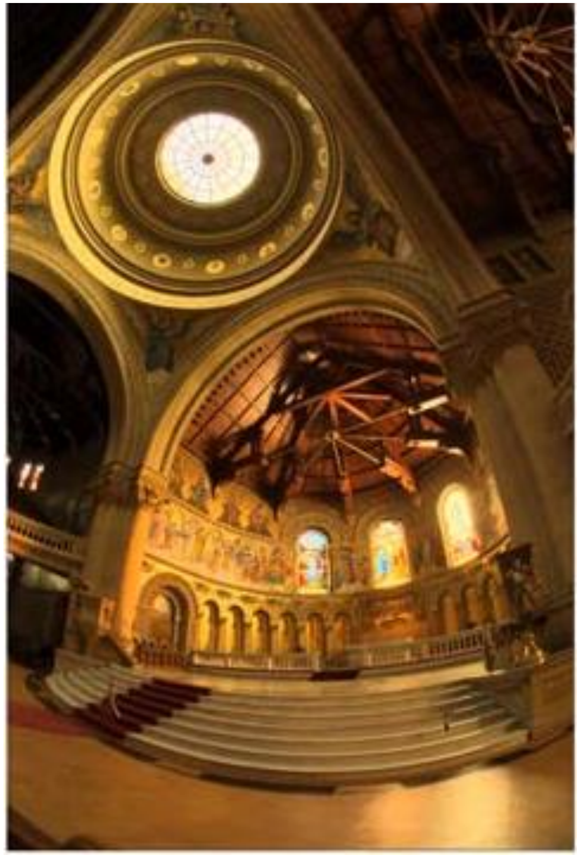
1. HDR imaging – which parts of the world to include to the 8-12 bits available to our device?
2. Tonemapping – which parts of the world to display in the 4-10 bits available to our device?

Key idea

1. Capture multiple LDR images at different exposures

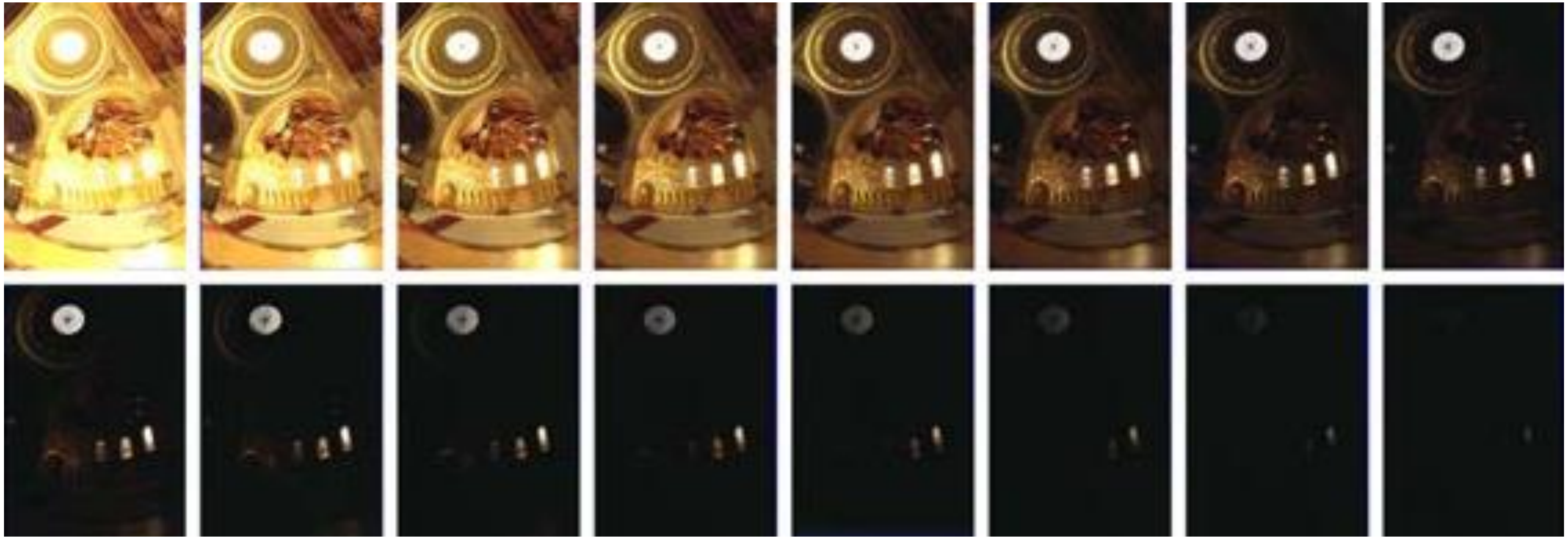


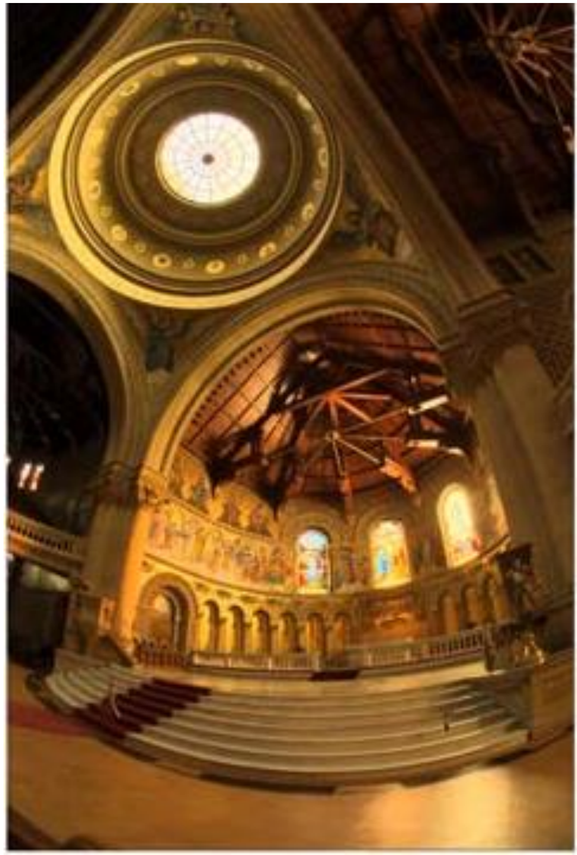
2. Merge them into a single HDR image



Key idea

1. Exposure bracketing: Capture multiple LDR images at different exposures

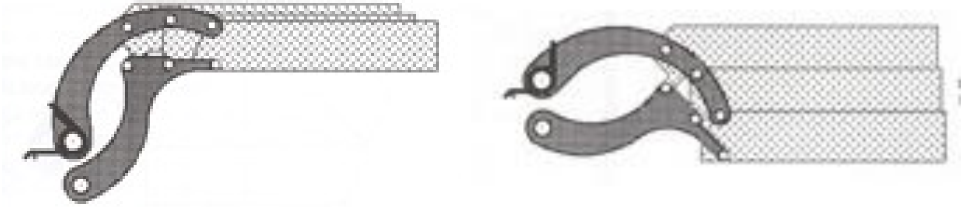




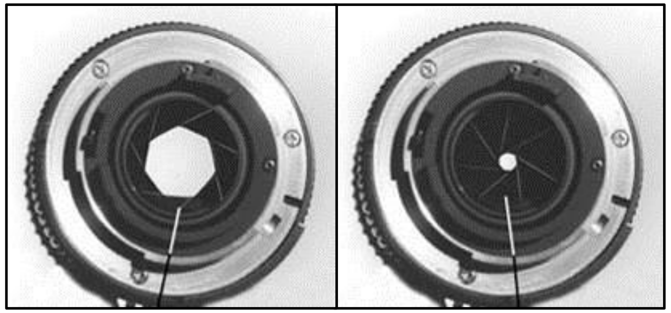
2. Merging: Combine them into a single HDR image

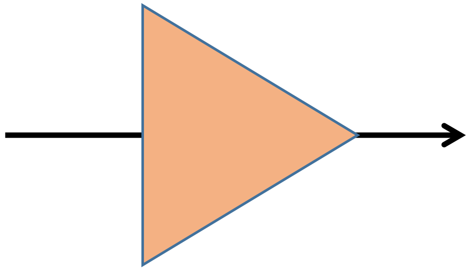
Ways to vary exposure

1. Shutter speed



2. F-stop (aperture, iris)





3. ISO

4. Neutral density (ND) filters

Pros and cons of each for HDR?

Ways to vary exposure

1. Shutter speed
   - Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   - Pros: repeatable, linear
   - Cons: noise and motion blur for long exposure
2. F-stop (aperture, iris)
   - Range: about f/0.98 to f/22 (3 orders of magnitude)
   - Pros: fully optical, no noise
   - Cons: changes depth of field
3. ISO
   - Range: about 100 to 1600 (1.5 orders of magnitude)
   - Pros: no movement at all
   - Cons: noise
4. Neutral density (ND) filters
   - Range: up to 6 densities (6 orders of magnitude)
   - Pros: works with strobe/flash
   - Cons: not perfectly neutral (color shift), extra glass (interreflections, aberrations), need to touch camera (shake)

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each “stop” is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec
usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Questions:

1. How many exposures?
2. What exposures?

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each "stop" is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

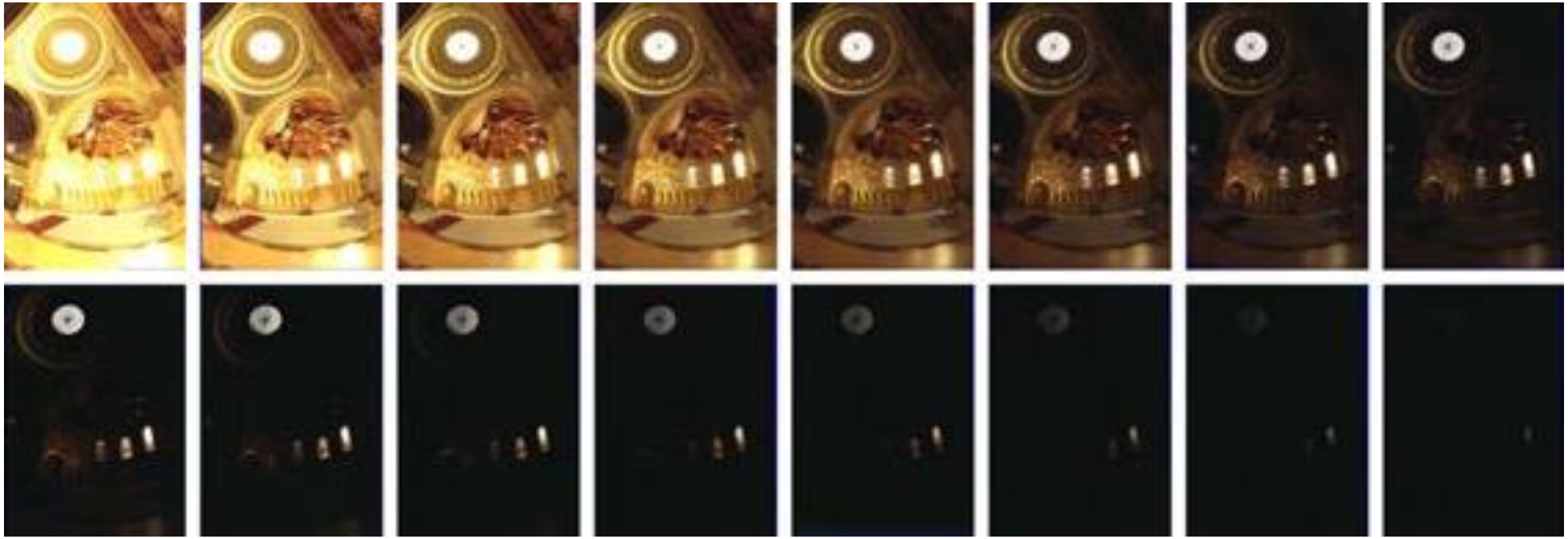
Questions:

