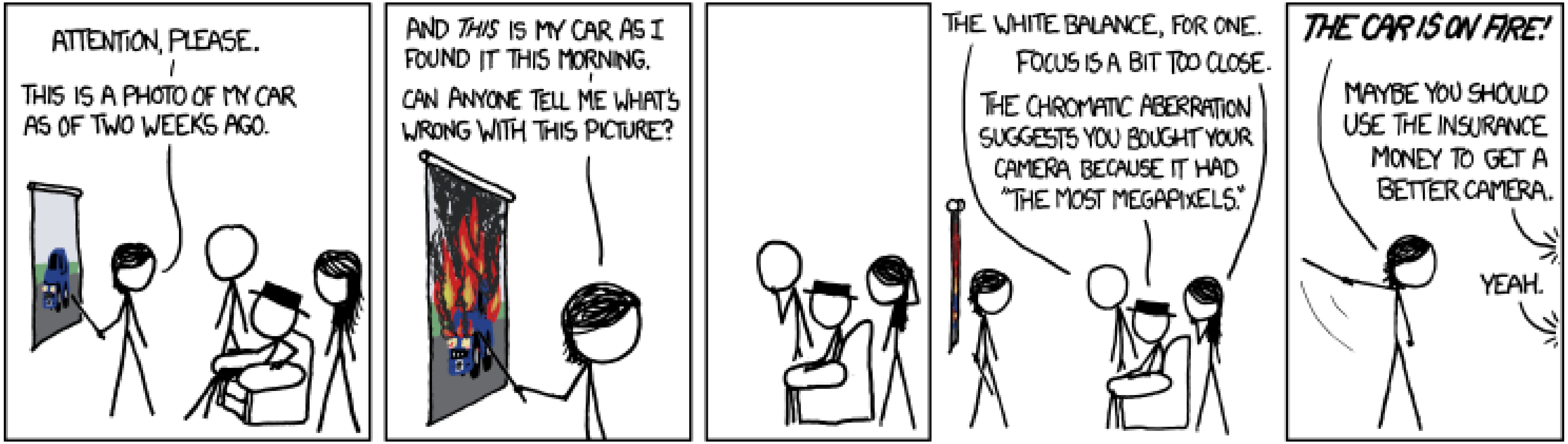
Digital photography


ATTENTION, PLEASE.
THIS IS A PHOTO OF MY CAR AS OF TWO WEEKS AGO.
AND *THIS* IS MY CAR AS I FOUND IT THIS MORNING.
CAN ANYONE TELL ME WHAT'S WRONG WITH THIS PICTURE?
THE WHITE BALANCE, FOR ONE.
FOCUS IS A BIT TOO CLOSE.
THE CHROMATIC ABERRATION SUGGESTS YOU BOUGHT YOUR CAMERA BECAUSE IT HAD "THE MOST MEGAPIXELS."
THE CAR IS ON FIRE!
MAYBE YOU SHOULD USE THE INSURANCE MONEY TO GET A BETTER CAMERA.
YEAH.

15-463, 15-663, 15-862
Computational Photography
Fall 2020, Lecture 2

http://graphics.cs.cmu.edu/courses/15-463

Course announcements

- No lecture on Monday (Labor day).
- Homework 1 will be posted on Friday, will be due September 17th at 23:59.
- If you are on waitlist, let me know.
- Office hours *for this week only* (will finalize starting next week based on survey results):
  Yannis – Friday 4-6 pm.

Course announcements

- Is there anyone not on Piazza?

  https://piazza.com/class/ksm9uc16vsg4bf

- Is there anyone not on Canvas?

  https://canvas.cmu.edu/courses/25153

Please take the start-of-semester survey and sign up for a camera before the weekend!

Survey link:

https://docs.google.com/forms/d/e/1FAIpQLSegKX5Wa9LJYxltC0D7lekOiOsEUAJd8kw4f1cVC2L6Rj1rOA/viewform

Camera sign up:

https://docs.google.com/spreadsheets/d/1BPx4YgTcm7SqPsPyppuIBniFXygAiJDnWBsQCzGKvKY/edit#gid=0

Both links available on Piazza.
We will use the survey results to finalize all logistics over the weekend.

Overview of today's lecture

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

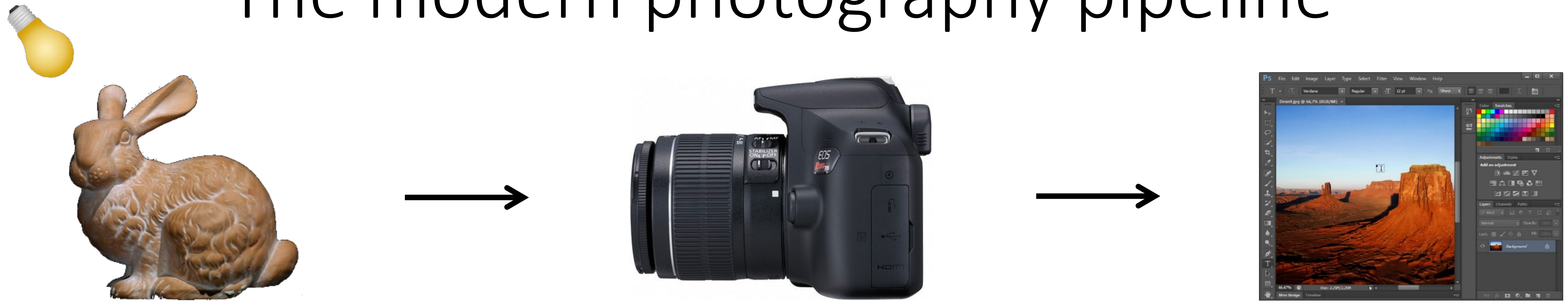
Take-home message: The values of pixels in a photograph and the values output by your camera's sensor are two very different things.

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).
- Marc Levoy (Stanford CS 178, Spring 2014).

The modern photography pipeline

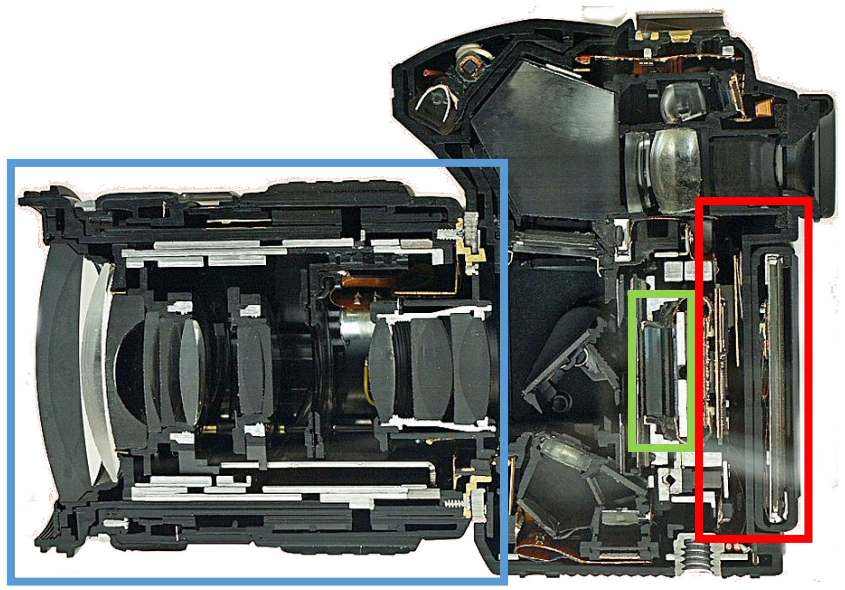


The modern photography pipeline

→

→

post-capture processing
(lectures 5-10)



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

(lectures 2-3, 11-20) (today, lecture 23) (today)

Imaging sensor primer

Imaging sensors

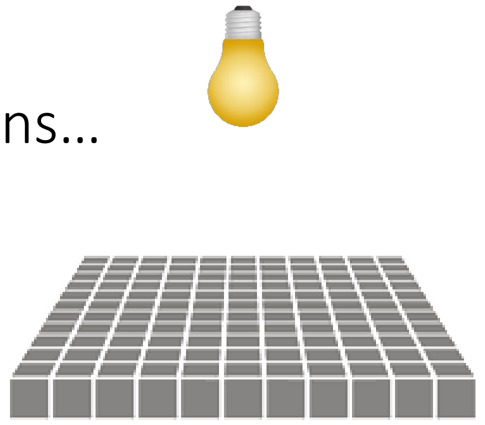
- Very high-level overview of digital imaging sensors.
- We could spend an entire course covering imaging sensors.
- Lecture 23 will cover sensors and noise issues in more detail.

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

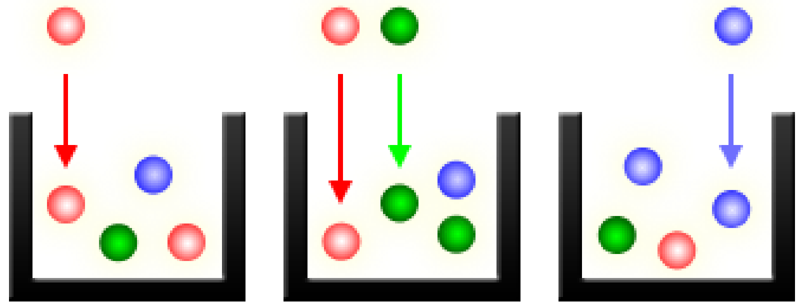
What does an imaging sensor do?

When the camera shutter opens…

… exposure begins…



array of photon buckets

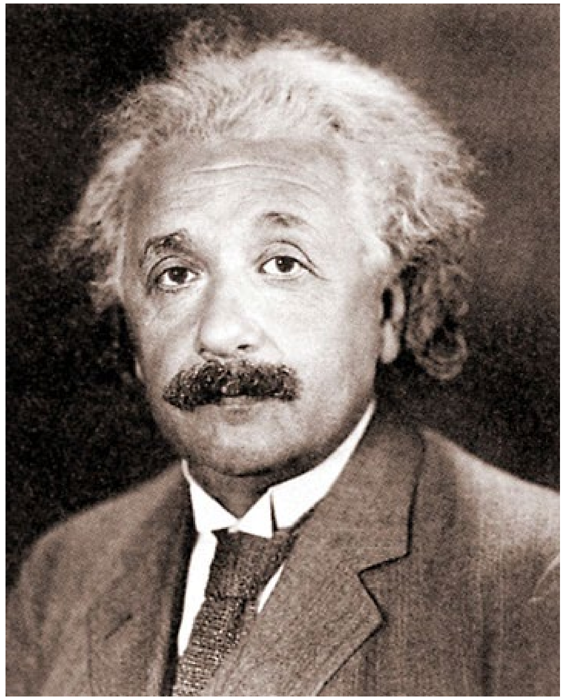


close-up view of photon buckets

… photon buckets begin to store photons...

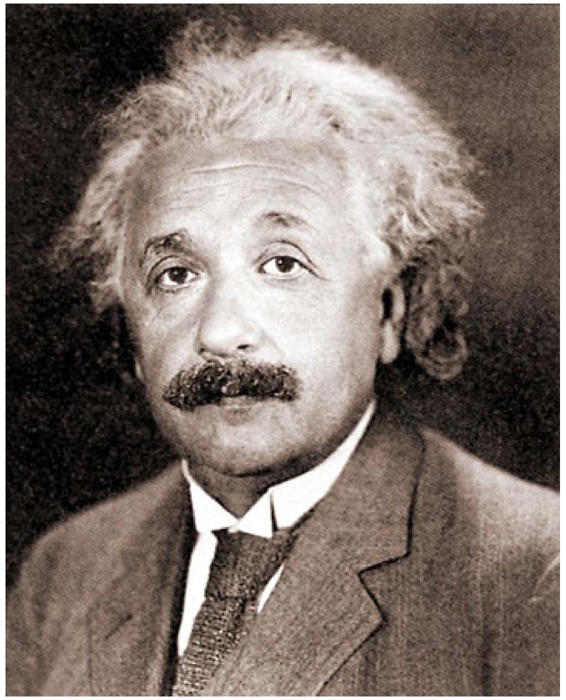
… until the camera shutter closes. Then, they convert stored photons to intensity values.

Nobel Prize in Physics



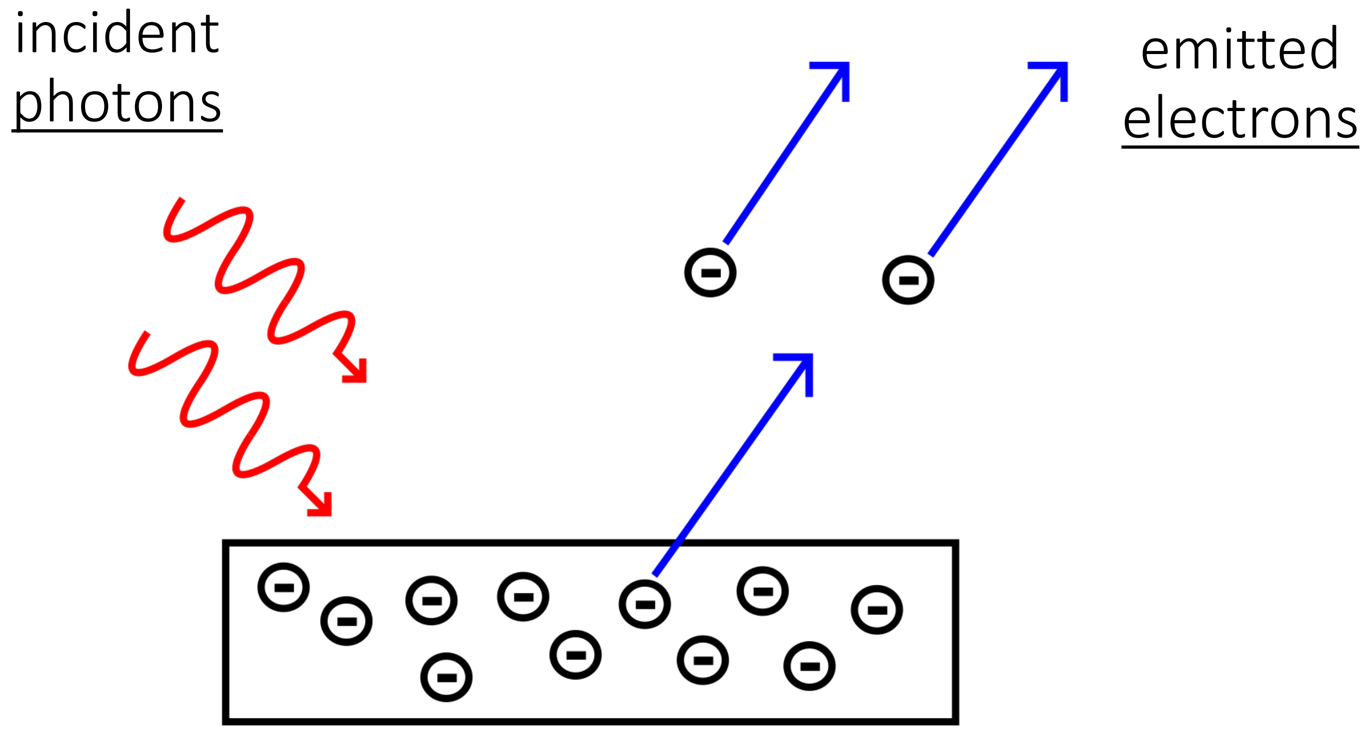
Who is this?

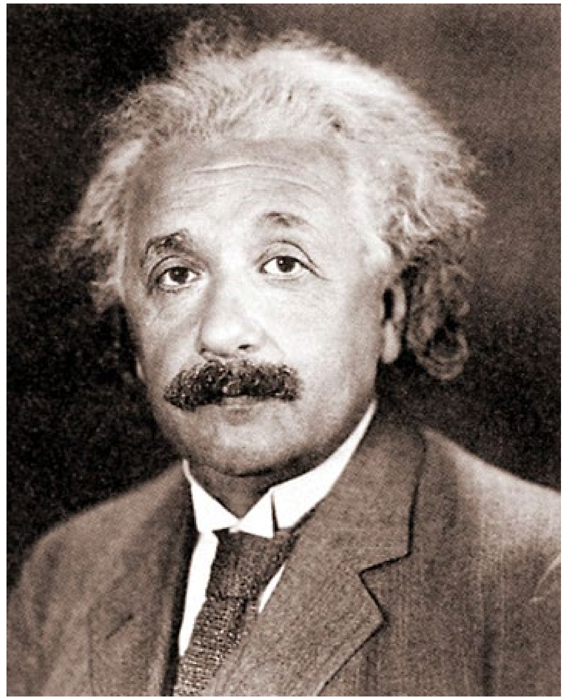
Nobel Prize in Physics



What is he known for?

Photoelectric effect

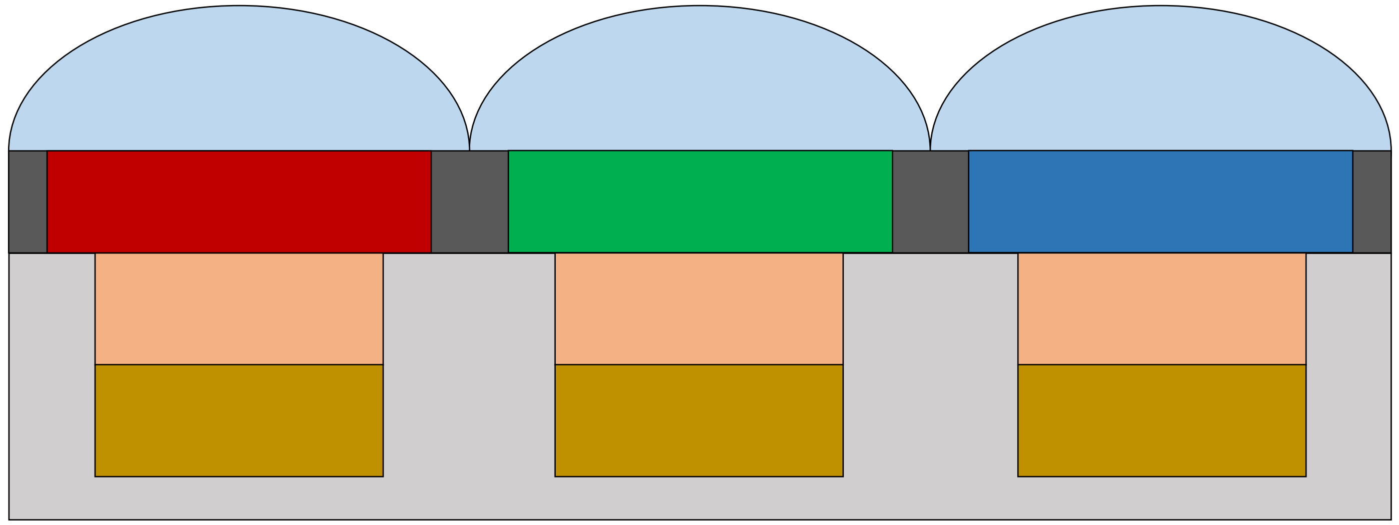


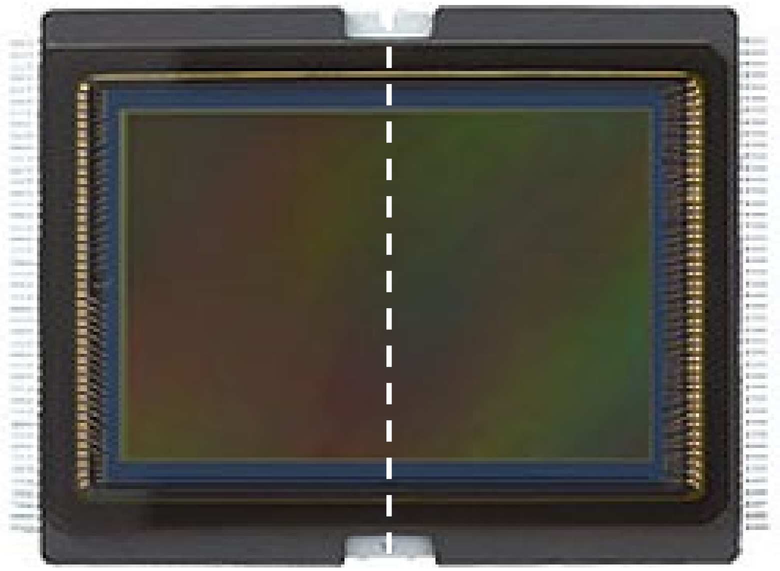


Albert Einstein

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

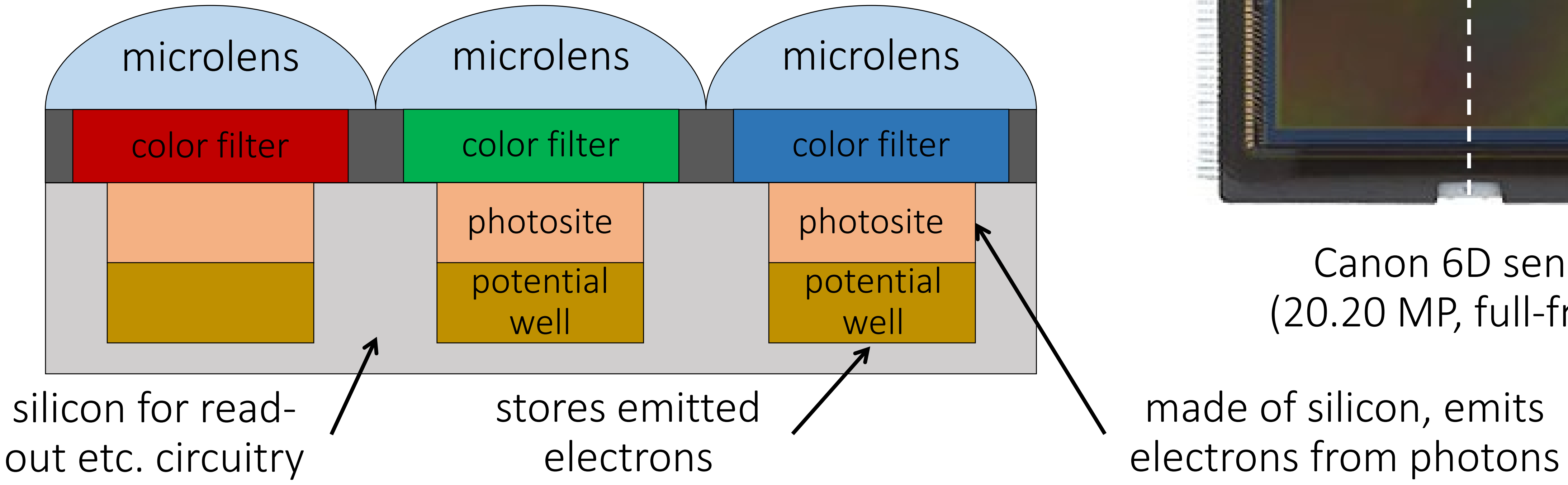
Basic imaging sensor design

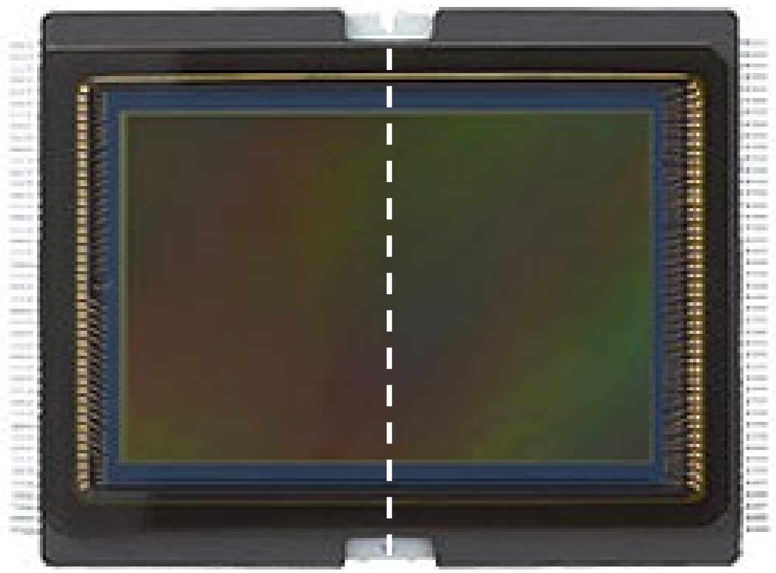




Canon 6D sensor
(20.20 MP, full-frame)

Basic imaging sensor design



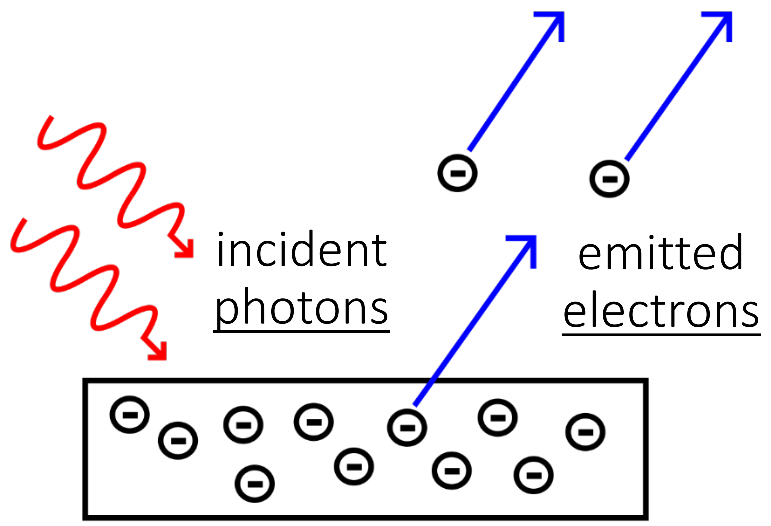

Canon 6D sensor
(20.20 MP, full-frame)

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?

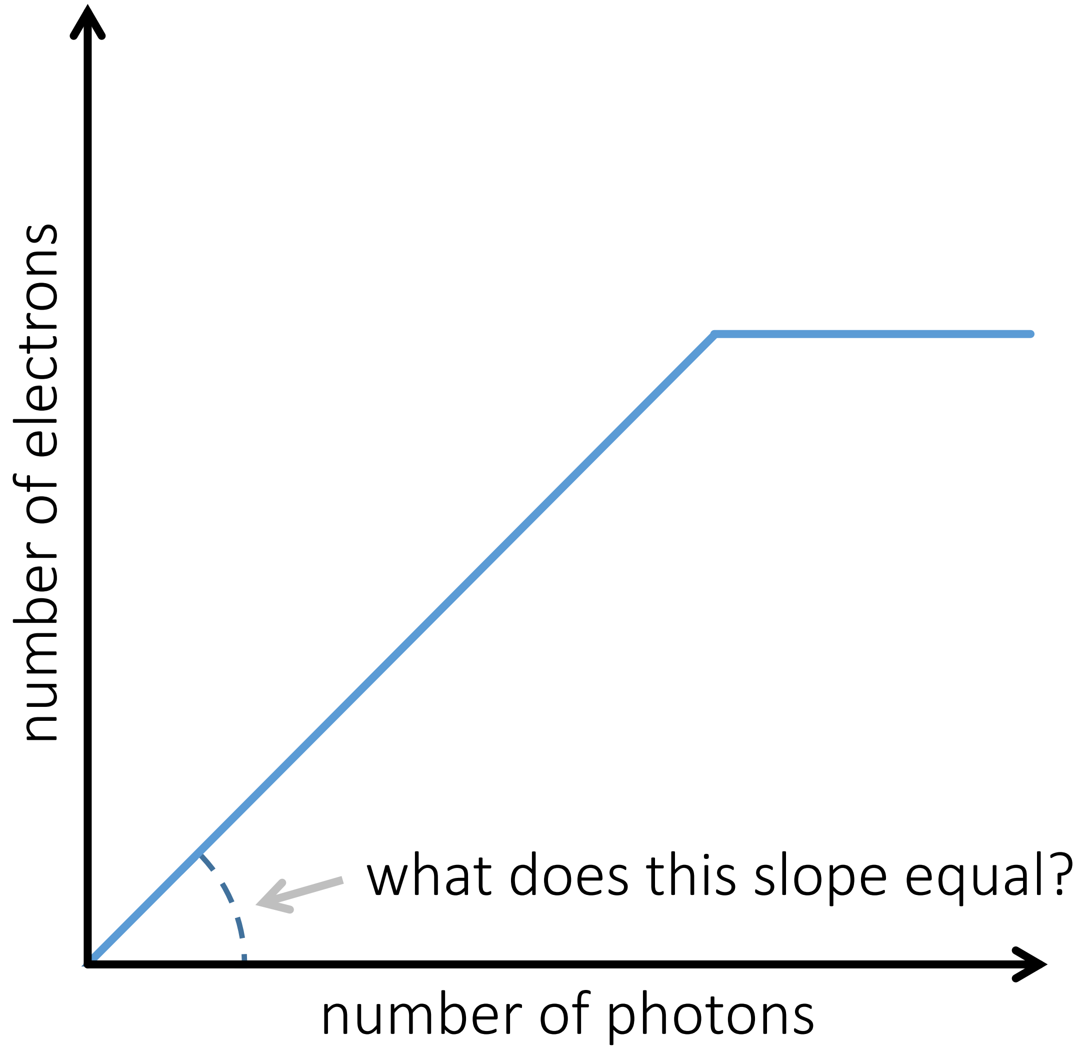
$$\boxed{QE = \frac{\#\ \text{electrons}}{\#\ \text{photons}}}$$



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

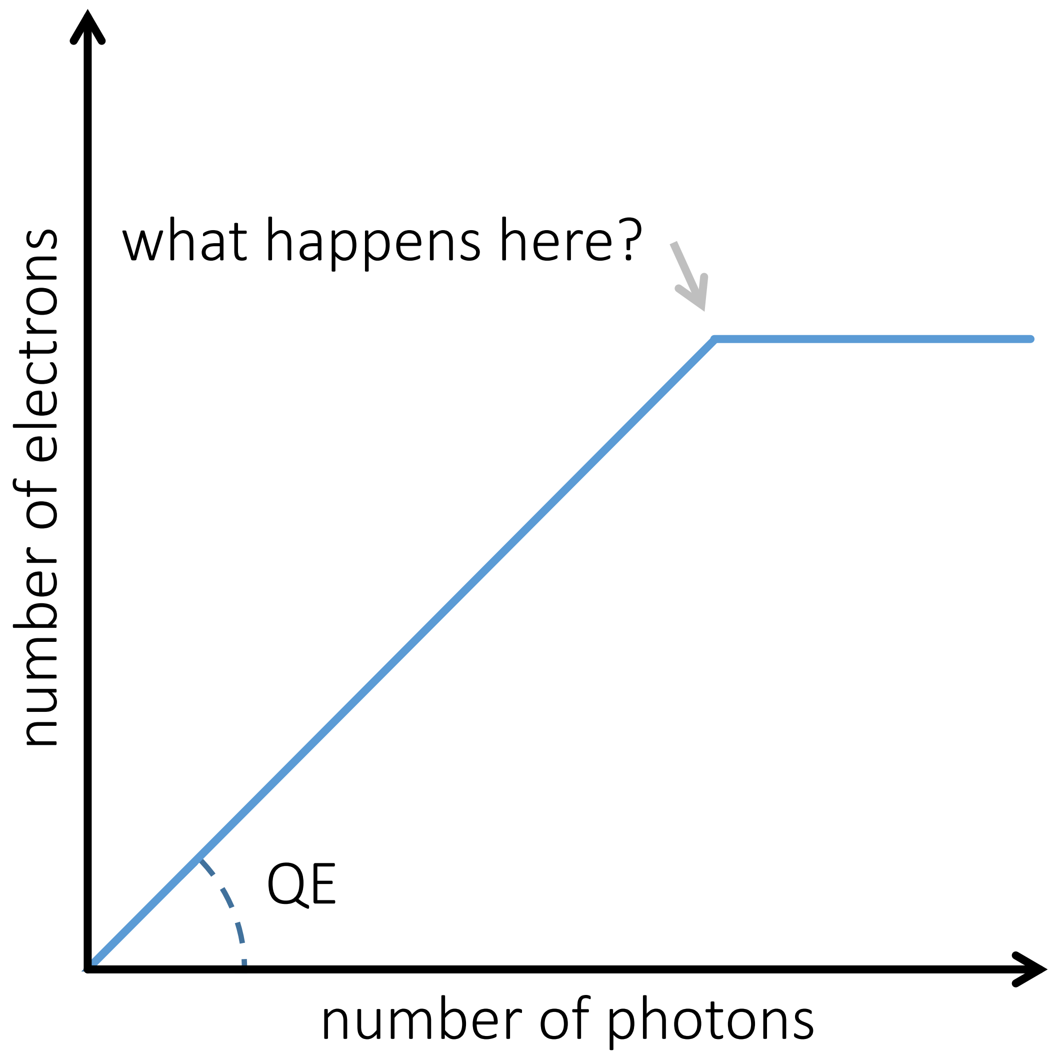
Photosite response

The photosite response is mostly *linear*



Photosite response

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

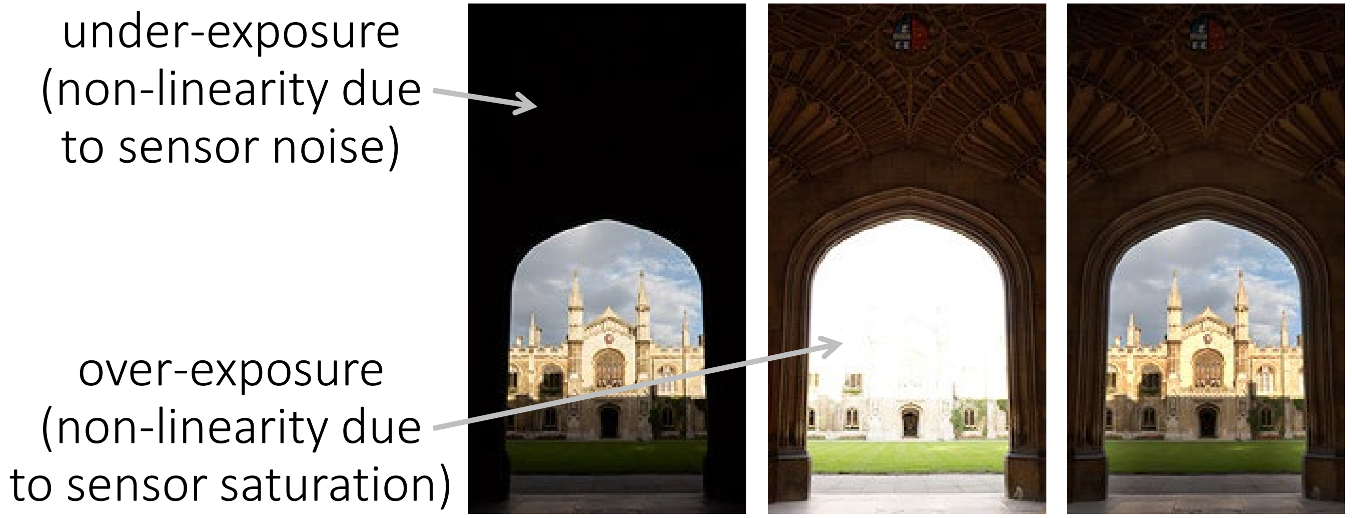


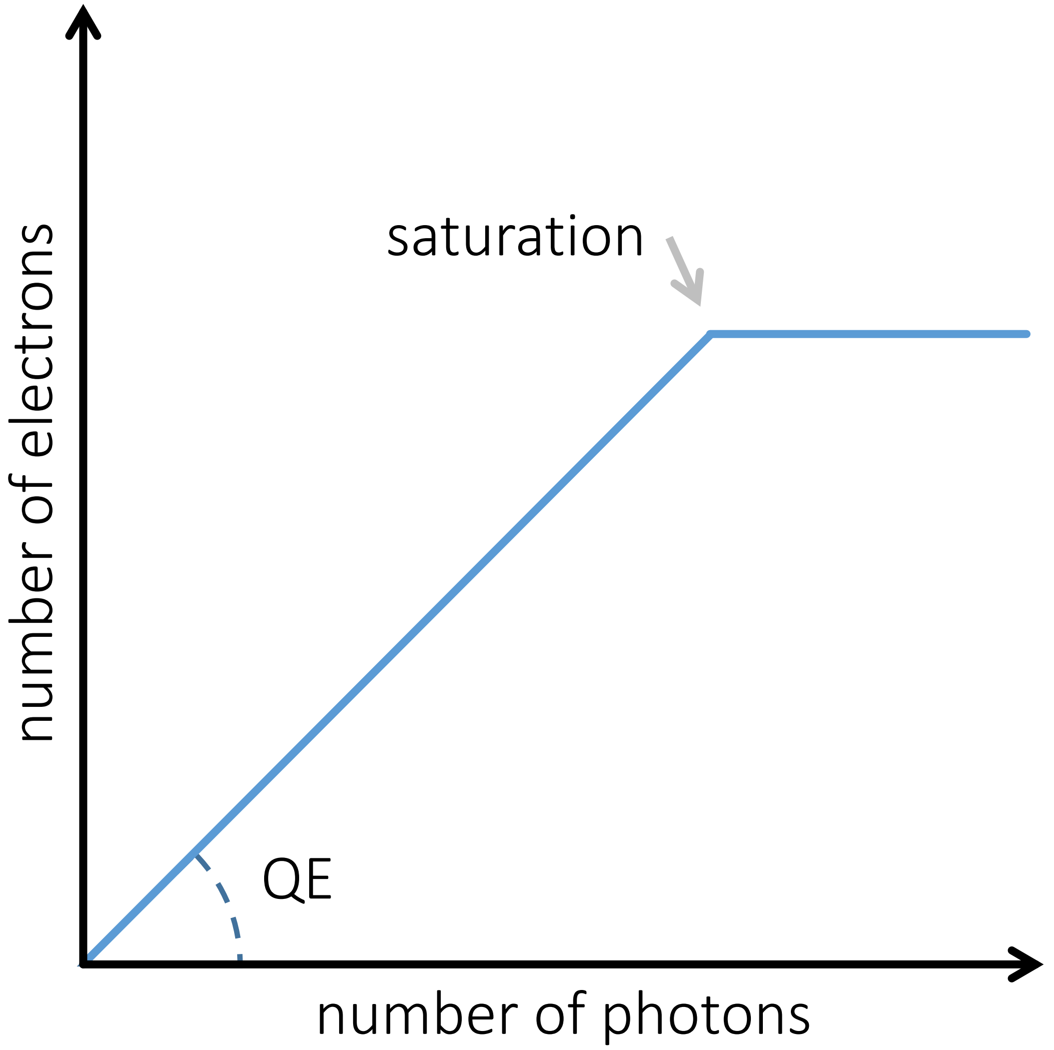
Photosite response

The photosite response is <u>mostly</u> 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).

