CS 89.15/189.5, Fall 2015

Computational Aspects of Digital Photography

Sensors & Demosaicing

Wojciech Jarosz

wojciech.k.jarosz@dartmouth.edu

Dartmouth

Today's agenda

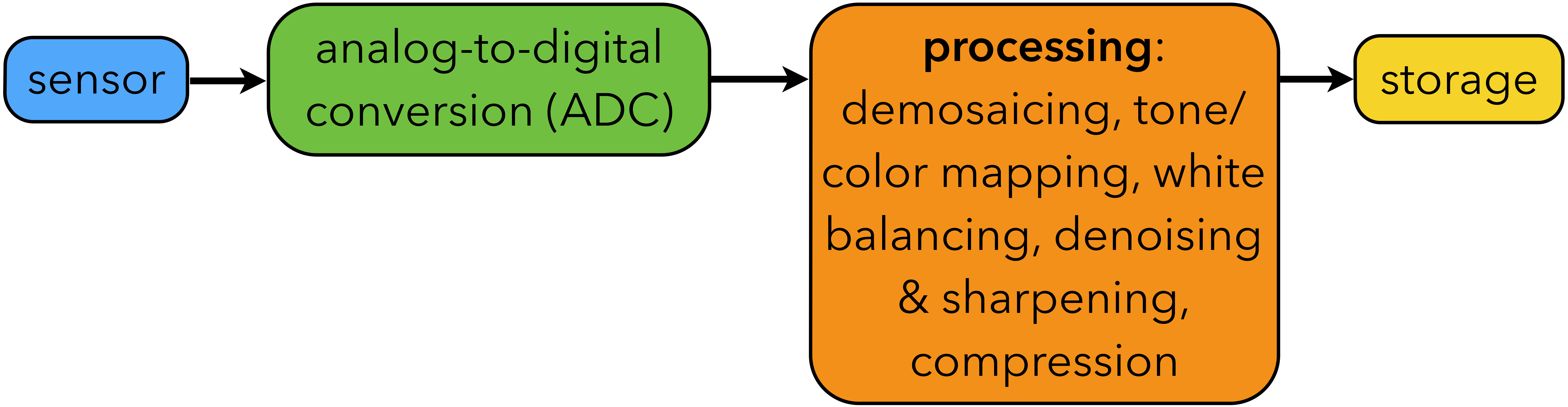
How do cameras record light?

How do cameras record color?

How can we transform that into color images?

How can we display those properly?

Camera pixel pipeline



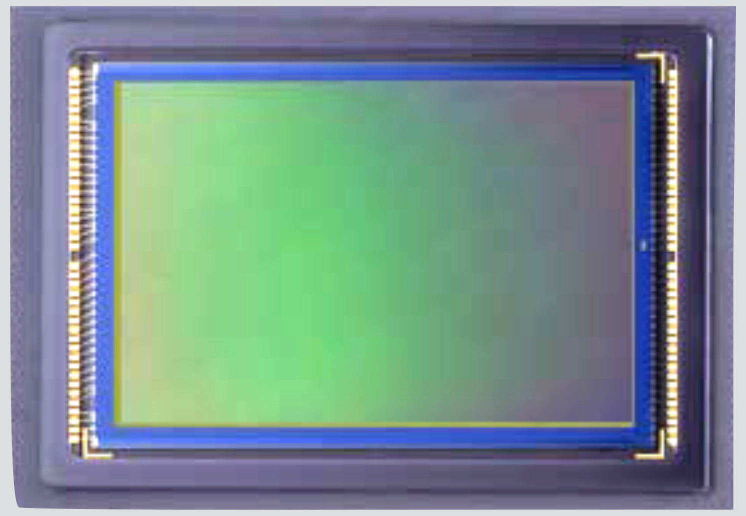
every camera uses different algorithms

the processing order may vary

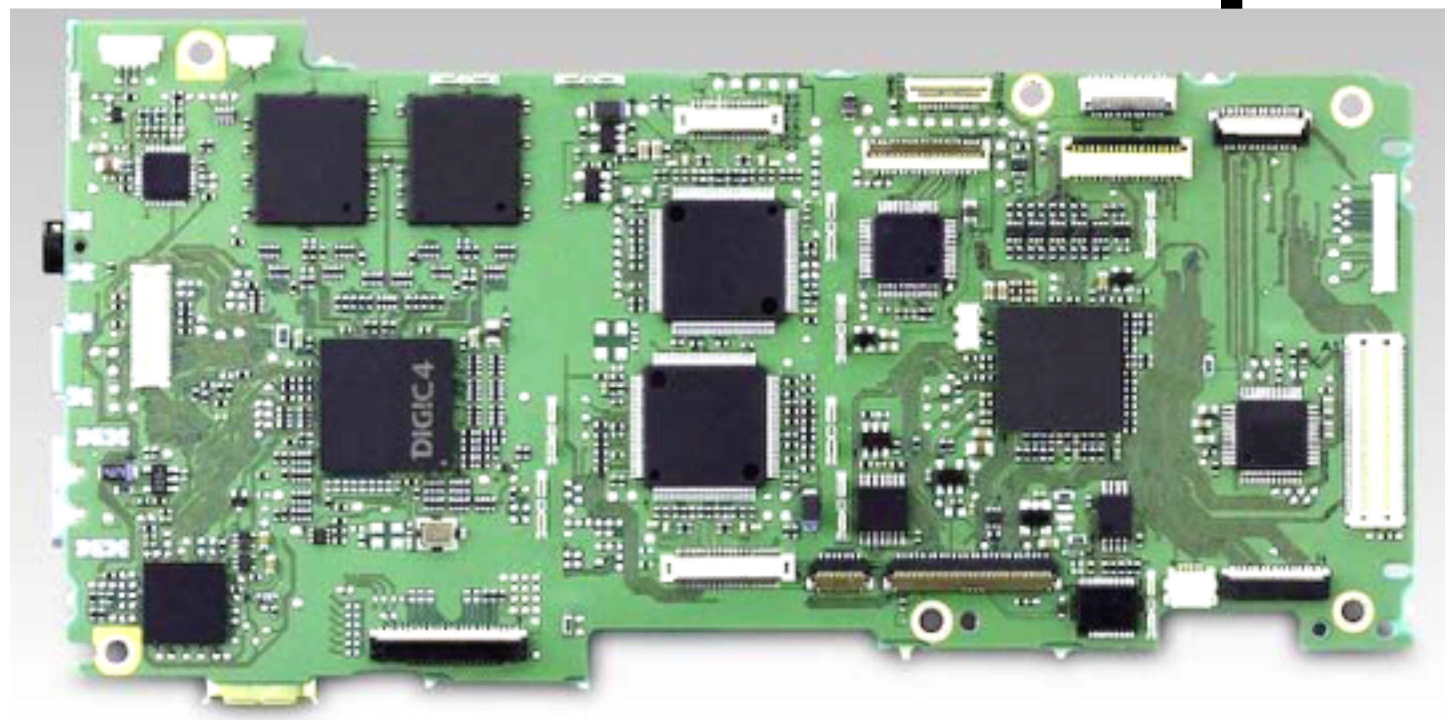
most of it is proprietary

Example pipeline

sensor → analog-to-digital conversion (ADC) → **processing**: demosaicing, tone/color mapping, white balancing, denoising & sharpening, compression → storage