1. How many exposures?
2. What exposures?

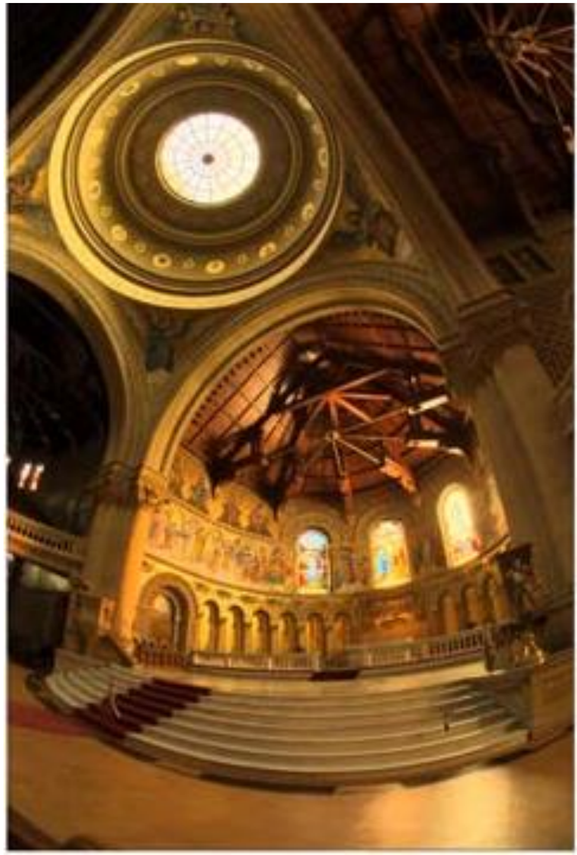
Answer: Depends on the scene, but a good default is 5 exposures, the metered exposure and +/- 2 stops around that.