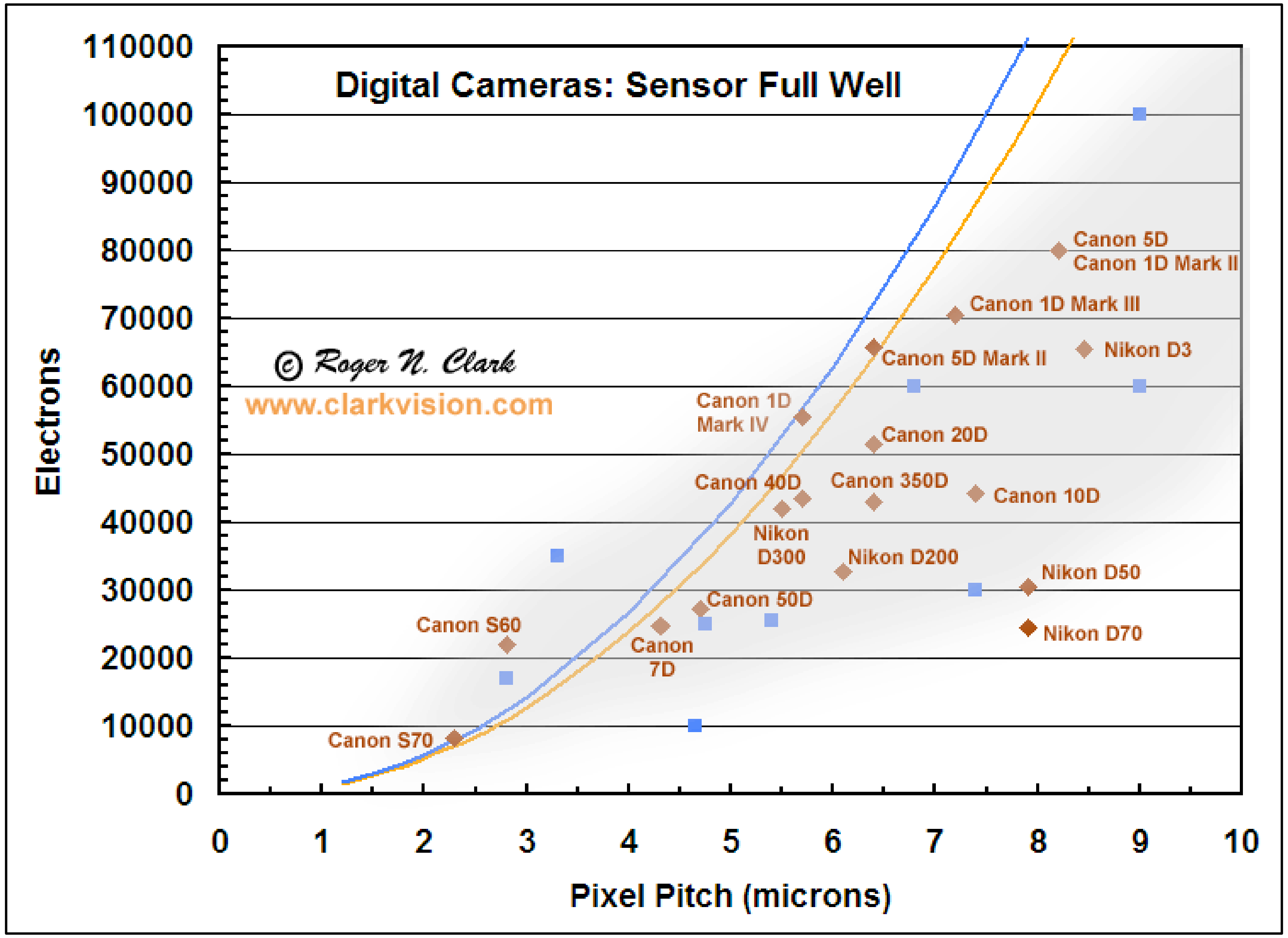

under-exposure (non-linearity due to sensor noise)
over-exposure (non-linearity due to sensor saturation)



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

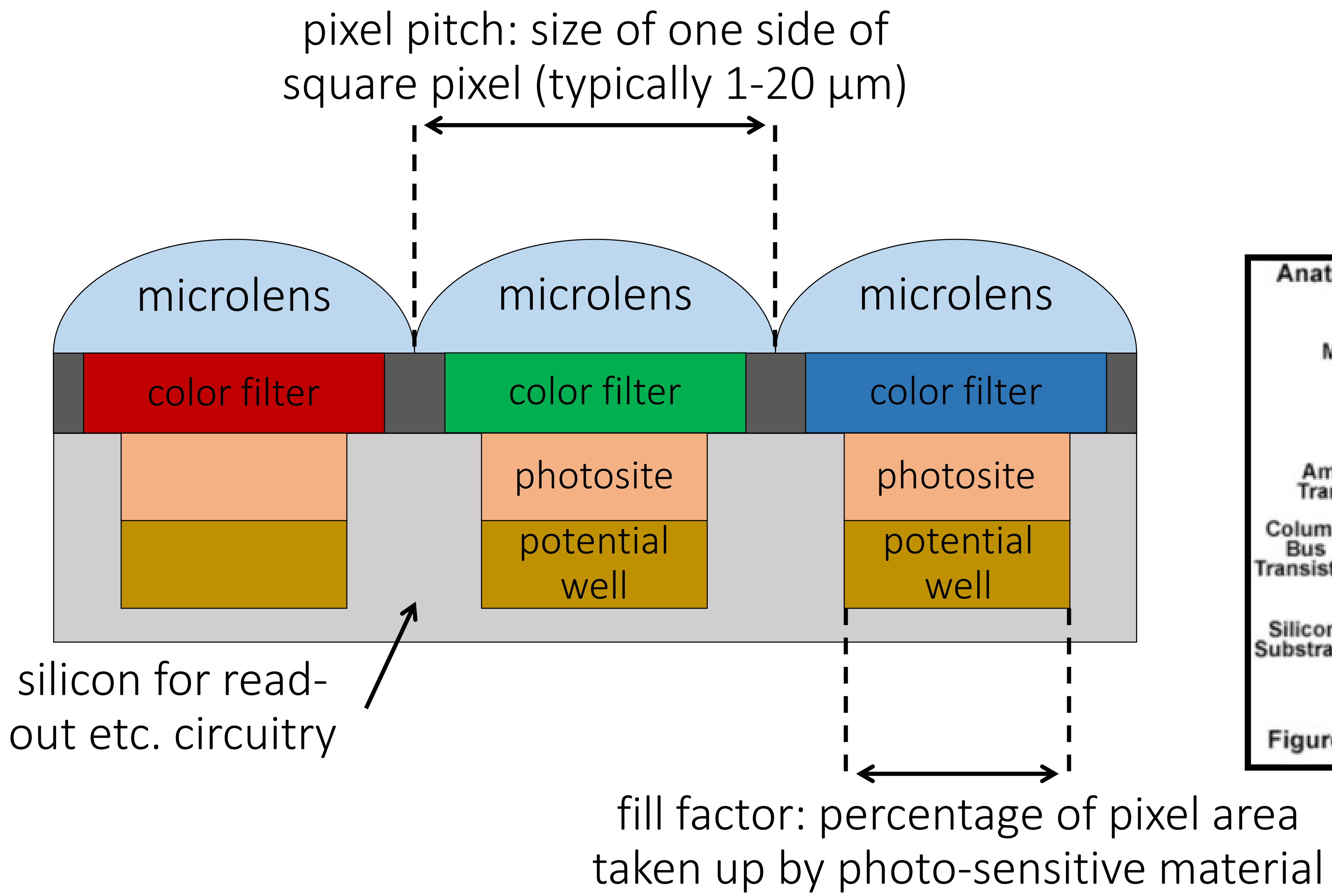
Photosite full-well capacity

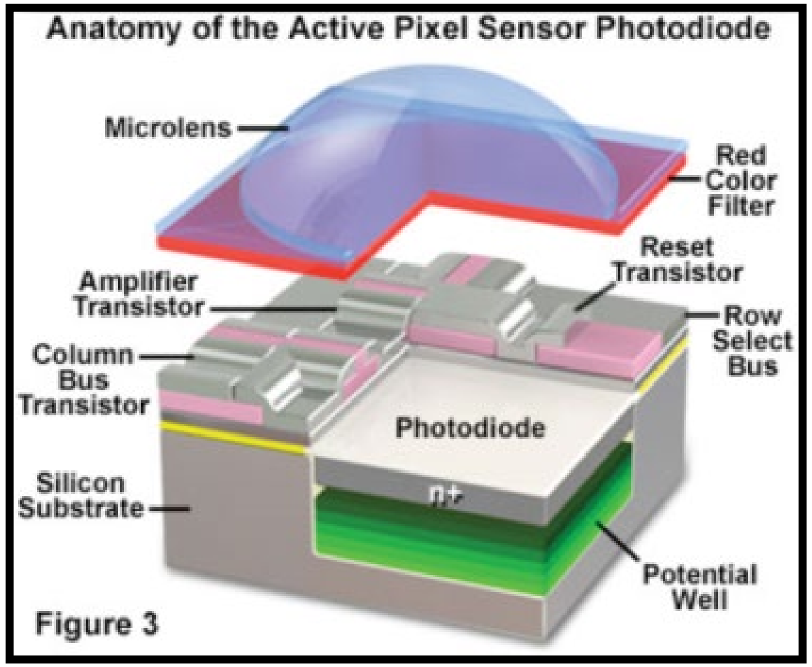
How many electrons can the photosite store before saturation?



- Another important optical performance metric of imaging sensors.

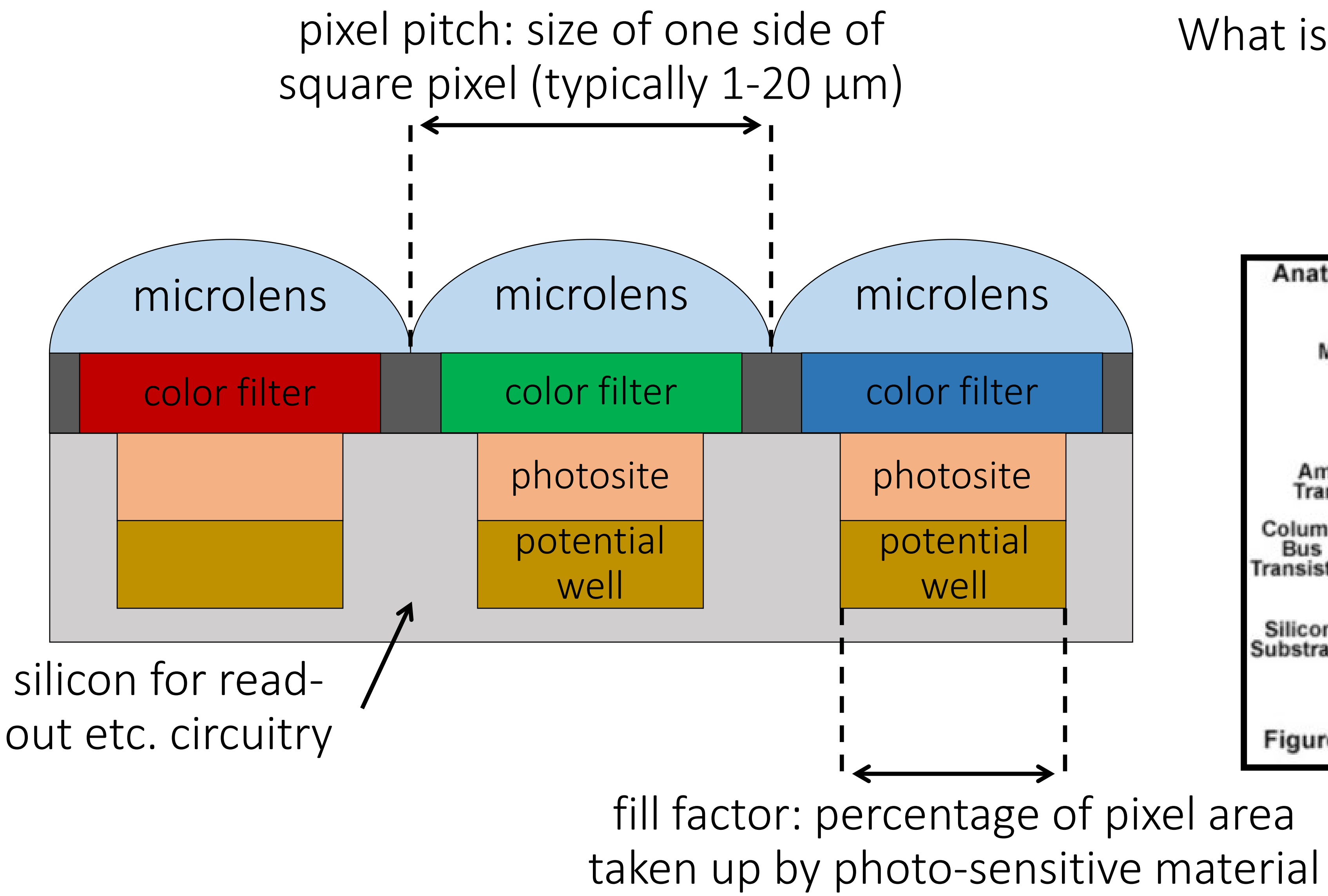
Pixel pitch and fill factor

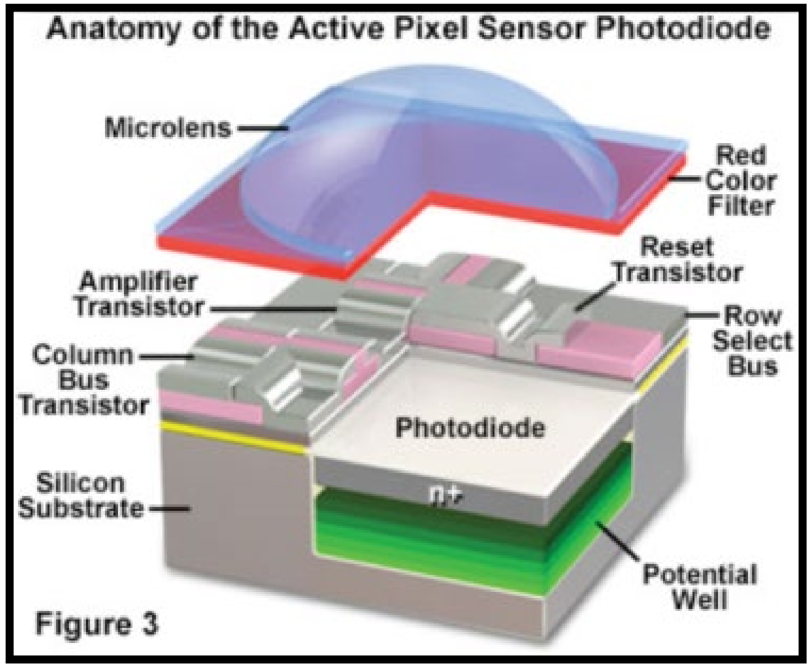

pixel pitch: size of one side of square pixel (typically 1-20 μm)
microlens
microlens
microlens
color filter
color filter
color filter
photosite
photosite
potential well
potential well
silicon for read-out etc. circuitry
fill factor: percentage of pixel area taken up by photo-sensitive material



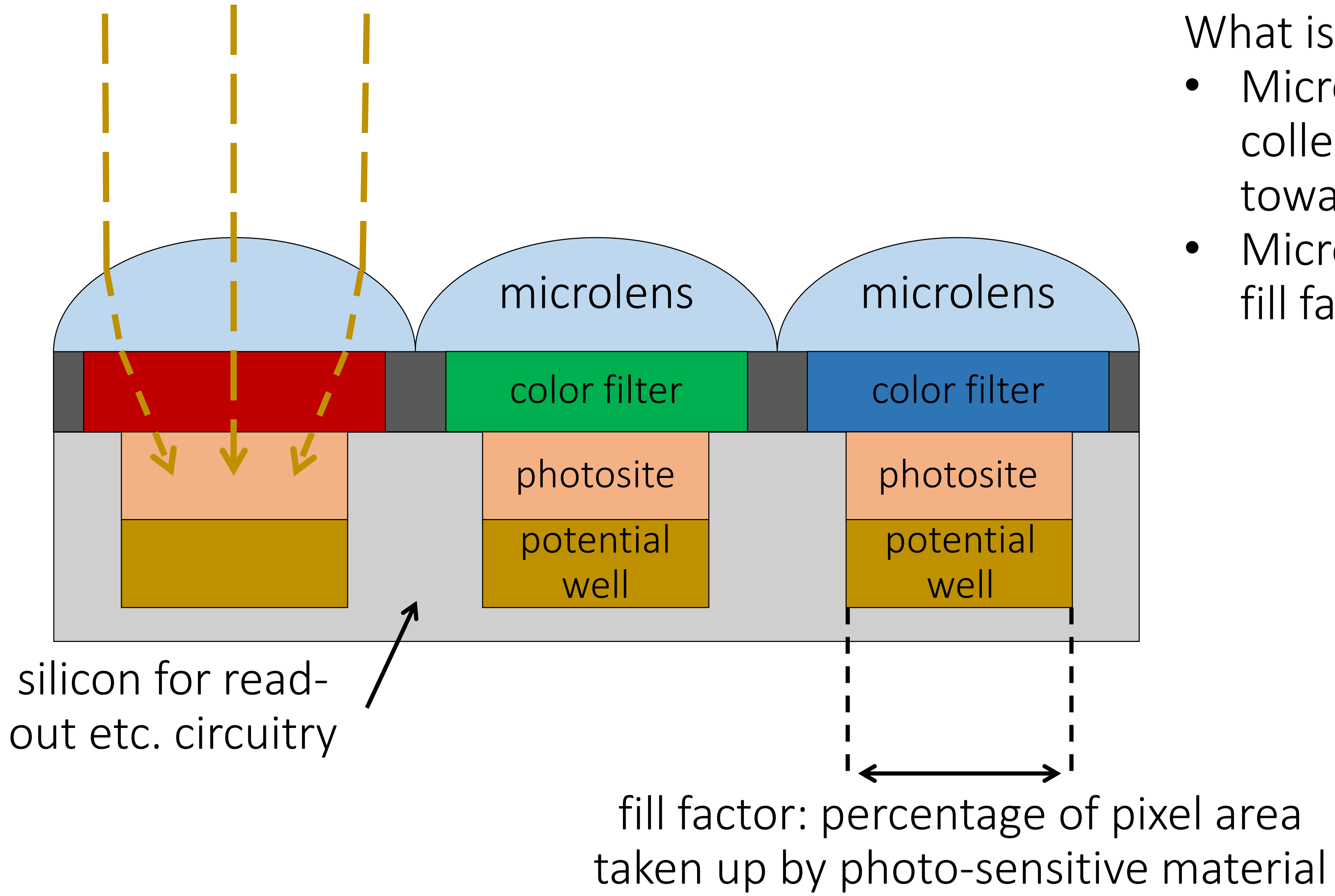
Microlenses (also called lenslets)

What is the role of the microlenses?


pixel pitch: size of one side of square pixel (typically 1-20 μm)
microlens
microlens
microlens
color filter
color filter
color filter
photosite
photosite
potential well
potential well
silicon for read-out etc. circuitry
fill factor: percentage of pixel area taken up by photo-sensitive material



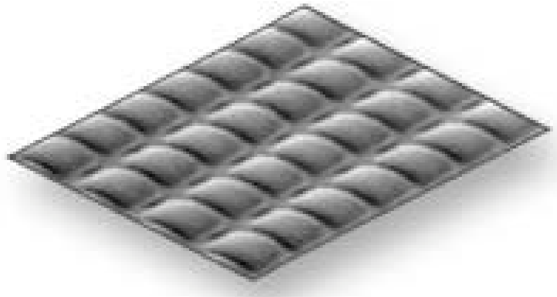
Microlenses (also called lenslets)


microlens
microlens
color filter
color filter
photosite
photosite
potential well
potential well
silicon for read-out etc. circuitry
fill factor: percentage of pixel area taken up by photo-sensitive material

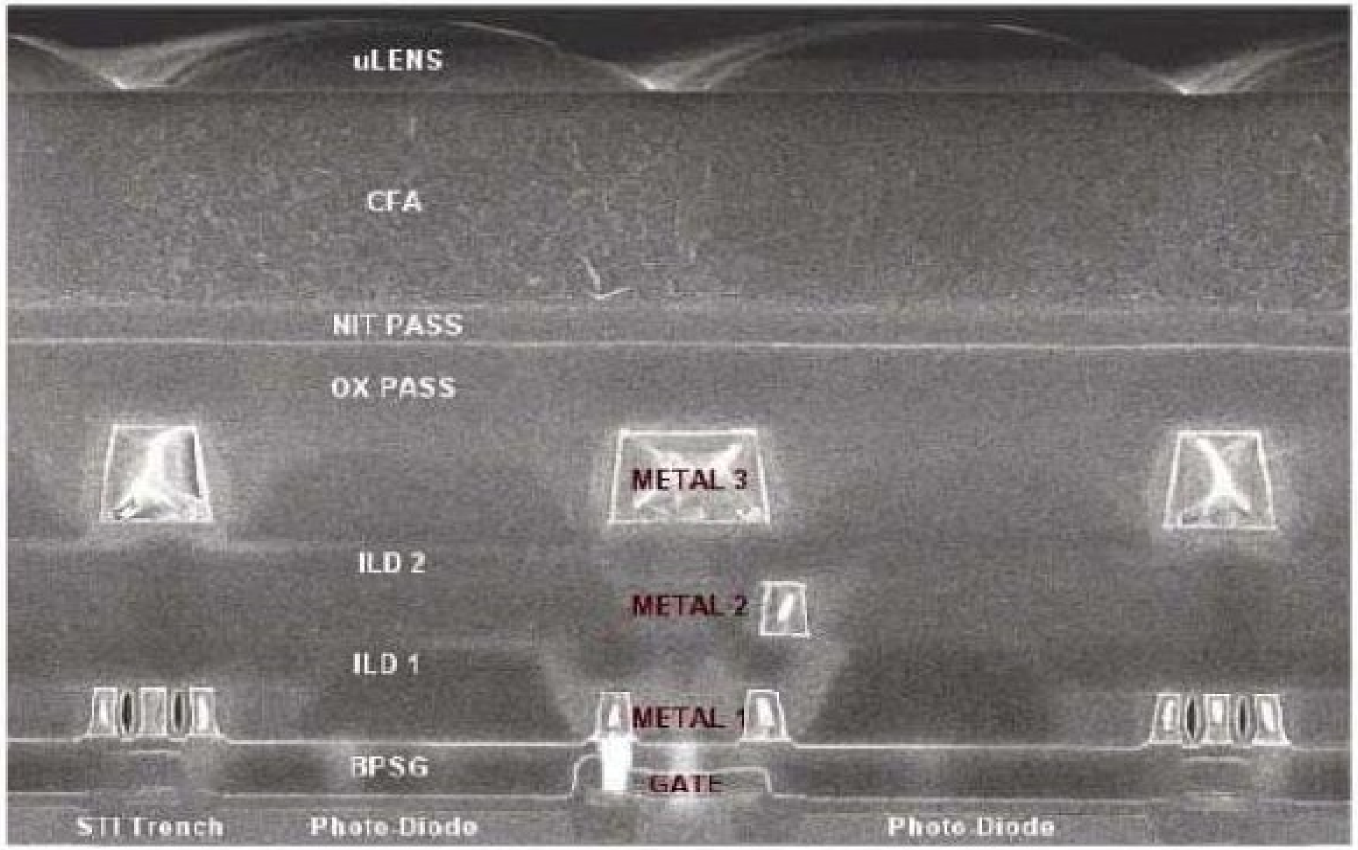
What is the role of the microlenses?

- Microlenses help photosite collect more light by bending rays towards photosensitive pixel area.
- Microlenses increase the *effective* fill factor.

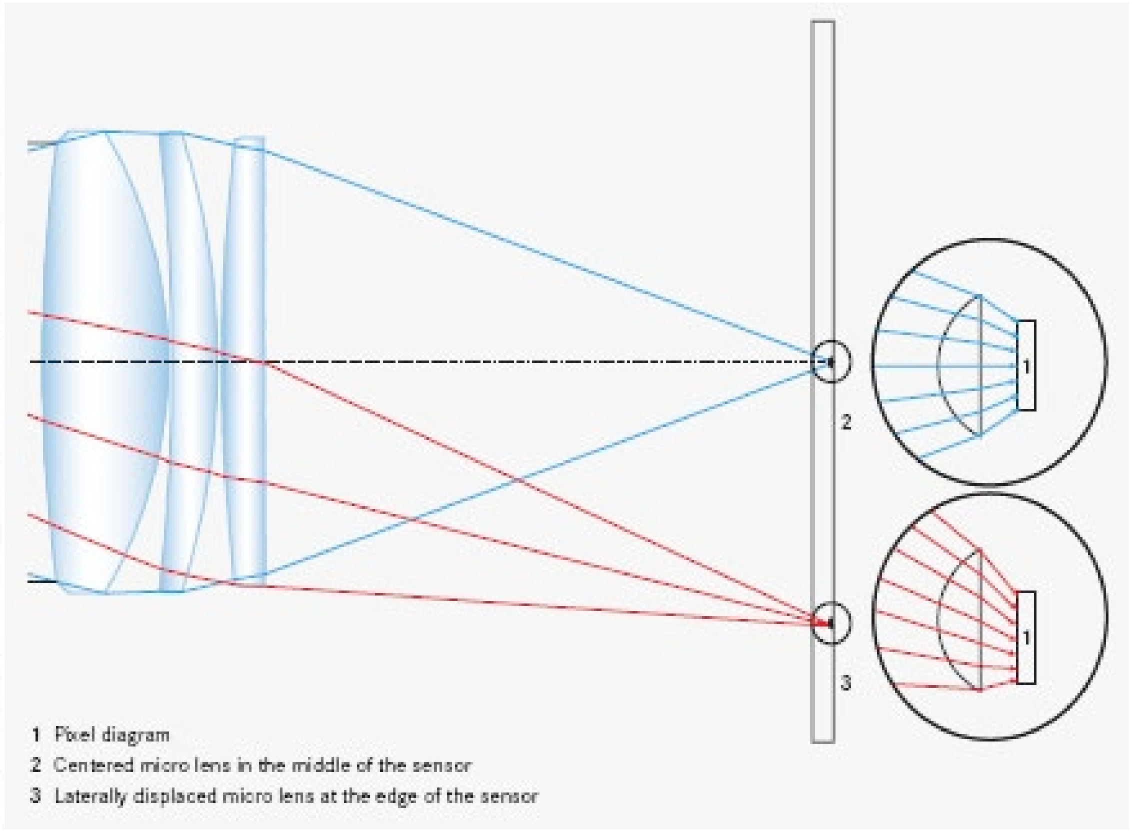
Microlenses



oblique view of microlens array

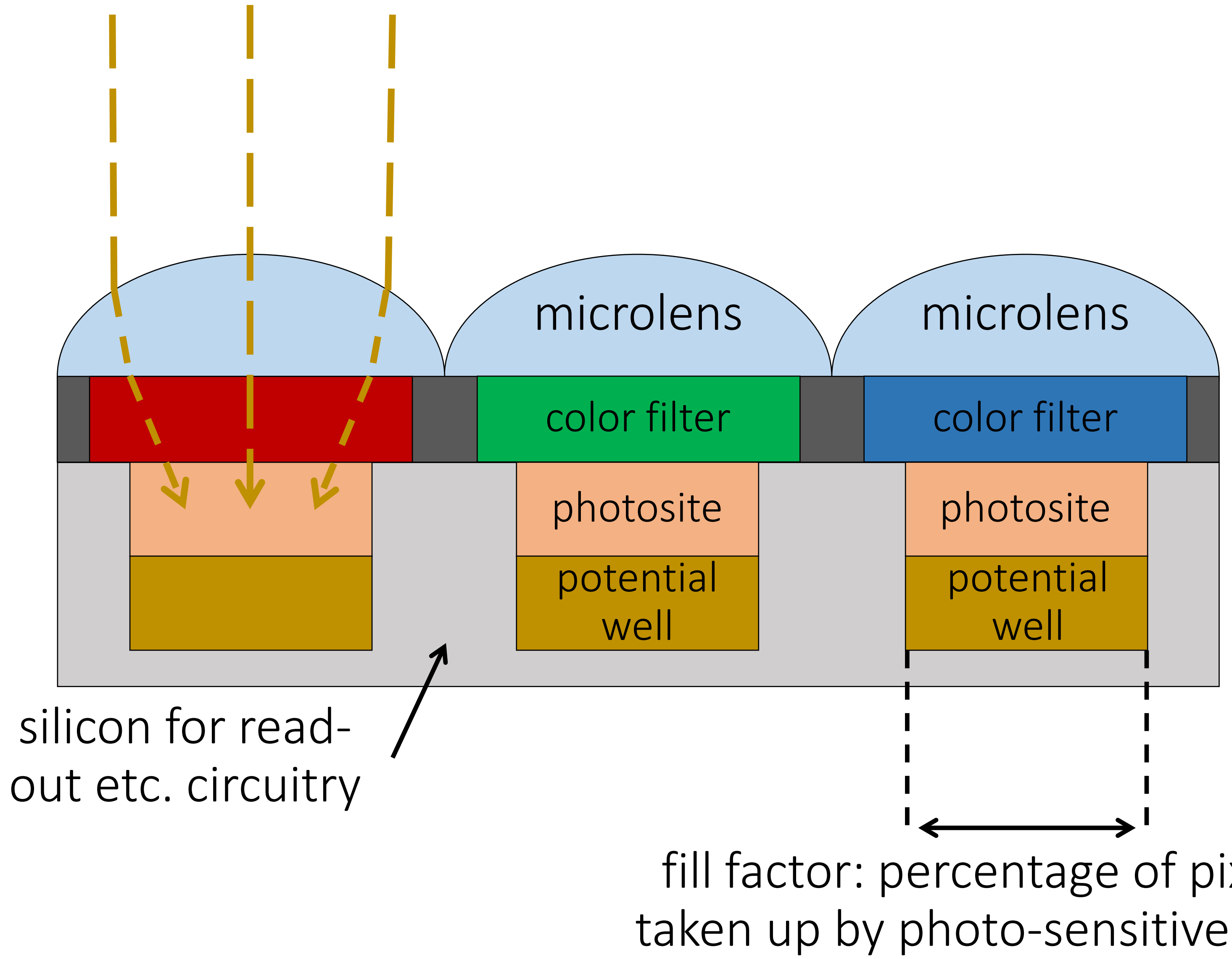


close-up of sensor cross-section


1
2
3
1
1 Pixel diagram
2 Centered micro lens in the middle of the sensor
3 Laterally displaced micro lens at the edge of the sensor

shifted microlenses for improved fill factor

Microlenses (also called lenslets)



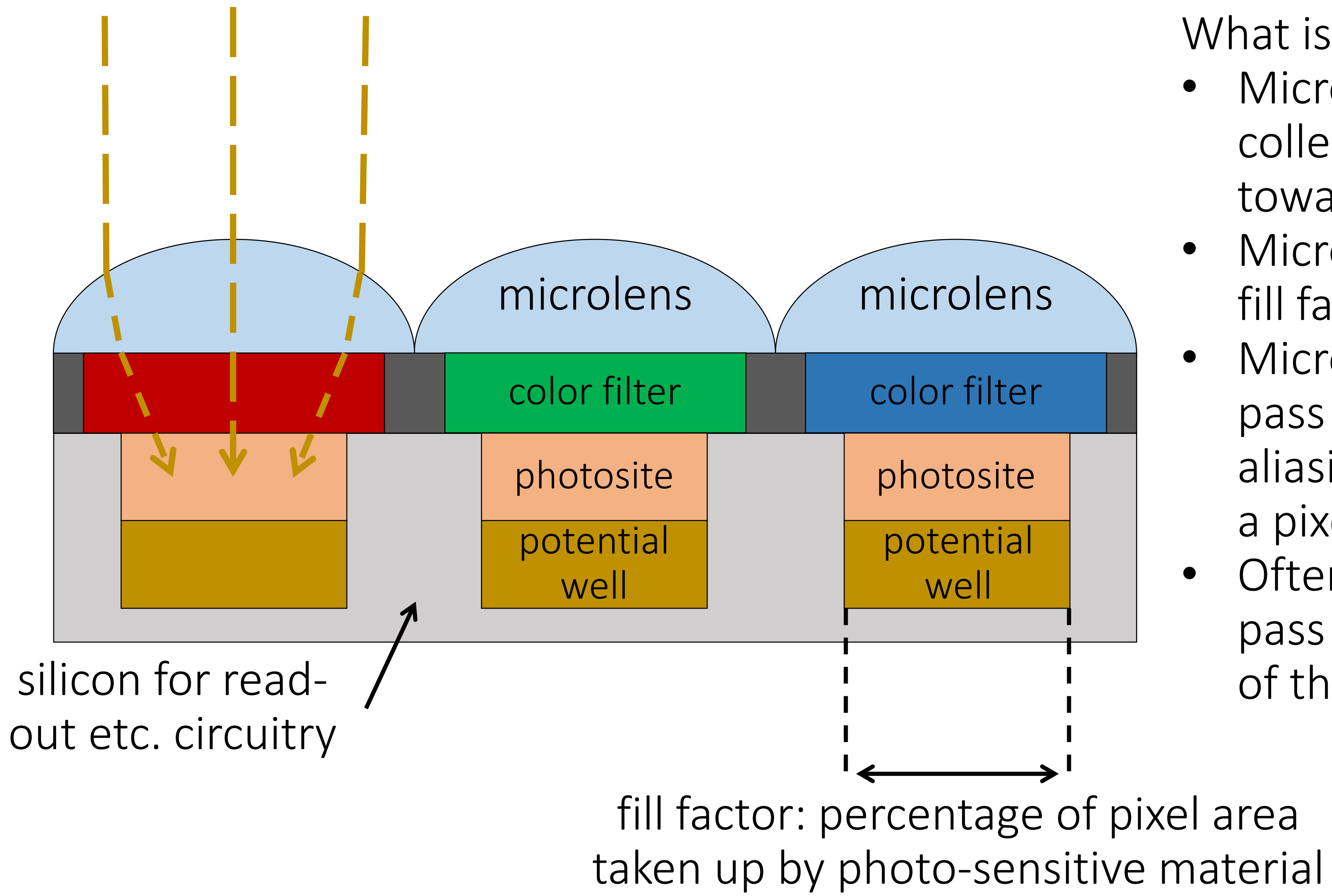
fill factor: percentage of pixel area taken up by photo-sensitive material

What is the role of the microlenses?

- Microlenses help photosite collect more light by bending rays towards photosensitive pixel area.
- Microlenses increase the *effective* fill factor.
- Microlenses also spatially low-pass filter the image to prevent aliasing artifacts.

What kind of spatial filter do the microlenses implement?

Microlenses (also called lenslets)



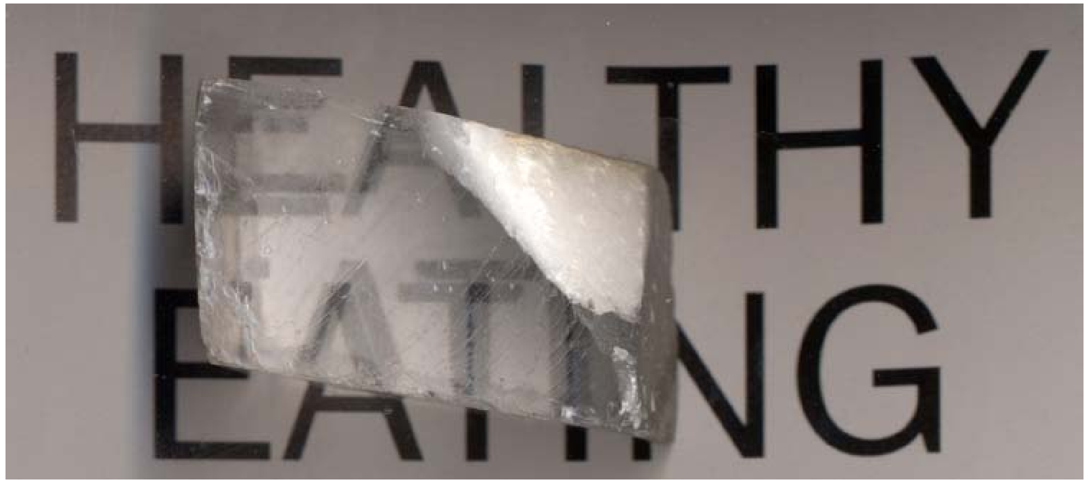
fill factor: percentage of pixel area taken up by photo-sensitive material

What is the role of the microlenses?