Canon 21 Mpix CMOS sensor



Canon DIGIC 4 processor

Compact Flash card

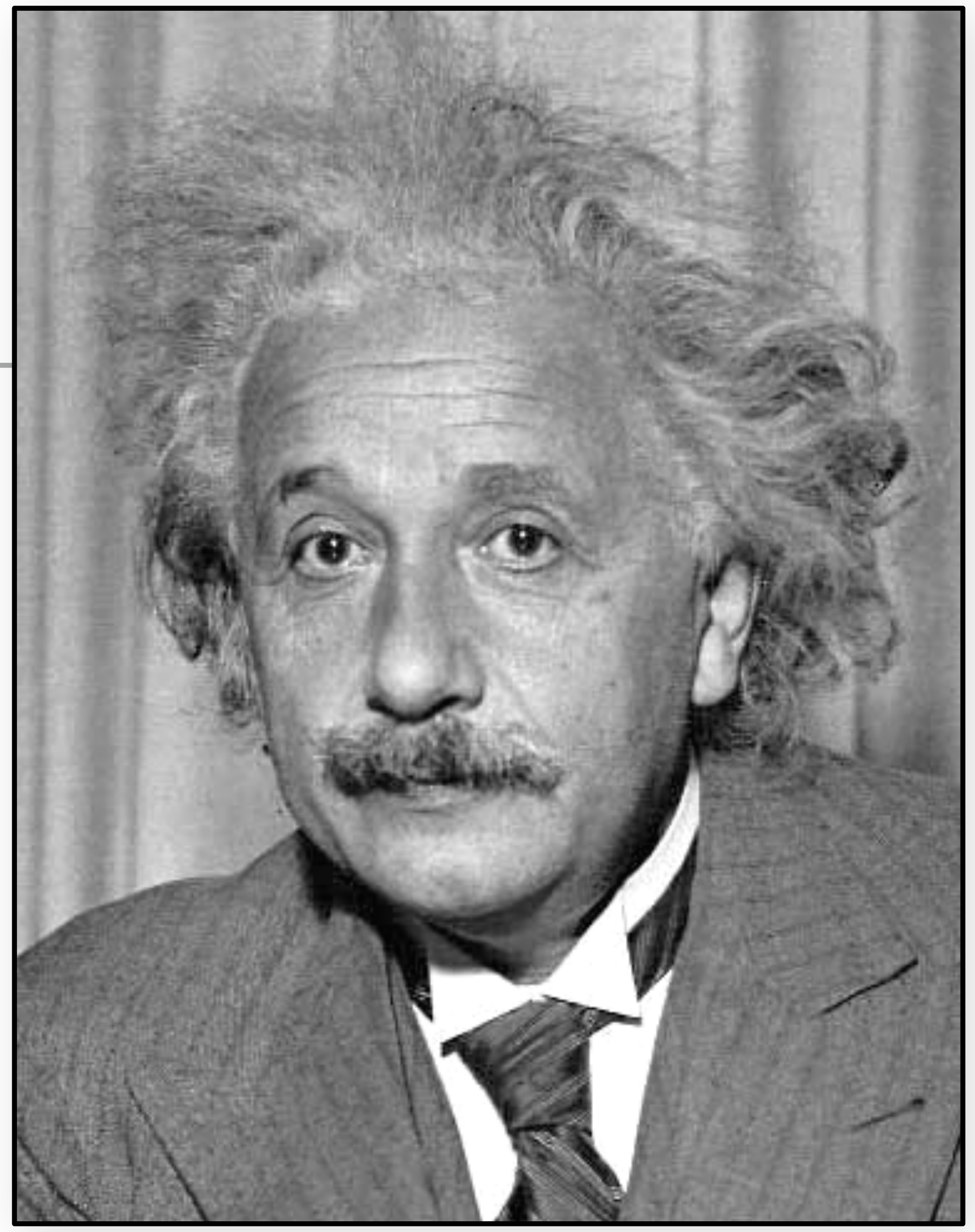
The photoelectric effect

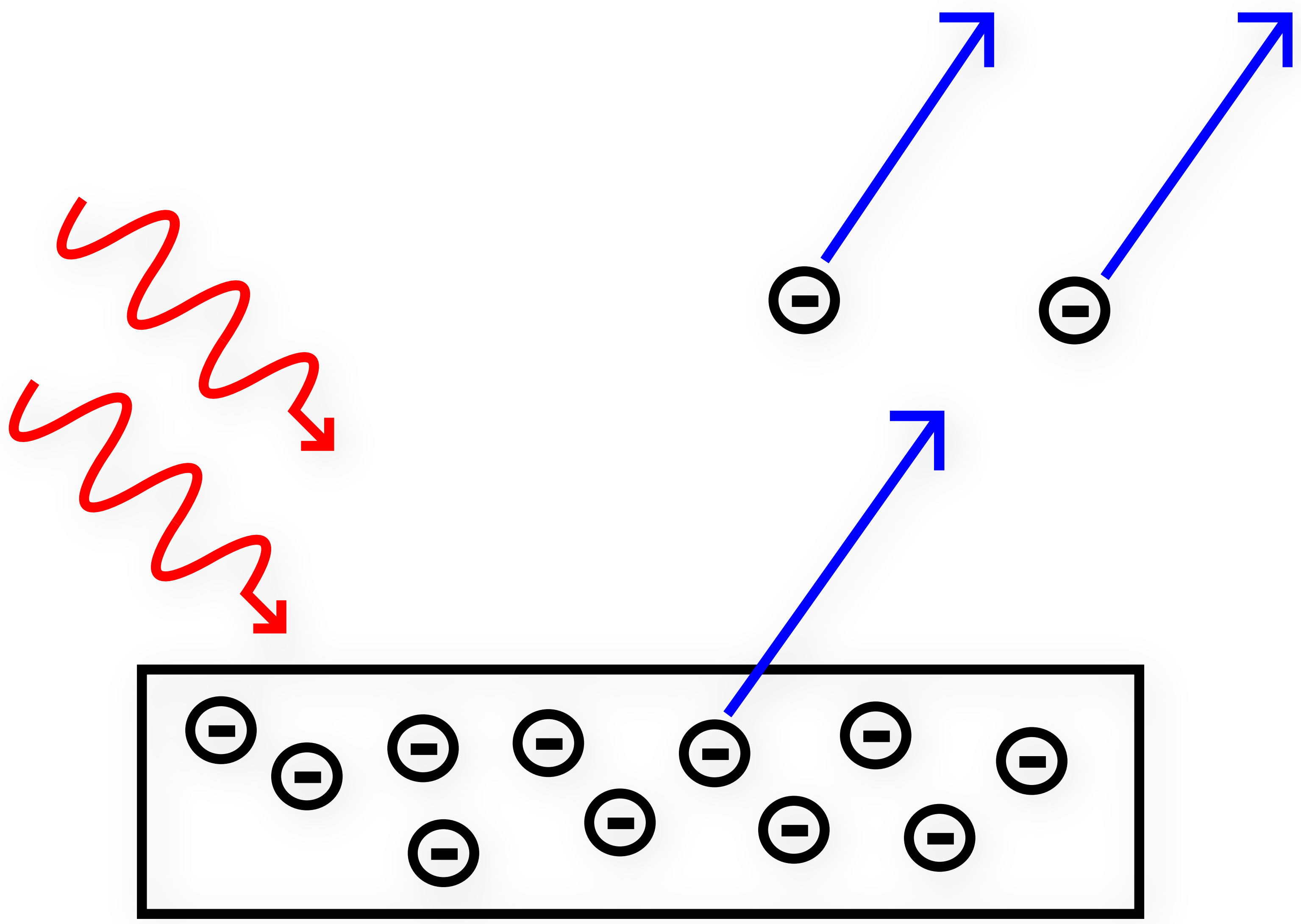
When a photon strikes a material, an electron may be emitted

- depends on the photon's energy, which depends on its wavelength

$$E_{\text{photon}} = \frac{h \times c}{\lambda}$$

- there is no notion of "brighter photons", only more or fewer of them





(wikipedia)