Key idea

1. Exposure bracketing: Capture multiple LDR images at different exposures

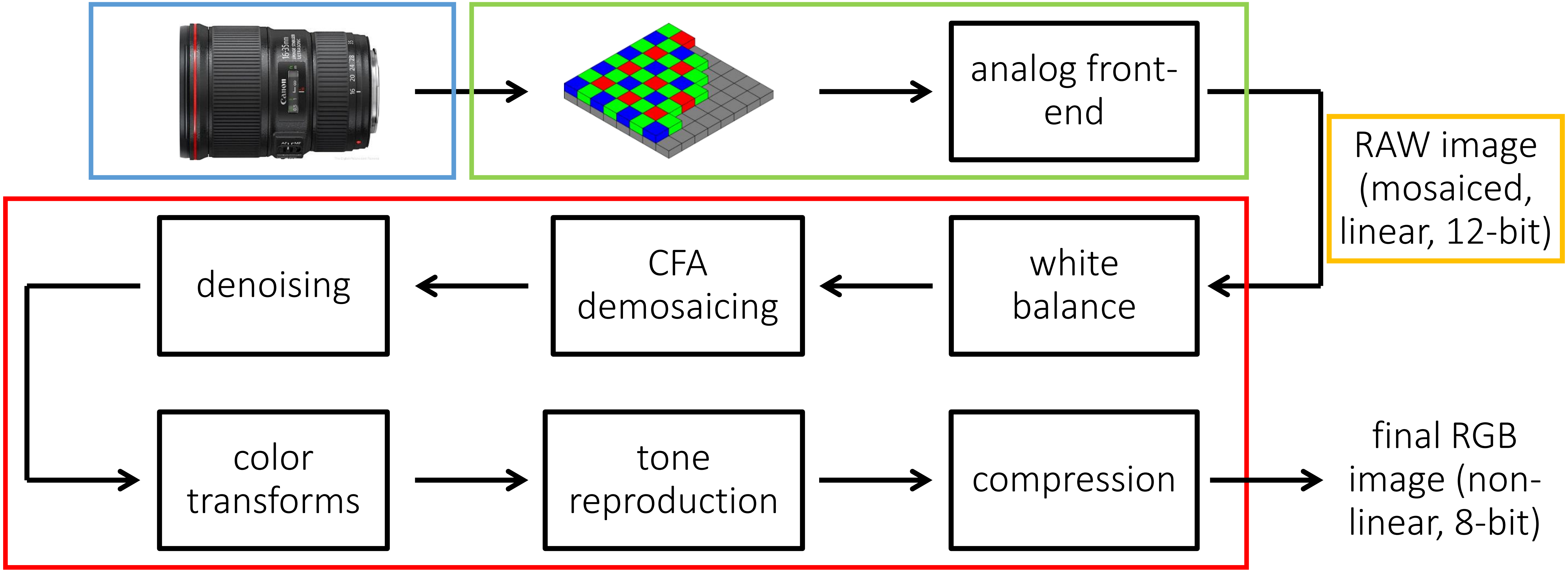


2. Merging: Combine them into a single HDR image



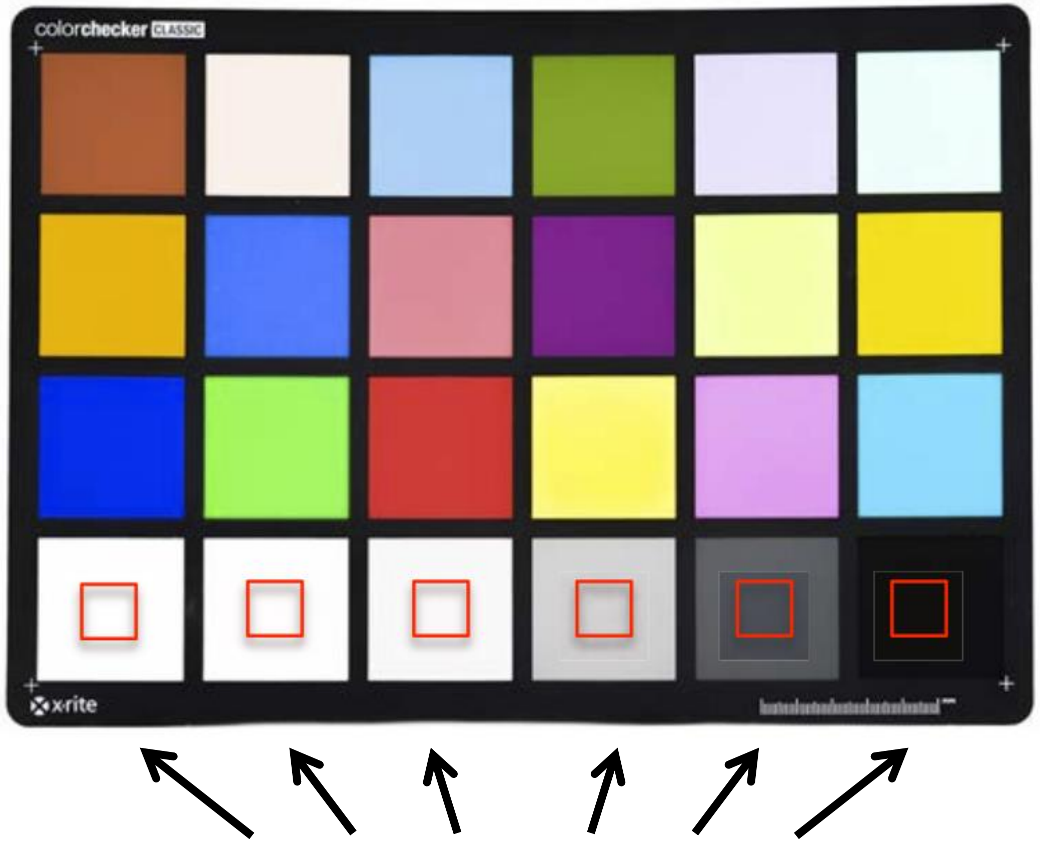
The image processing pipeline

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

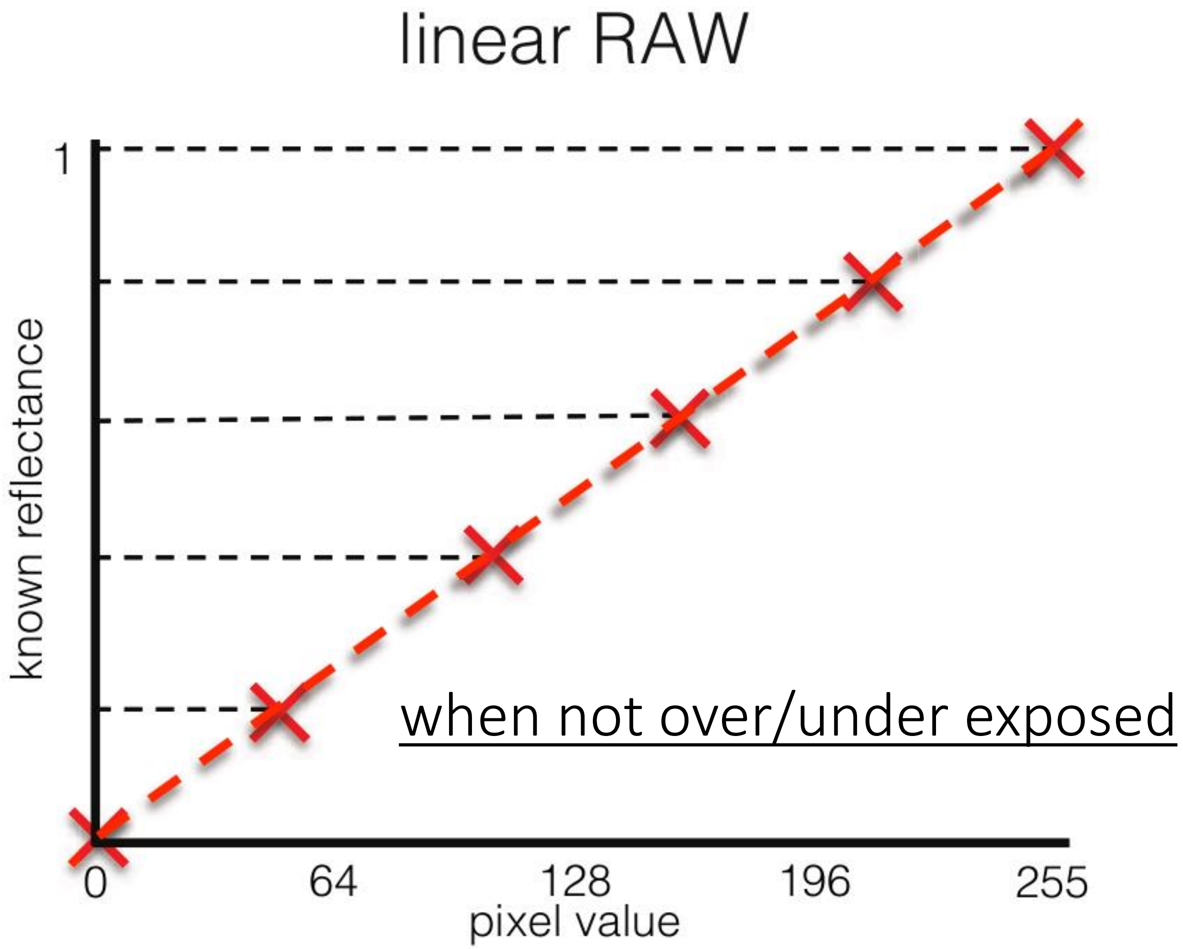


RAW images have a linear response curve

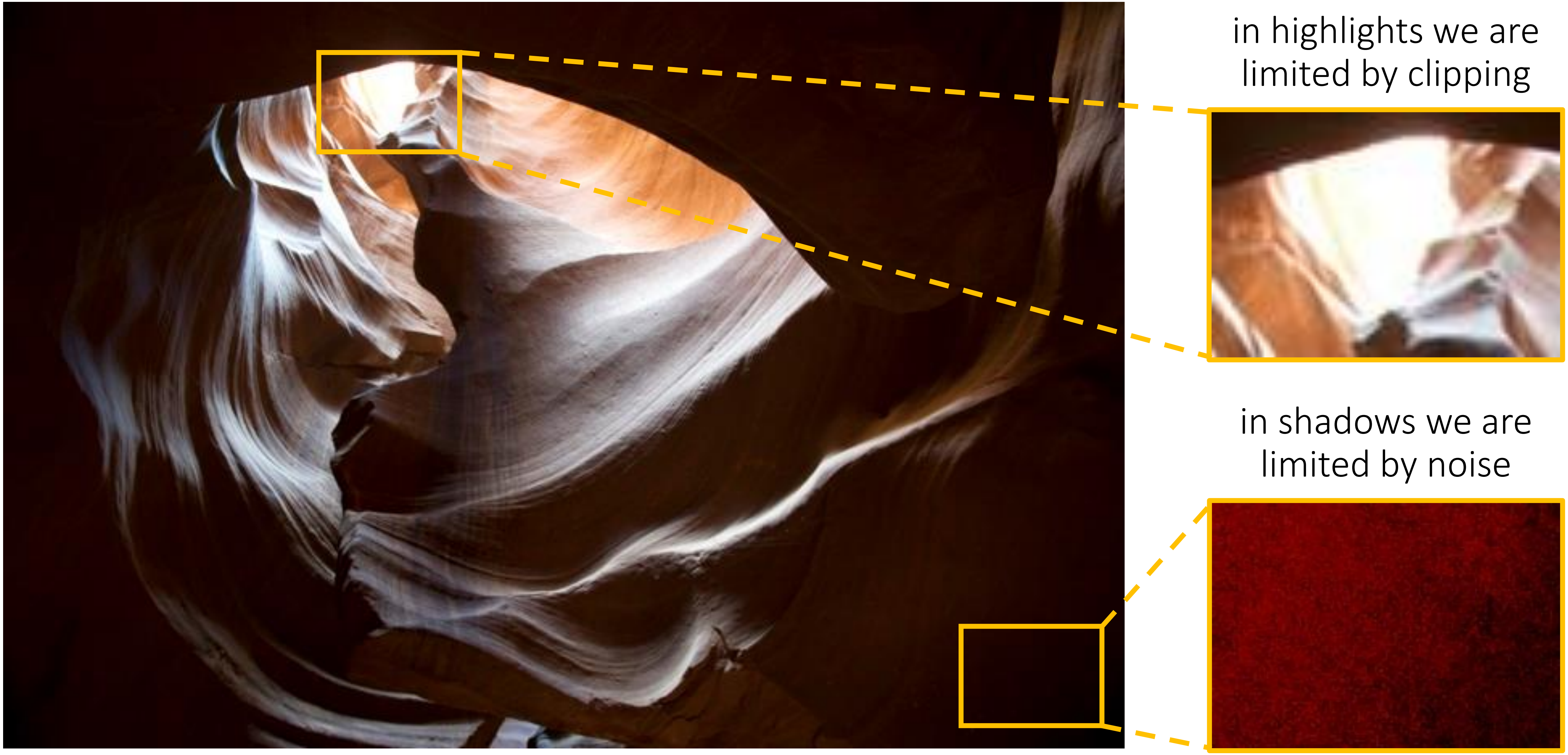
Colorchecker: Great tool for radiometric and color calibration.



Patches at bottom row have reflectance that increases linearly.



Over/under exposure



in highlights we are limited by clipping

in shadows we are limited by noise

RAW (linear) image formation model

Real scene radiance for image pixel (x,y): $L(x, y)$

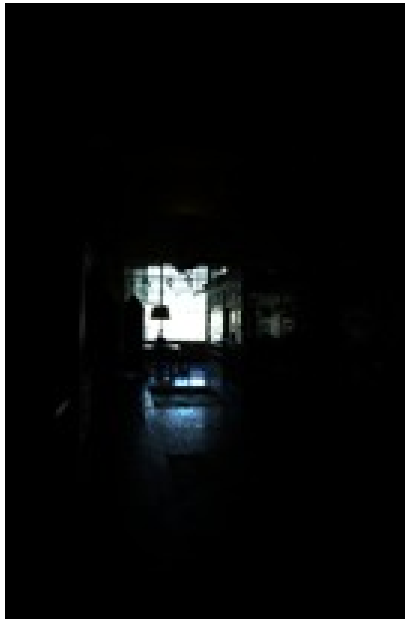
Exposure time:

t_5

t_4

t_3

t_2



t_1

What is an expression for the image $I_{linear}(x,y)$ as a function of $L(x,y)$?

RAW (linear) image formation model

Real scene radiance for image pixel (x,y): $L(x, y)$

Exposure time:

t_5 t_4 t_3 t_2 t_1

What is an expression for the image $I_{linear}(x,y)$ as a function of $L(x,y)$?

$$I_{linear}(x,y) = \text{clip}[\, t_i \cdot L(x,y) + \text{noise} \,]$$

How would you merge these images into an HDR one?

Merging RAW (linear) exposure stacks

For each pixel:

1. Find “valid” images

2. Weight valid pixel values appropriately

3. Form a new pixel value as the weighted average of valid pixel values

How would you implement steps 1-2?

t_5 t_4 t_3 t_2 t_1

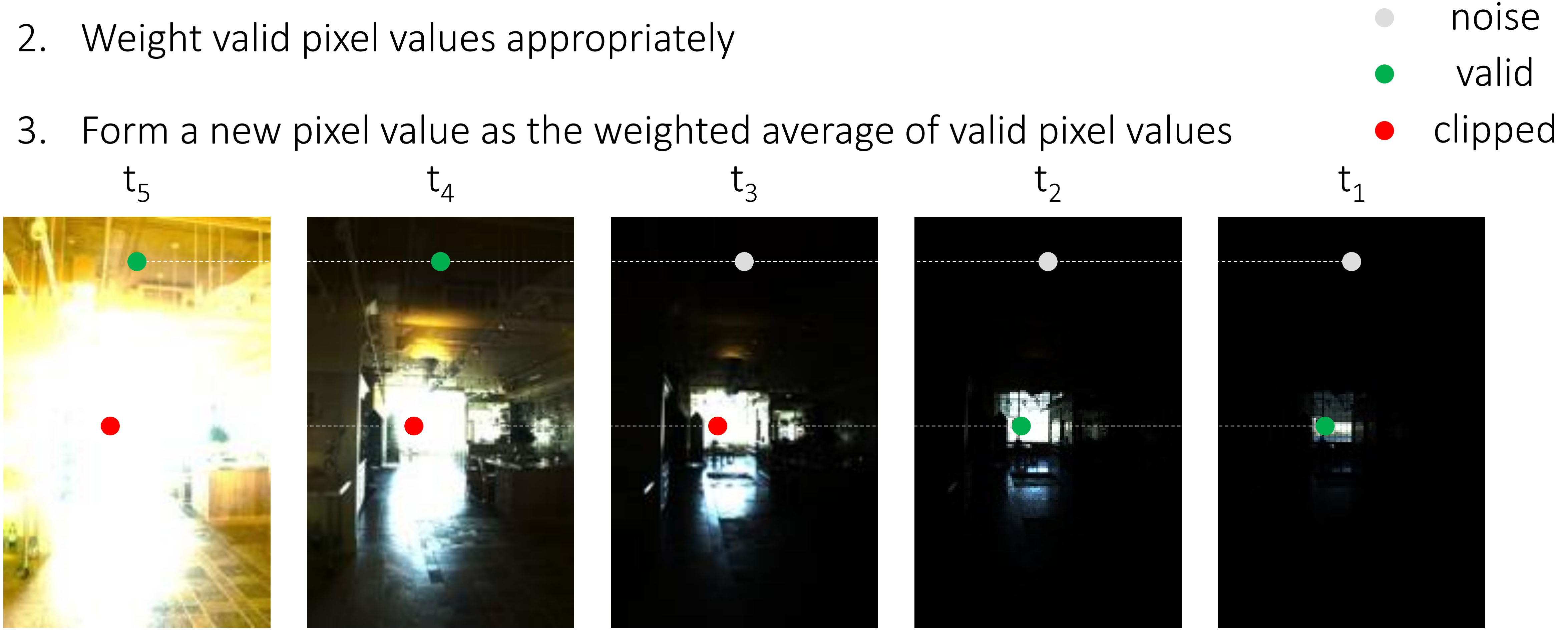
Merging RAW (linear) exposure stacks

For each pixel:

1. Find “valid” images ⟵ (noise) 0.05 < pixel < 0.95 (clipping)

2. Weight valid pixel values appropriately

3. Form a new pixel value as the weighted average of valid pixel values



Merging RAW (linear) exposure stacks

For each pixel:

1. Find “valid” images ⟵ (noise) 0.05 < pixel < 0.95 (clipping)

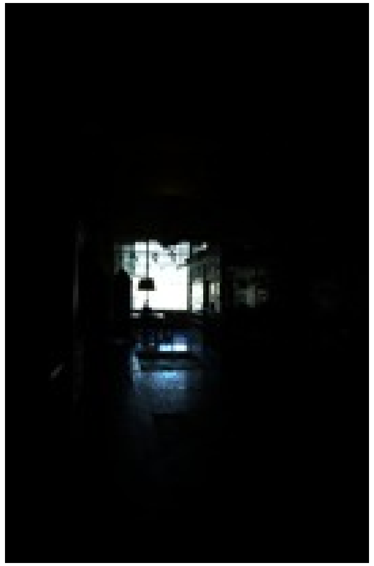
2. Weight valid pixel values appropriately ⟵ (pixel value) / t_i