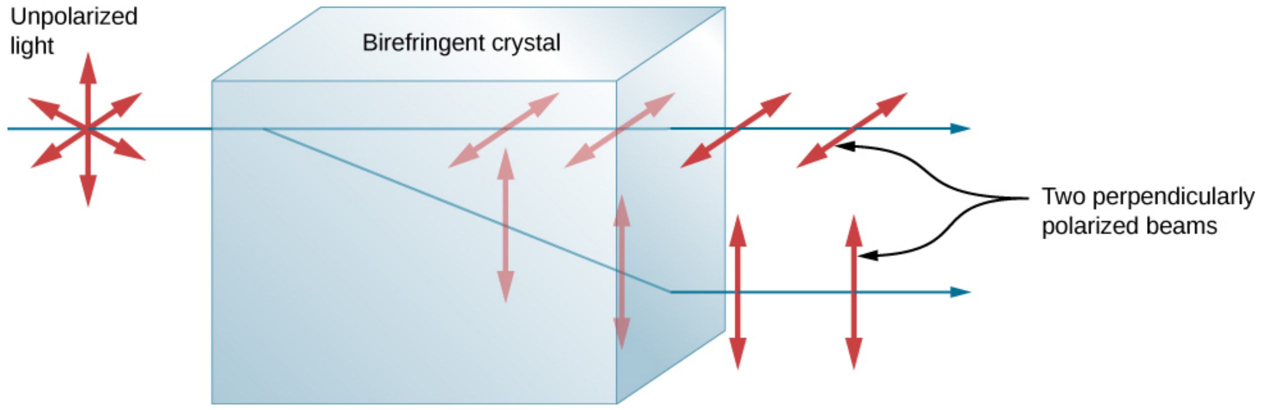
- Microlenses help photosite collect more light by bending rays towards photosensitive pixel area.
- Microlenses increase the *effective* fill factor.
- Microlenses also spatially low-pass filter the image to prevent aliasing artifacts by implementing a pixel-sized 2D rect (box) filter.
- Often an additional optical low-pass filter (OPLF) is placed in front of the sensor to improve prefilter.

Quick aside: optical low-pass filter

- Sensors often have a separate glass sheet in front of them acting as an optical low-pass filter (OLPF, also known as optical anti-aliasing filter).
- The OLPF is typically implemented as two birefringent layers, combined with the infrared filter.
- The two layers split 1 ray into 4 rays, implementing a 4-tap discrete convolution filter kernel.



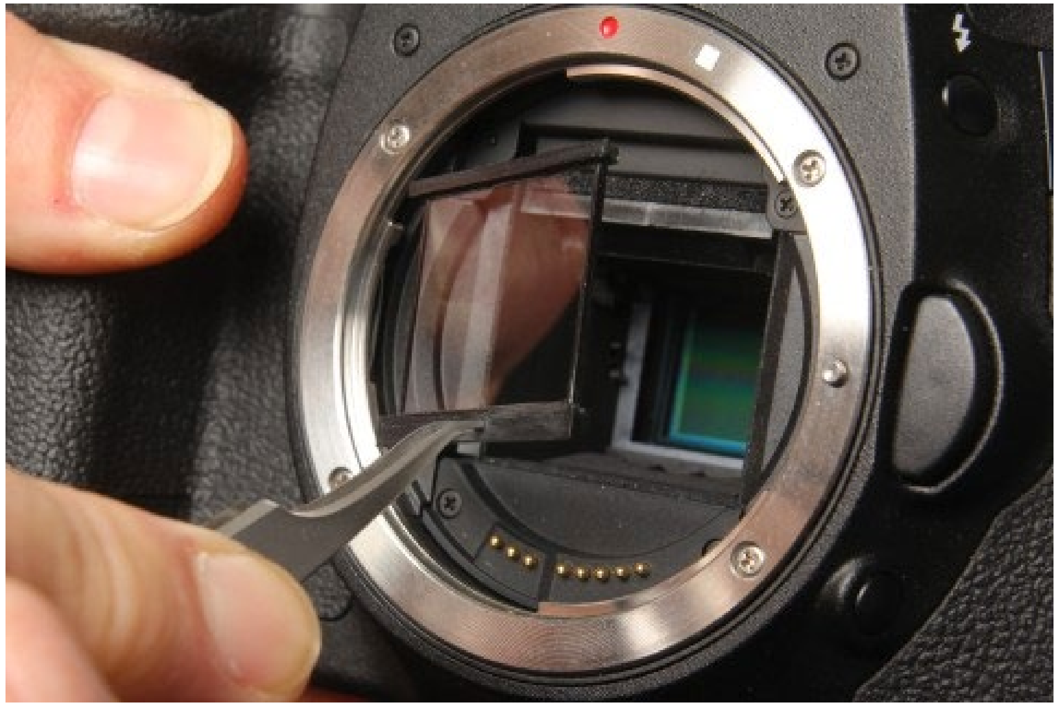
birefringence in a calcite crystal



birefringence ray diagram

Quick aside: optical low-pass filter

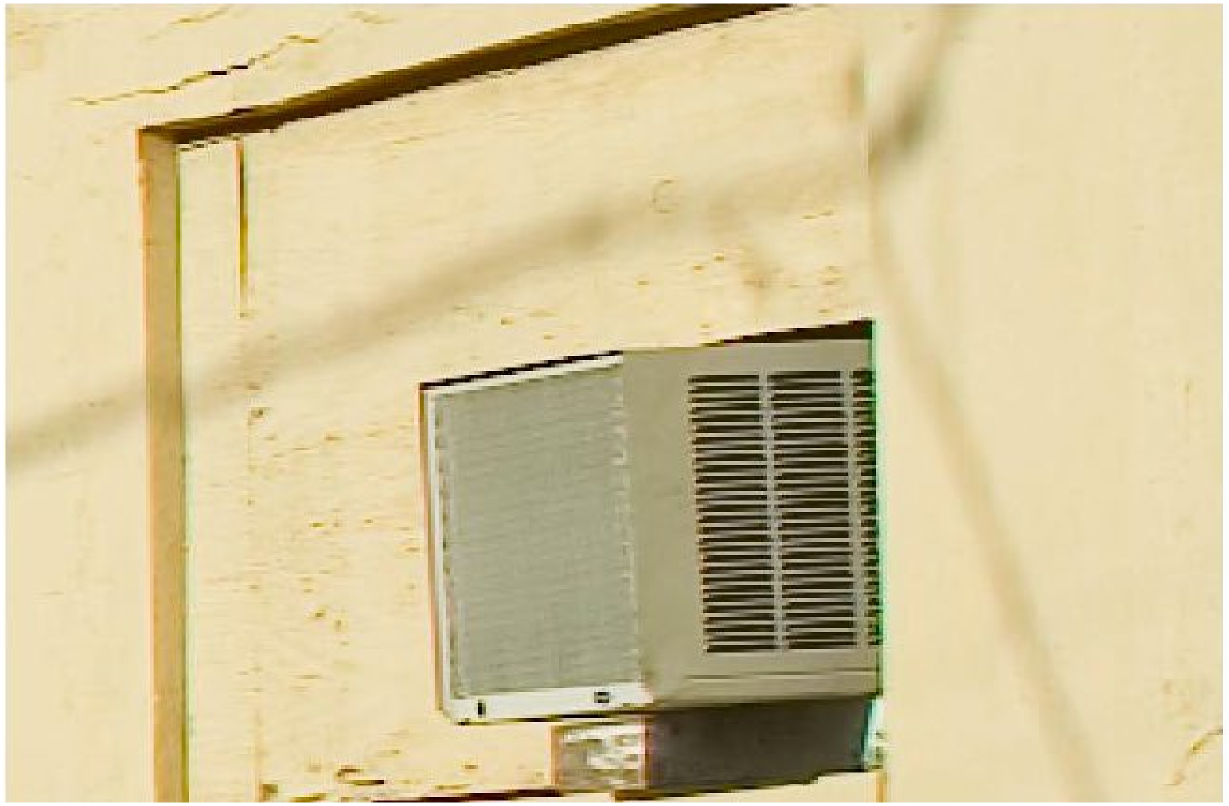
- Sensors often have a separate glass sheet in front of them acting as an optical low-pass filter (OLPF, also known as optical anti-aliasing filter).
- The OLPF is typically implemented as two birefringent layers, combined with the infrared filter.
- The two layers split 1 ray into 4 rays, implementing a 4-tap discrete convolution filter kernel.

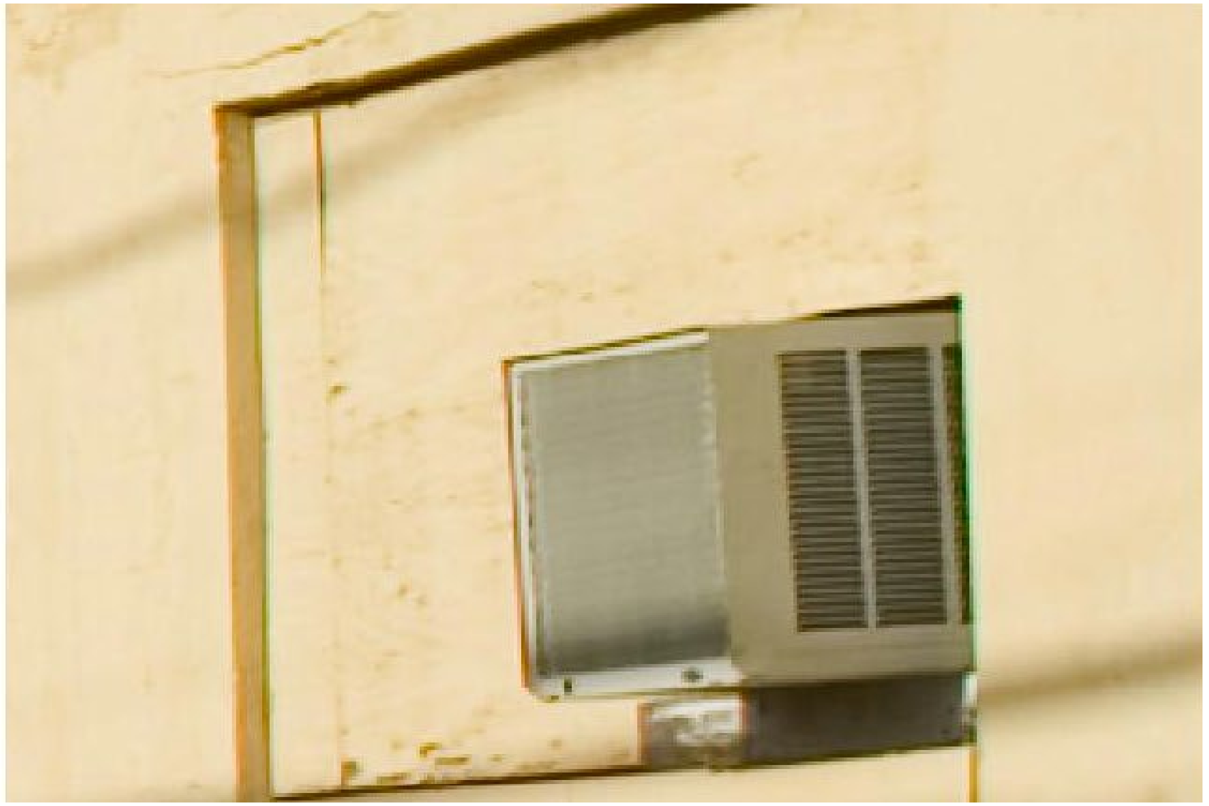


- However, the OLPF means you also lose resolution.
- Nowadays, due the large number of pixels, OLPF are becoming unnecessary.
- Photographers often hack their cameras to remove the OLPF, to avoid the loss of resolution (“hot rodding”).
- Camera manufacturers offer camera versions with and without an OLPF.
- The OLPF can be problematic also when working with coherent light (spurious fringes).

Quick aside: optical low-pass filter

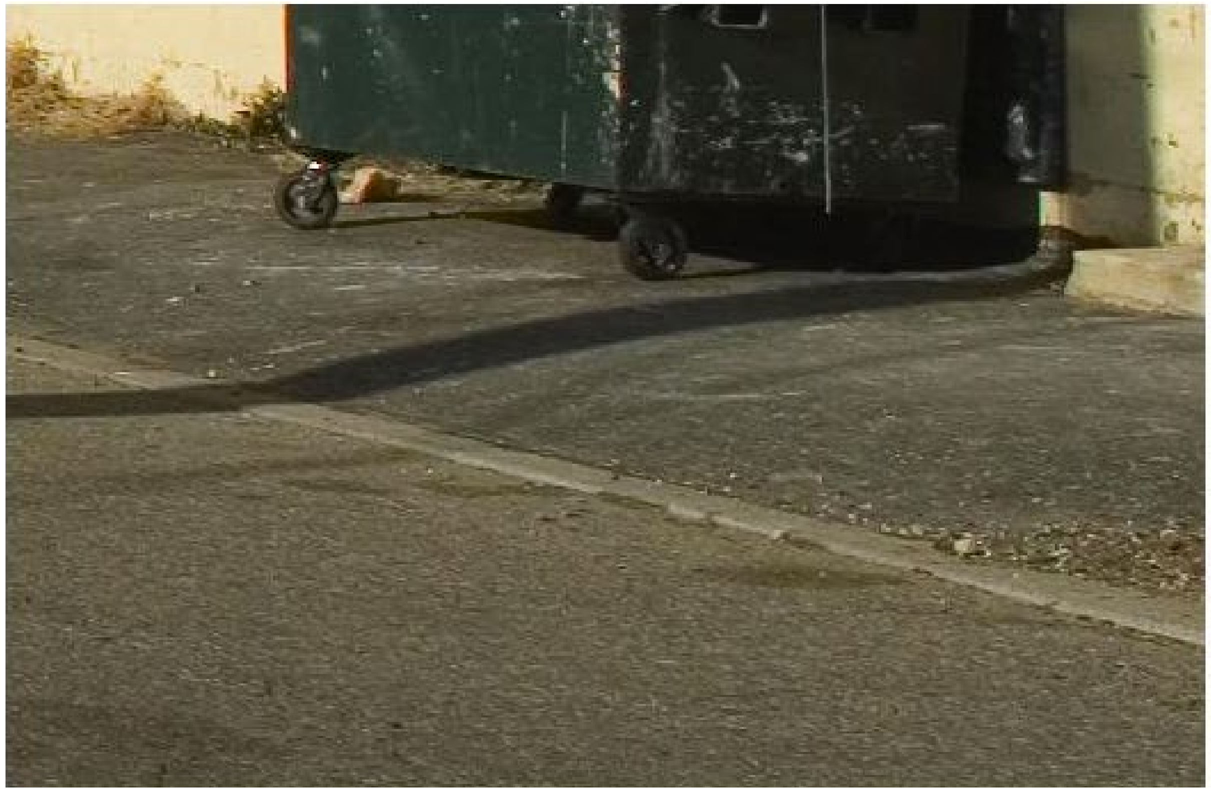
Example where OLPF is needed


without OLPF

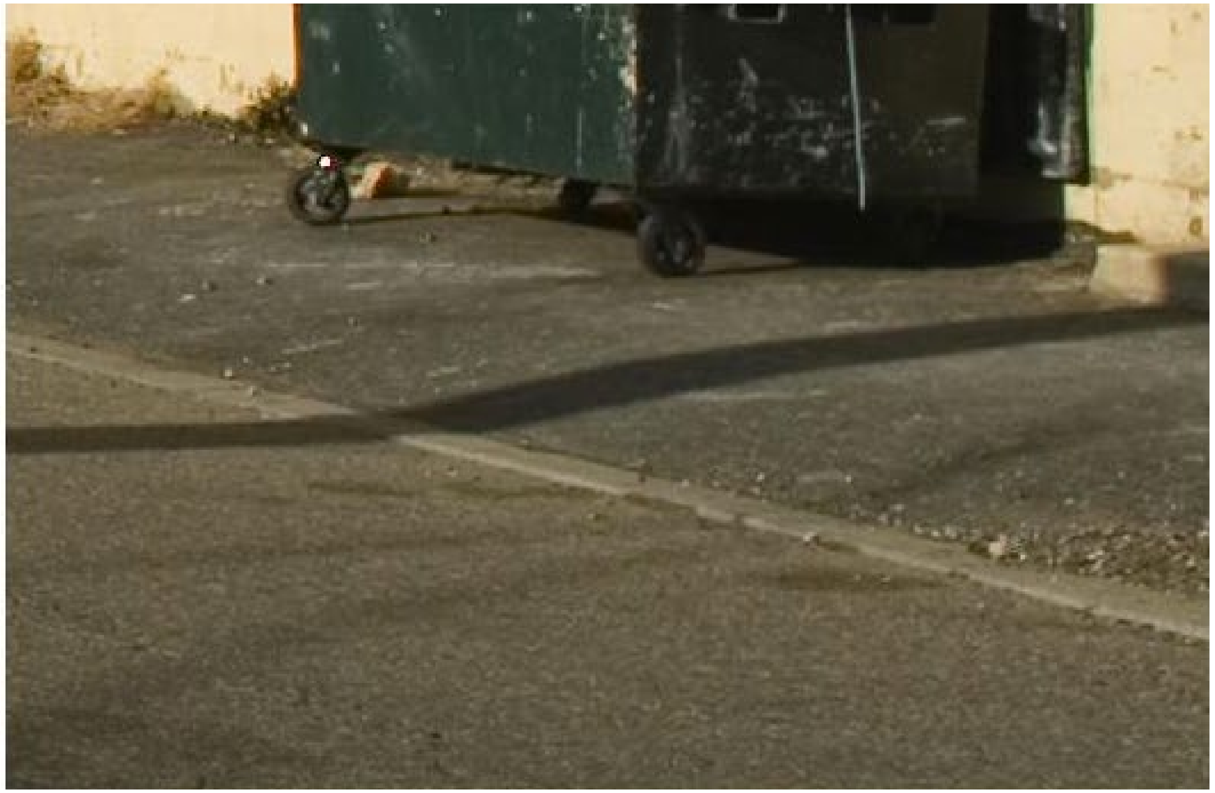

with OLPF

Quick aside: optical low-pass filter

Example where OLPF is unnecessary



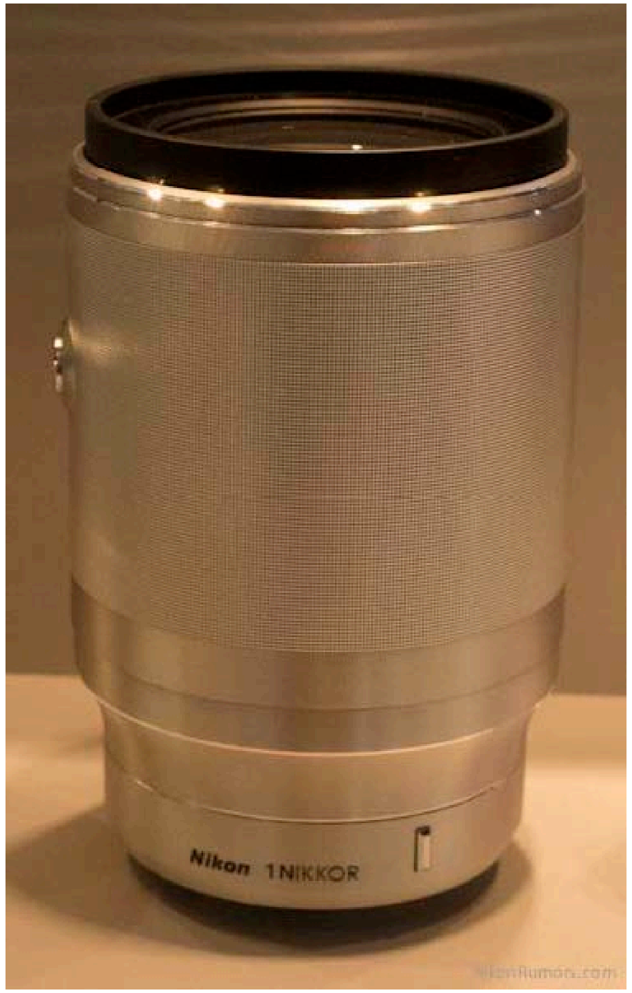
without OLPF



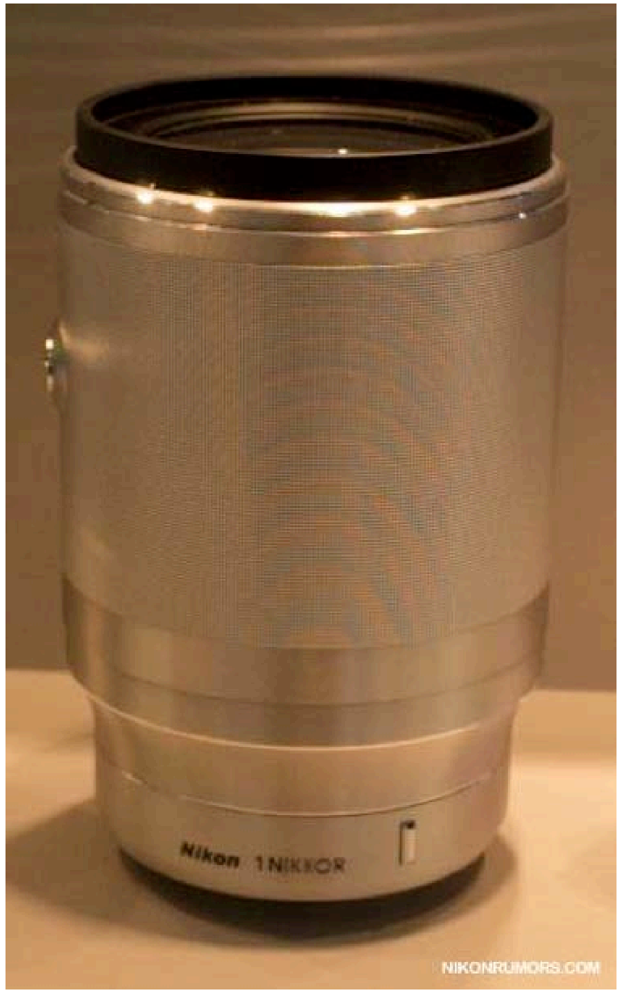
with OLPF

Quick aside: optical low-pass filter

Identical camera model with and without an OLPF (no need for customization).

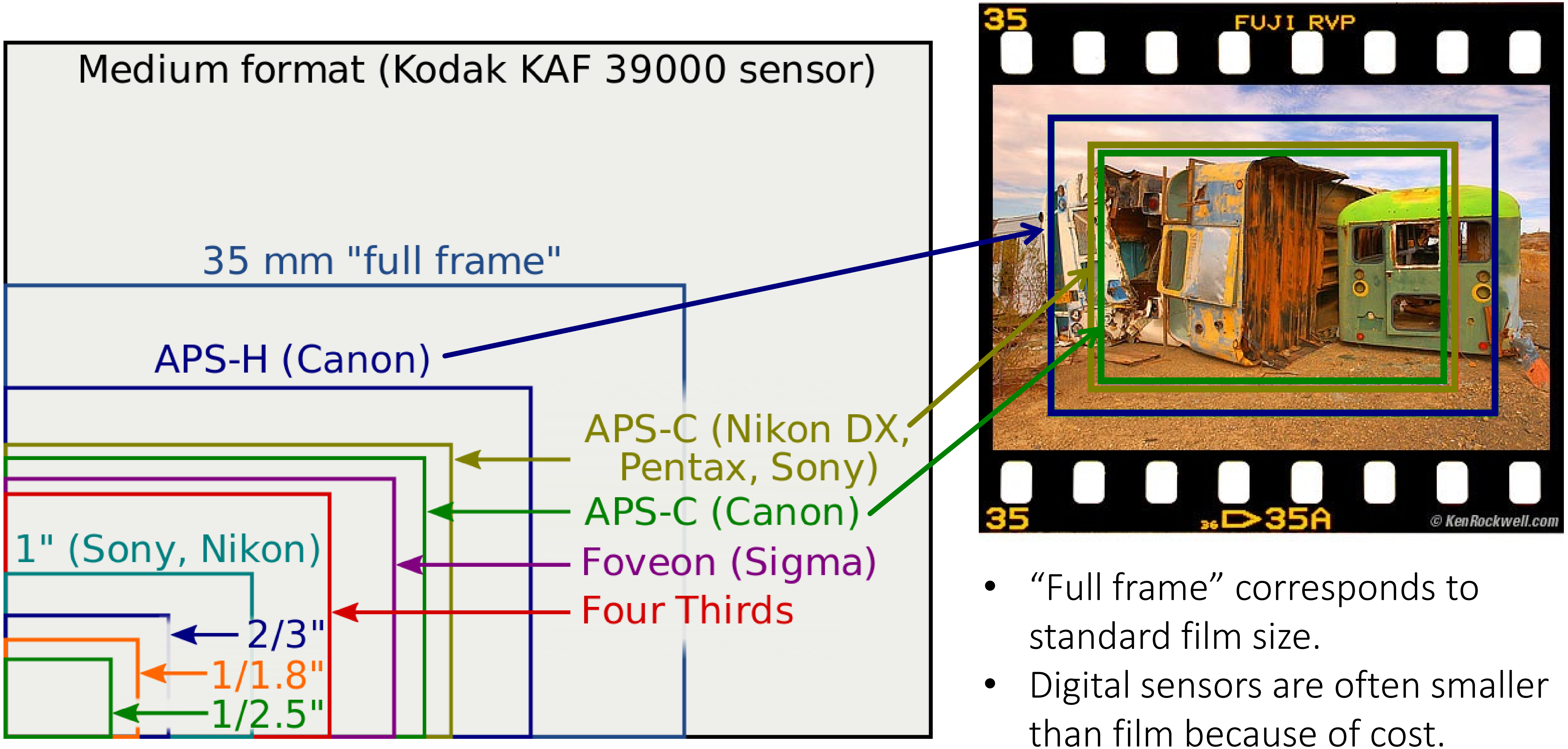


Nikon D800



Nikon D800E

Sensor size

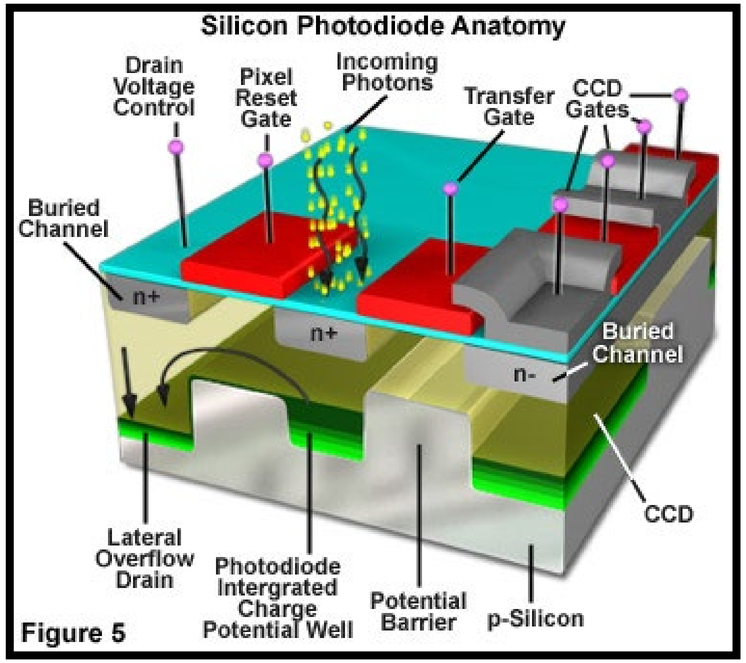


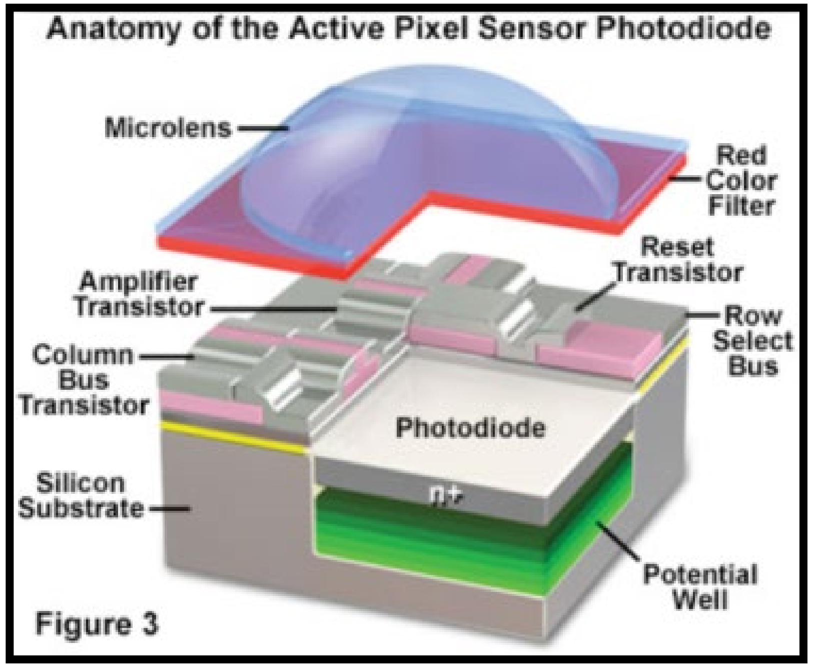
- "Full frame" corresponds to standard film size.
- Digital sensors are often smaller than film because of cost.

Two main types of imaging sensors

Do you know them?

Two main types of imaging sensors



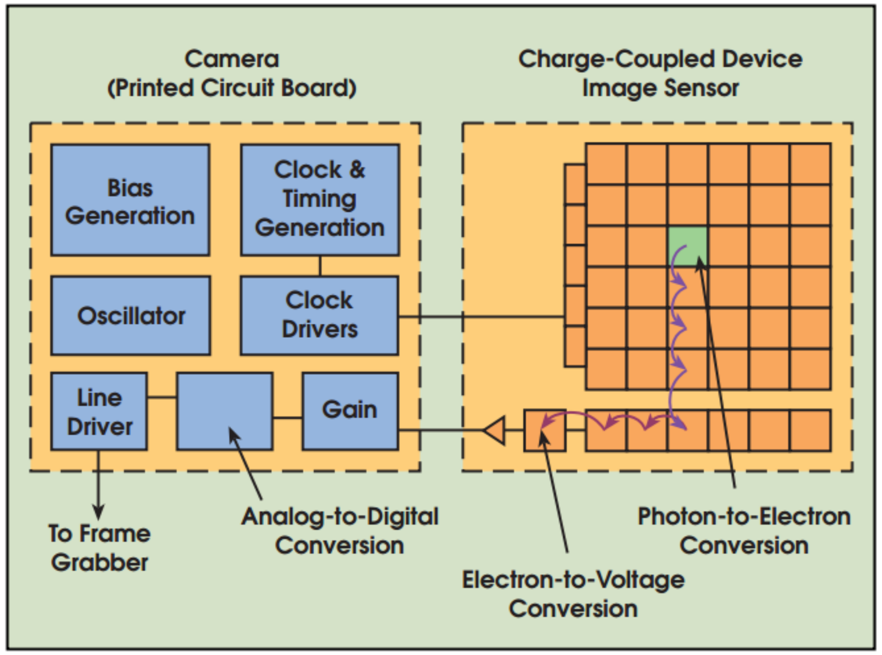


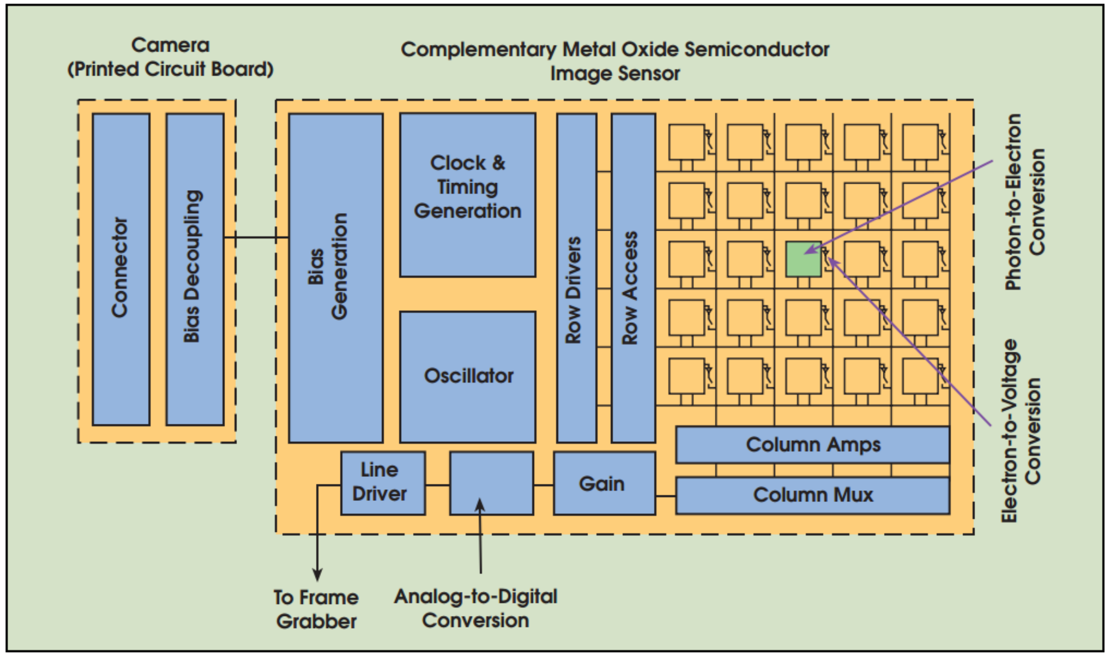
Charged coupled device (CCD):
- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

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



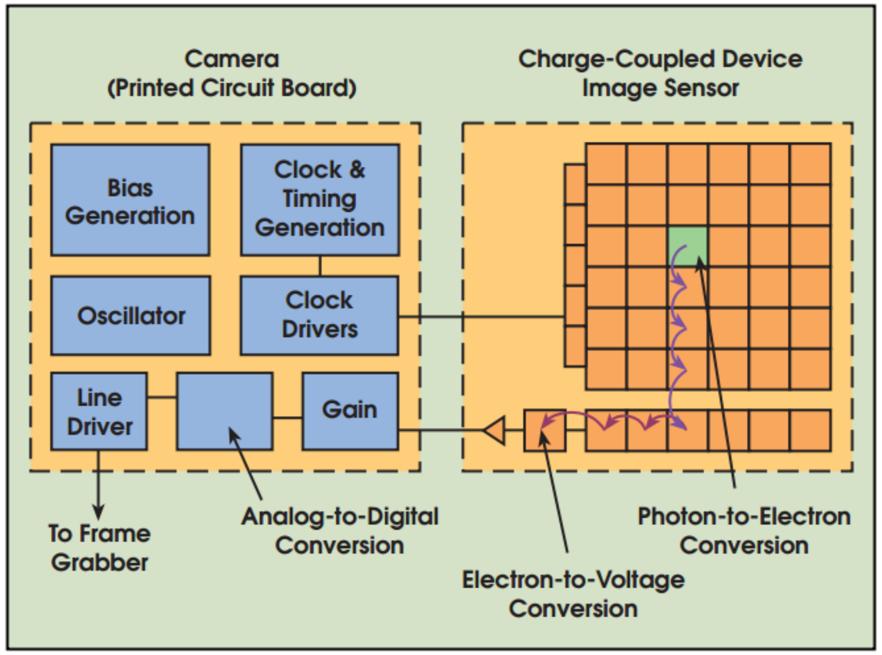


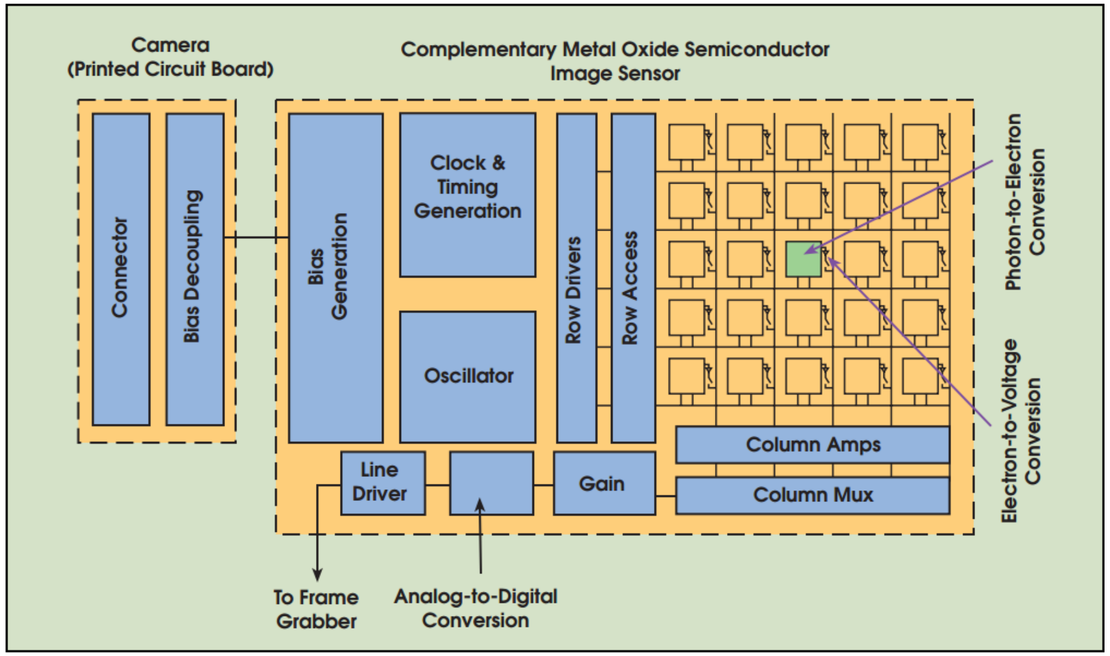
Charged coupled device (CCD):
- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