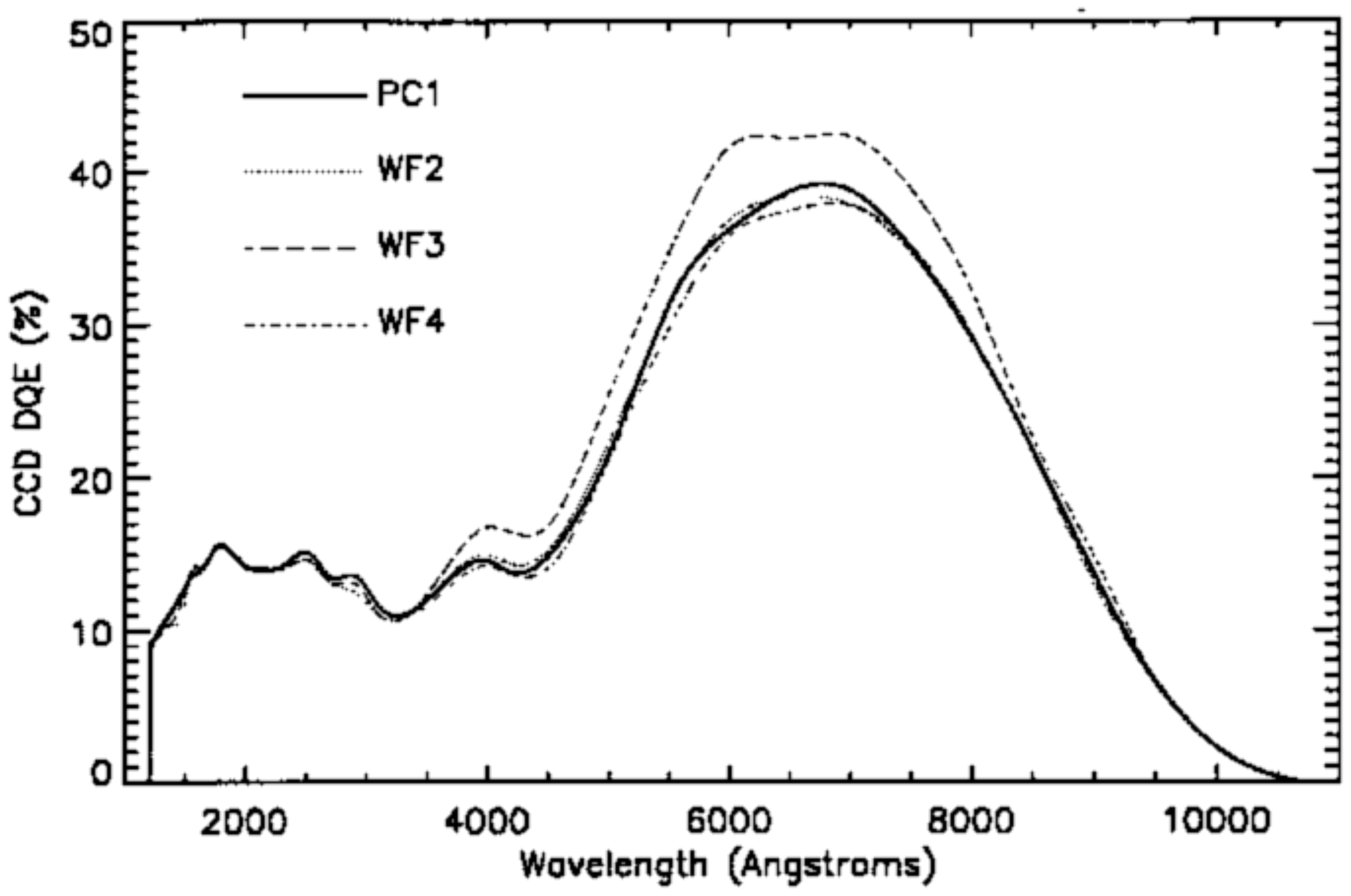
Quantum efficiency

Not all photons will produce an electron

- depends on quantum efficiency of the device

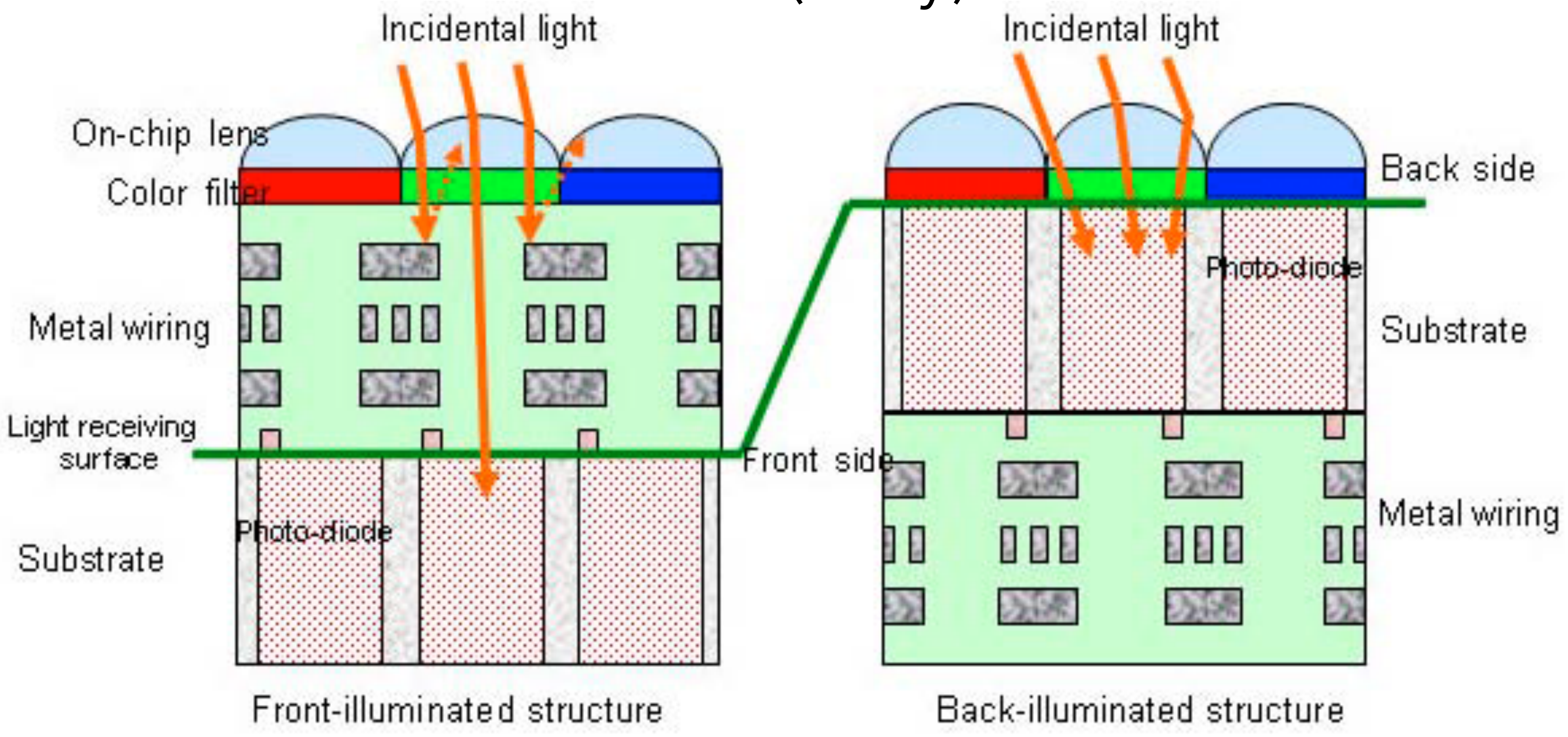
$$QE = \frac{\#\, electrons}{\#\, photons}$$

- human vision: ~15%
- typical digital camera: < 50%
- best back-thinned CCD: > 90%

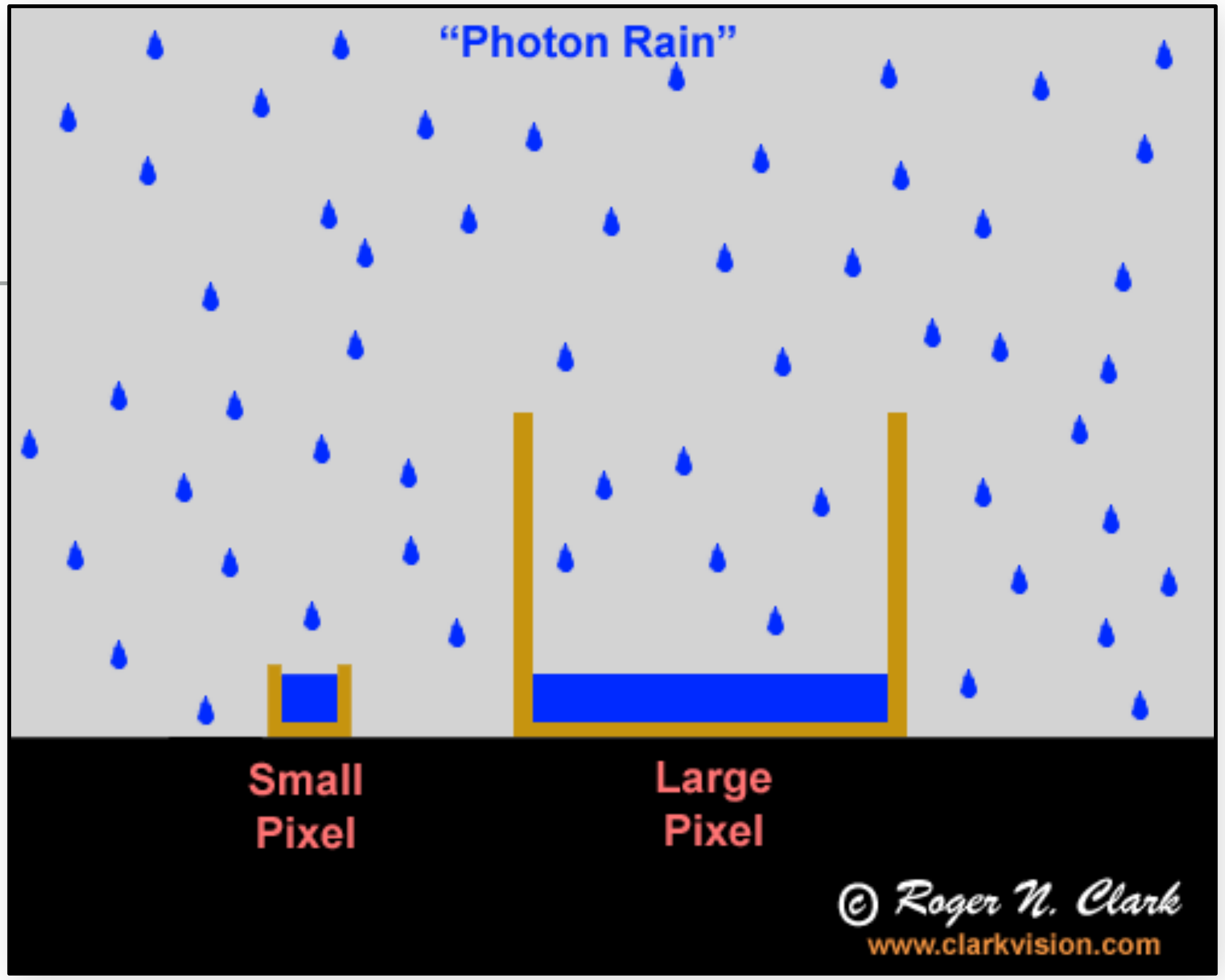


Hubble Space Telescope Camera 2

back-illuminated CMOS (Sony)



Pixel size



The current from one electron is small (10–100 fA)

- so integrate over space and time (pixel area × exposure time)
- larger pixel × longer exposure means more accurate measure

Typical pixel sizes:

- iPhone 5s (4.89×3.67mm@3264×2448pixels) = 1.5μ×1.5μ = $2.25\mu^2$
- Canon 5D II (36.00×24.00mm@5616×3744pixels) = 6.4μ×6.4μ = $41\mu^2$