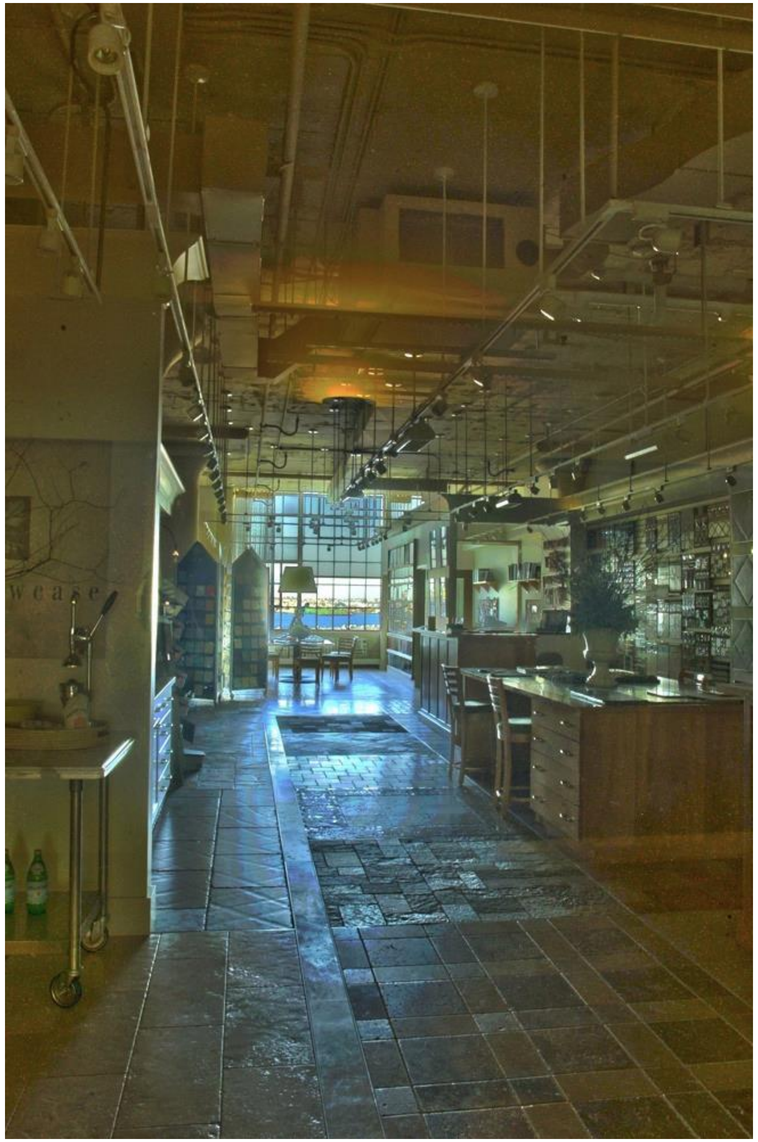
3. Form a new pixel value as the weighted average of valid pixel values

t_5 t_4 t_3 t_2 t_1

Merging result (after tonemapping)



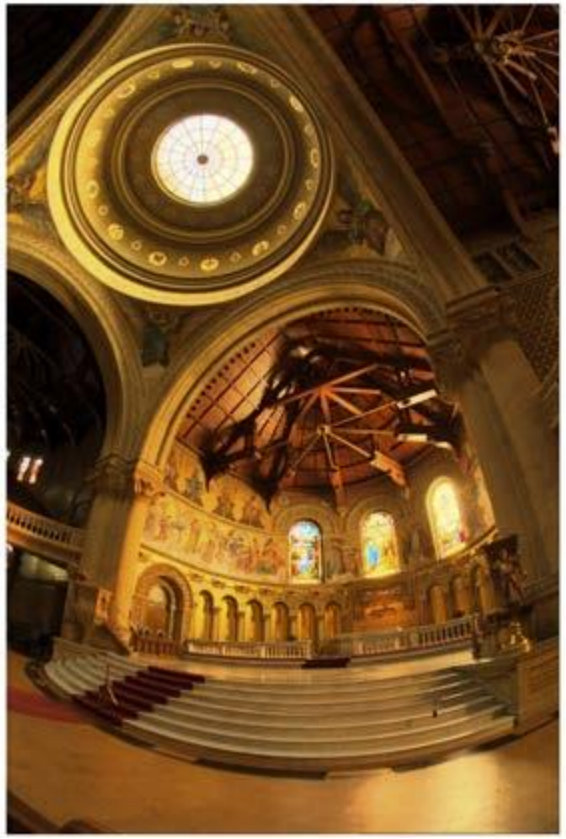





Relative vs absolute radiance

Final fused HDR image gives radiance only up to a global scale

- If we know exact radiance at one point, we can convert relative HDR image to absolute radiance map

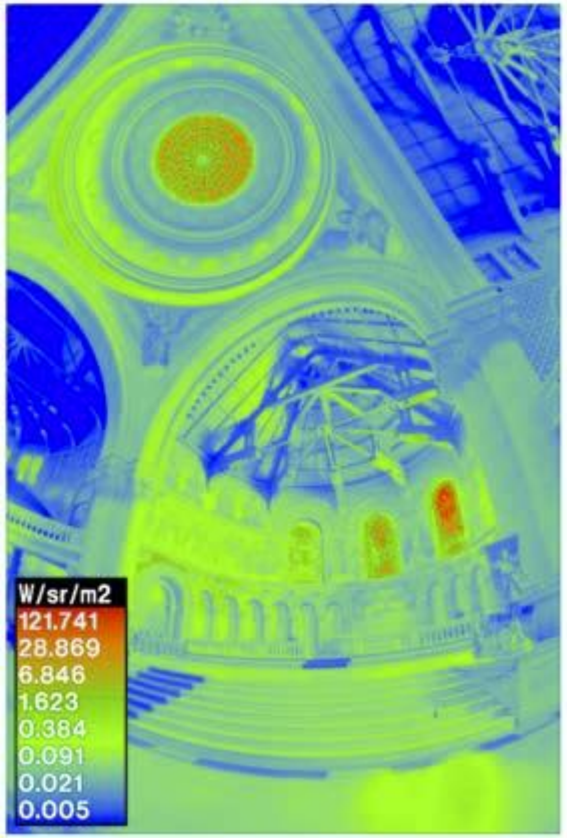


HDR image (relative radiance)

→

spotmeter (absolute radiance at one point)

→

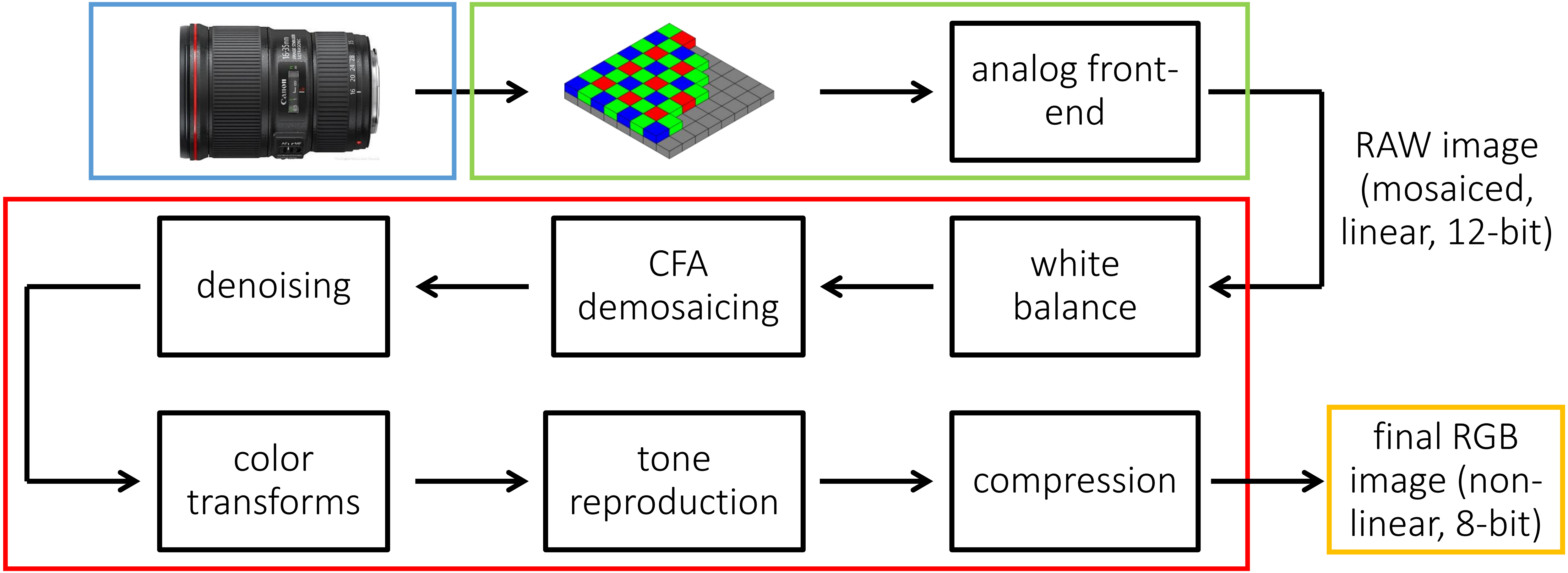


absolute radiance map

What if I cannot use raw?

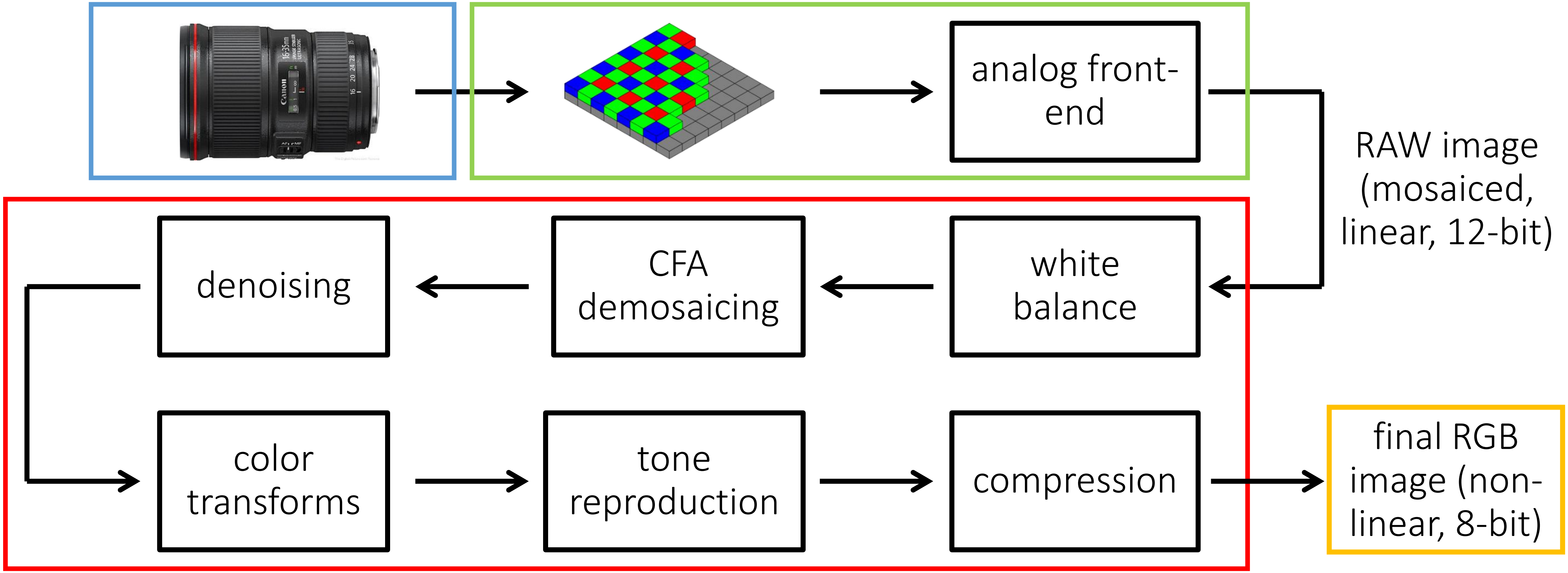
The image processing pipeline

- Can you foresee any problem when we switch from RAW to rendered images?