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


Camera (Printed Circuit Board)
Charge-Coupled Device Image Sensor
Bias Generation
Clock & Timing Generation
Oscillator
Clock Drivers
Line Driver
Gain
To Frame Grabber
Analog-to-Digital Conversion
Photon-to-Electron Conversion
Electron-to-Voltage Conversion


Camera (Printed Circuit Board)
Complementary Metal Oxide Semiconductor Image Sensor
Connector
Bias Decoupling
Bias Generation
Clock & Timing Generation
Oscillator
Row Drivers
Row Access
Column Amps
Column Mux
Line Driver
Gain
To Frame Grabber
Analog-to-Digital Conversion
Photon-to-Electron Conversion
Electron-to-Voltage Conversion

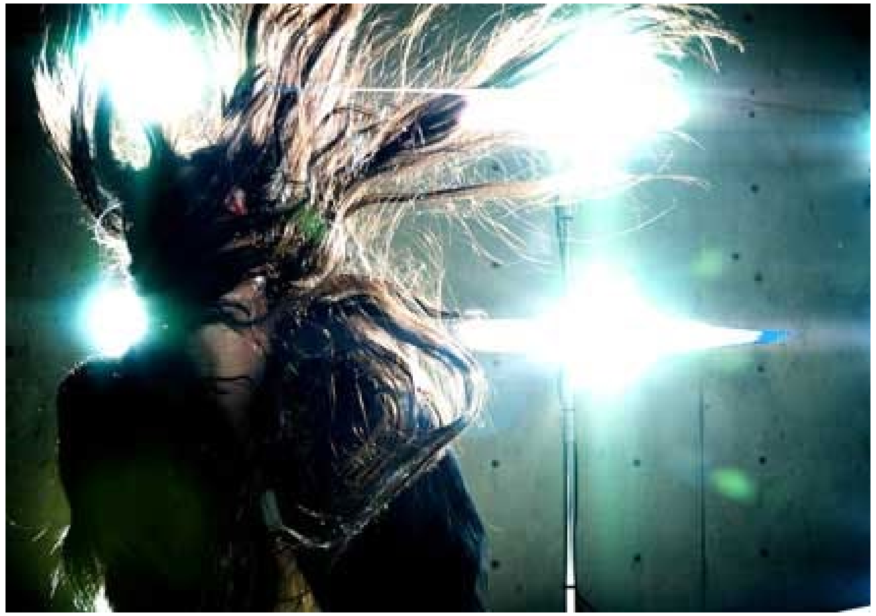
Charged coupled device (CCD):
- row brigade shifts charges row-by-row
- amplifiers convert charges to voltages row-by-row

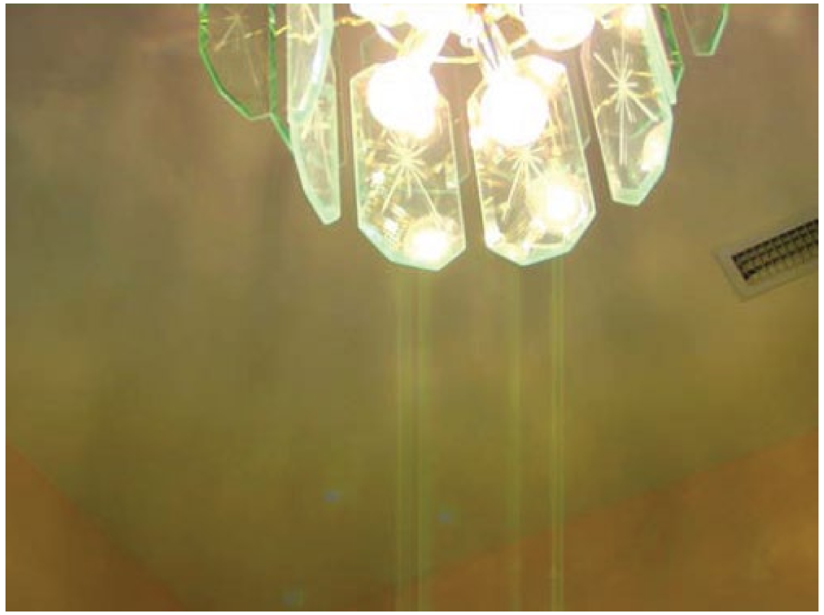
✓ higher sensitivity
✓ lower noise

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

✓ faster read-out
✓ lower cost

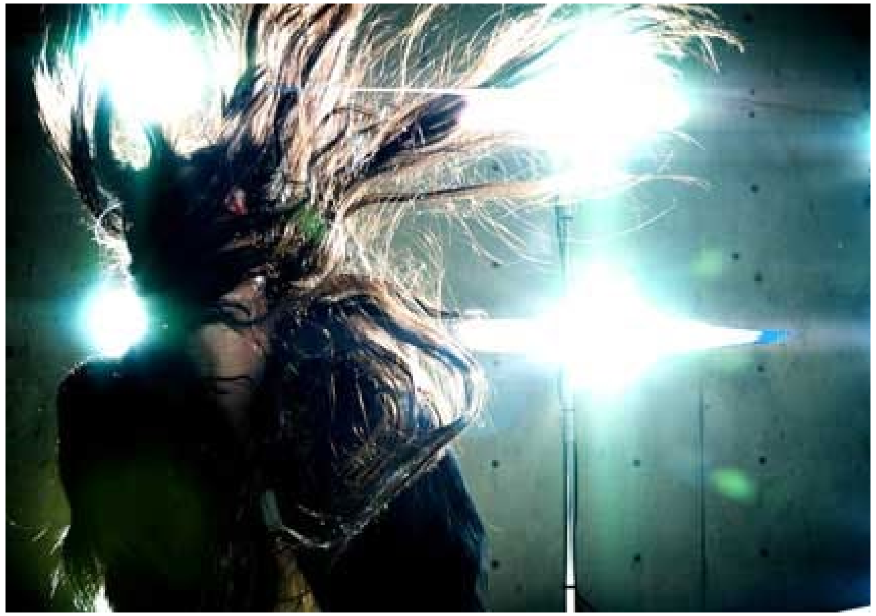
Artifacts of the two types of sensors

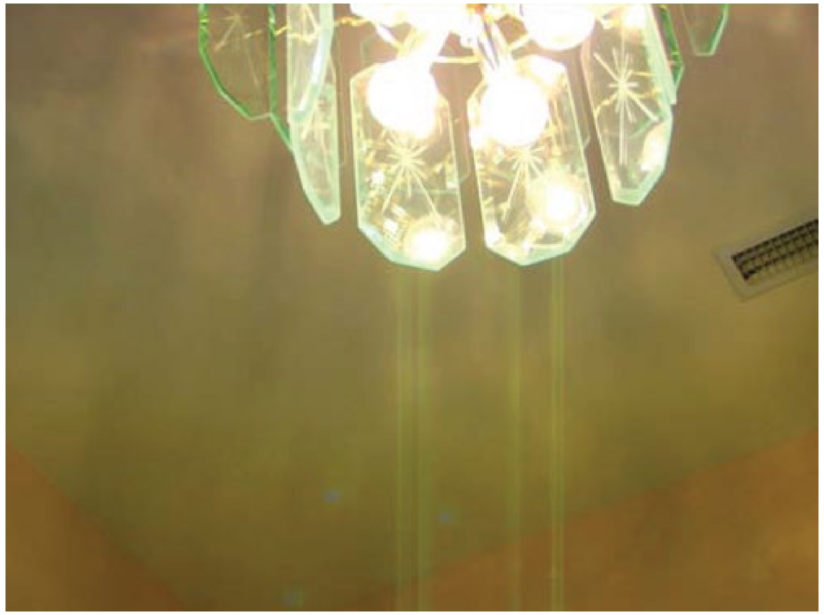

sensor bloom


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)

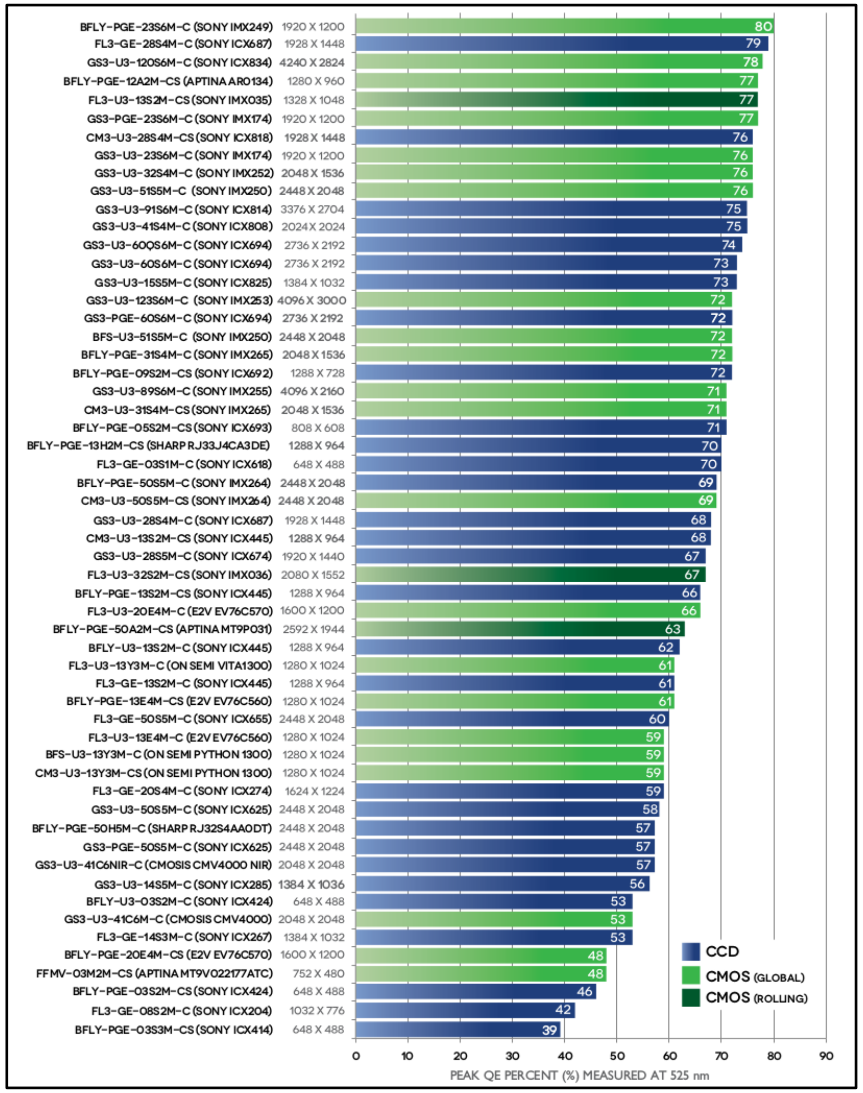
Overflow from saturated pixels

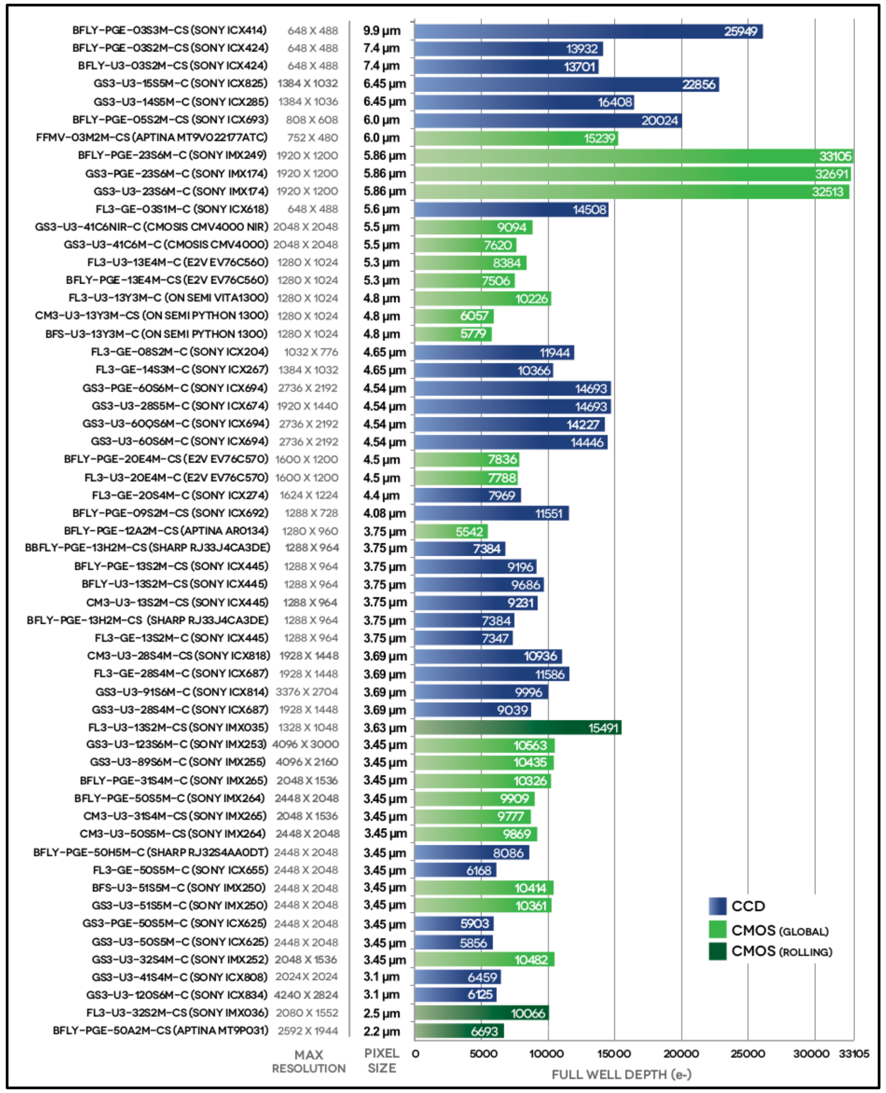
- mitigated by more electronics to contain charge
  (at the cost of photosensitive area)

CCD vs CMOS

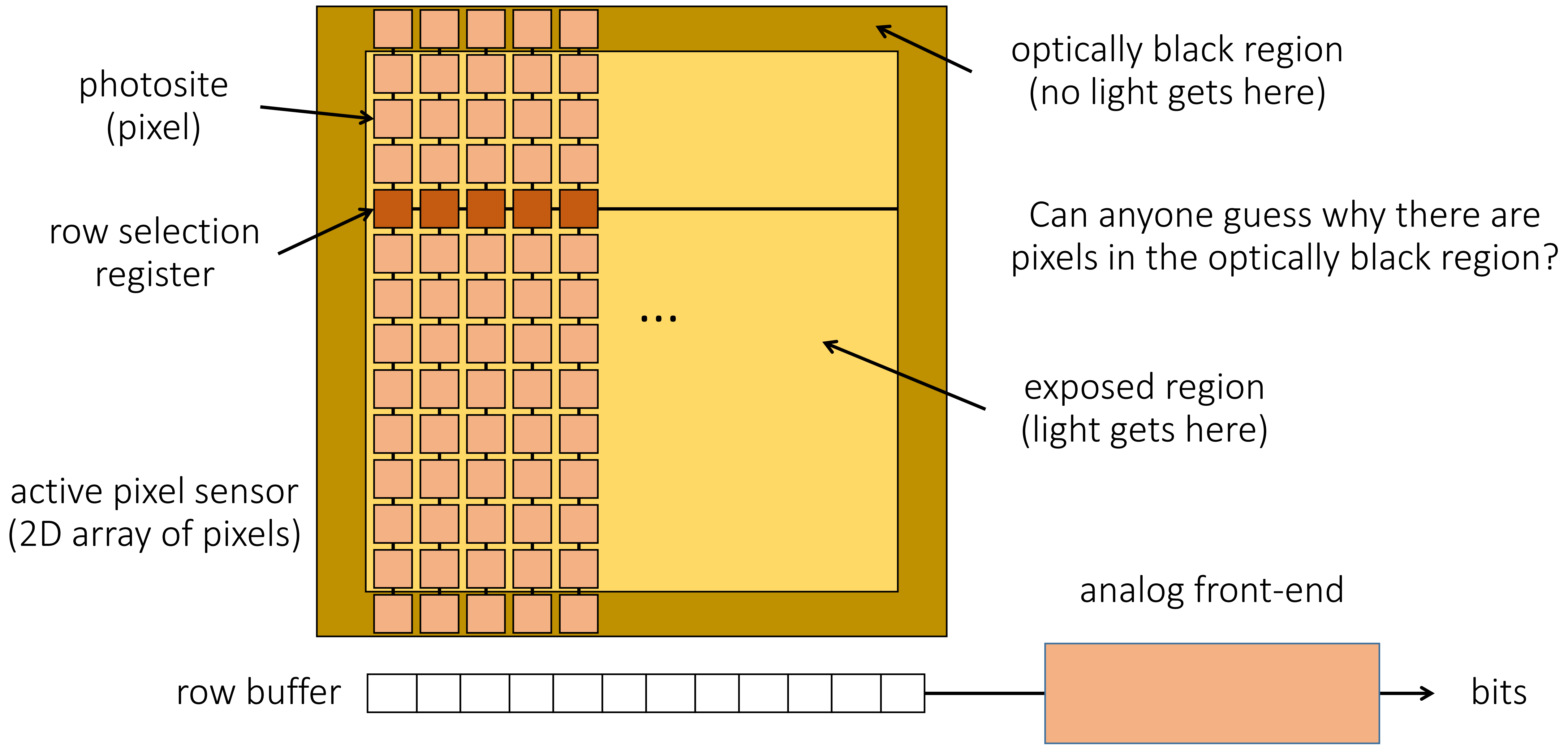
- Modern CMOS sensors have optical performance comparable to CCD sensors.

- Most modern commercial and industrial cameras use CMOS sensors.



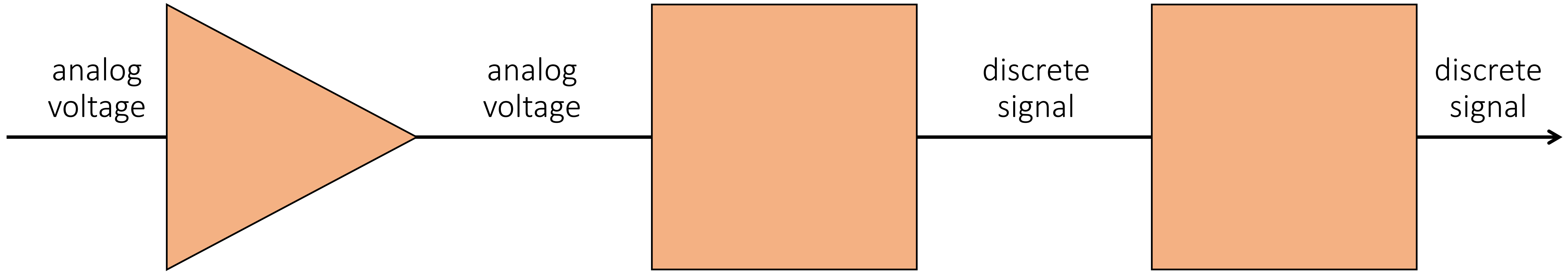


CMOS sensor (very) simplified layout



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

Analog front-end



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

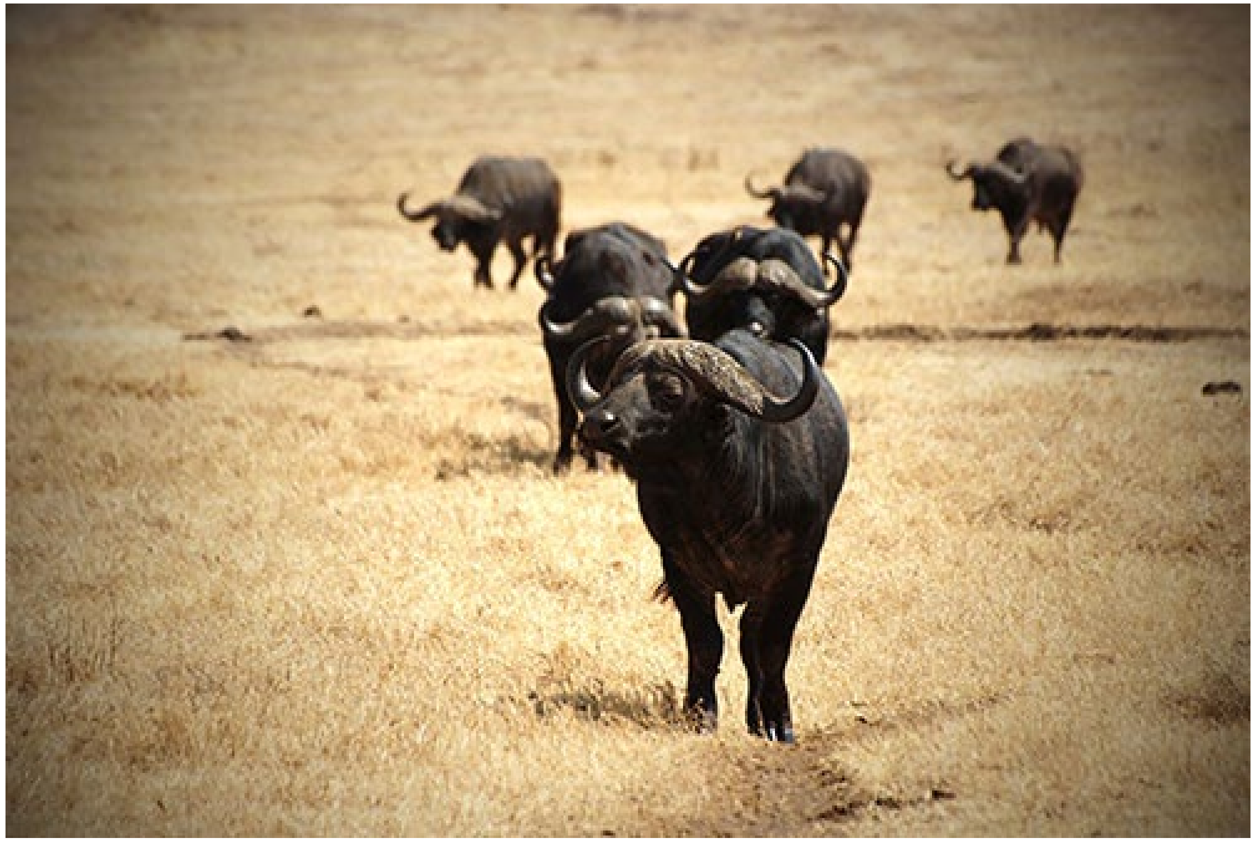
<u>analog-to-digital converter (ADC)</u>:
- depending on sensor, output has 10-16 bits.
- most often (?) 12 bits.

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

Vignetting

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

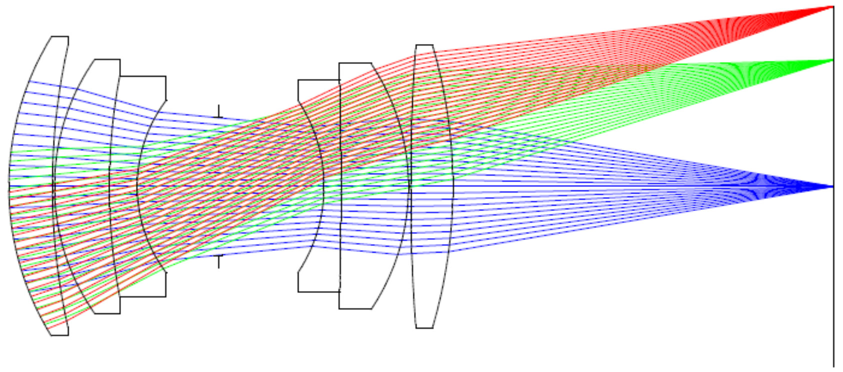
white wall under uniform light


more interesting example of vignetting

Vignetting

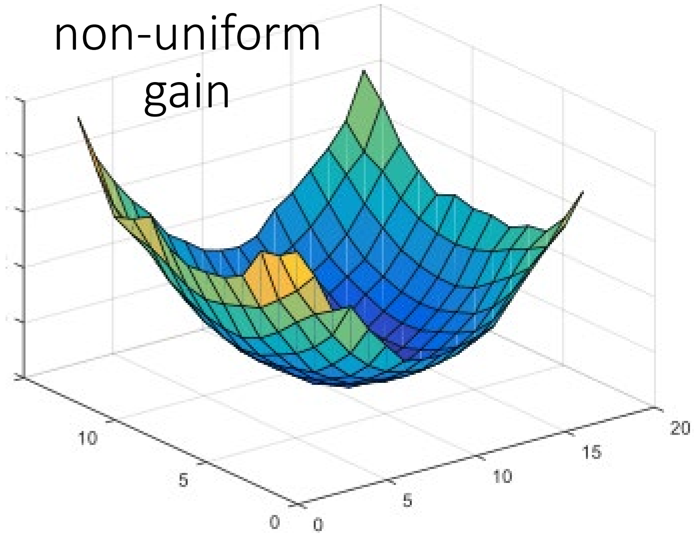
Four types of vignetting:

- 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 photosites.



$\longrightarrow$

non-uniform gain



$\longrightarrow$

What does an imaging sensor do?

When the camera shutter opens, the sensor:

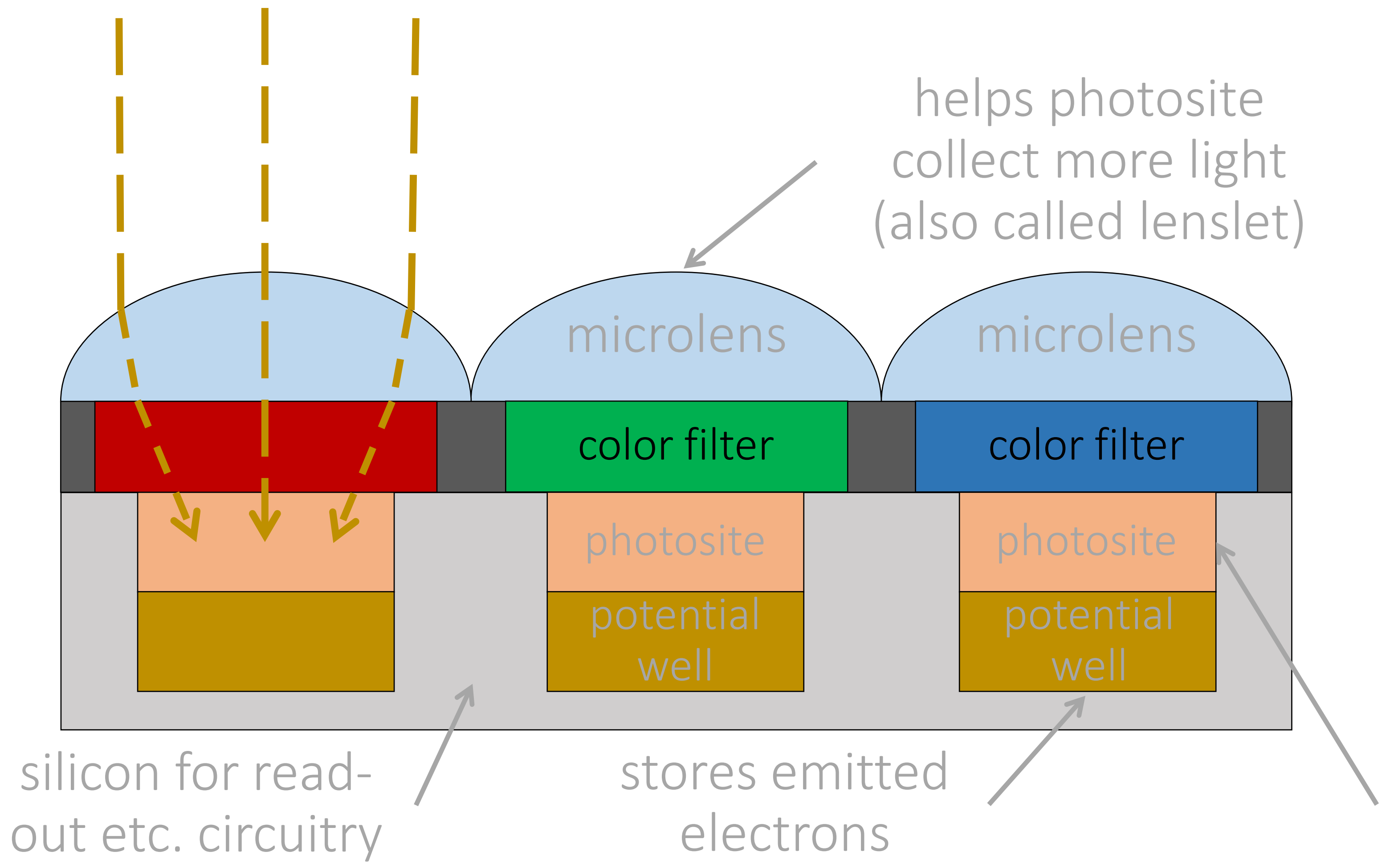
- at every photosite, converts incident photons into electrons
- stores electrons into the photosite’s potential well while it is not full

… 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?


helps photosite collect more light (also called lenslet)
microlens
microlens
color filter
color filter
photosite
potential well
photosite
potential well
silicon for read-out etc. circuitry
stores emitted electrons
made of silicon, emits electrons from photons

- 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 primer

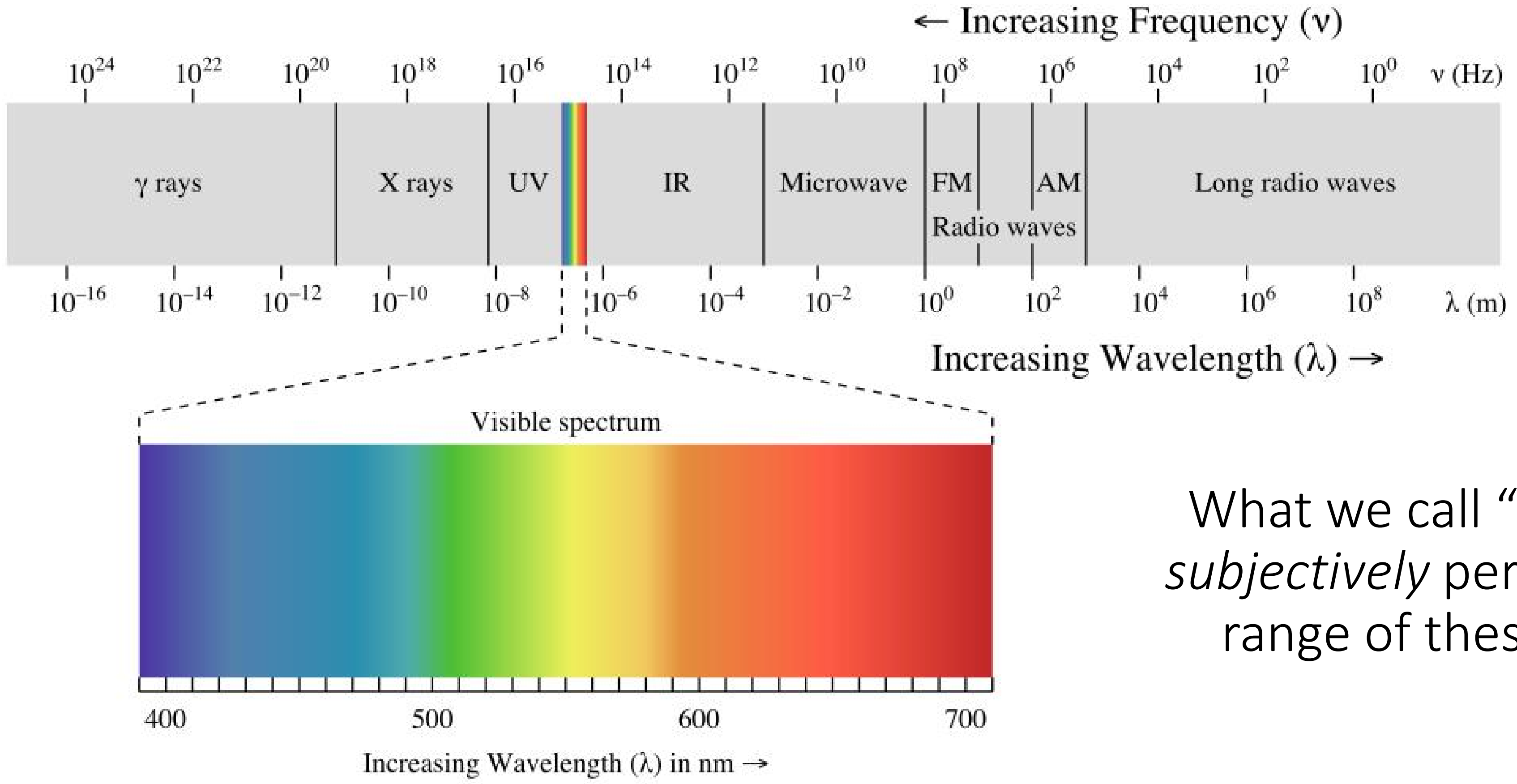
Color

- 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 a later lecture.

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.

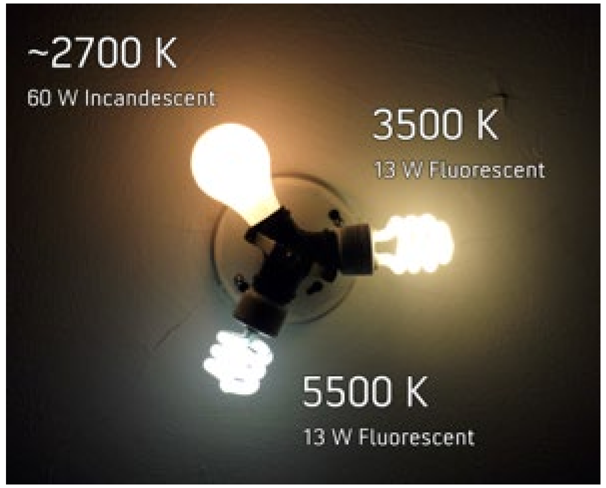

← Increasing Frequency (ν)
10^{24} 10^{22} 10^{20} 10^{18} 10^{16} 10^{14} 10^{12} 10^{10} 10^{8} 10^{6} 10^{4} 10^{2} 10^{0} ν (Hz)
γ rays
X rays
UV
IR
Microwave
FM
AM
Radio waves
Long radio waves
10^{-16} 10^{-14} 10^{-12} 10^{-10} 10^{-8} 10^{-6} 10^{-4} 10^{-2} 10^{0} 10^{2} 10^{4} 10^{6} 10^{8} λ (m)
Increasing Wavelength (λ) →
Visible spectrum
400
500
600
700
Increasing Wavelength (λ) in nm →

electromagnetic spectrum

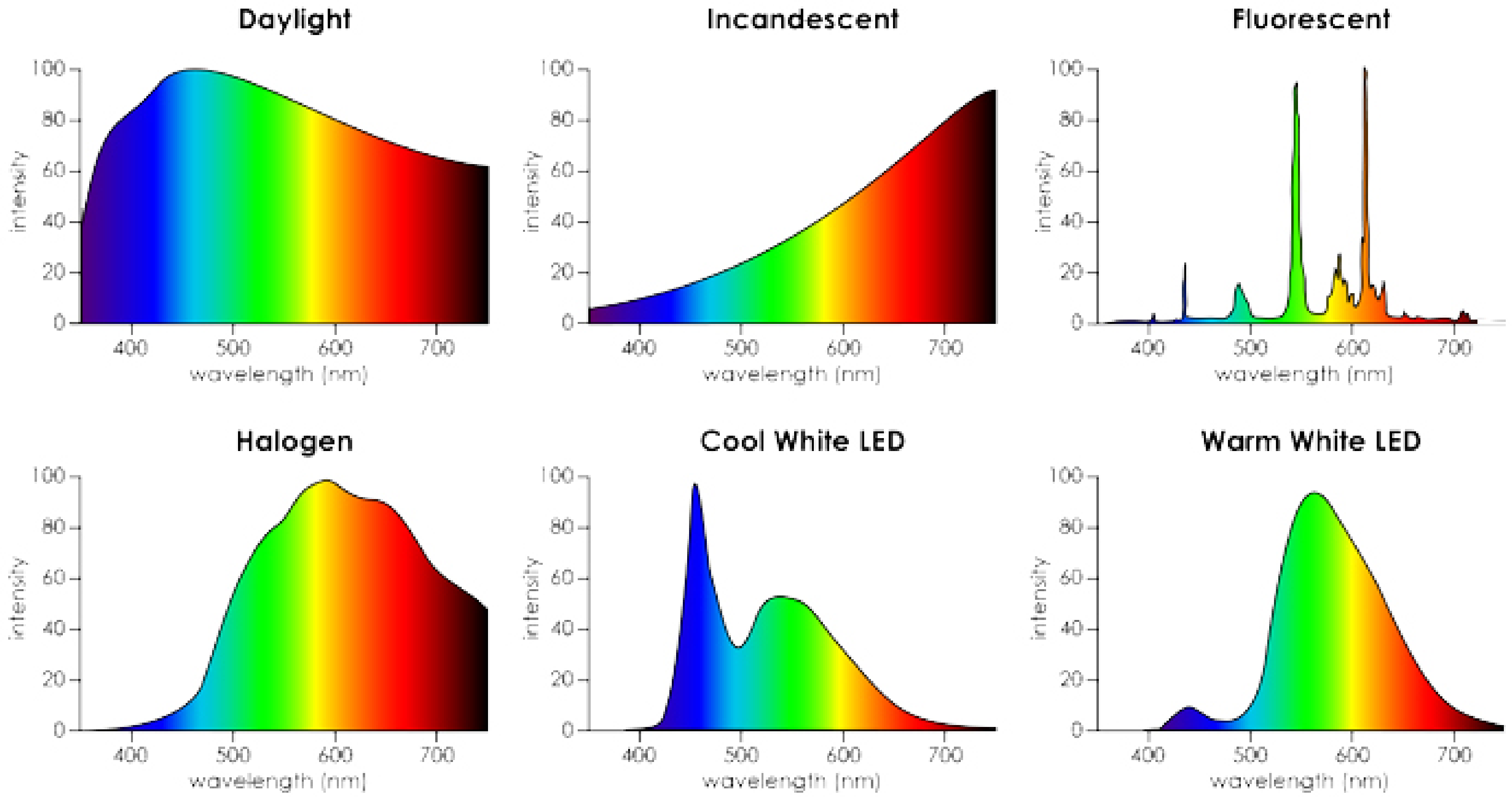
What we call "color" is how we *subjectively* perceive a very small range of these wavelengths.