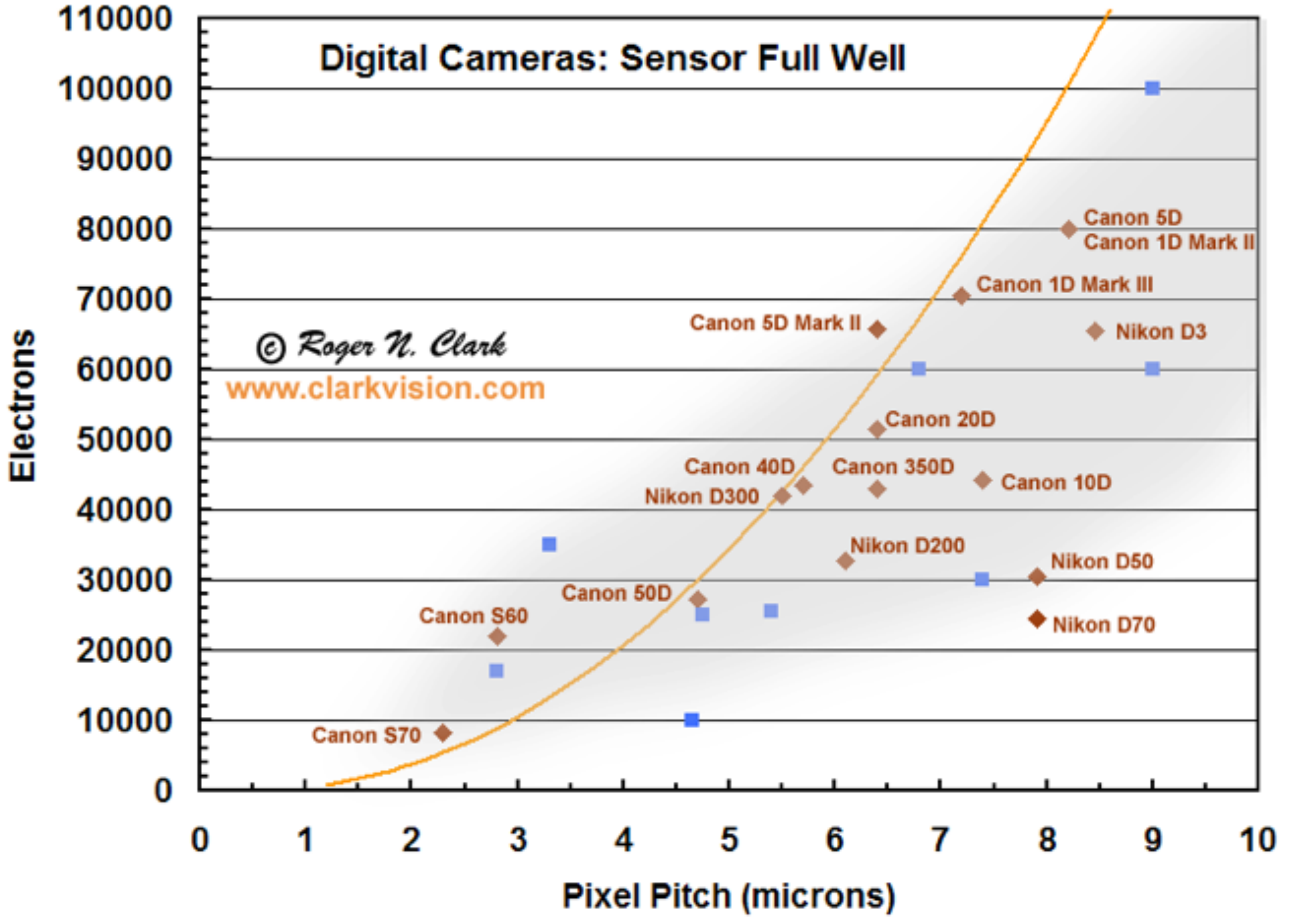
Full well capacity

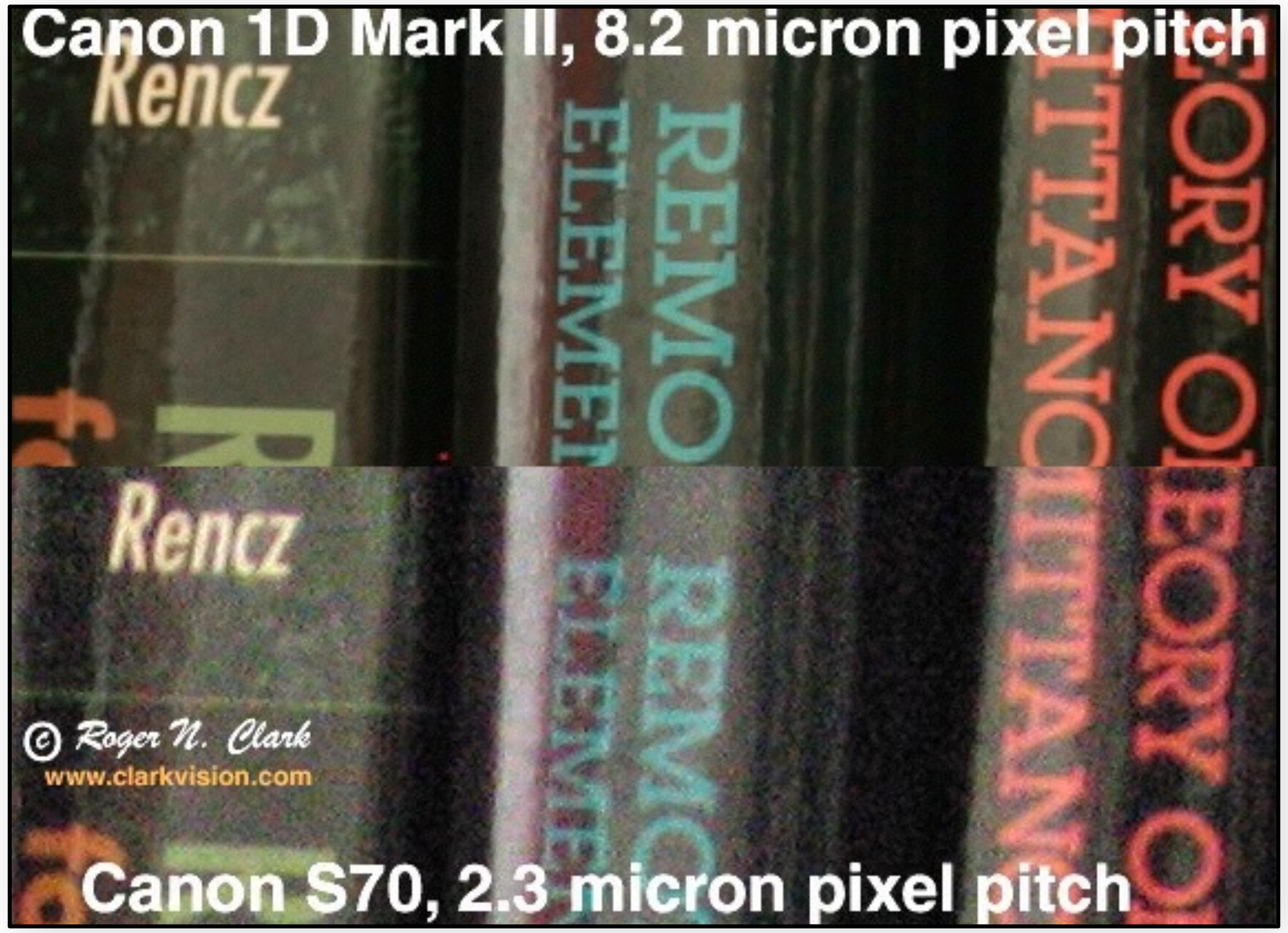
How many electrons can a pixel hold?

- depends mainly on the size of the pixel (but fill factor is important)

Too many photons causes *saturation*

- larger capacity leads to higher *dynamic range* between the brightest scene feature that won't saturate and the darkest that isn't too noisy



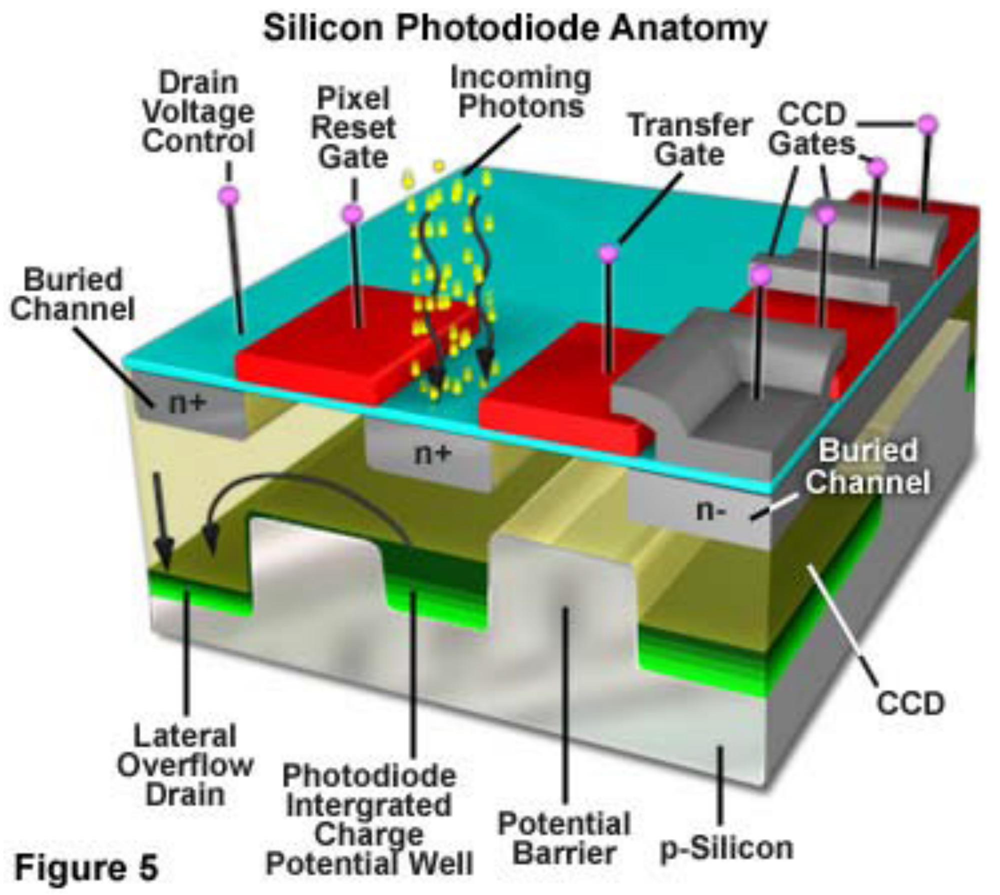


Blooming

Charge spilling over to nearby pixels

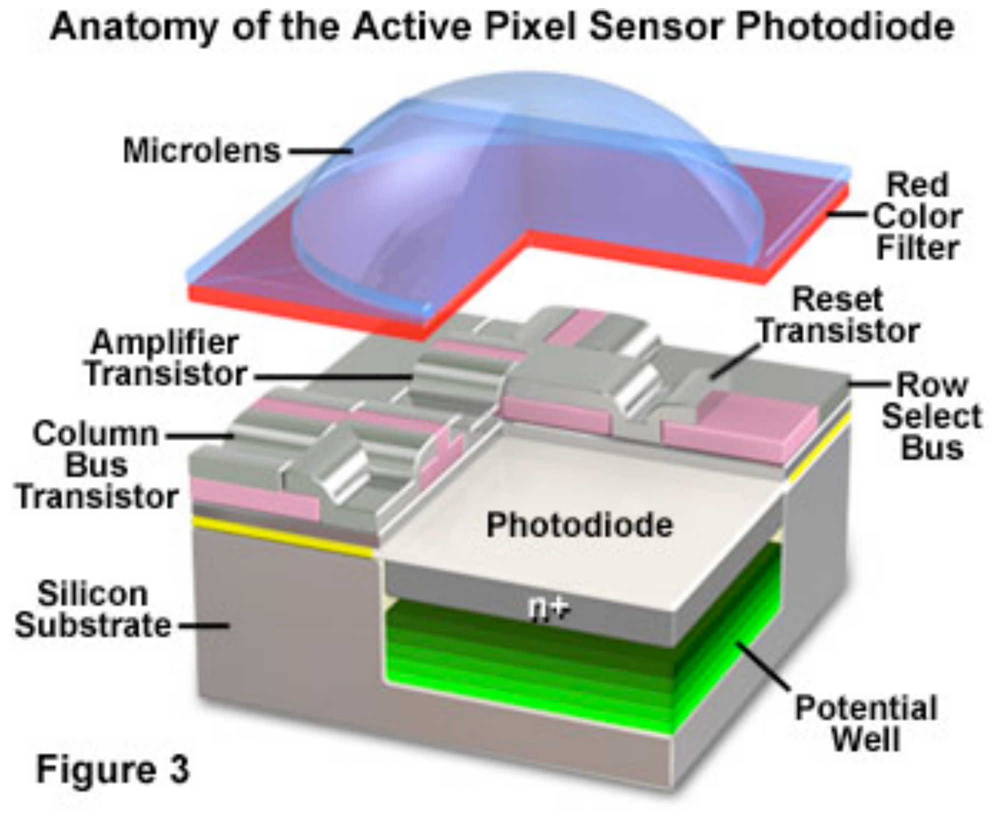
- can happen on CCD *and* CMOS sensors
- don't confuse with glare or other image artifacts