The image processing pipeline

- Can you foresee any problem when we switch from RAW to rendered images?
- How do we deal with the nonlinearities?



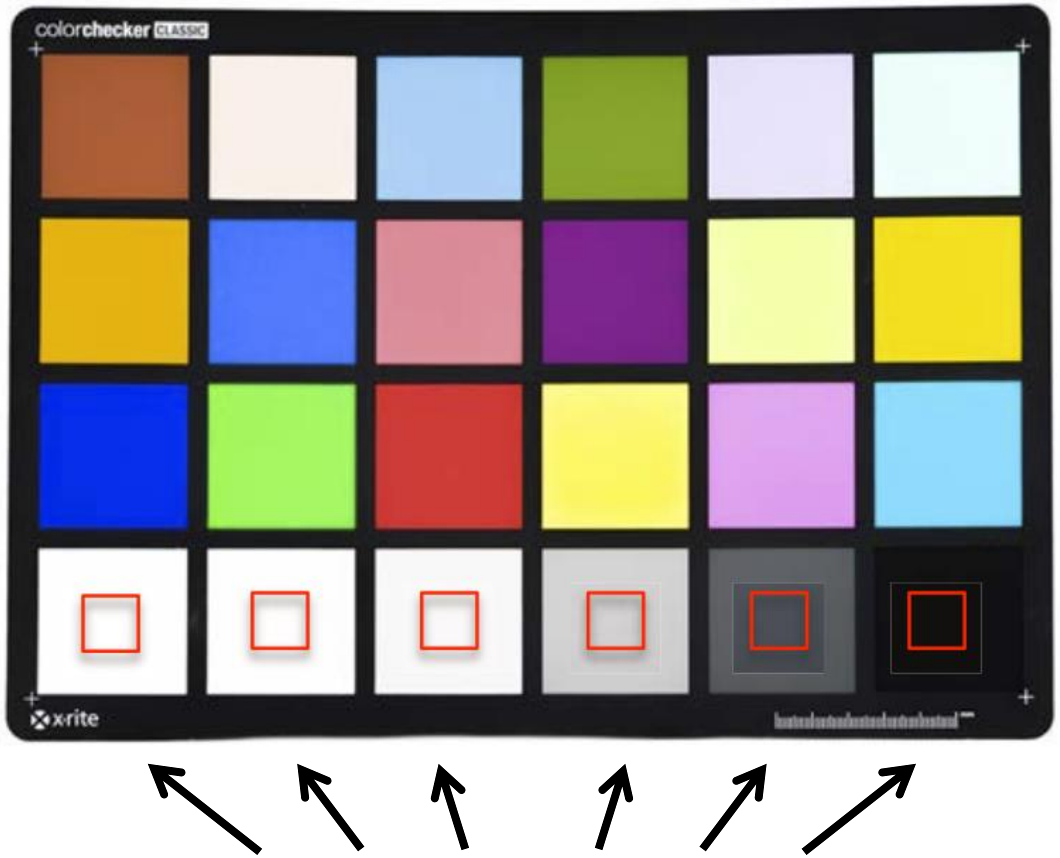
Radiometric calibration

The process of measuring the camera's response curve. Can be two in three ways:

- Take images of scenes with different irradiance while keeping exposure the same.
- Takes images under different exposures while keeping irradiance the same.
- Takes images of scenes with different irradiance and under different exposures.

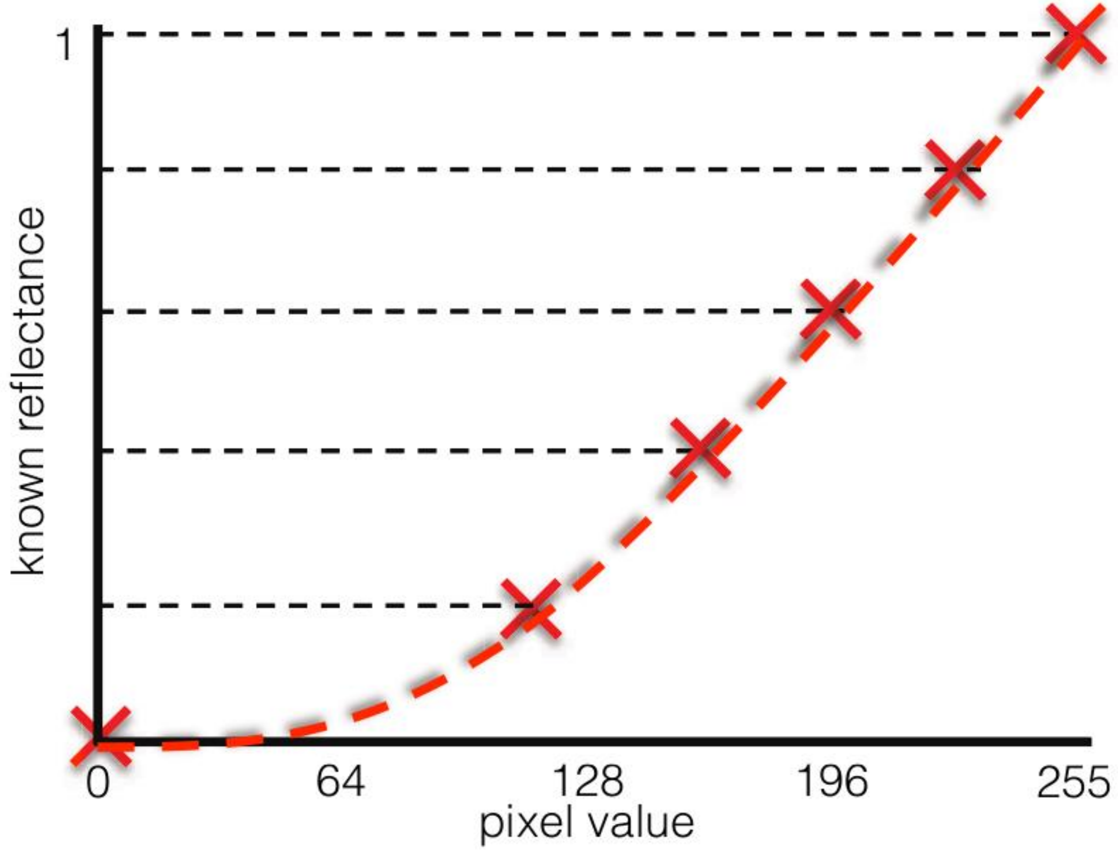
Same camera exposure, varying scene irradiance

Colorchecker: Great tool for radiometric and color calibration.



Patches at bottom row have reflectance that increases linearly.

e.g. JPEG



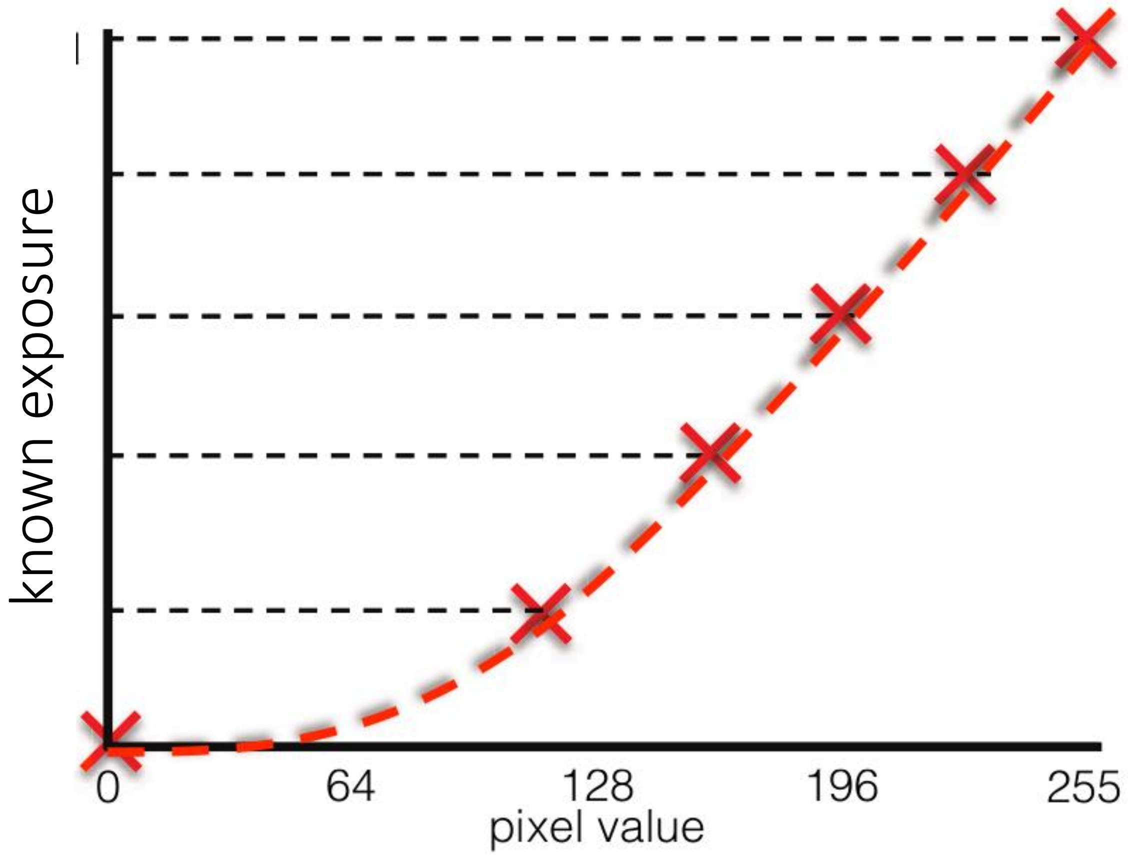
Different values correspond to patches of increasing reflected irradiance.