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)

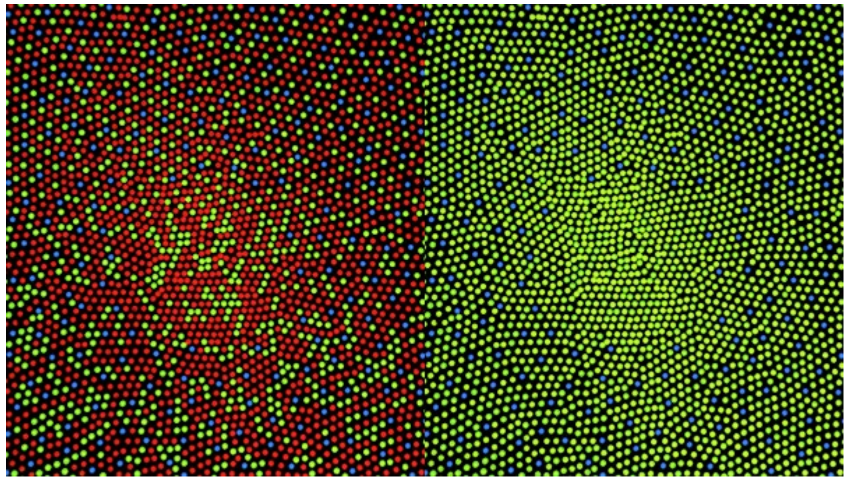
- *Any* light sensor (digital or not) has different sensitivity to different wavelengths.
- This is described by the sensor's *spectral sensitivity function* $f(\lambda)$.
- When measuring light of a some SPD $\Phi(\lambda)$, the sensor produces a *scalar* response:

$$\text{sensor response} \longrightarrow R = \int_\lambda \overset{\text{light SPD}}{\Phi(\lambda)} \overset{\text{sensor SSF}}{f(\lambda)} d\lambda$$

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*).

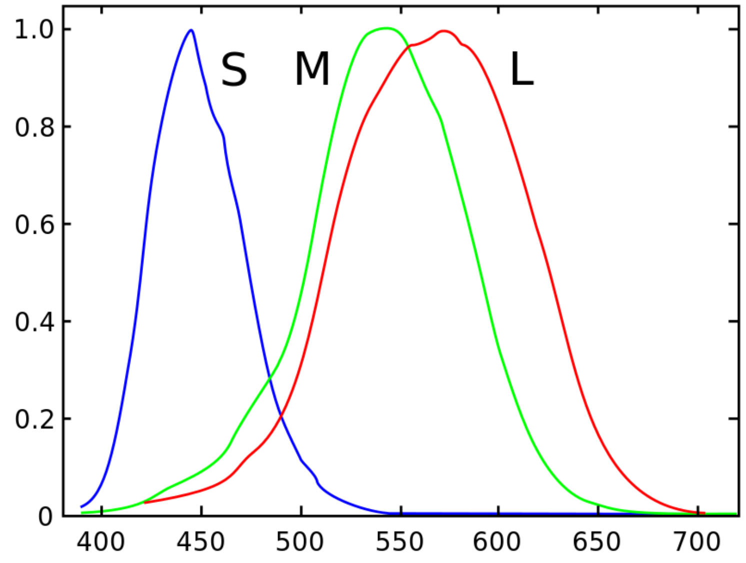


cone distribution for normal vision (64% L, 32% M)

“short” $$S = \int_\lambda \Phi(\lambda) S(\lambda) d\lambda$$

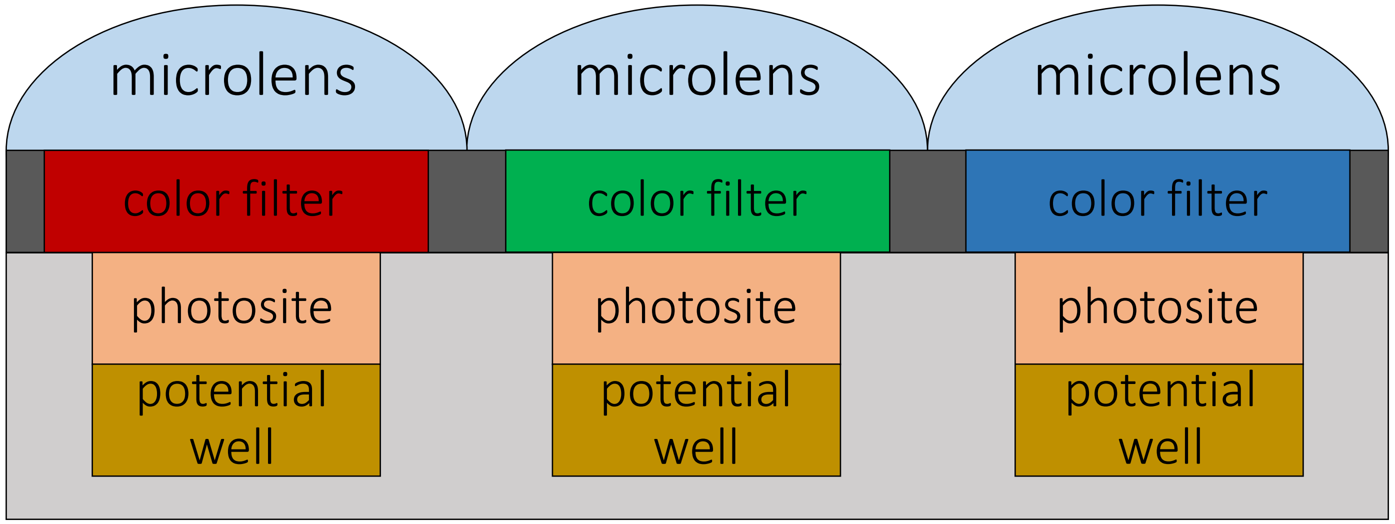
“medium” $$M = \int_\lambda \Phi(\lambda) M(\lambda) d\lambda$$

“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.

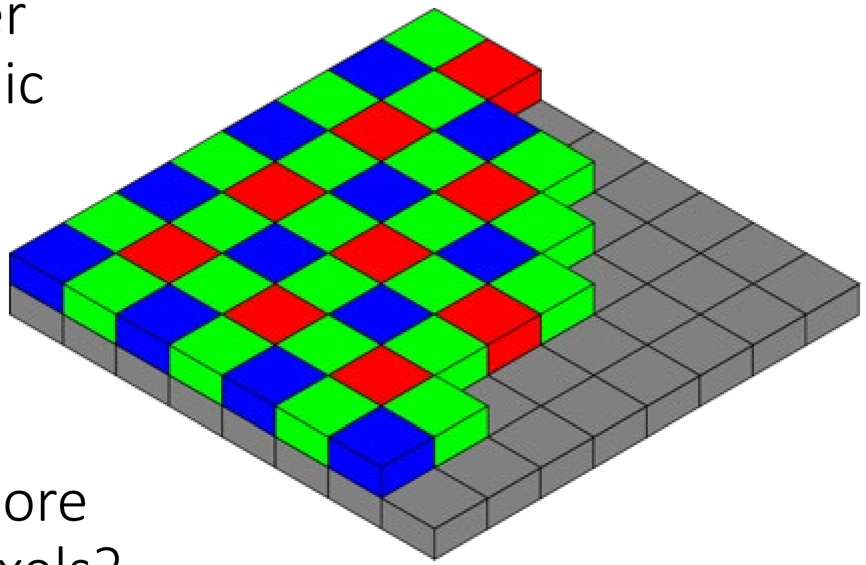


What color filters to use?

Two design choices:

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

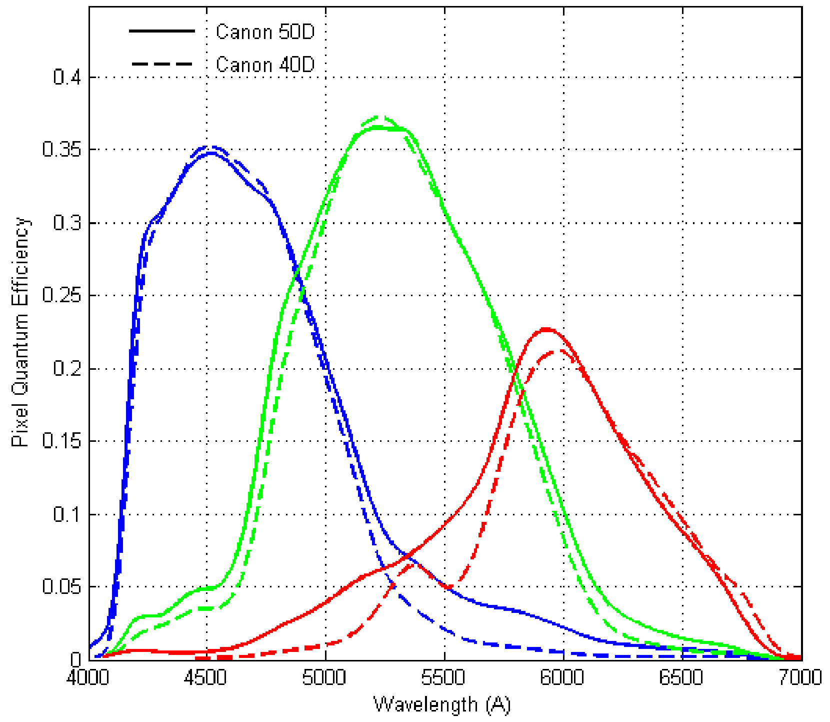
Bayer mosaic



Why more green pixels?

SSF for Canon 50D

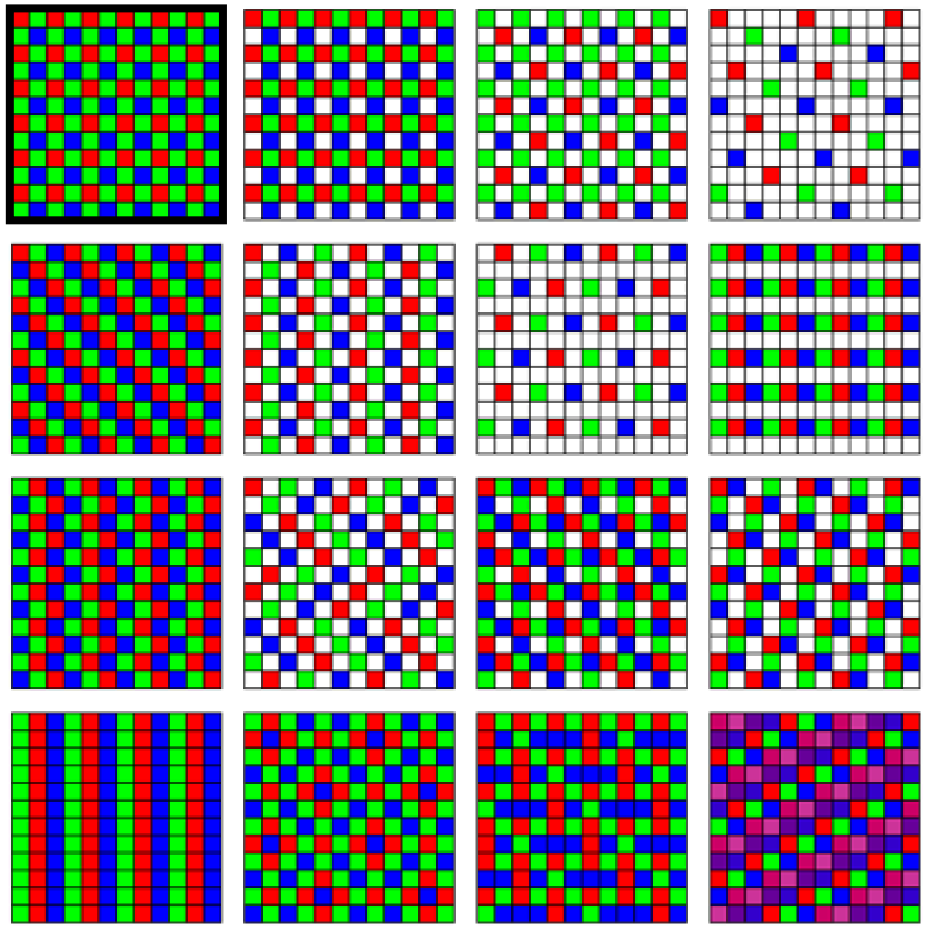
Generally do not match human LMS.

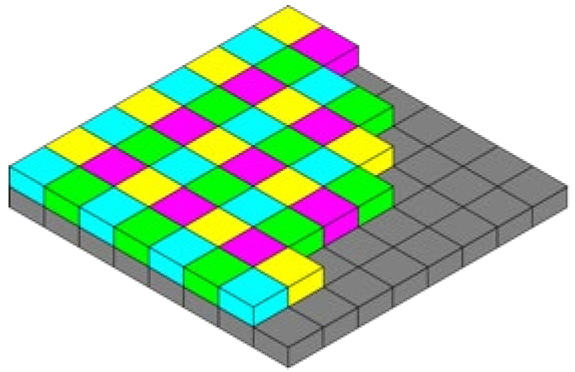


$f(\lambda)$

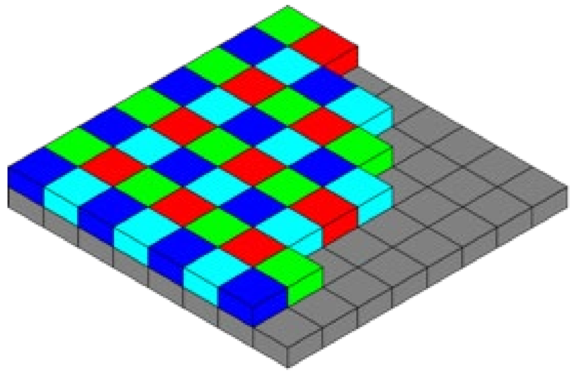
Many different CFAs

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





CYGM
Canon IXUS, Powershot



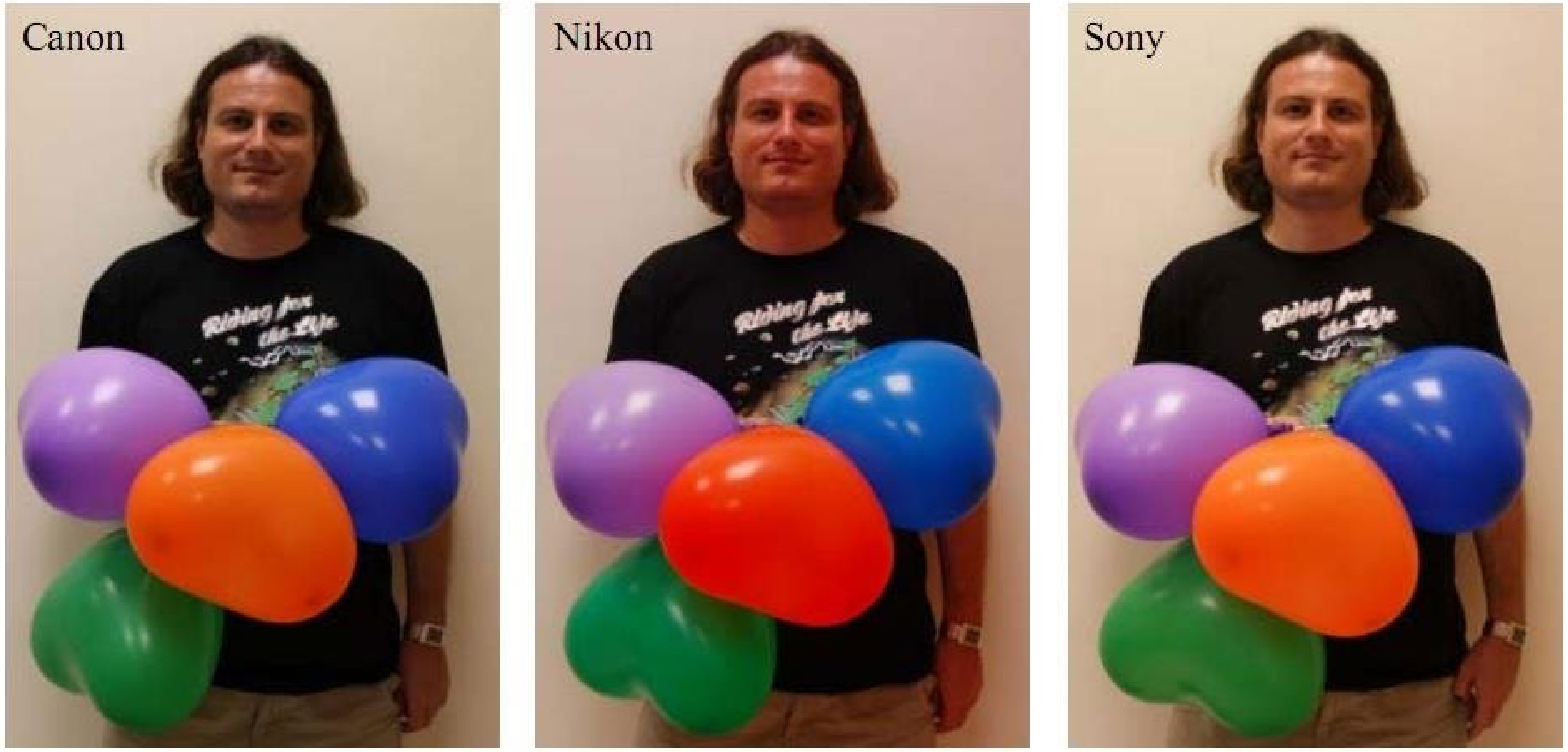
RGBE
Sony Cyber-shot

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.

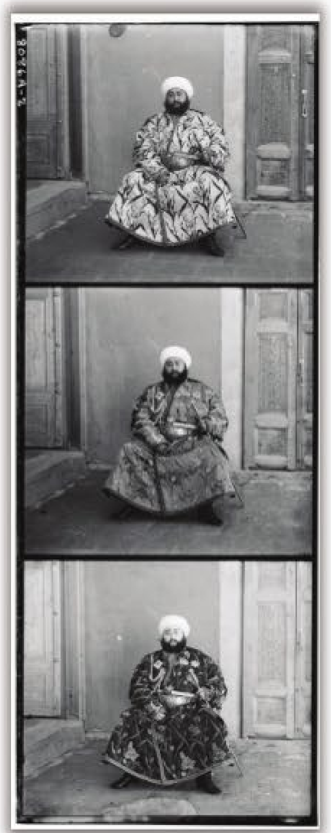


Images of the same scene captured using 3 different cameras with identical settings.

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

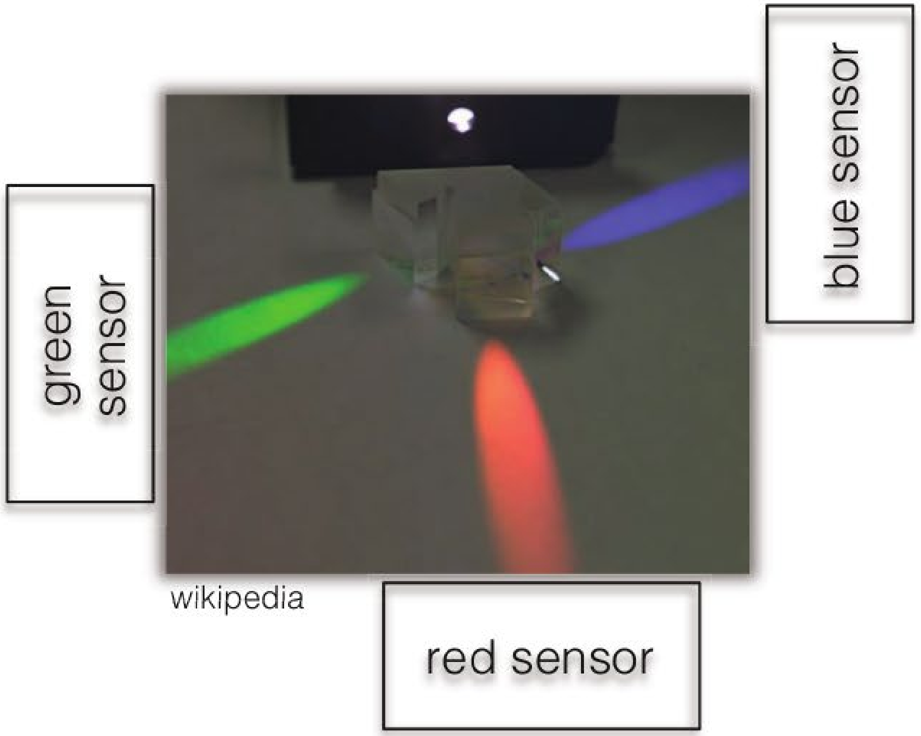
Aside: can you think of other ways to capture color?

field sequential



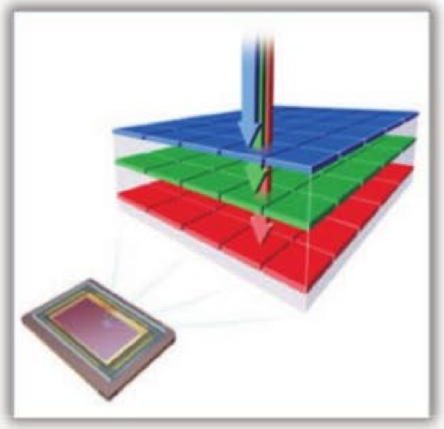
Prokudin-Gorsky

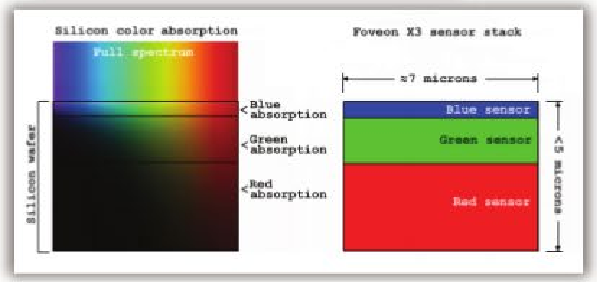
multiple sensors



wikipedia

vertically stacked





Foveon X3

[Slide credit: Gordon Wetzstein]

What does an imaging sensor do?

When the camera shutter opens, the sensor:

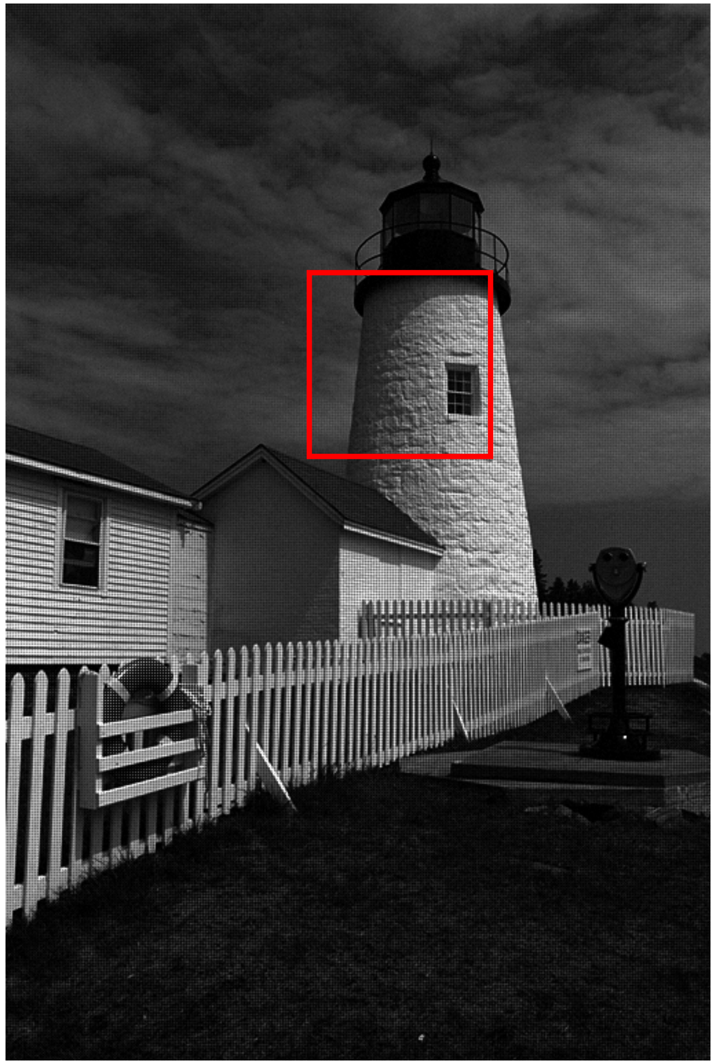
- at every photosite, converts incident photons into electrons $\boxed{\text{using mosaic's SSF}}$
- stores electrons into the photosite's potential well while it is not full

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

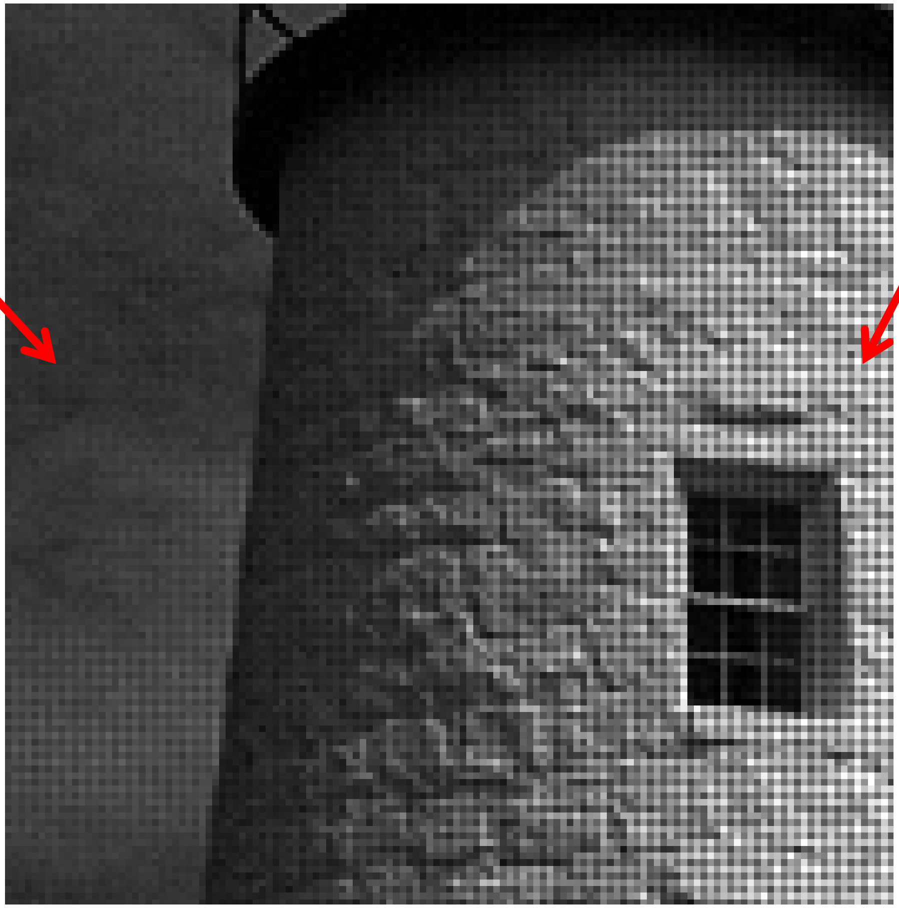
- reads out photosites' wells, row-by-row, and converts them to analog signals
- applies a (possibly non-uniform) gain to these analog signals
- converts them to digital signals
- corrects non-linearities

… and finally returns an image.

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



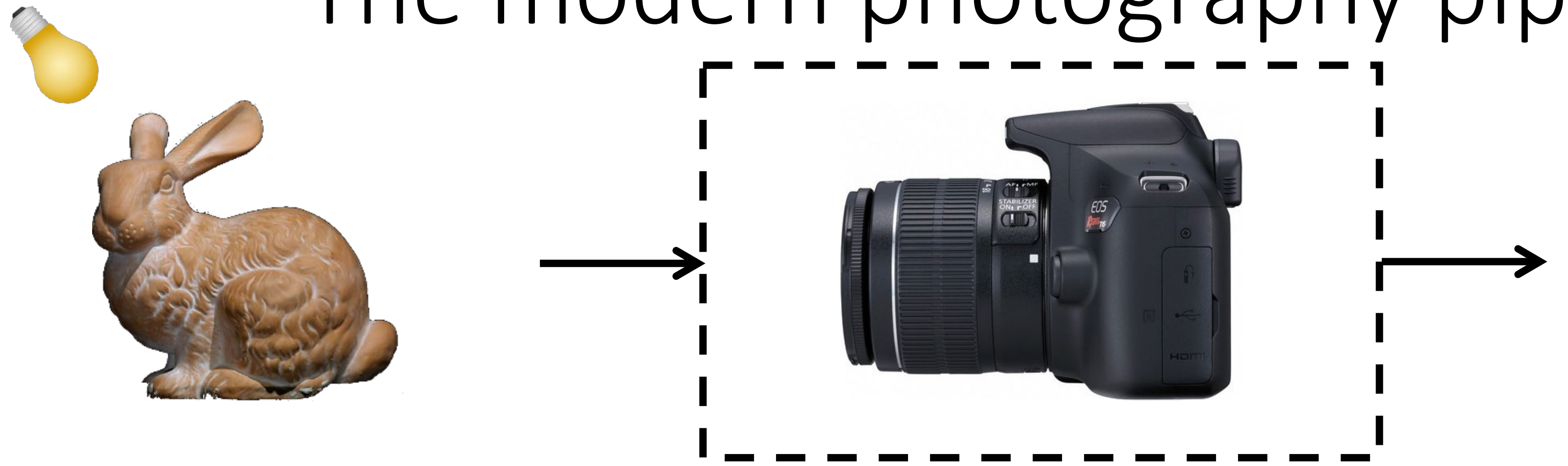
lots of noise



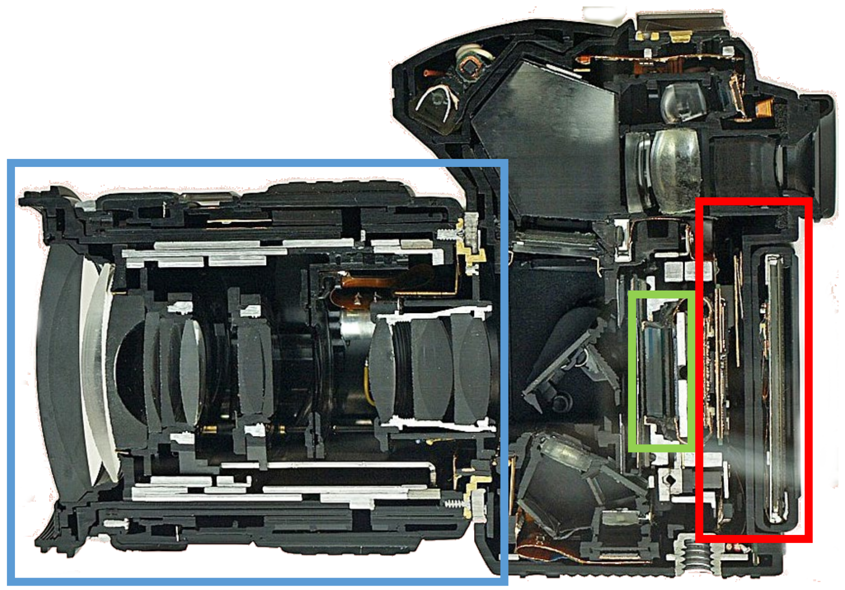
mosaicking artifacts

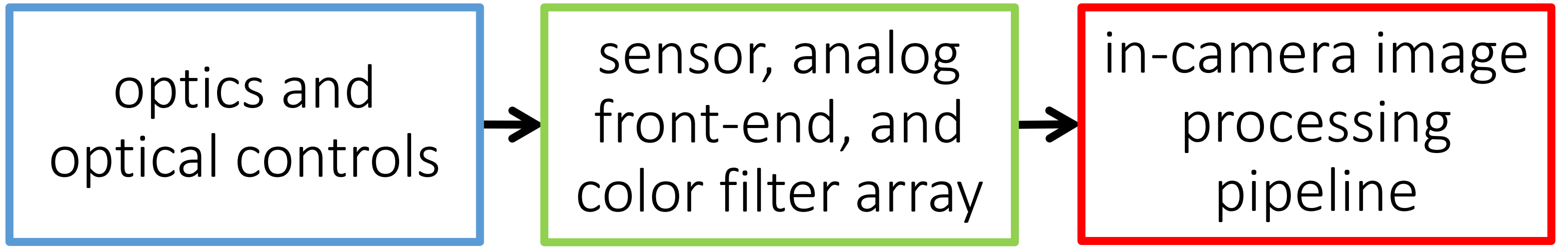
- Kind of disappointing.
- We call this the *RAW* image.