CMOS vs CCD sensors



CCD: charge-coupled device

- oldest solid-state image sensor technology
- charge shifted along columns to an output amplifier
- highest image quality, but not as flexible or cheap

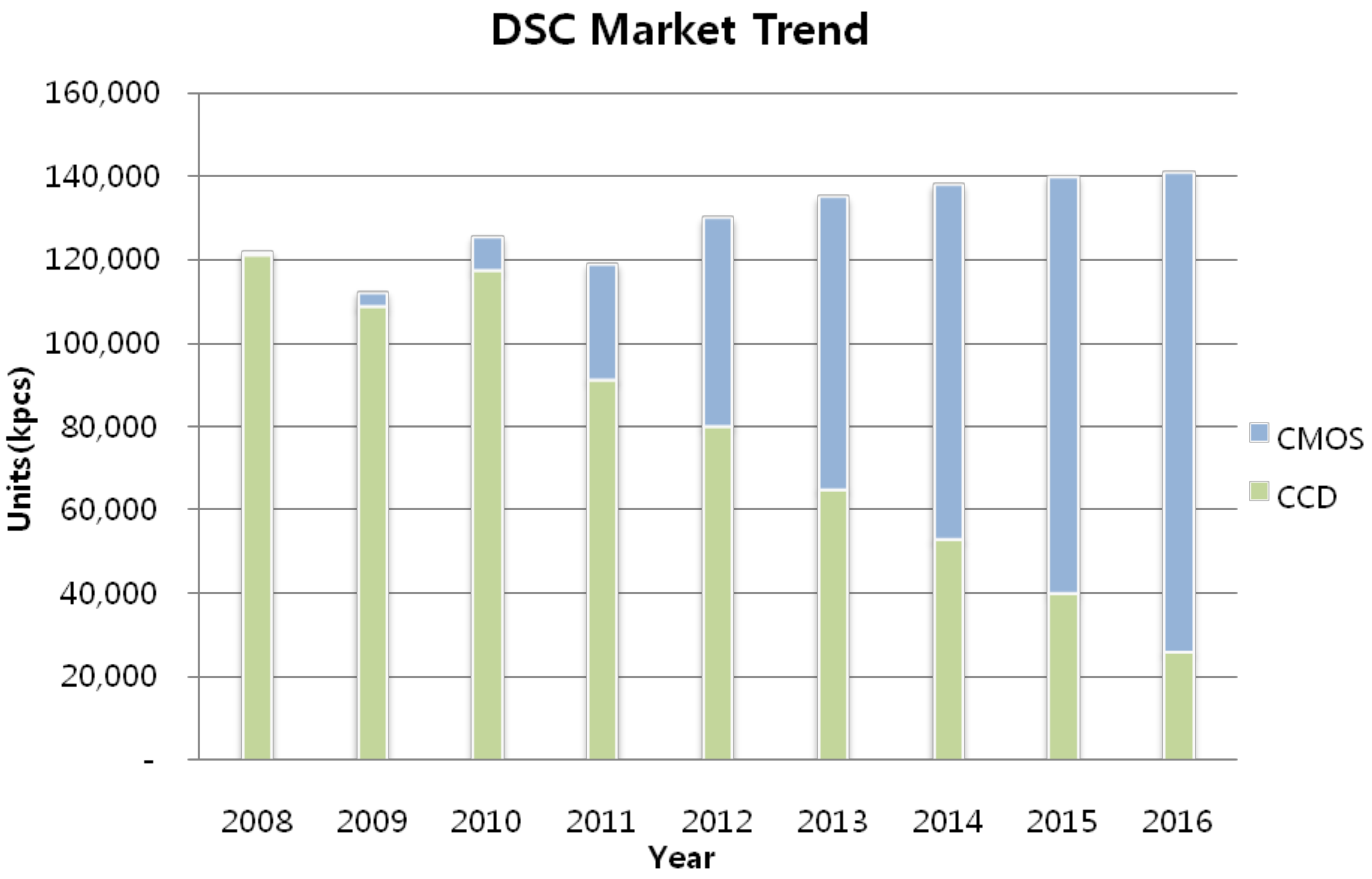


CMOS: complementary metal-oxide semiconductor

- newer, currently taking over
- each pixel has own charge amplifier, read out by row/column addressing
- same process used for CPUs and other VLSI chips
- low power, but noisy (but getting better)