Same scene irradiance, varying camera exposure

Colorchecker: Great tool for white balancing and radiometric calibration.

All points on (the white part of) the target have the same reflectance.

e.g. JPEG

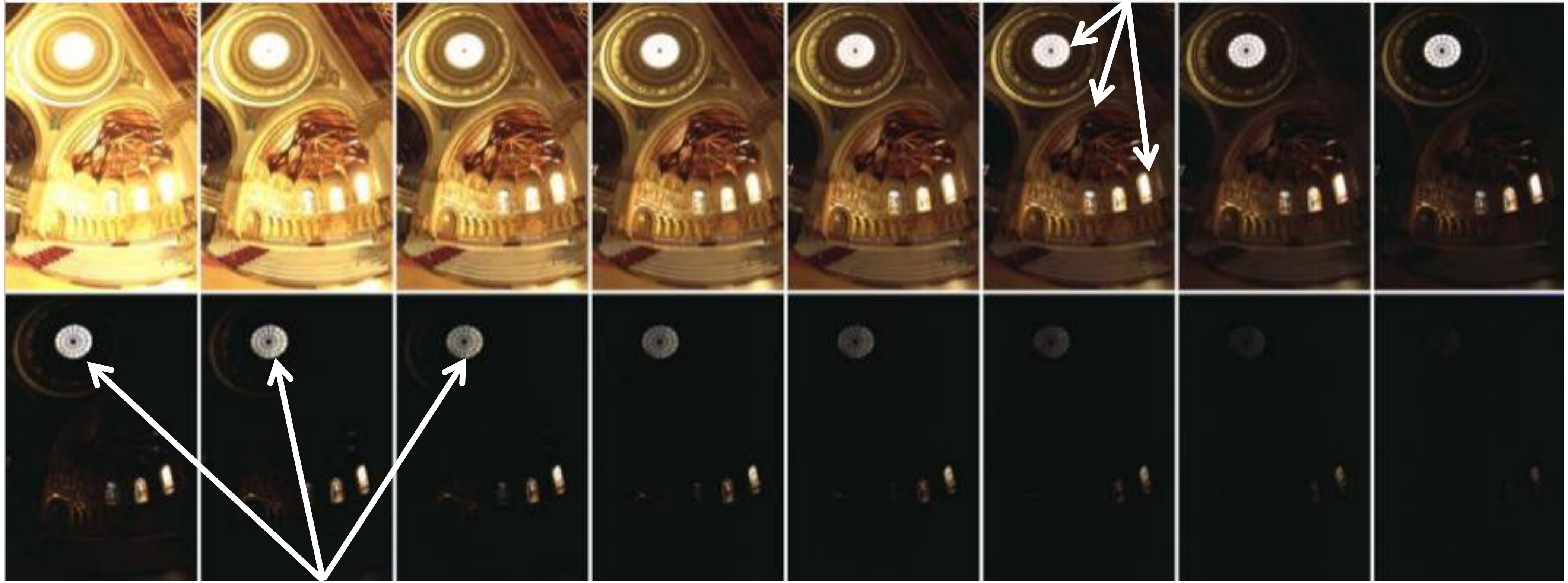


Different values correspond to images taken under increasing camera exposure.

Varying both scene irradiance and camera exposure

You can do this using the LDR exposure stack itself.

Same scene irradiance, different camera exposure

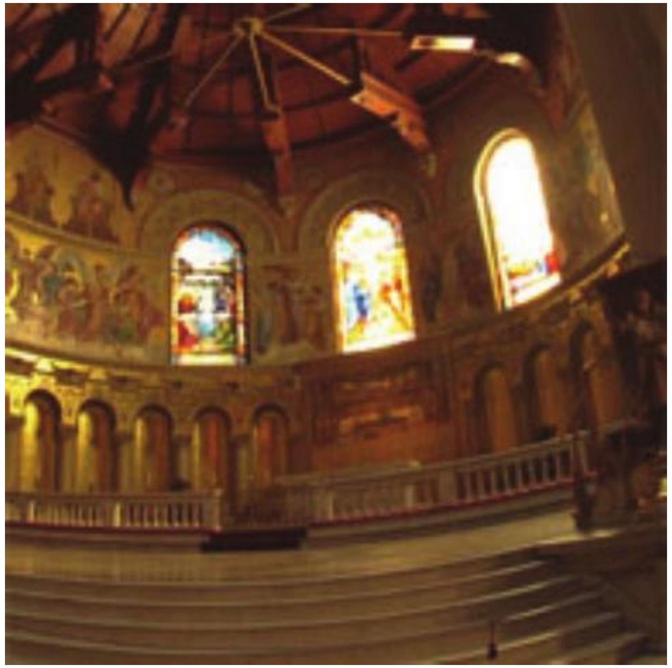


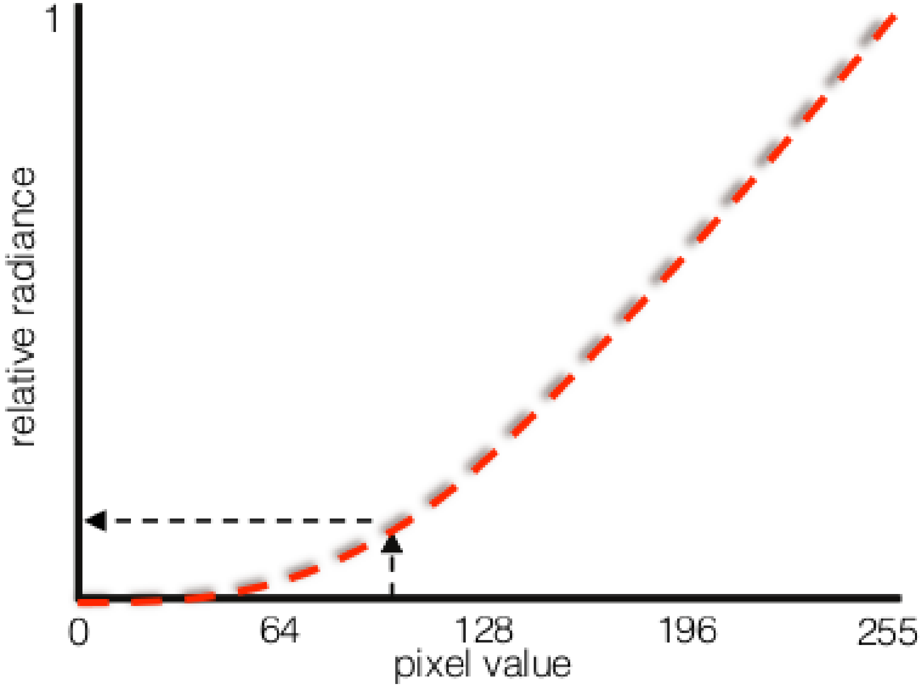
Same scene irradiance, different camera exposure

Non-linear image formation model

Real scene radiance for image pixel (x,y): $L(x, y)$

Exposure time: t_i





$$I_{linear}(x,y) = \text{clip}[\, t_i \cdot L(x,y) + \text{noise} \,]$$

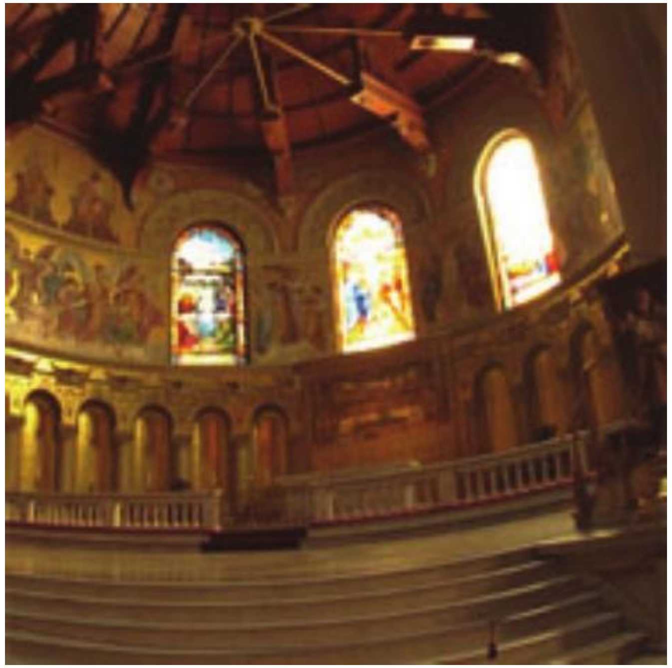
$$I_{non\text{-}linear}(x,y) = f[\, I_{linear}(x,y) \,]$$

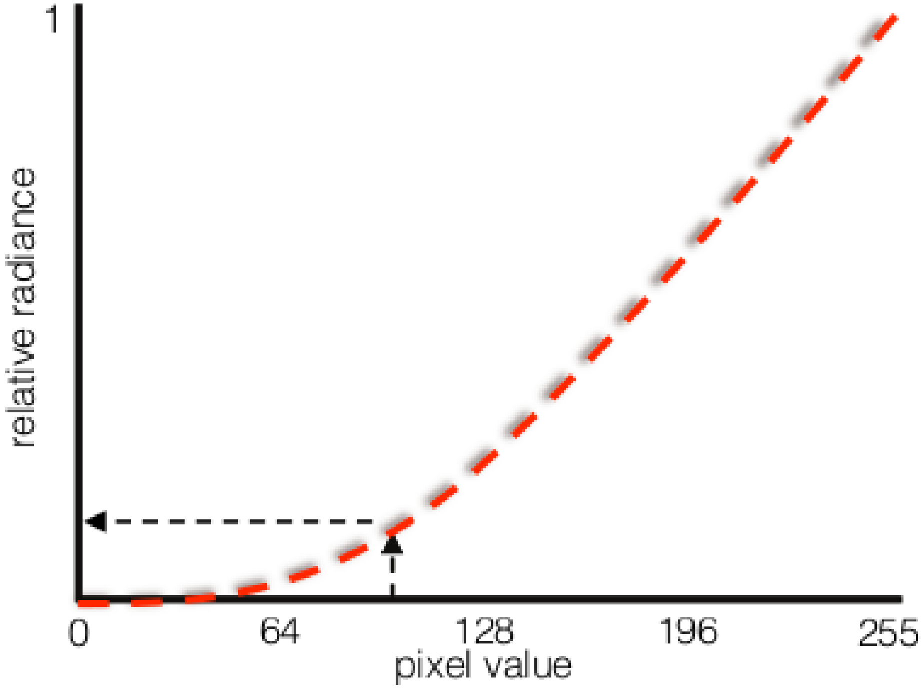
How would you merge the non-linear images into an HDR one?