The modern photography pipeline



post-capture processing
(lectures 5-10)



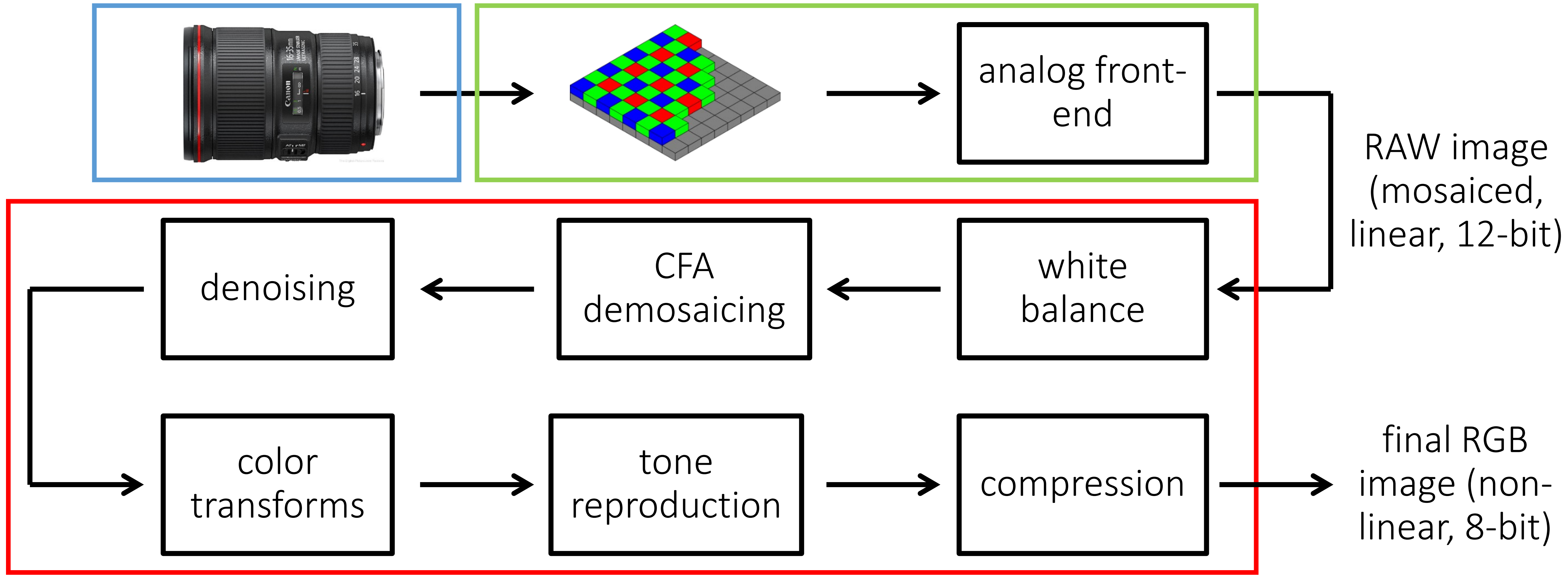

optics and optical controls
sensor, analog front-end, and color filter array
in-camera image processing pipeline

(lectures 2-3, 11-20) (today, lecture 23) (today)

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

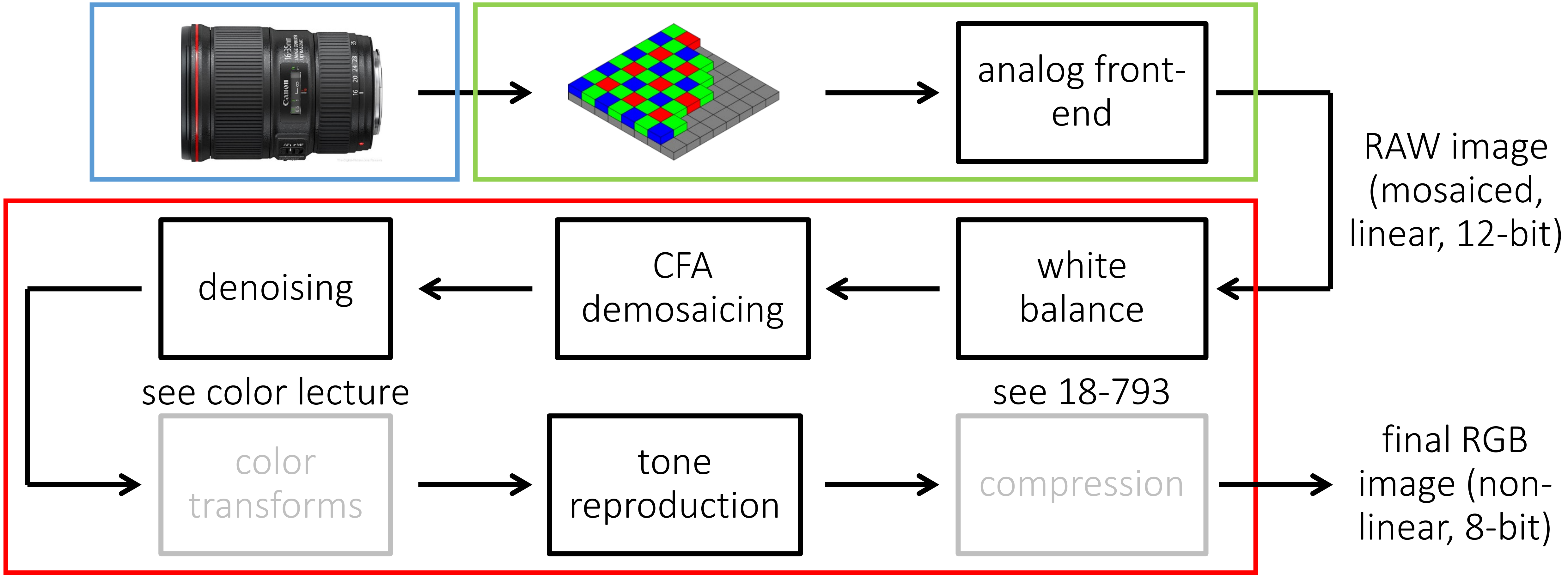
- Sometimes the term *image signal processor* (ISP) is used to refer to the image processing pipeline itself.

- 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*.

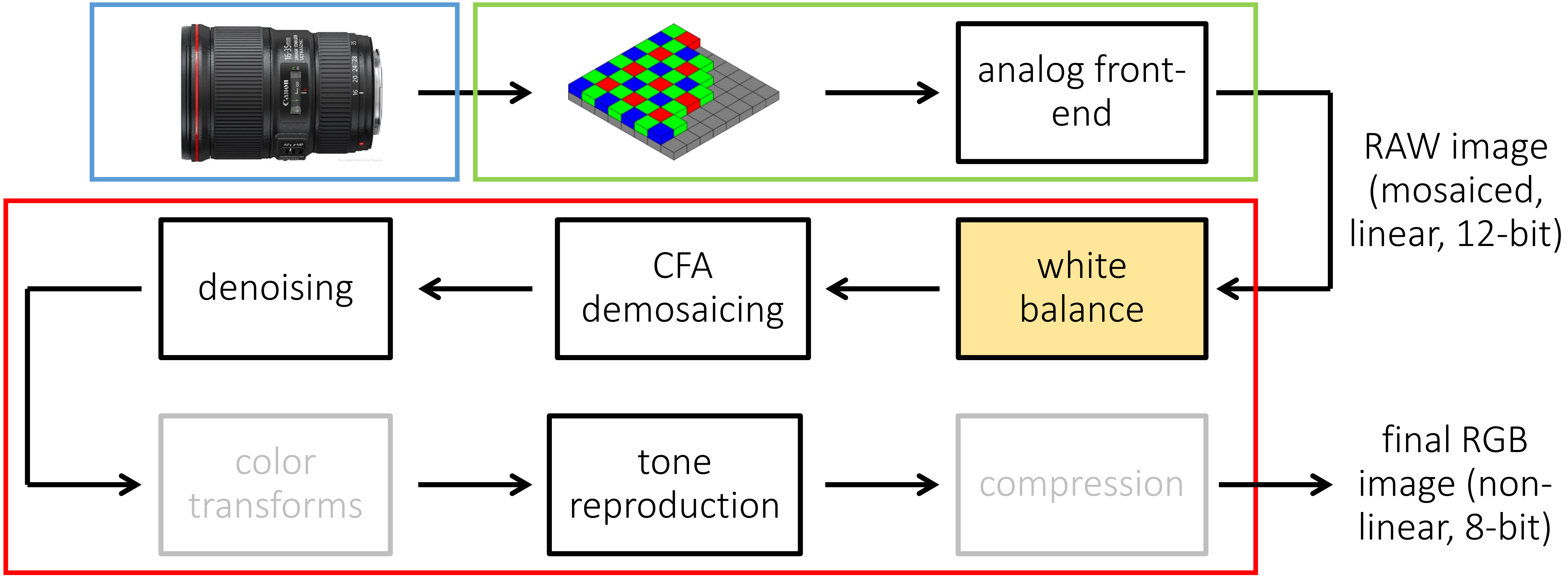
The (in-camera) image processing pipeline

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

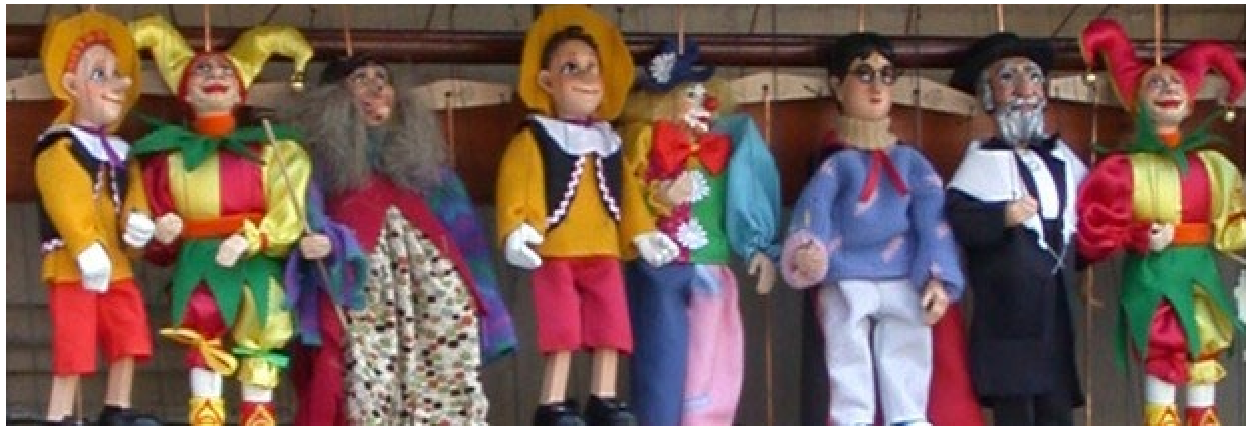
Human visual system has *chromatic adaptation*:

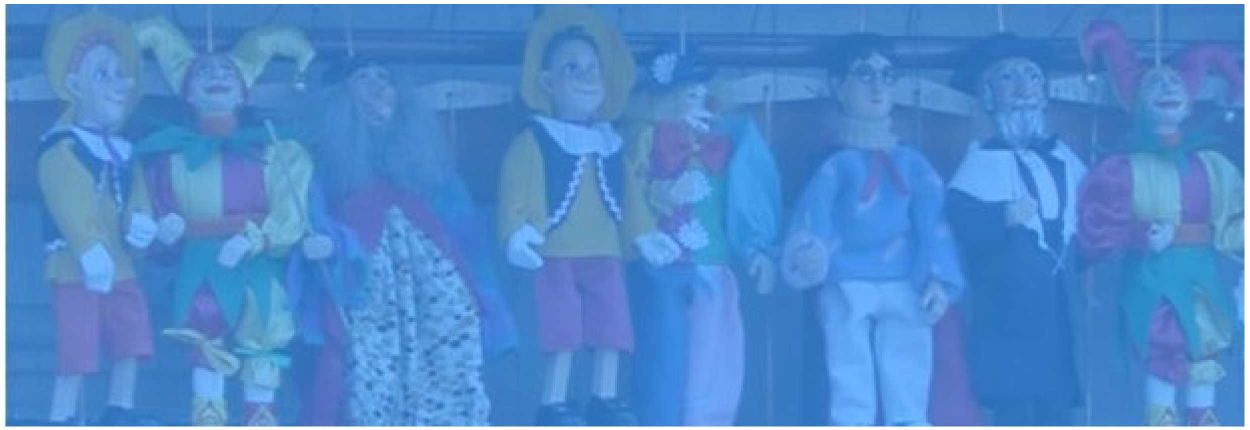
- We can perceive white (and other colors) correctly under different light sources.

[Slide credit: Todd Zickler]

White balancing

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





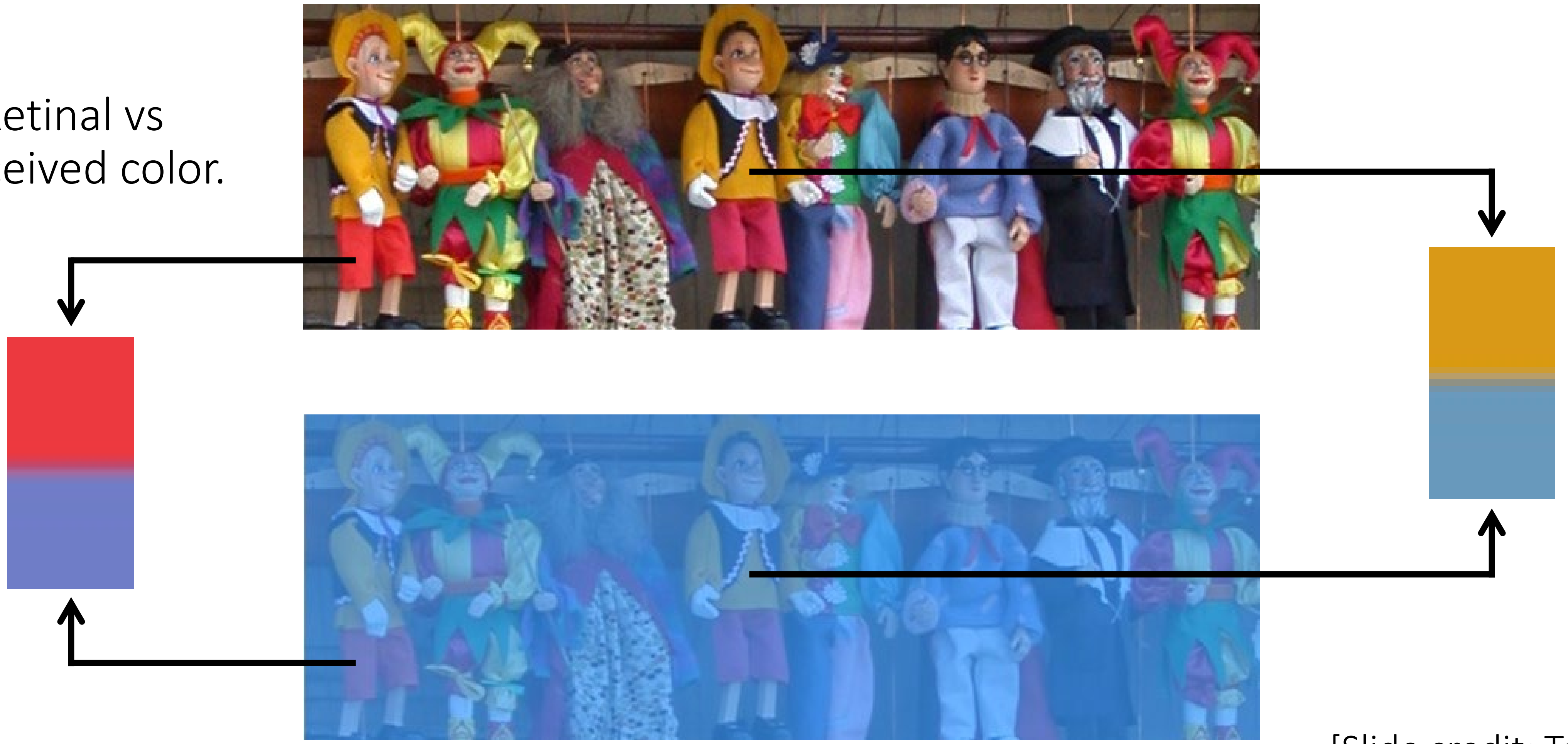
[Slide credit: Todd Zickler]

White balancing

Human visual system has *chromatic adaptation*:

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

Retinal vs perceived color.



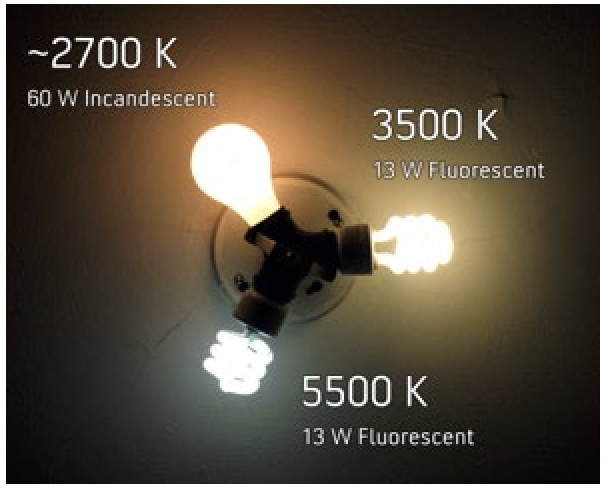
[Slide credit: Todd Zickler]

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

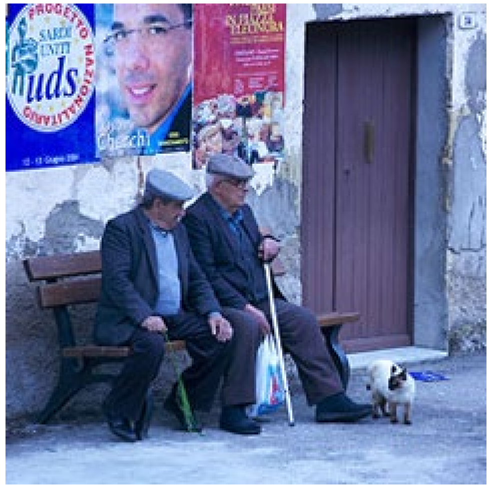


image captured under fluorescent

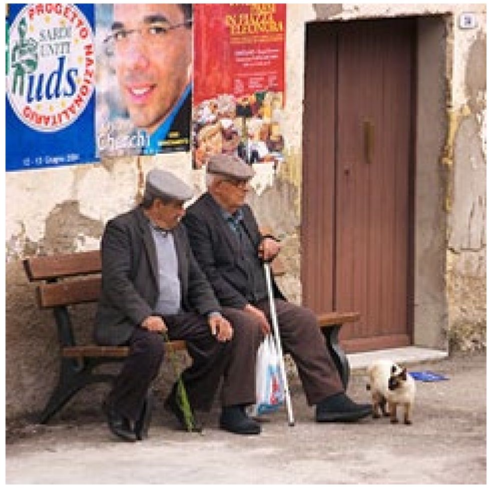
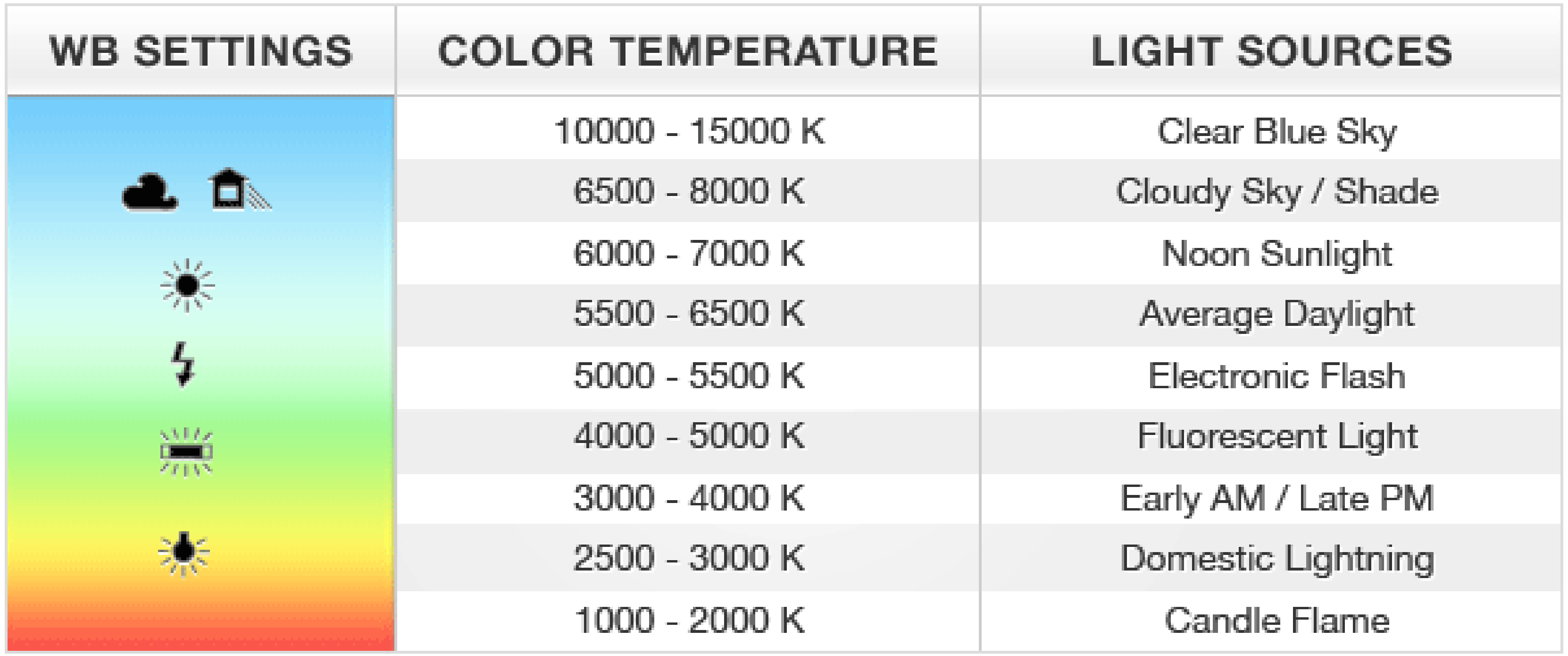


image white-balanced to daylight

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.

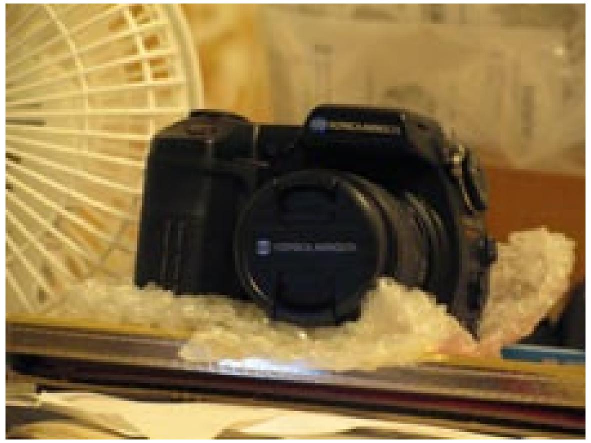


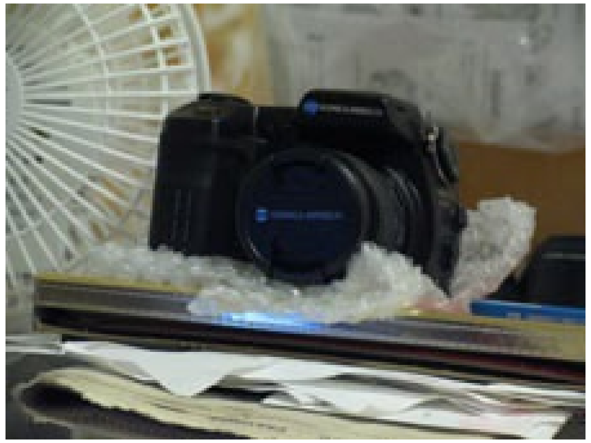
WB SETTINGS	COLOR TEMPERATURE	LIGHT SOURCES
	10000 - 15000 K	Clear Blue Sky
☁ ⌂	6500 - 8000 K	Cloudy Sky / Shade
☀	6000 - 7000 K	Noon Sunlight
	5500 - 6500 K	Average Daylight
ϟ	5000 - 5500 K	Electronic Flash
▬	4000 - 5000 K	Fluorescent Light
	3000 - 4000 K	Early AM / Late PM
☀	2500 - 3000 K	Domestic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

Manual vs automatic white balancing

Manual white balancing:

- Select a camera preset based on lighting.



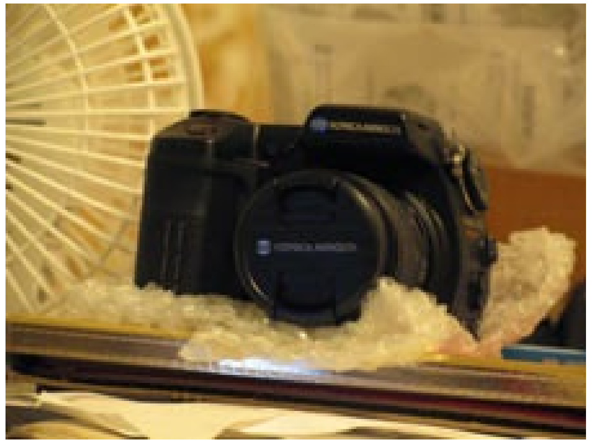


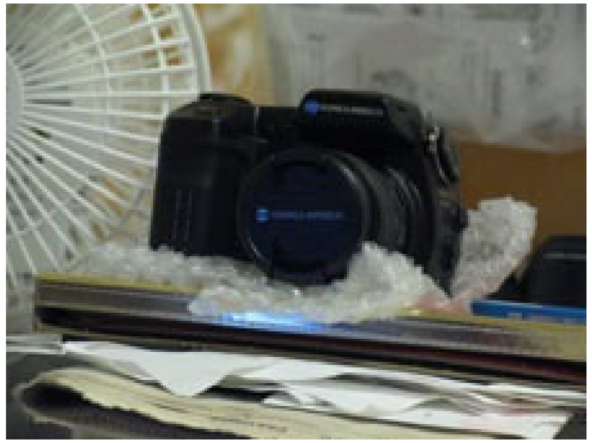
Can you think of any other way to do manual white balancing?

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.



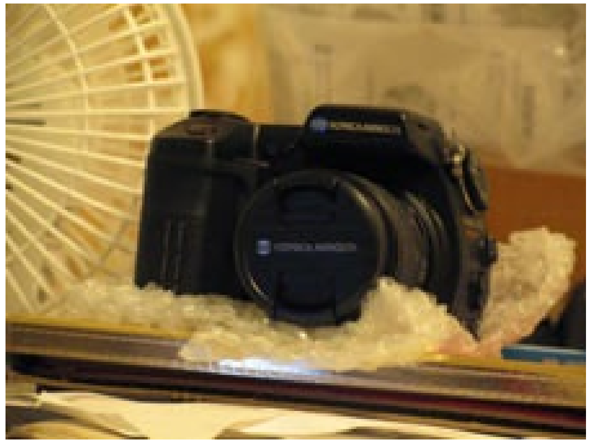


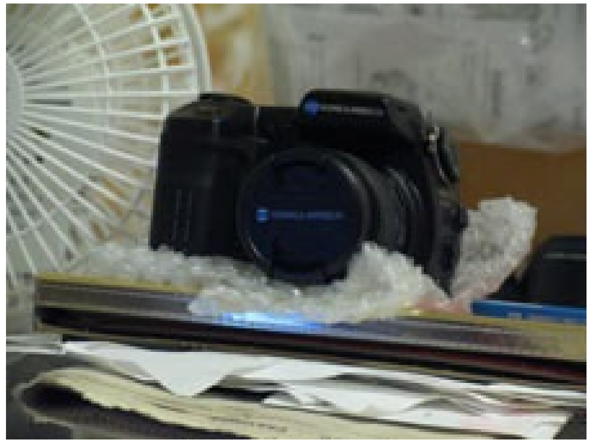
How can we do automatic white balancing?

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.

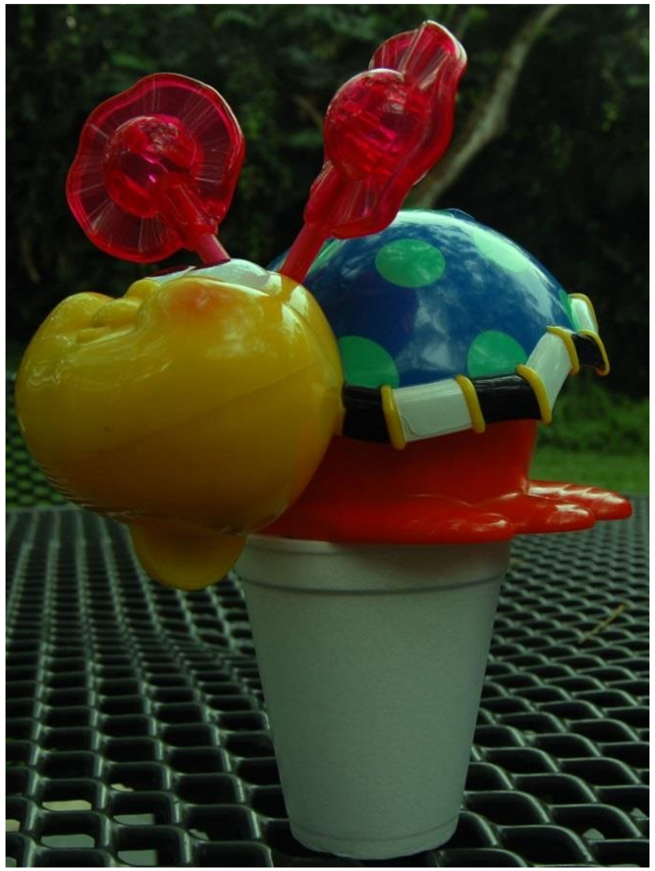
$$\text{white-balanced RGB} \rightarrow \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} G_{avg}/R_{avg} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & G_{avg}/B_{avg} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \leftarrow \text{sensor RGB}$$

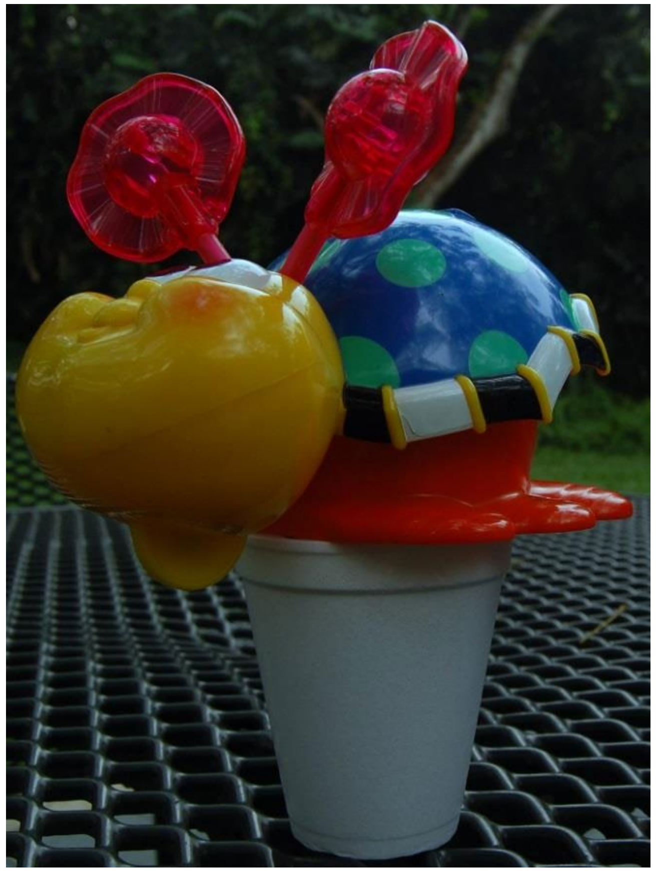
White world assumption:

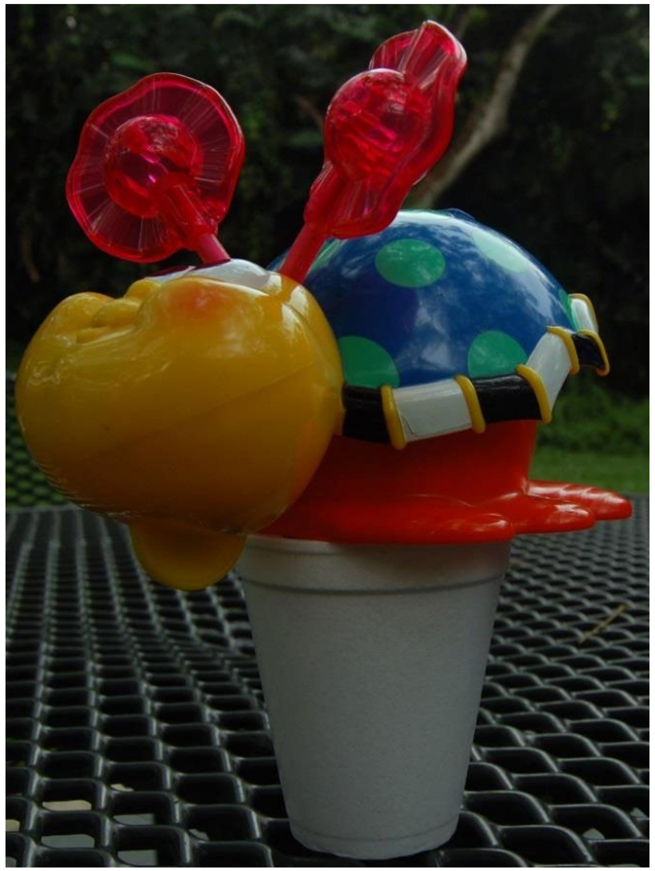
- Compute per-channel maximum.
- Normalize each channel by its maximum.
- Normalize by green channel maximum.

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

Automatic white balancing example

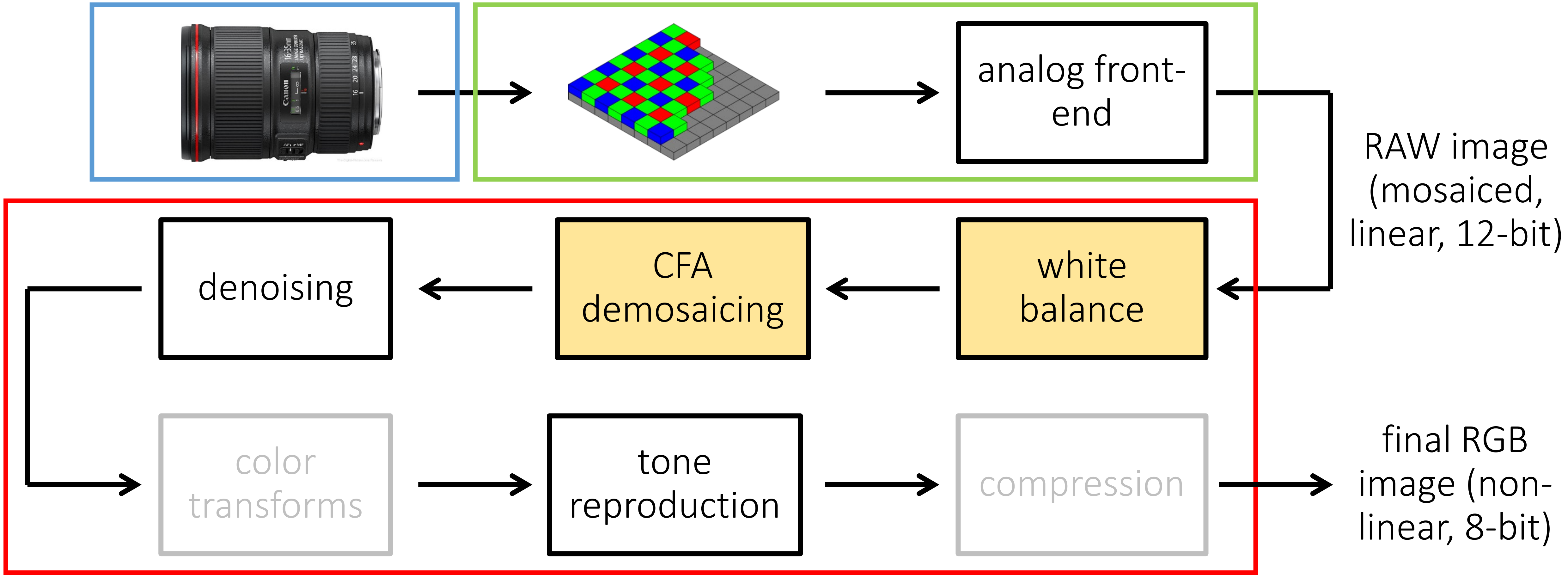

input image


grey world


white world

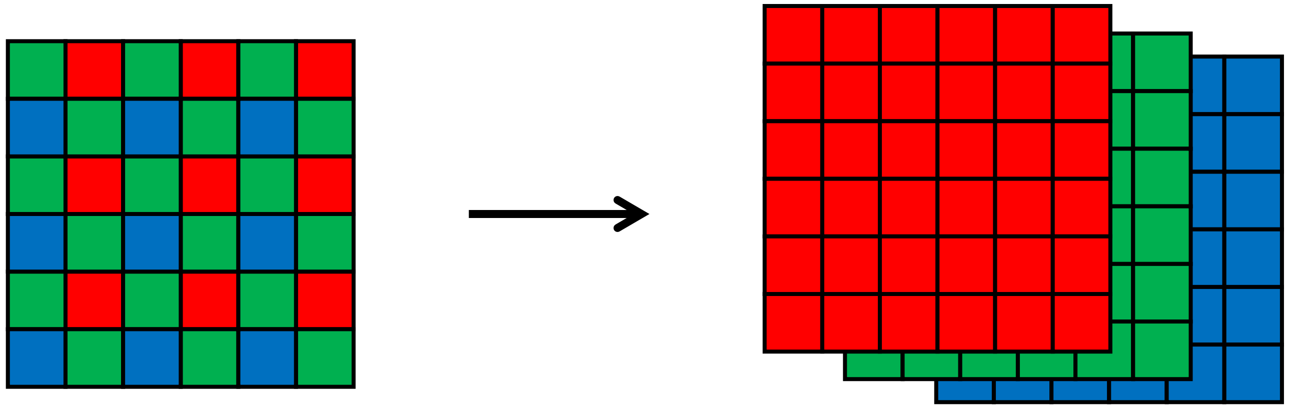
The (in-camera) image processing pipeline

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

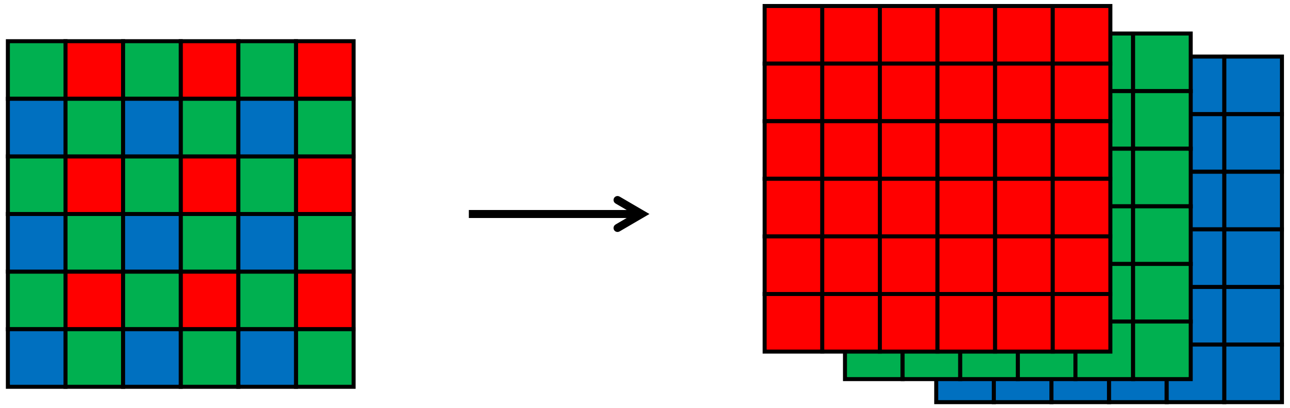
Produce full RGB image from mosaiced sensor output.



Any ideas on how to do this?