Market trend

Samsung white paper, Current Status and Future Perspectives of CMOS Image Sensor

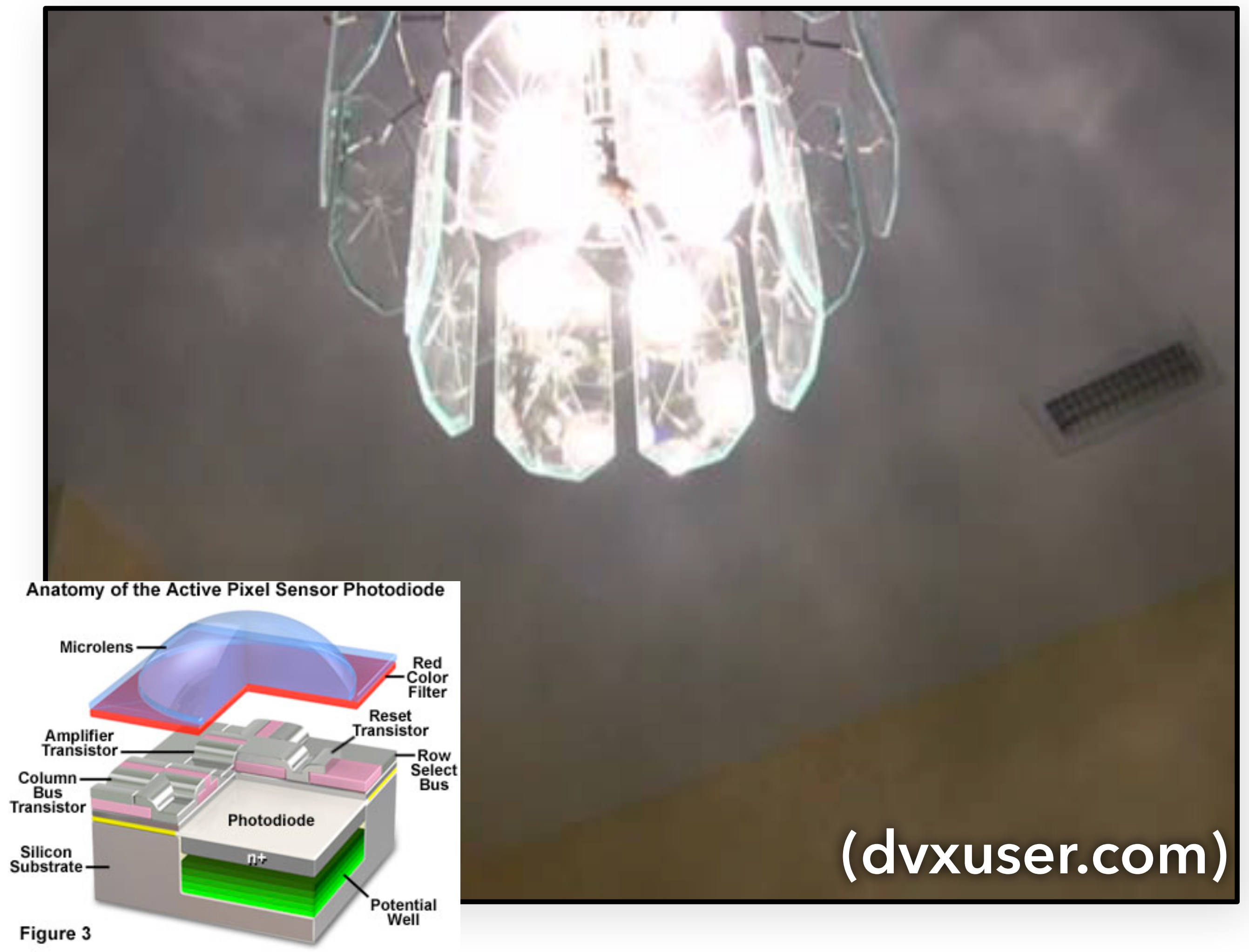


Smearing

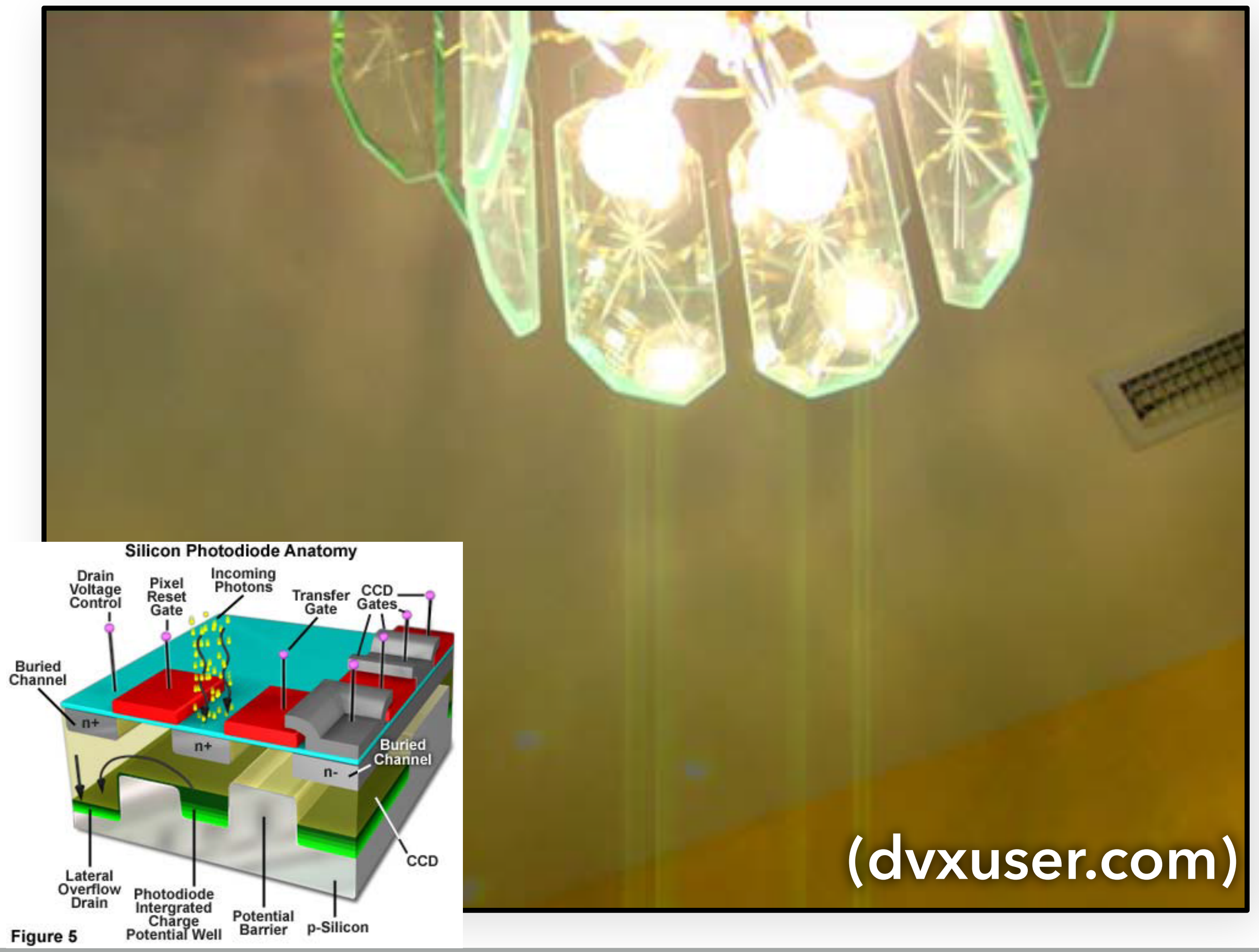
Side effect of readout on CCD sensors

- along columns; looks different than bloom
- only happens if pixels saturate
- doesn't happen on CMOS sensors

CMOS

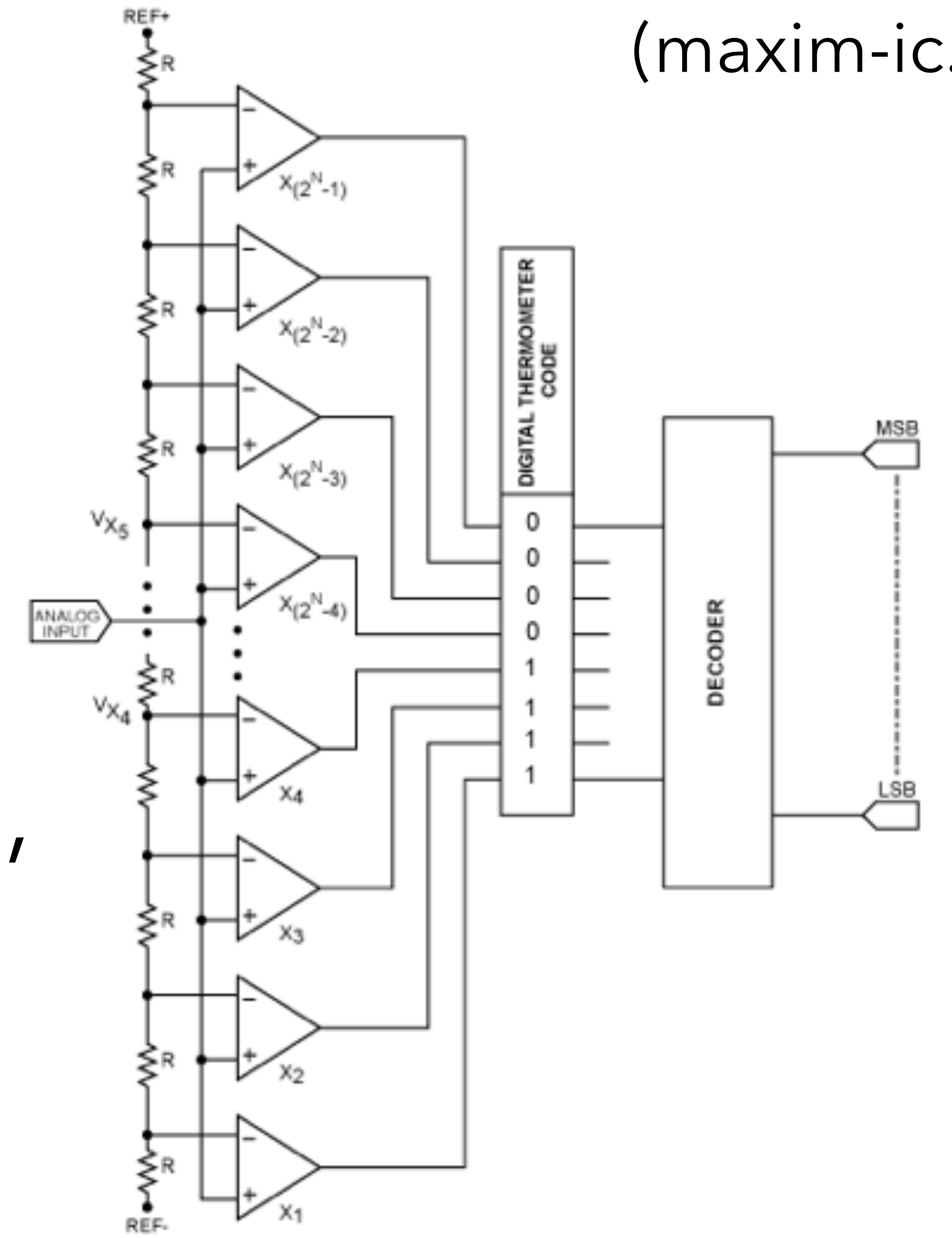


CCD



Analog to digital conversion (ADC)

(maxim-ic.com)

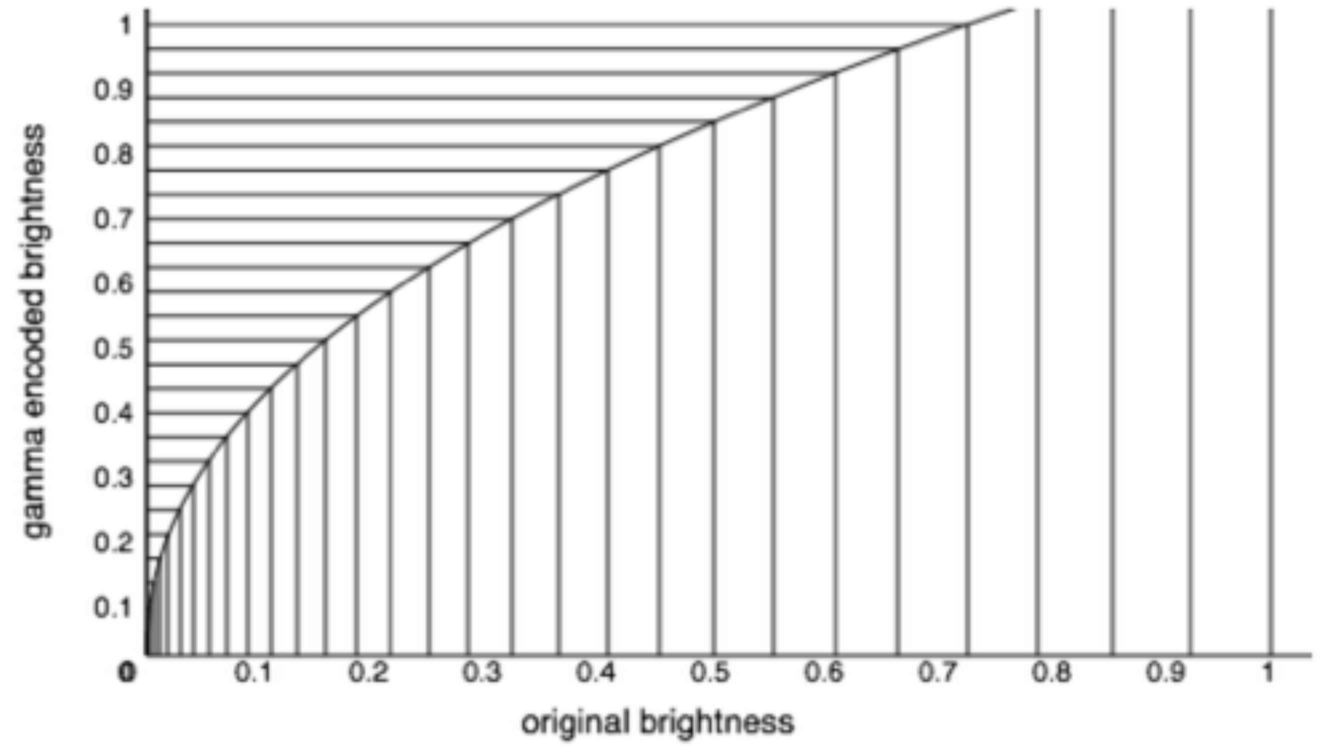


Convert analog signal to digital values

Recent sensors have one ADC per column of pixels

Must output more bits than JPEG stores (due to gamma)