Non-linear image formation model

Real scene radiance for image pixel (x,y): $L(x, y)$

Exposure time: t_i





$$I_{linear}(x,y) = \text{clip}[\, t_i \cdot L(x,y) + \text{noise} \,]$$

$$I_{non\text{-}linear}(x,y) = f[\, I_{linear}(x,y) \,] \qquad I_{est}(x,y) = f^{-1}[\, I_{non\text{-}linear}(x,y) \,]$$

Use inverse transform to estimate linear image, then proceed as before

Linearization

$I_{\text{non-linear}}(x,y) = f[\ I_{\text{linear}}(x,y)\ ]$

$I_{\text{est}}(x,y) = f^{-1}[\ I_{\text{non-linear}}(x,y)\ ]$

Merging non-linear exposure stacks

1. Calibrate response curve

2. Linearize images

For each pixel:

3. Find "valid" images ⟵ (noise) $0.05 < \text{pixel} < 0.95$ (clipping)

4. Weight valid pixel values appropriately ⟵ (pixel value) / t_i

5. Form a new pixel value as the weighted average of valid pixel values

Same steps as in the RAW case.

Merging non-linear exposure stacks

1. Calibrate response curve
2. Linearize images

For each pixel:

3. Find "valid" images ← (noise) 0.05 < pixel < 0.95 (clipping)
4. Weight valid pixel values appropriately ← (pixel value) / t_i
5. Form a new pixel value as the weighted average of valid pixel values

→ Same steps as in the RAW case.

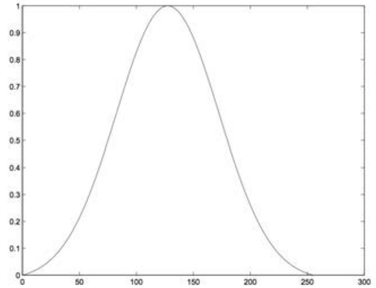
Note: many possible weighting schemes

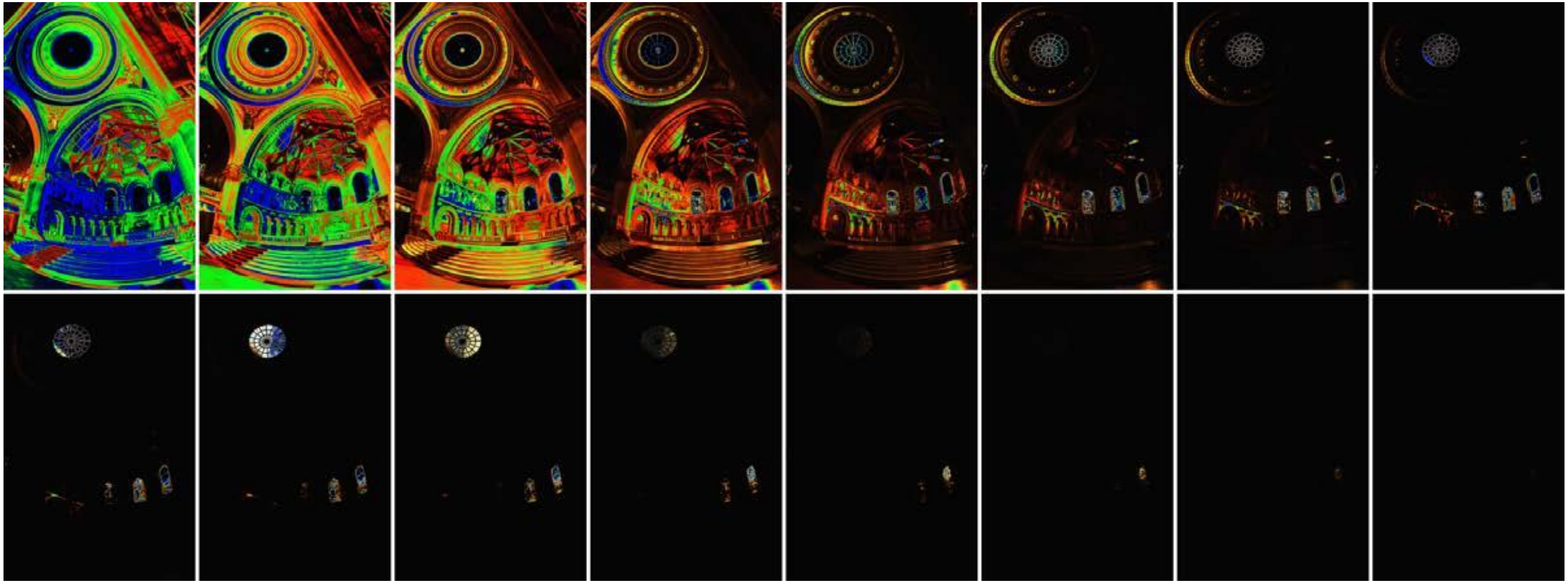
Many possible weighting schemes

“Confidence” that pixel is noisy/clipped

- We can derive optimal weights by modeling the sensor noise.

$$w_{ij} = \exp\left(-4 \frac{\left(I_{lin_{ij}} - 0.5 \right)^2}{0.5^2} \right)$$





What if I cannot measure response curve?

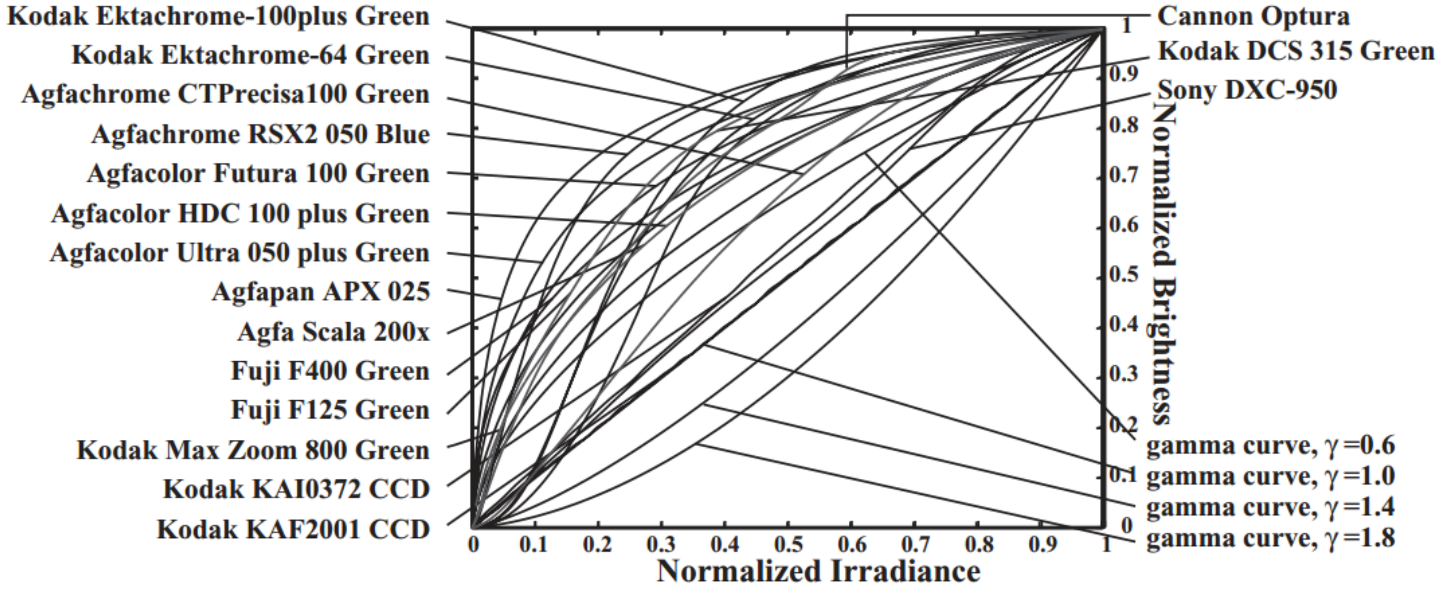
Tone reproduction curves

The exact tone reproduction 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

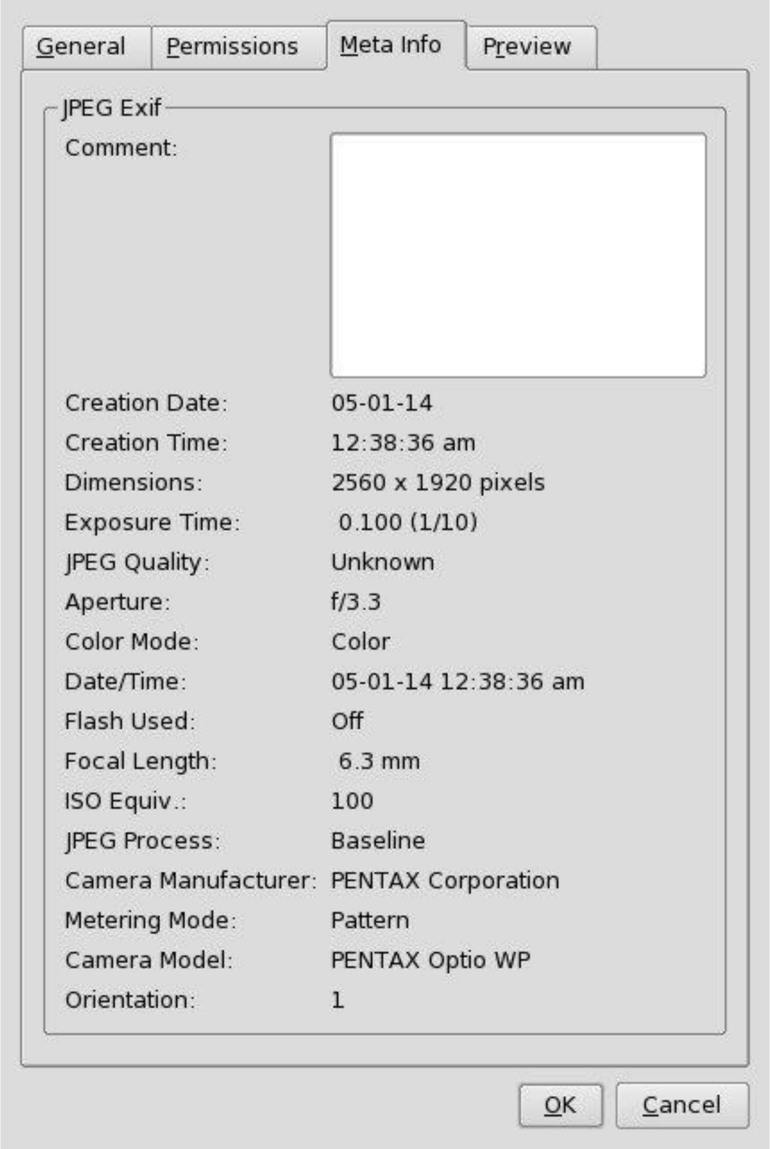


If nothing else, take the square of your image to approximately remove effect of tone reproduction curve.

You may find information in the image itself

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.



Basic HDR approach

1. Capture multiple LDR images at different exposures

2. Merge them into a single HDR image

Any problems with this approach?

Basic HDR approach

1. Capture multiple LDR images at different exposures

2. Merge them into a single HDR image

Problem: Very sensitive to movement

- Scene must be completely static

- Camera must not move

Most modern automatic HDR solutions include an alignment step before merging exposures

How do we store HDR images?