CFA demosaicing

Produce full RGB image from mosaiced sensor output.

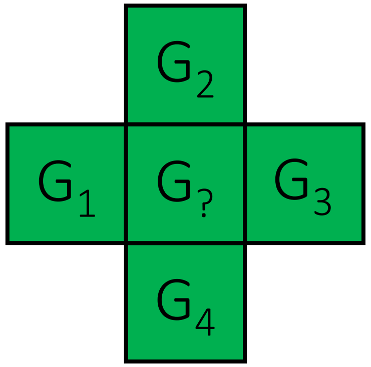


Interpolate from neighbors:

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

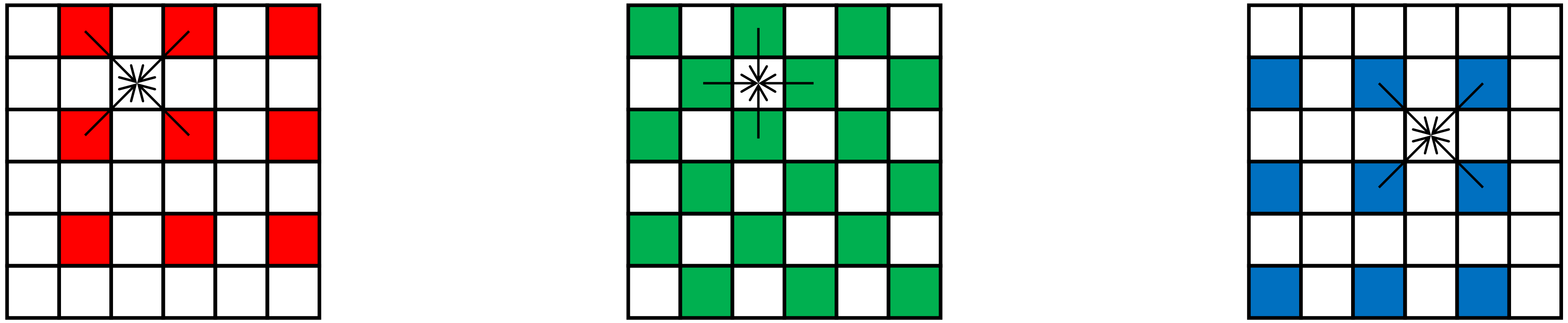
Demosaicing by bilinear interpolation

Bilinear interpolation: Simply average your 4 neighbors.



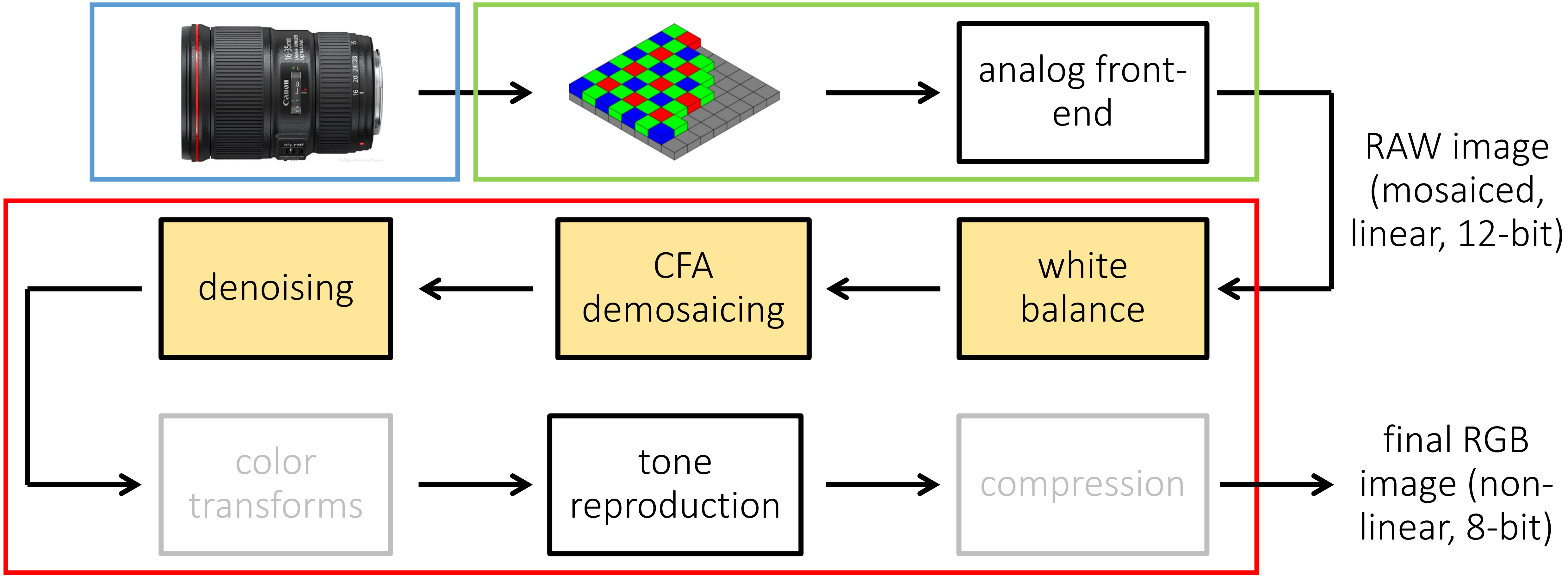
$$\boxed{G_?} = \frac{\boxed{G_1} + \boxed{G_2} + \boxed{G_3} + \boxed{G_4}}{4}$$

Neighborhood changes for different channels:



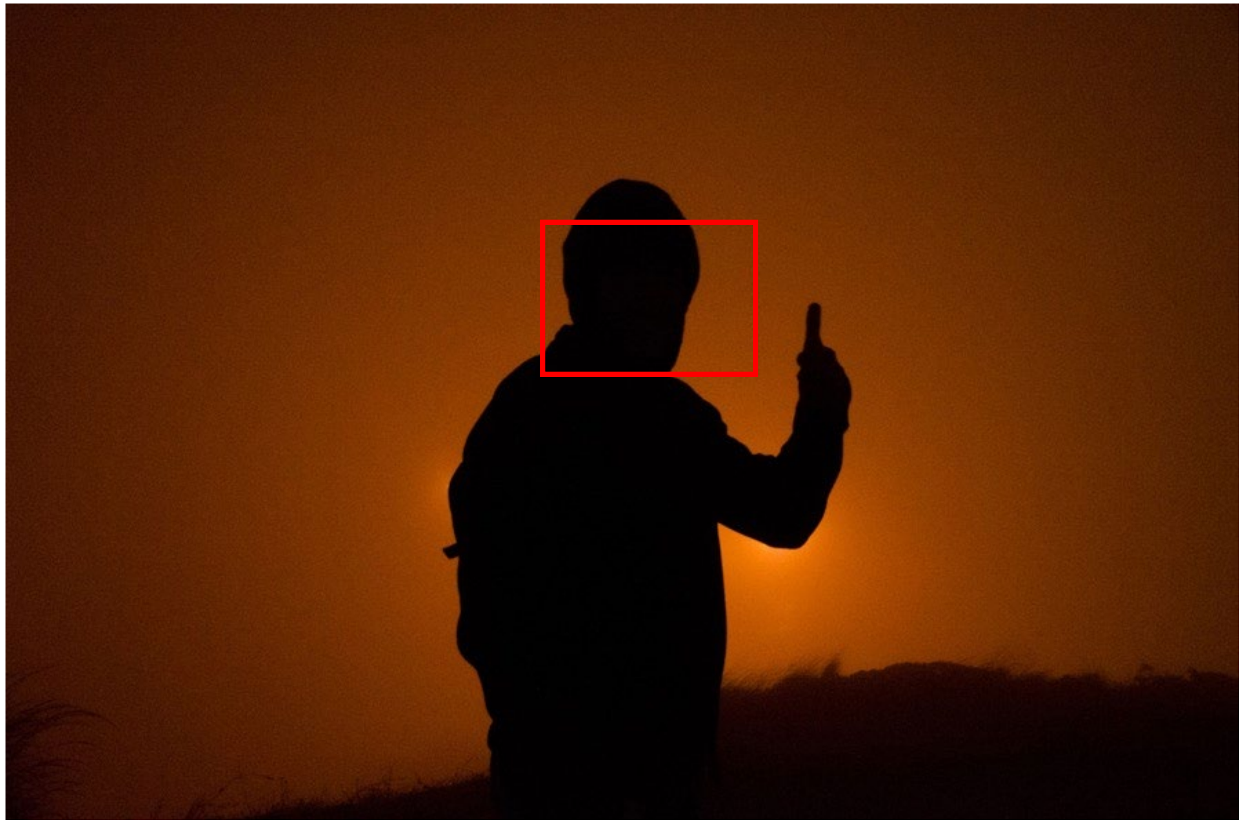
The (in-camera) image processing pipeline

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:

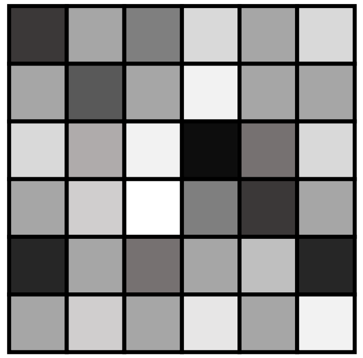
- Emitted electrons due to thermal activity (becomes worse as sensor gets hotter.)

3) Read noise:

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

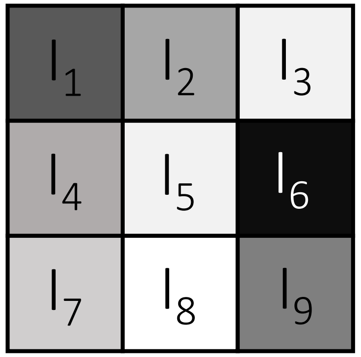
Bright scene and large pixels: photon shot noise is the main noise source.

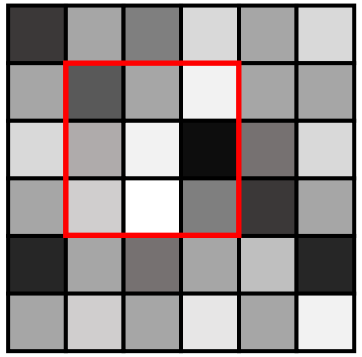
How to denoise?



How to denoise?

Look at the neighborhood around you.





- Mean filtering (take average):

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

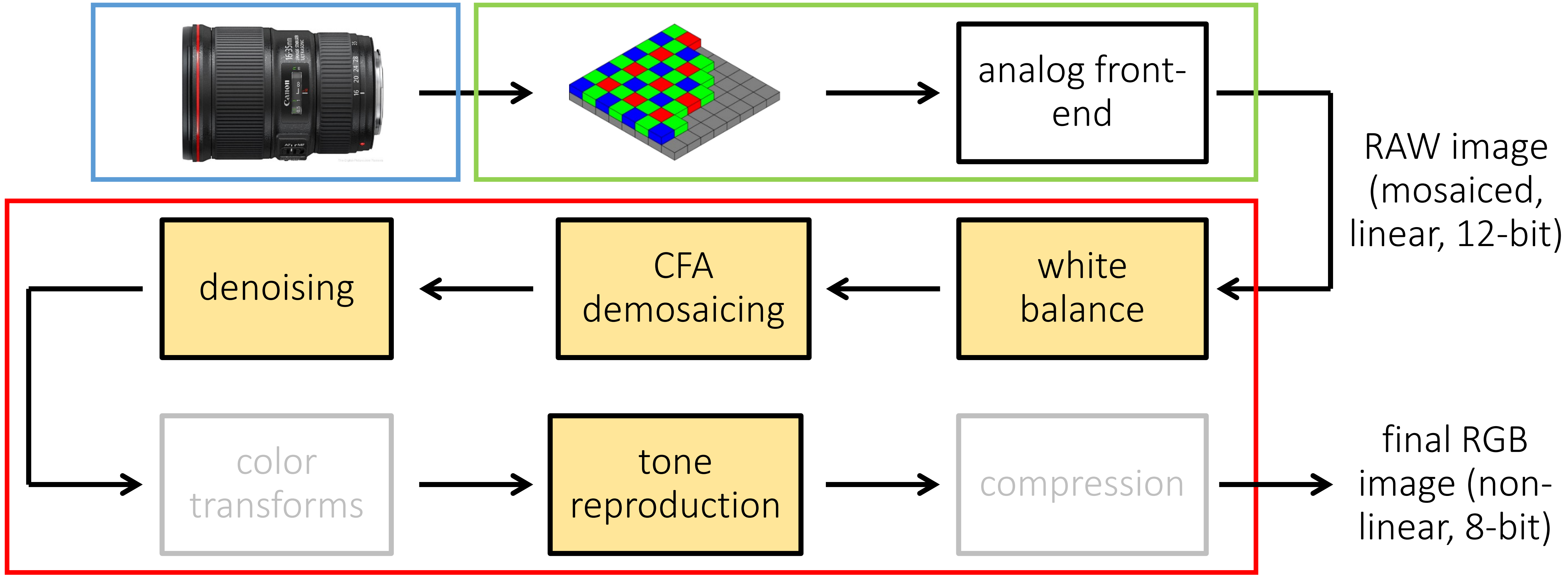
- Median filtering (take median):

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

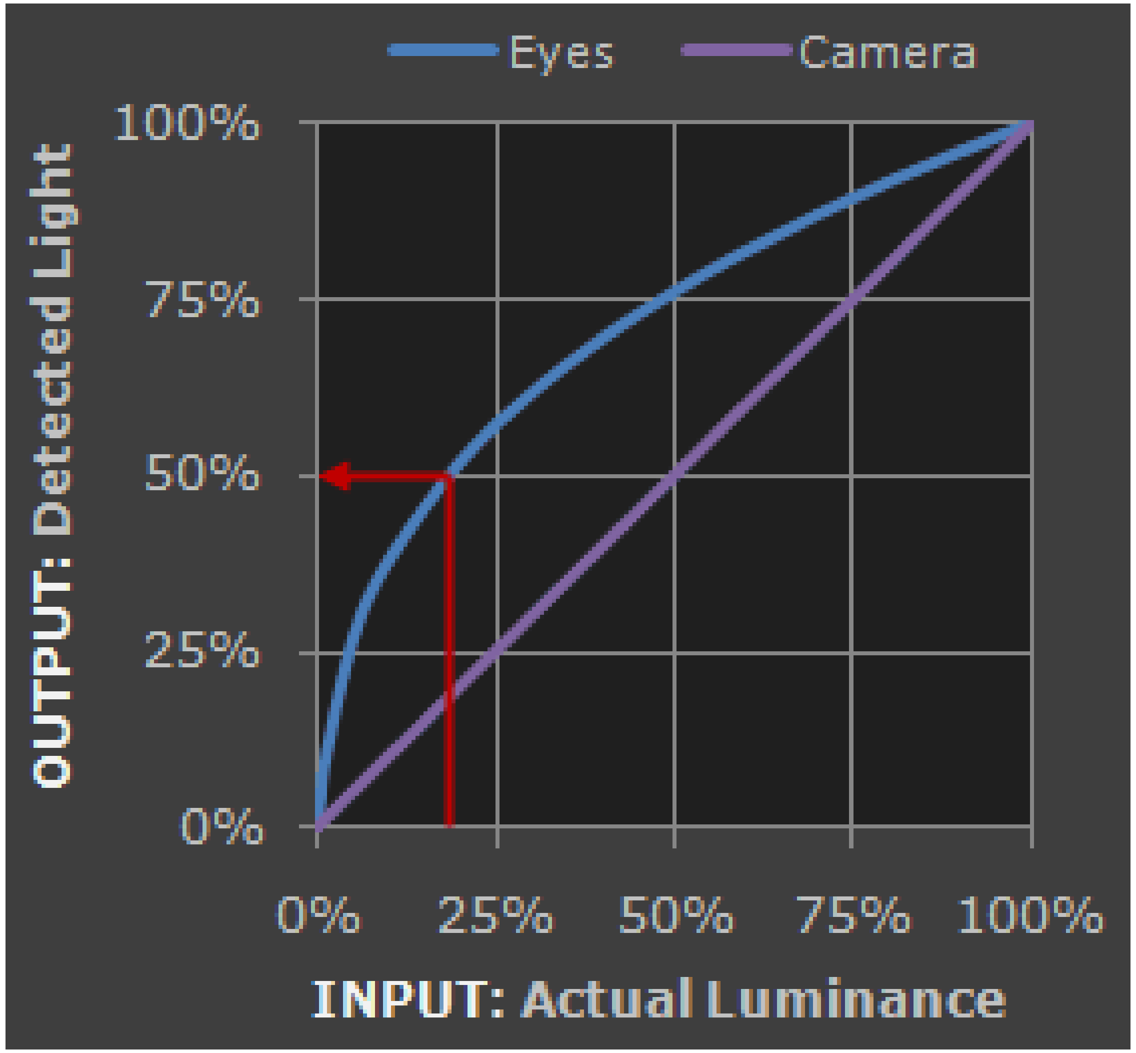
Large area of research. We will see some more about filtering in a later lecture.

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.

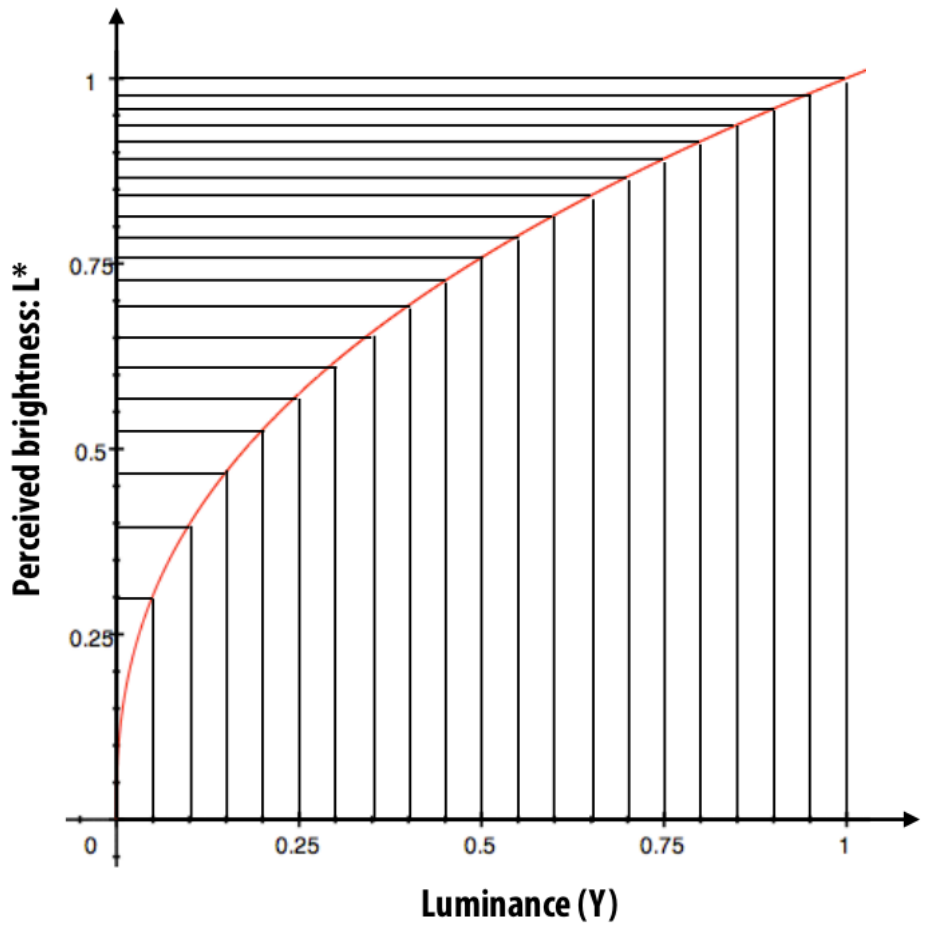
However, human-eye *perception* (perceived brightness) is *non-linear*:

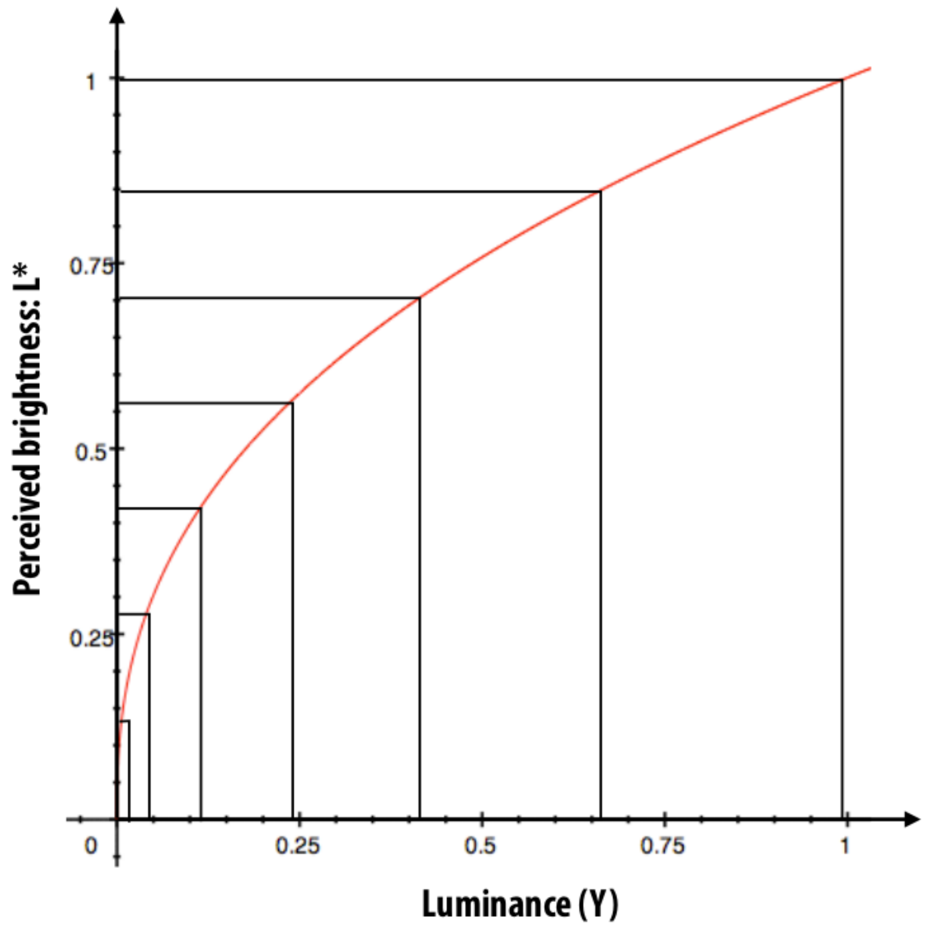
- More sensitive to dark tones.
- Approximately a Gamma function.

Gamma encoding

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

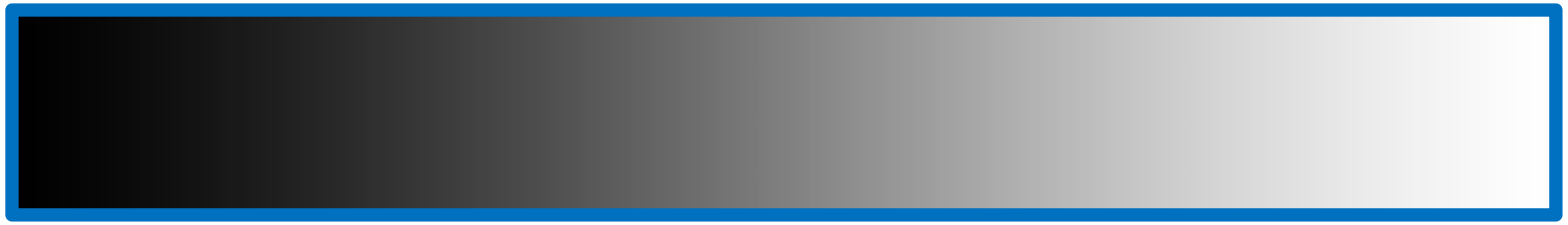
- Apply non-linear curve to use available bits to better encode the information human vision is more sensitive to.





Demonstration

original (8-bits, 256 tones)

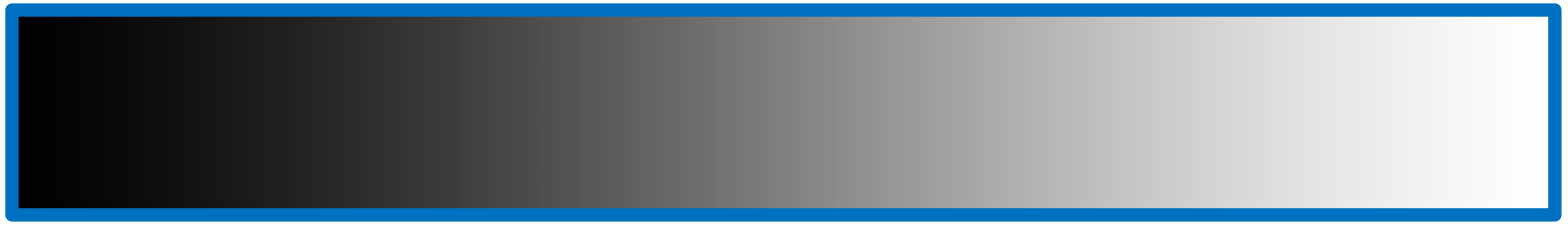


Can you predict what will happen if we linearly encode this tone range with only 5 bits?

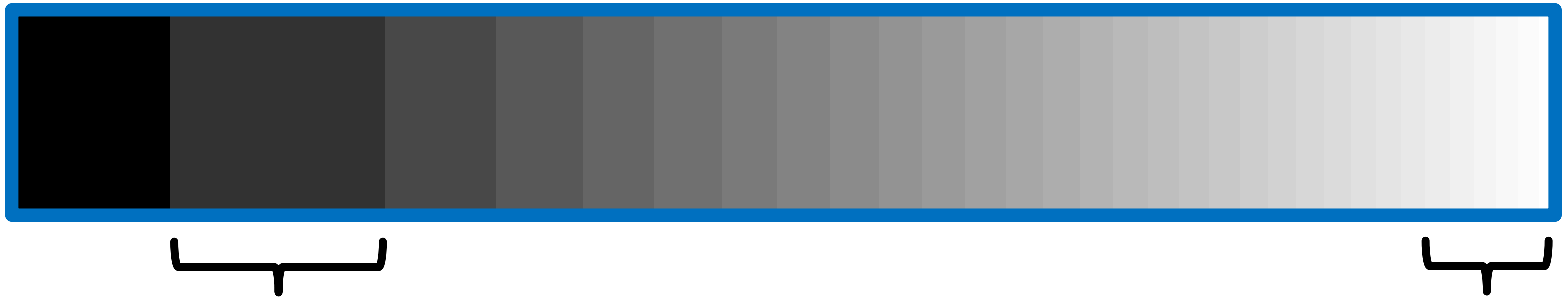
Can you predict what will happen if we gamma encode this tone range with only 5 bits?

Demonstration

original (8-bits, 256 tones)



linear encoding (5-bits, 32 tones)



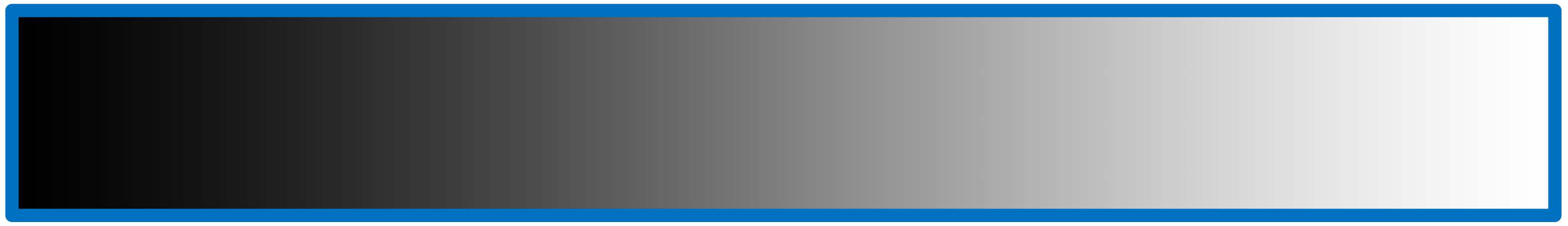
all of this range gets mapped to just one tone

all of these tones look the same

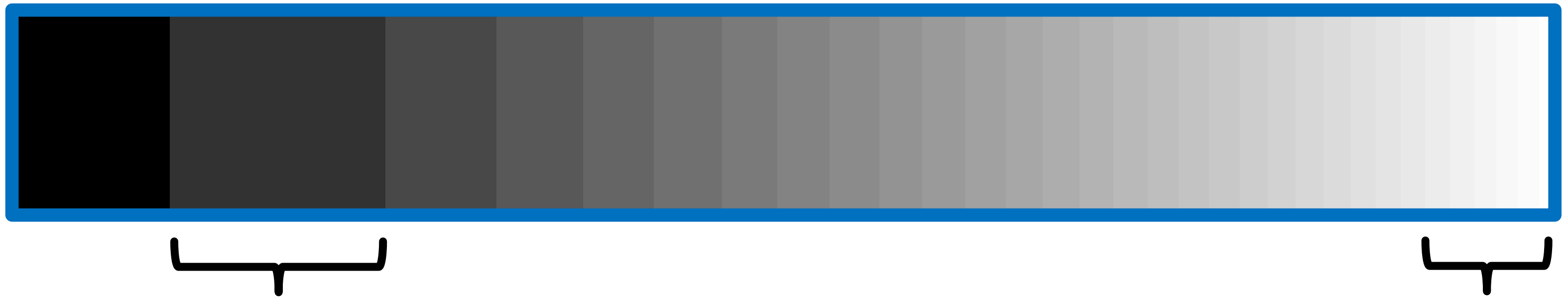
Can you predict what will happen if we gamma encode this tone range with only 5 bits?

Demonstration

original (8-bits, 256 tones)



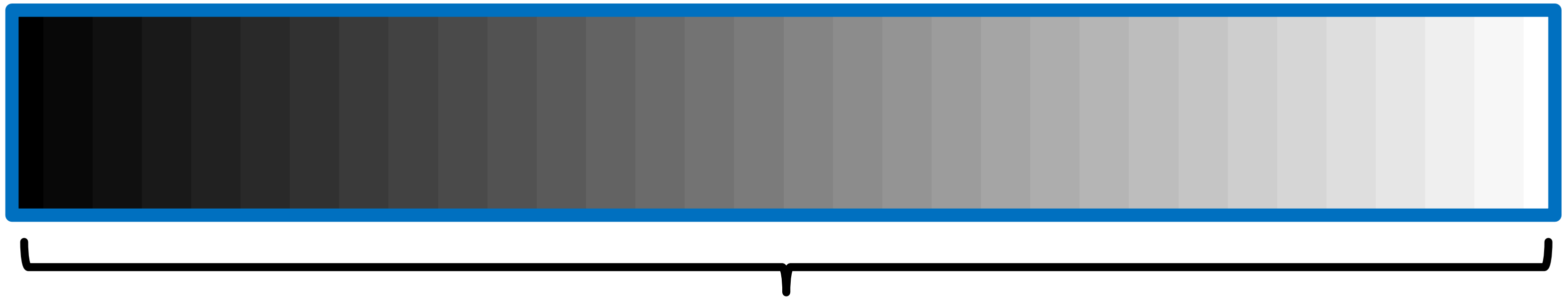
linear encoding (5-bits, 32 tones)



all of this range gets mapped to just one tone

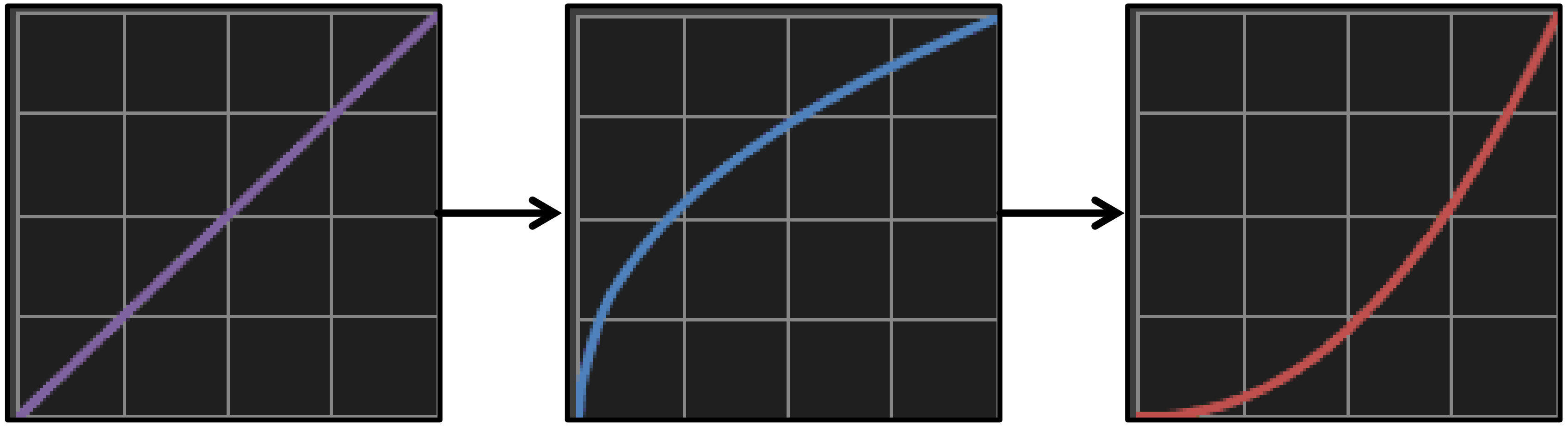
all of these tones look the same

gamma encoding (5-bits, 32 tones)