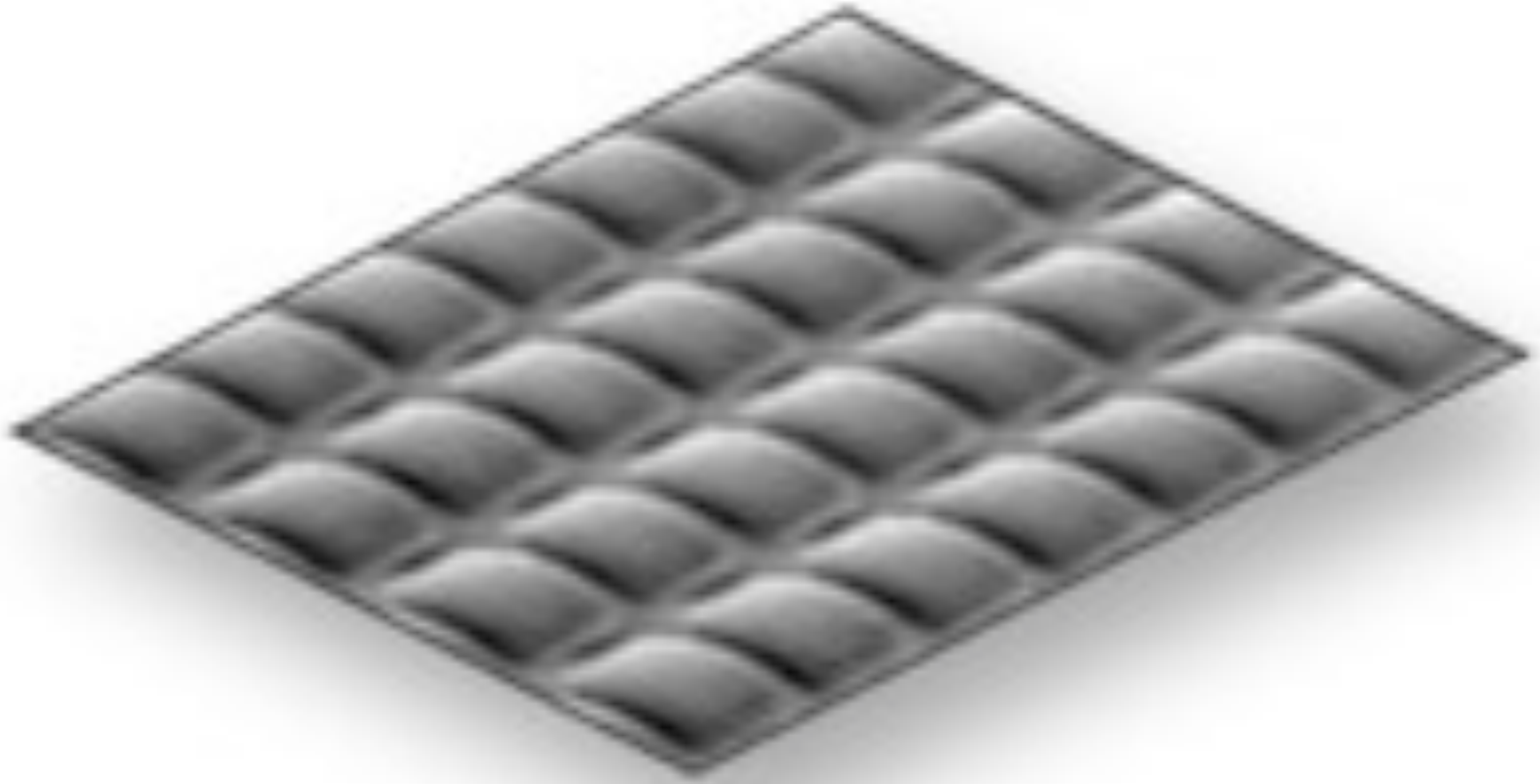
- converting ADC values (as stored in a RAW file) to the values stored in a JPEG file includes a step called gamma correction, which has the form output = input$^{\gamma}$ (0.0 ≤ input ≤ 1.0)
- since JPEG files only store 8 bits/pixel per channel, in order for a smooth gray ramp to fill each of these 256 buckets, the camera's ADC needs to output ≥ ~10 bits; otherwise, dark parts of the ramp will exhibit banding after applying gamma correction and re-quantizing