- Most standard image formats store integer 8-bit images
- Some image formats store integer 12-bit or 16-bit images
- HDR images are floating point 32-bit or 64-bit images

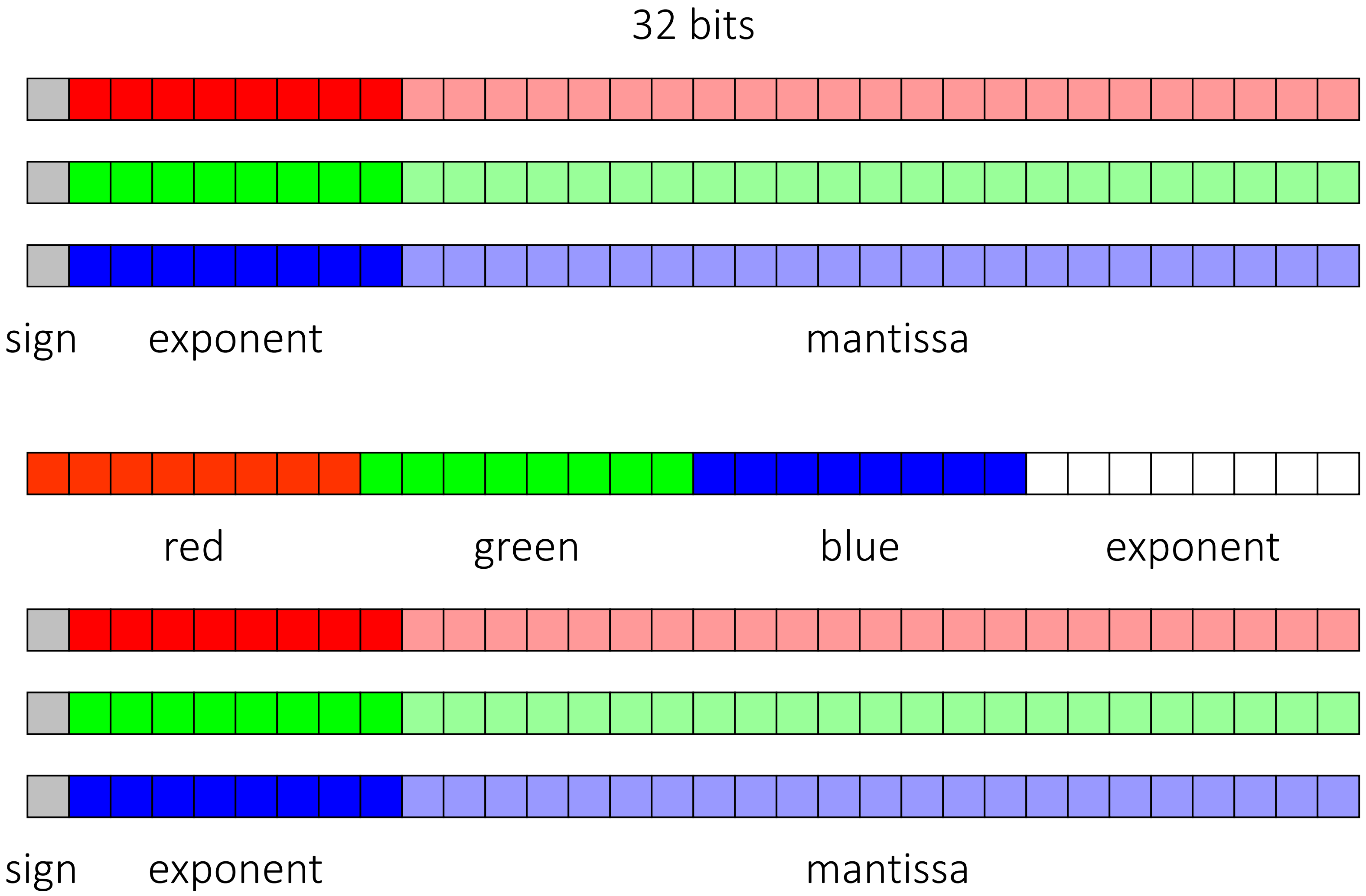
How do we store HDR images?

Use specialized image formats for HDR images

portable float map (.pfm)
- very simple to implement

Radiance format (.hdr)
- supported by Matlab

OpenEXR format (.exr)
- multiple extra features



Another type of HDR images

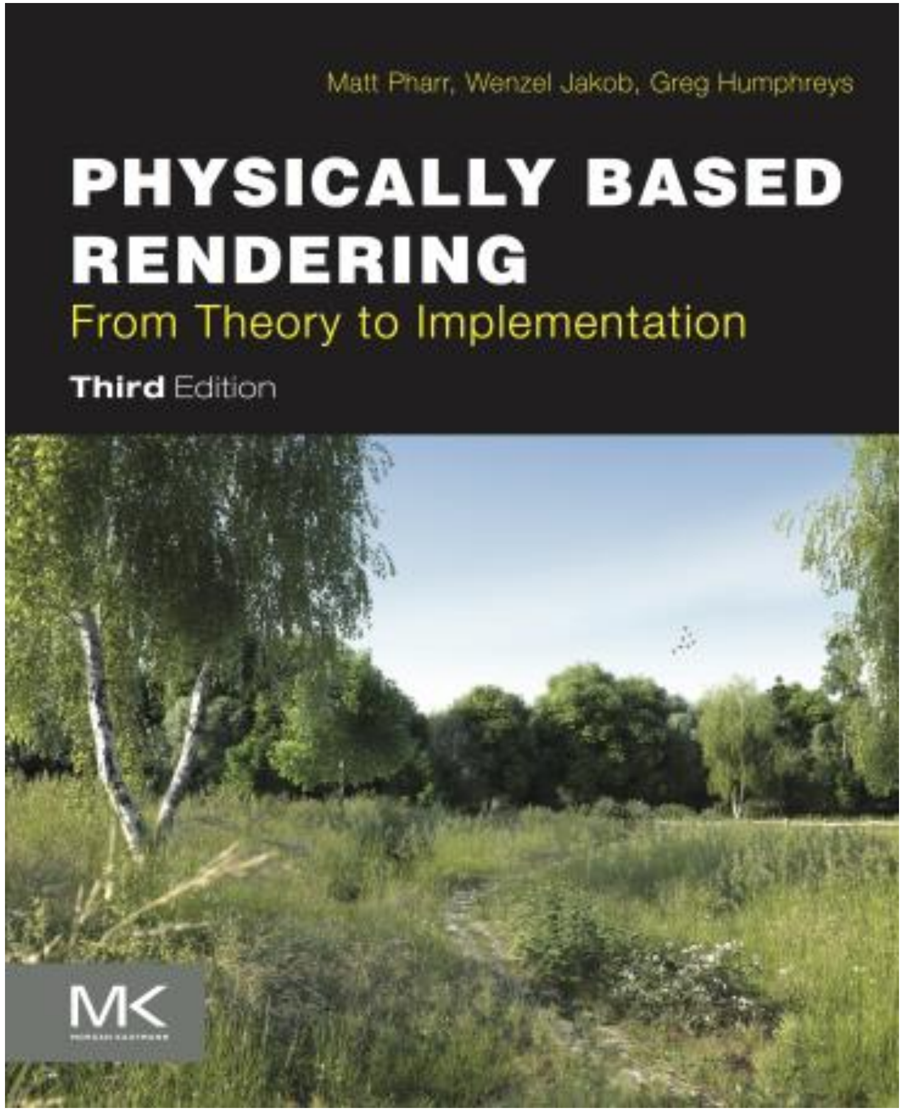
Light probes: place a chrome sphere in the scene and capture an HDR image
- Used to measure real-world illumination environments (“environment maps”)

Application: image-based relighting (later lecture)

Another way to create HDR images

Physics-based renderers simulate radiance maps (relative or absolute)

- Their outputs are very often HDR images



A note about HDR today

- Most cameras (even phone cameras) have automatic HDR modes/apps
- Popular-enough feature that phone manufacturers are actively competing about which one has the best HDR
- The technology behind some of those apps (e.g., Google's HDR+) is published in SIGGRAPH and SIGGRAPH Asia conferences

Burst photography for high dynamic range and low-light imaging on mobile cameras

Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams
Jonathan T. Barron, Florian Kainz, Jiawen Chen, Marc Levoy
Google Research

Figure 1: *A comparison of a conventional camera pipeline (left, middle) and our burst photography pipeline (right) running on the same cell-phone camera. In this low-light setting (about 0.7 lux), the conventional camera pipeline underexposes (left). Brightening the image (middle) reveals heavy spatial denoising, which results in loss of detail and an unpleasantly blotchy appearance. Fusing a burst of images increases the signal-to-noise ratio, making aggressive spatial denoising unnecessary. We encourage the reader to zoom in. While our pipeline excels in low-light and high-dynamic-range scenes (for an example of the latter see figure 10), it is computationally efficient and reliably artifact-free, so it can be deployed on a mobile camera and used as a substitute for the conventional pipeline in almost all circumstances. For readability the figure has been made uniformly brighter than the original photographs.*

Abstract

Cell phone cameras have small apertures, which limits the number of photons they can gather, leading to noisy images in low light. They also have small sensor pixels, which limits the number of electrons each pixel can store, leading to limited dynamic range. We describe a computational photography pipeline that captures, aligns, and merges a burst of frames to reduce noise and increase dynamic range. Our system has several key features that help make it robust and efficient. First, we do not use bracketed exposures. Instead, we capture frames of constant exposure, which makes alignment more robust, and we set this exposure low enough to avoid blowing out highlights. The resulting merged image has clean shadows and high bit depth, allowing us to apply standard HDR tone mapping methods. Second, we begin from Bayer raw frames rather than the demosaicked RGB (or YUV) frames produced by hardware Image Signal Processors (ISPs) common on mobile platforms. This gives us more bits per pixel and allows us to circumvent the ISP's unwanted tone mapping and spatial denoising. Third, we use a novel FFT-based alignment algorithm and a hybrid 2D/3D Wiener filter to denoise and merge the frames in a burst. Our implementation is built atop Android's Camera2 API, which provides per-frame camera control and access to raw imagery, and is written in the Halide domain-specific language (DSL). It runs in 4 seconds on device (for a 12 Mpix image), requires no user intervention, and ships on several mass-produced cell phones.

Keywords: computational photography, high dynamic range

Concepts: •**Computing methodologies** → **Computational photography; Image processing;**

1 Introduction

The main technical impediment to better photographs is lack of light. In indoor or night-time shots, the scene as a whole may provide insufficient light. The standard solution is either to apply analog or digital gain, which amplifies noise, or to lengthen exposure time, which causes motion blur due to camera shake or subject motion. Surprisingly, daytime shots with high dynamic range may also suffer from lack of light. In particular, if exposure time is reduced to avoid

SA '16 Technical Papers, December 05 - 08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2980254

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, and we often need to account for it when doing computer vision.

References

Basic reading:

- Szeliski textbook, Section 2.3
- Michael Brown, “Understanding the In-Camera Image Processing Pipeline for Computer Vision,” CVPR 2016,
  very detailed discussion of issues relating to color photography and management, slides available at:
  http://www.comp.nus.edu.sg/~brown/CVPR2016_Brown.html
- Nine Degrees Below, https://ninedegreesbelow.com/
  amazing resource for color photography, reproduction, and management.