tone encoding becomes a lot more perceptually uniform

Tone reproduction pipeline

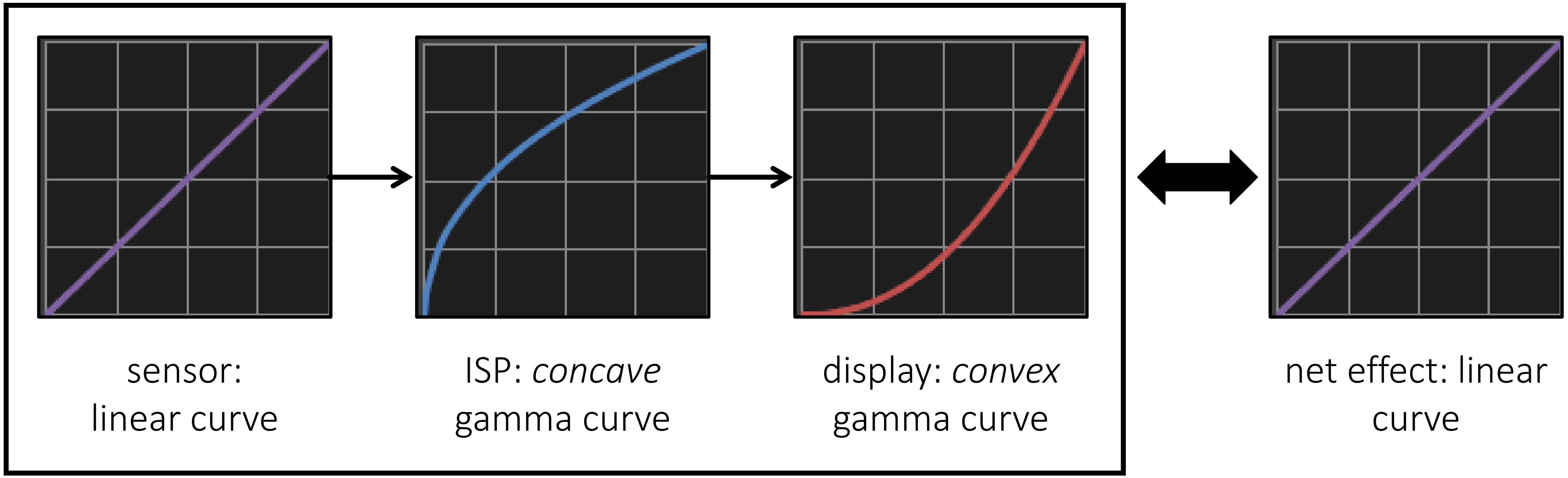


sensor:
linear curve

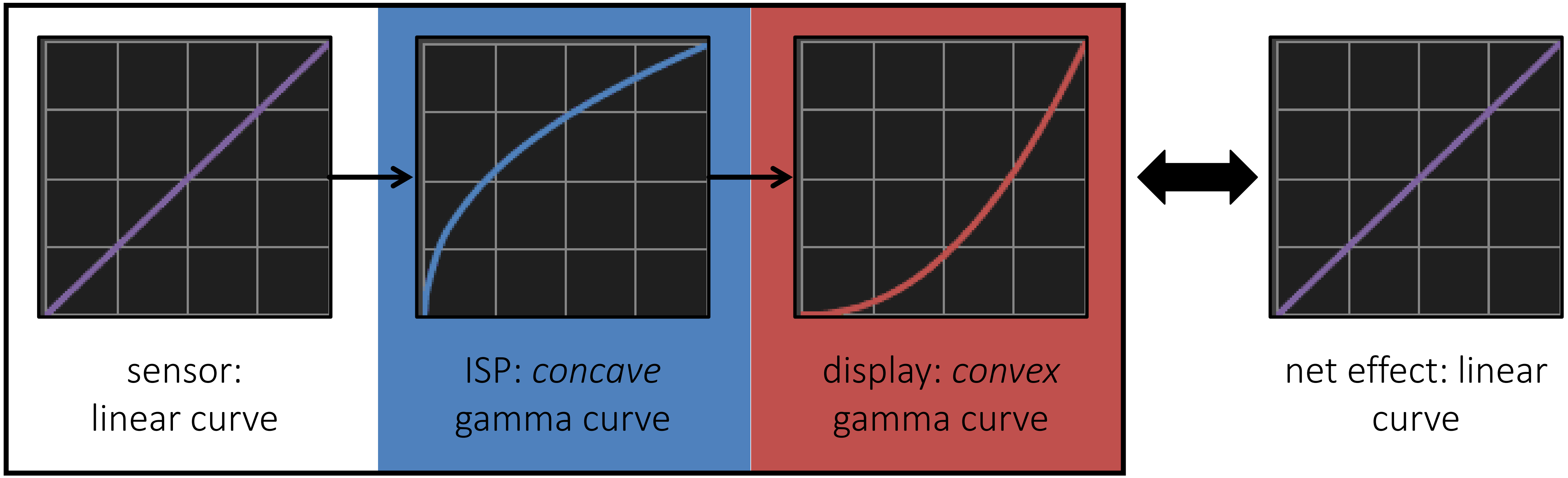
ISP: *concave*
gamma curve

display: *convex*
gamma curve

Tone reproduction pipeline



Tone reproduction pipeline



sensor:
linear curve

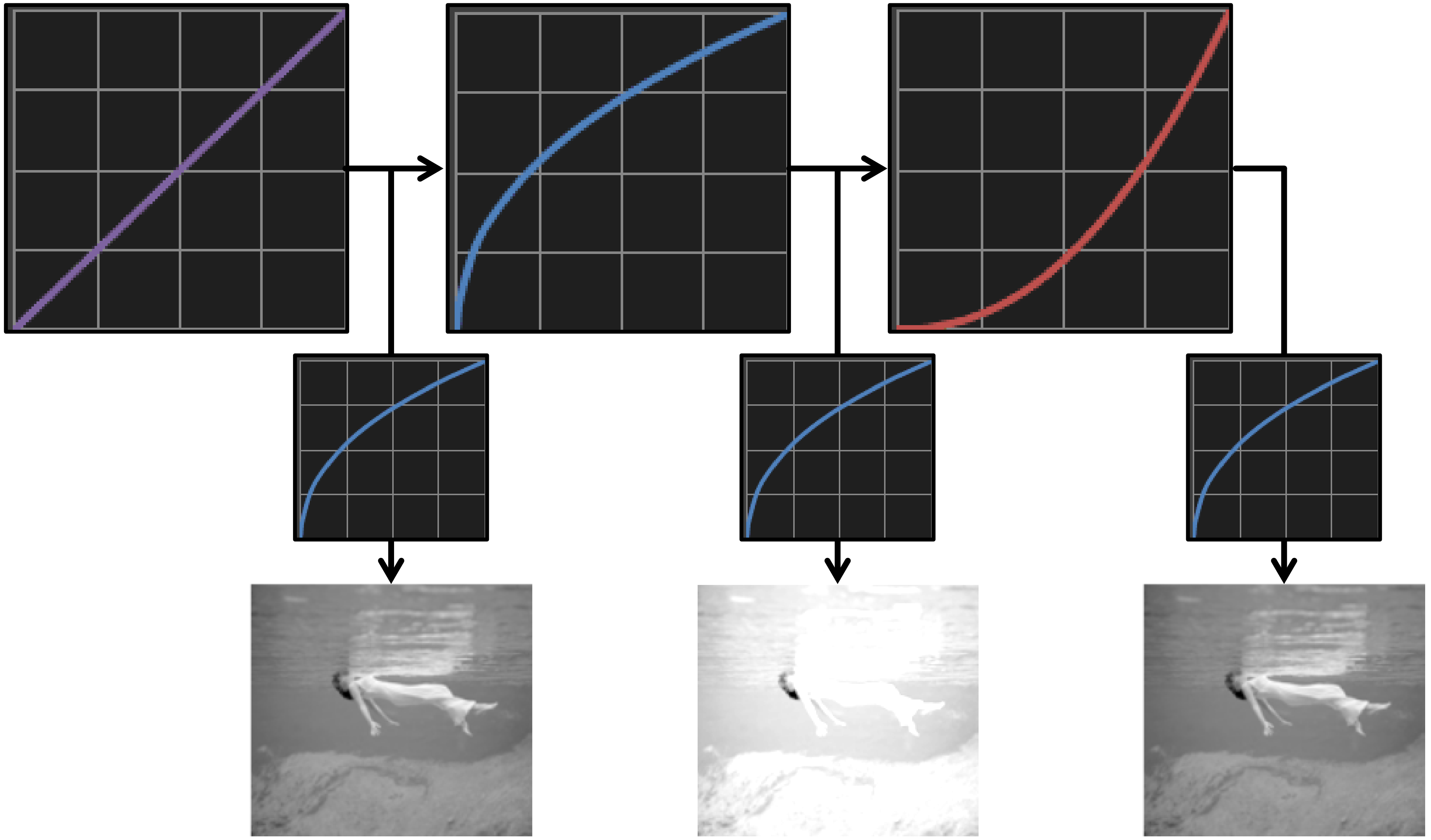
ISP: *concave*
gamma curve

display: *convex*
gamma curve

net effect: linear
curve

gamma encoding gamma correction

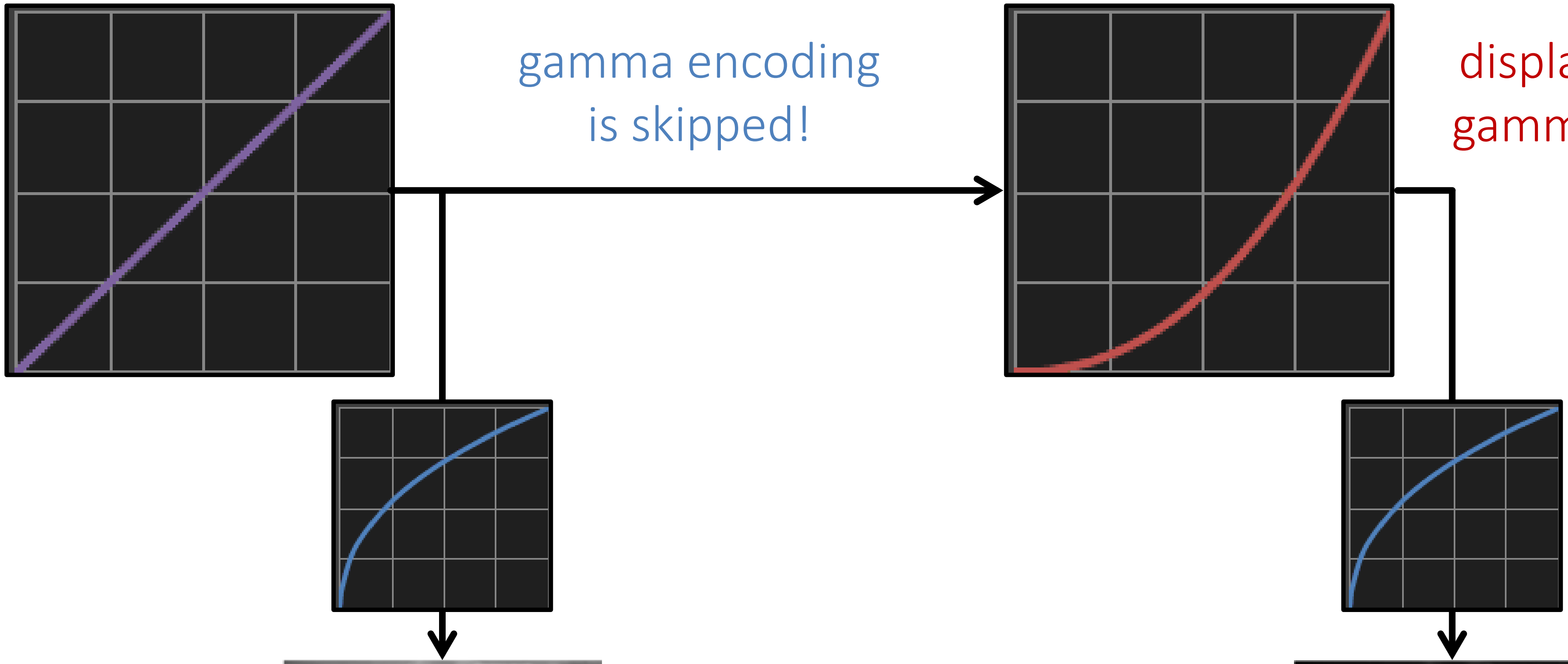
Tone reproduction pipeline



human visual system: *concave* gamma curve

image a human would see at different stages of the pipeline

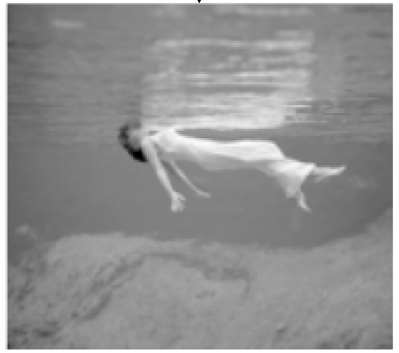
RAW pipeline



gamma encoding is skipped!

display still applies gamma correction!

human visual system: *concave* gamma curve



RAW image appears very dark! (Unless you are using a RAW viewer)

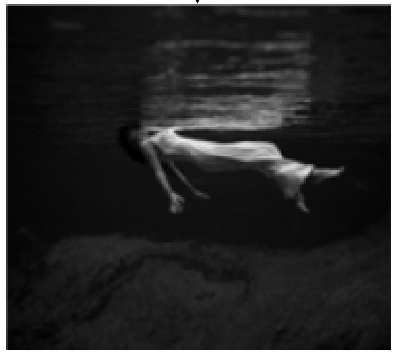


image a human would see at different stages of the pipeline

Historical note

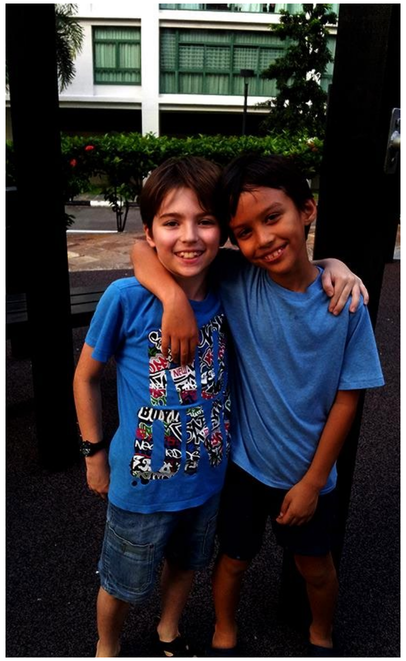
- CRT displays used to have a response curve that was (almost) exactly equal to the inverse of the human sensitivity curve. Therefore, displays could skip gamma correction and display directly the gamma-encoded images.

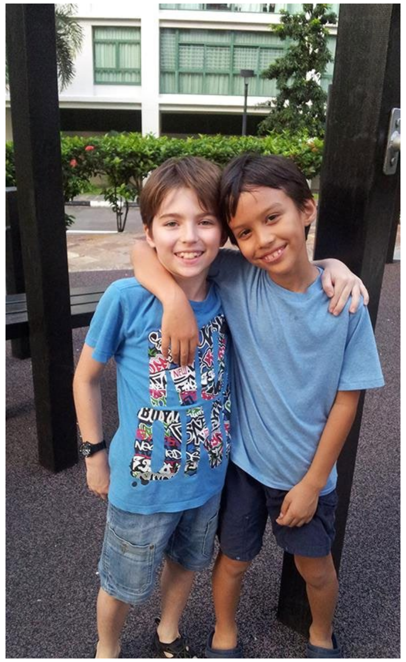
- It is sometimes mentioned that gamma encoding is done to undo the response curve of a display. This used to (?) be correct, but it is not true nowadays. Gamma encoding is performed to ensure a more perceptually-uniform use of the final image’s 8 bits.

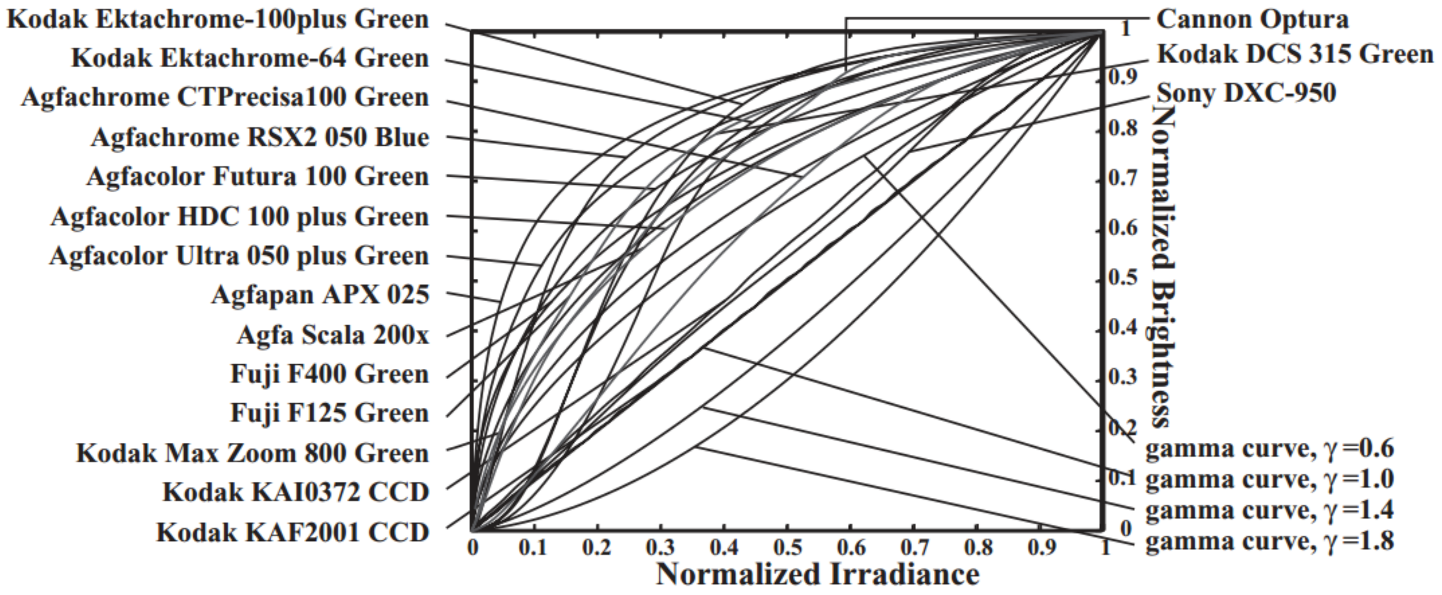
Gamma encoding curves

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 γ = 1 / 2.2.


before gamma

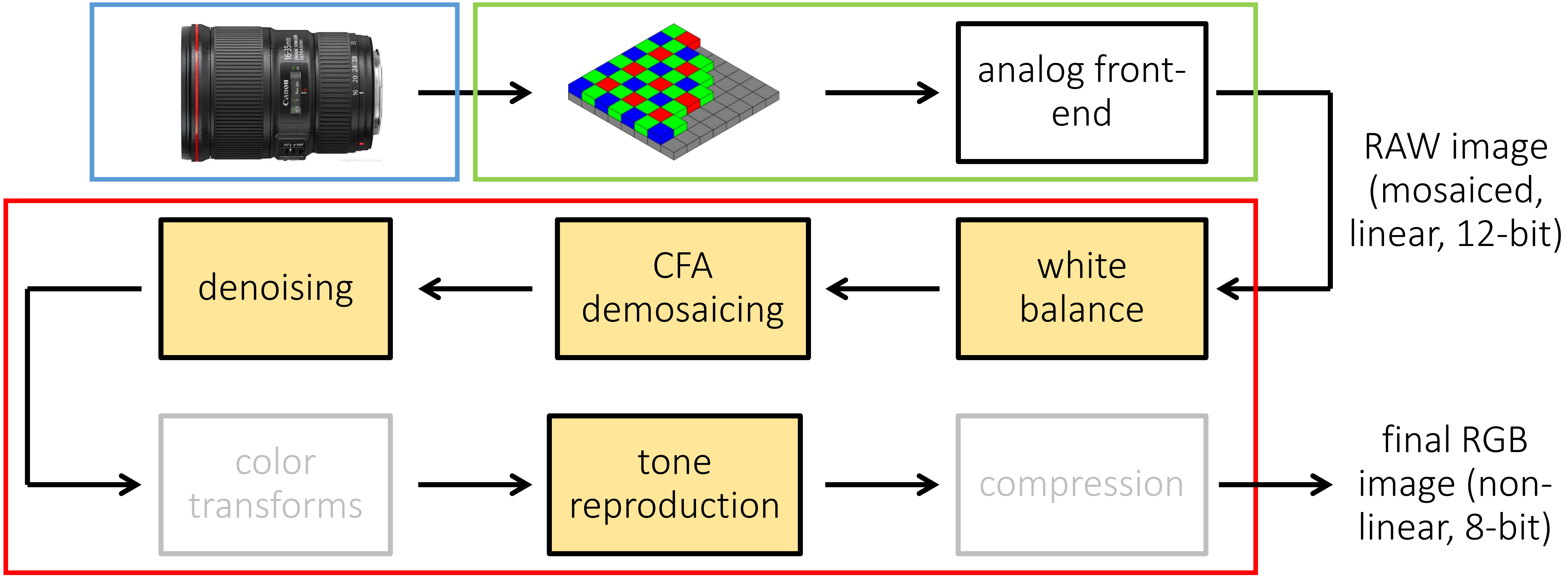

after gamma



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?

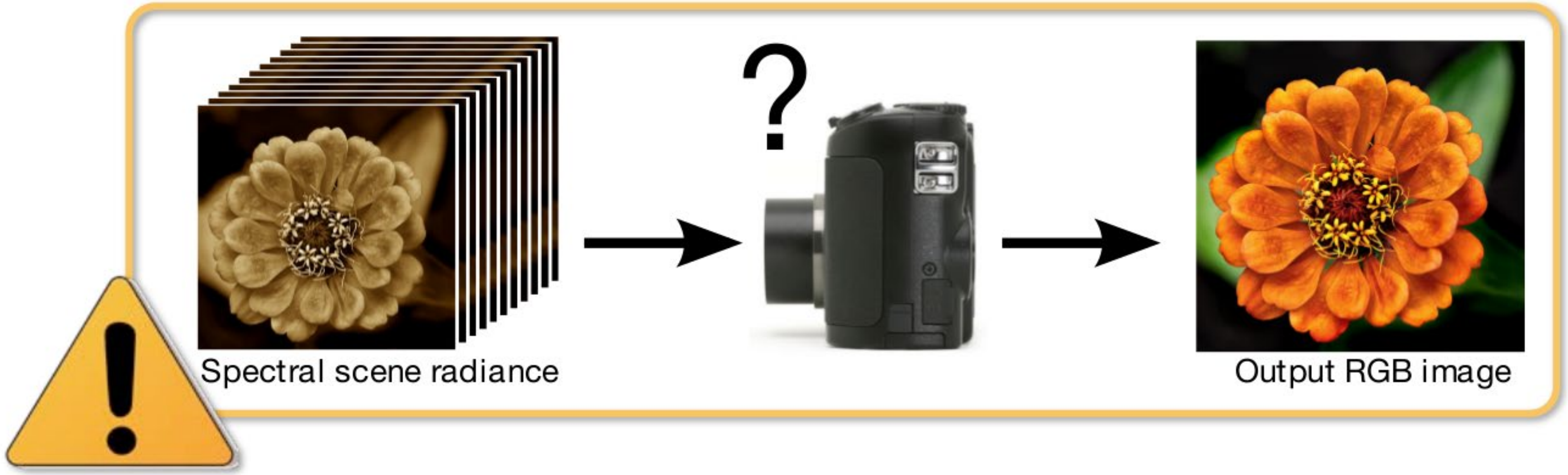
Nope, tough luck.

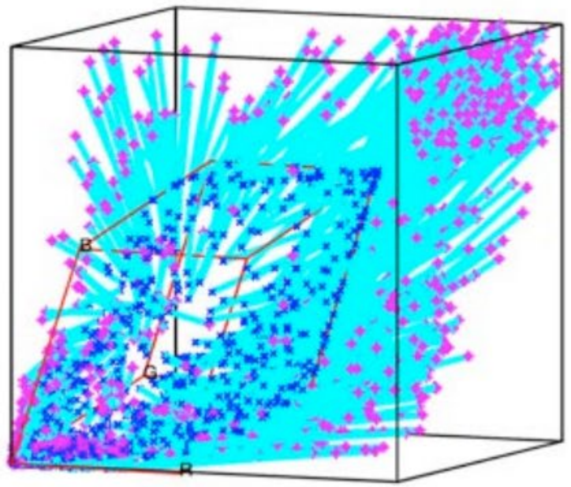
- The image processing pipeline is lossy: After all the steps, information about the original image is lost.

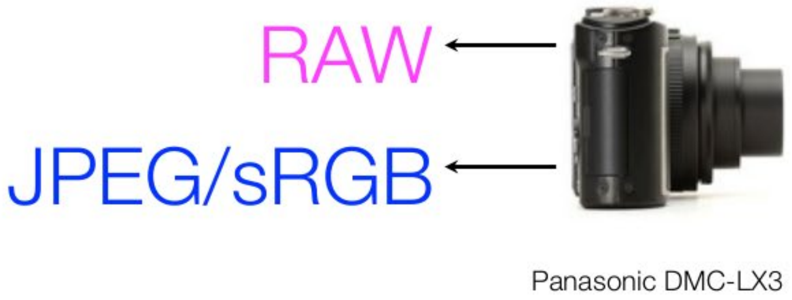
- Sometimes we may be able to reverse a camera's image processing pipeline *if we know exactly what it does* (e.g., by using information from other similar RAW images).

- The conversion of PNG/JPG back to RAW is known as "derendering" and is an active research area.

Derendering







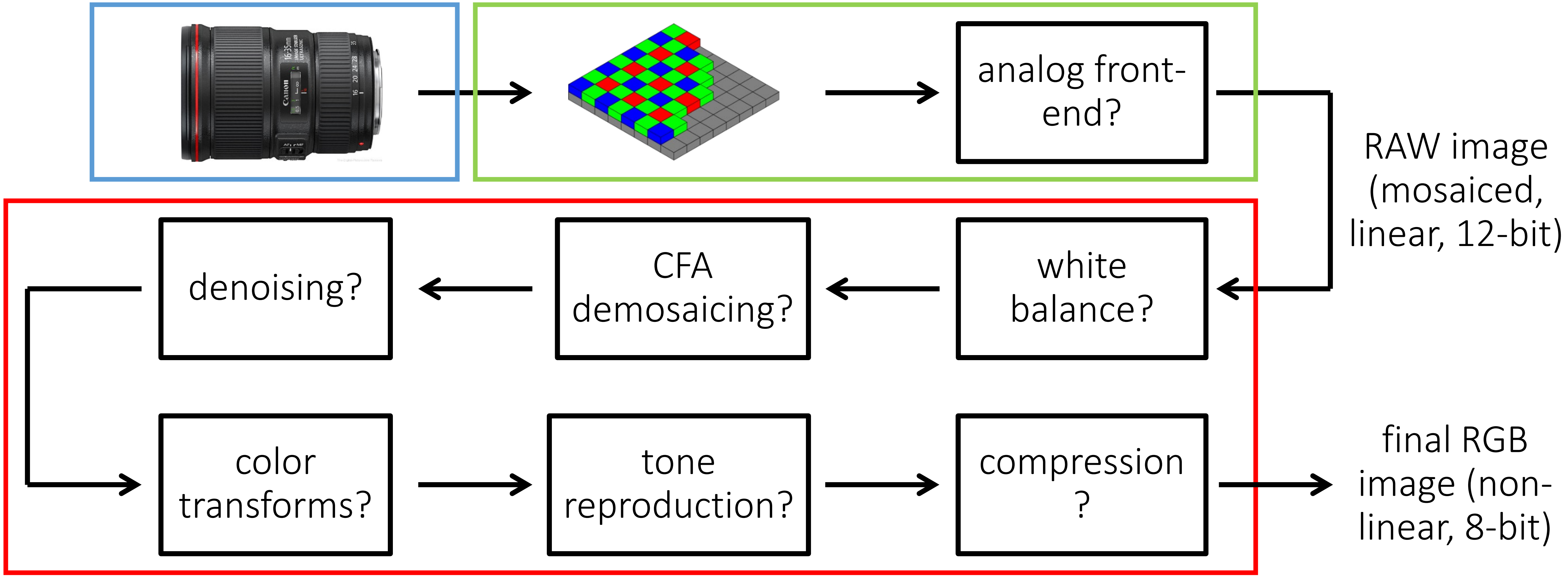
Why did you use italics in the previous slide?

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

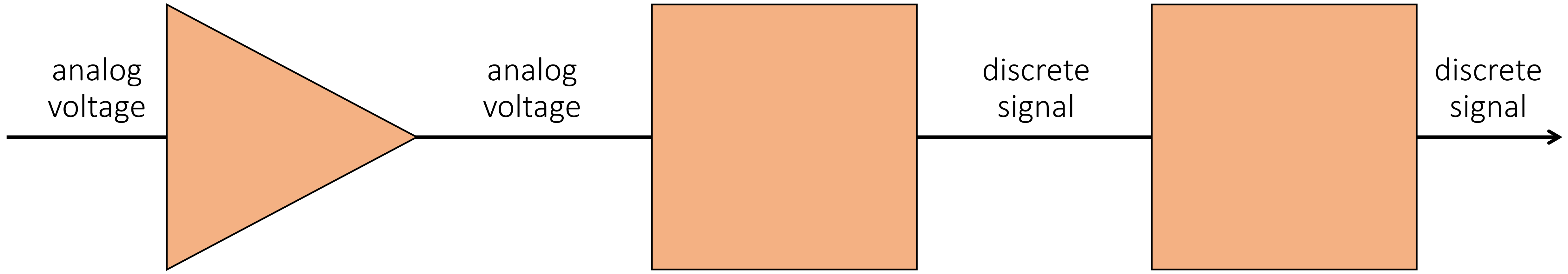
- Almost all of the steps in both the sensor and image processing pipeline I described earlier are camera-dependent.
- Even if we know the basic steps, the implementation details are proprietary information that companies actively try to keep secret.
- I will go back to a few of my slides to show you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



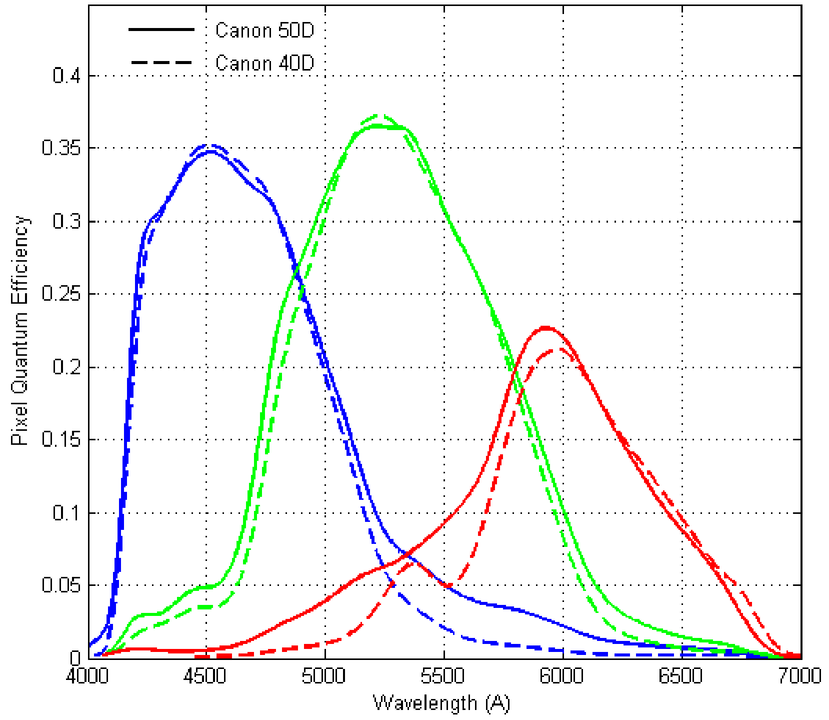
<u>analog amplifier (gain)</u>:
- gets voltage in range needed by A/D converter?
- accommodates ISO settings?
- accounts for vignetting?

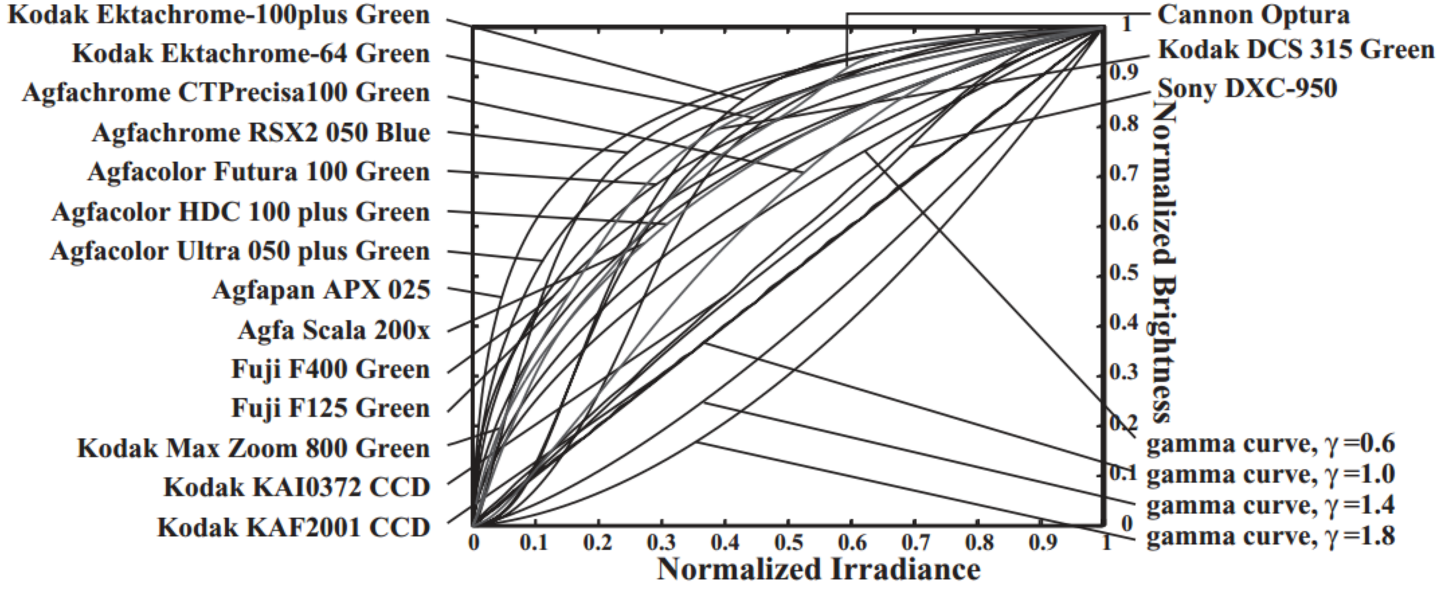
<u>analog-to-digital converter (ADC)</u>:
- depending on sensor, output has 10-16 bits.
- most often (?) 12 bits.

<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

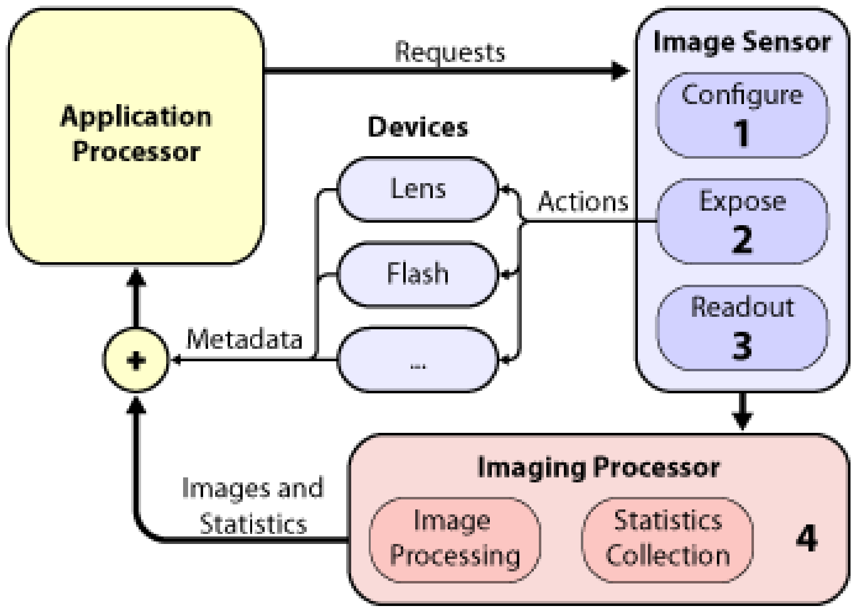
- Very difficult to get access to ground-truth data at intermediate stages of the pipeline.
- Very difficult to evaluate effect of new algorithms for specific pipeline stages.

…but things are getting better

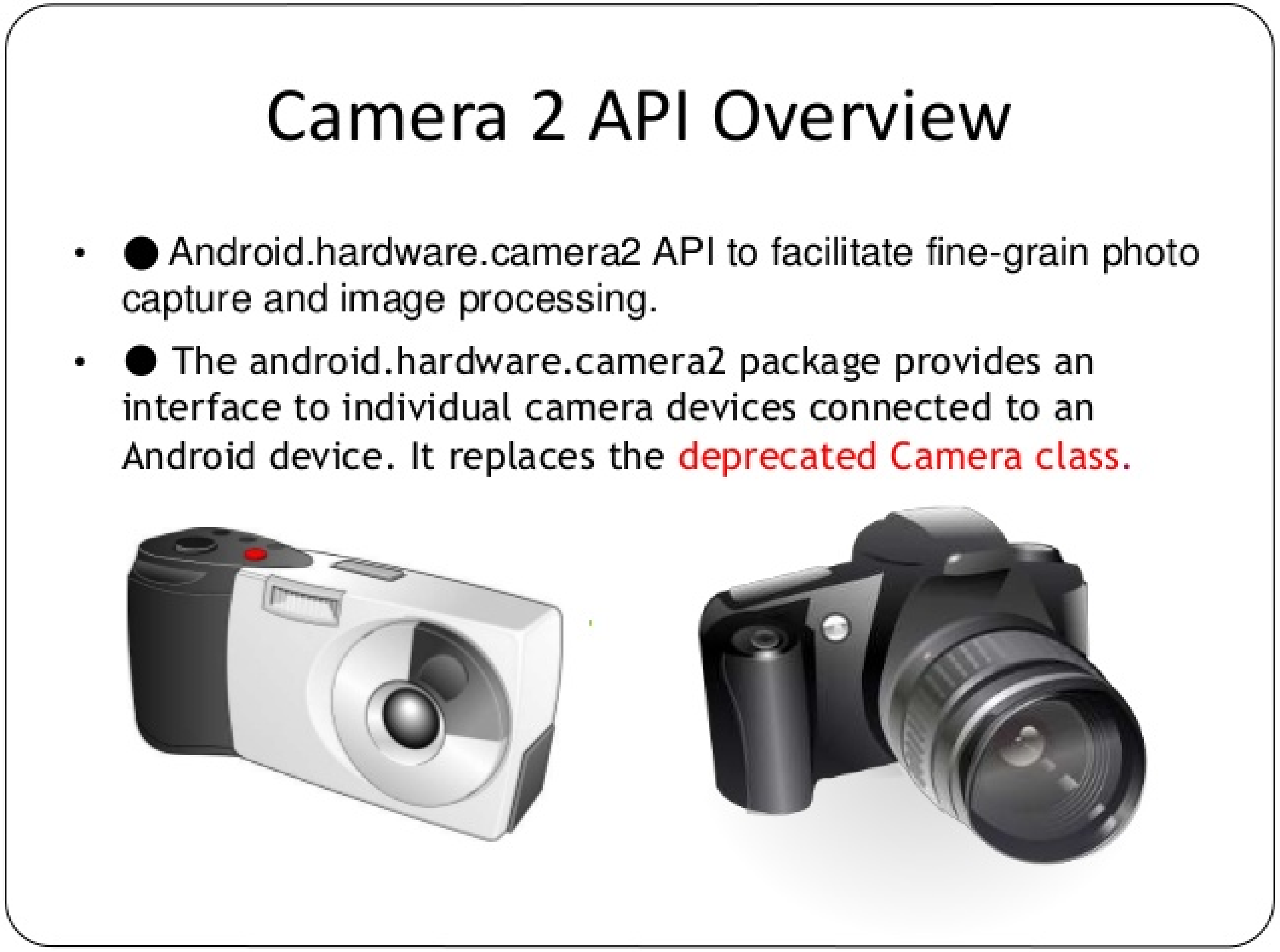
The Frankencamera: An Experimental Platform for Computational Photography

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 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.

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.
- Kim et al., "A New In-Camera Imaging Model for Color Computer Vision and Its Application," PAMI 2012.
- Chakrabarti et al., "Probabilistic Derendering of Camera Tone-mapped Images," PAMI 2014.
  Two papers that discuss in detail how to model and calibrate the image processing pipeline, how to (attempt to) derender an image that has already gone through the pipeline, and how to rerender an image under a different camera's pipeline.