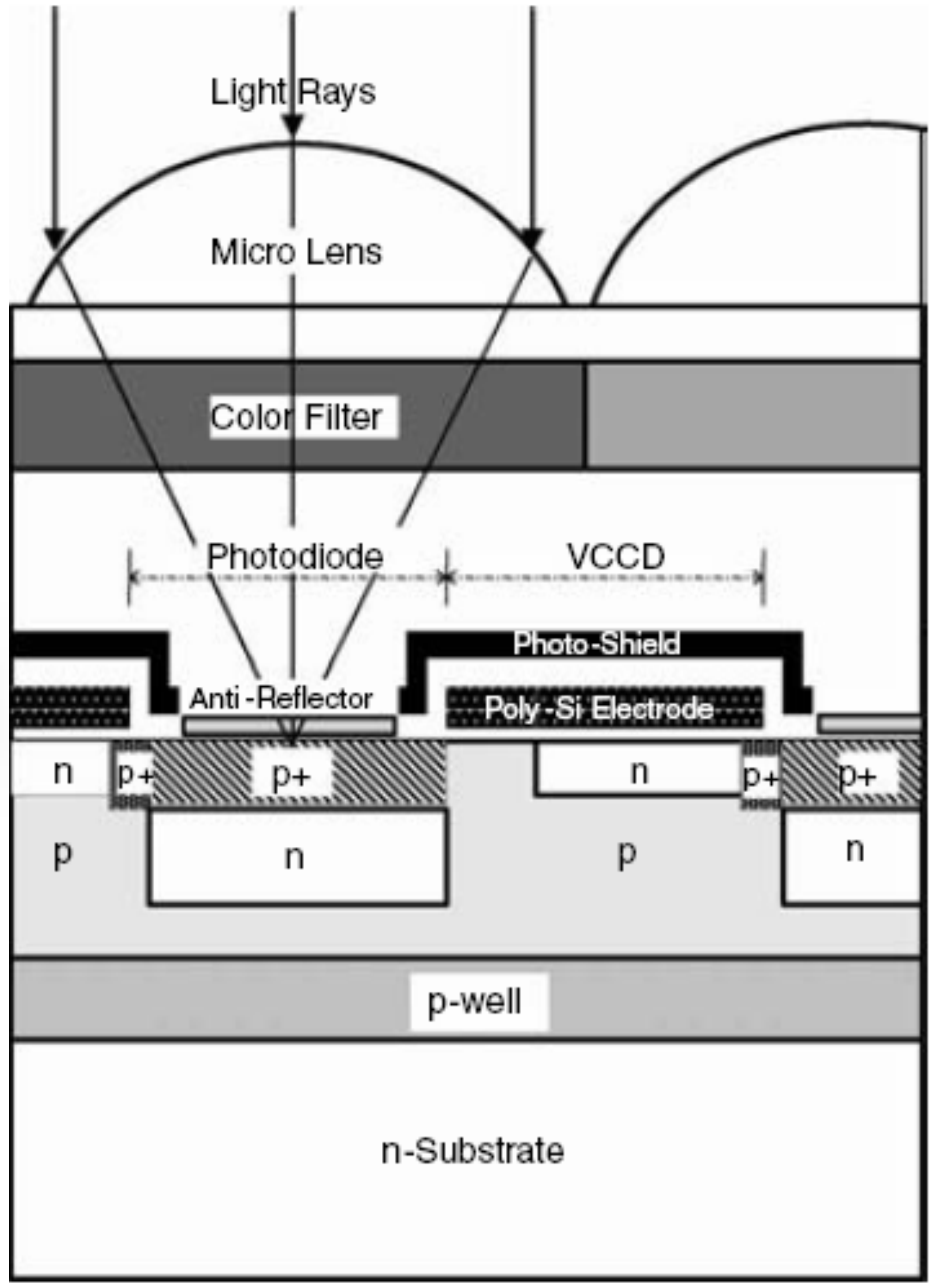


Fill factor

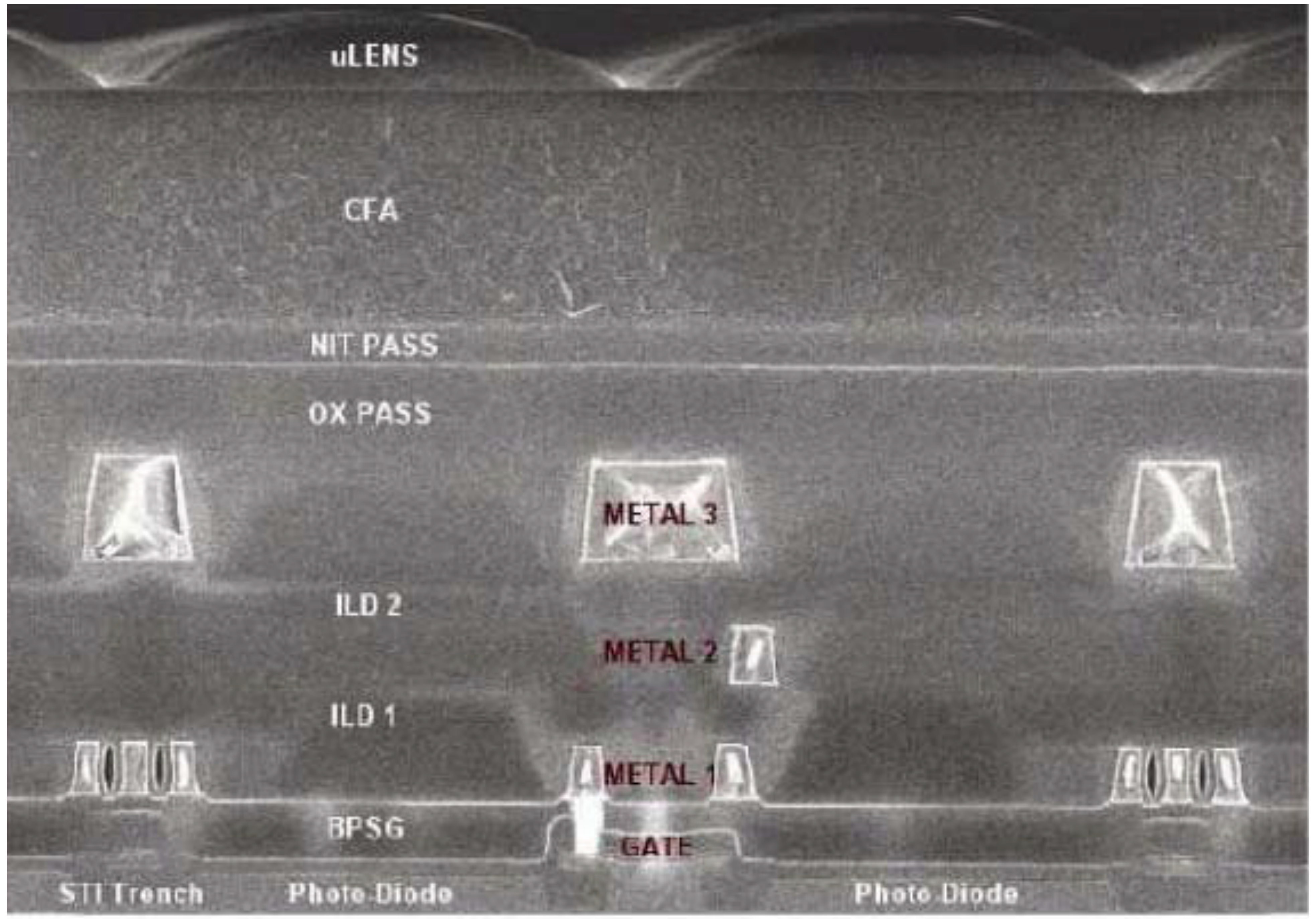
Fraction of sensor surface available to collect photons

- can be improved using per-pixel microlenses





on a CCD sensor



on a (front-illuminated) CMOS sensor

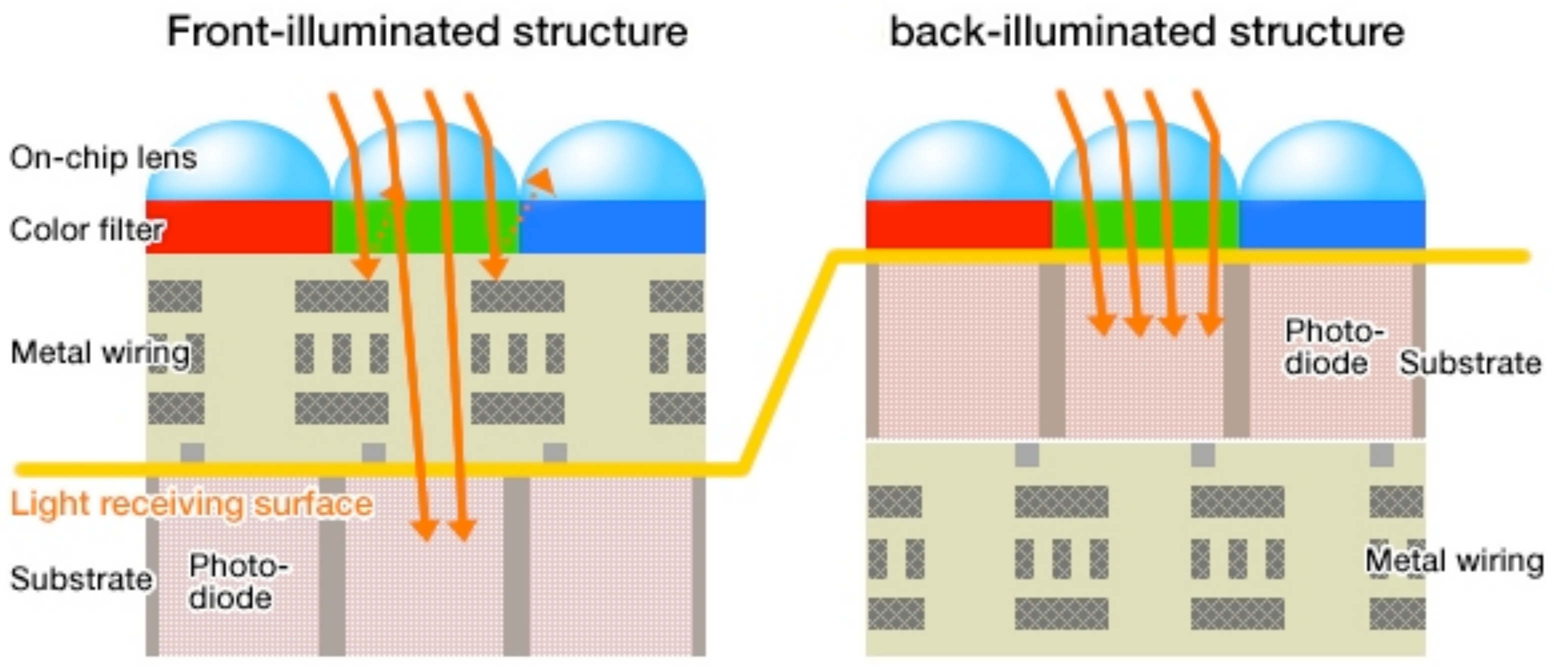
Front vs. back illumination

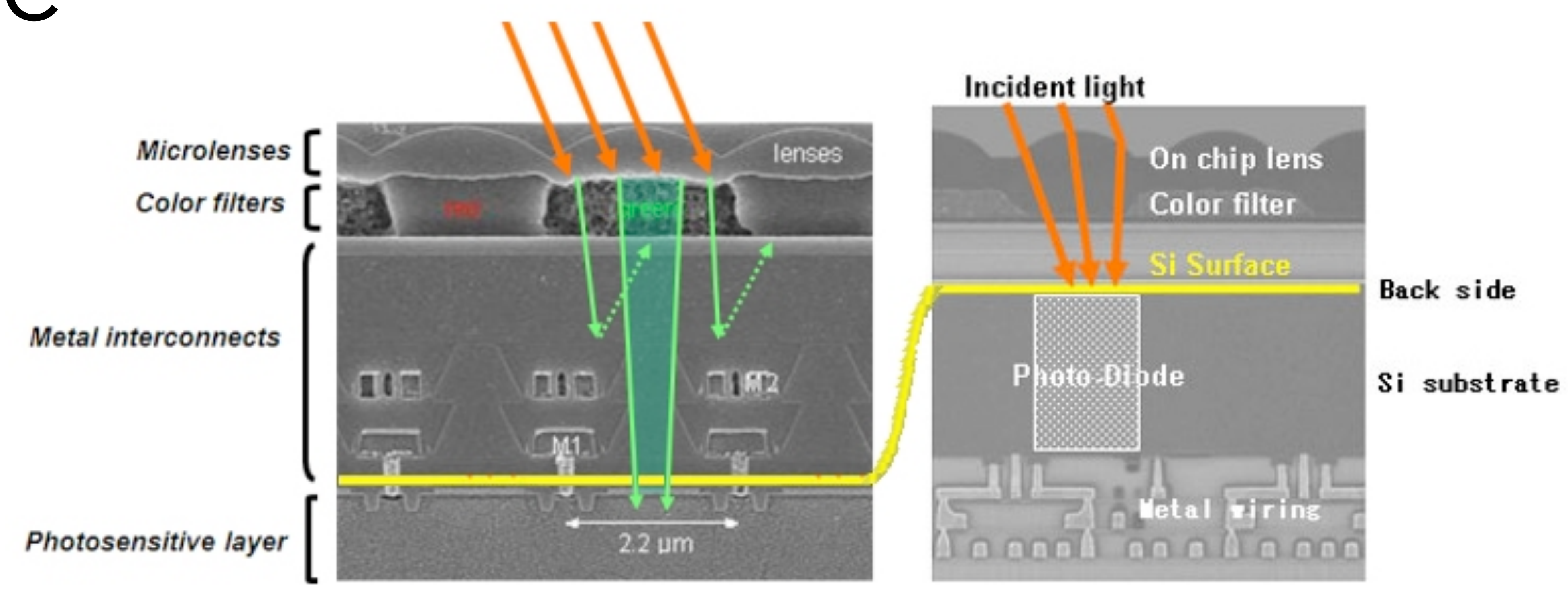
Front illuminated

- conventional design has interconnects and circuitry in front
- causes reduced fill factor and QE (particularly for blue)

Back illuminated

- originally an esoteric product for astronomy
- grind away back of chip and illuminate the photosensors directly
- now becoming popular in small-format CMOS sensors (iPhone 5)





Spatial prefiltering

Integrating light over an area at each pixel instead of point sampling serves two functions:

1. captures more photons, to improve dynamic range
2. convolves the image with a prefilter, to avoid aliasing

Microlenses gather more light and improve the prefilter

- microlenses ensure that the spatial prefilter is a 2D rect of width roughly equal to the pixel spacing

Antialiasing filters are typically added to further reduce aliasing

Removing the antialiasing filter

"hot rodding" your digital camera ($450 + shipping)

anti-aliasing filter removed

normal

Removing the antialiasing filter

"hot rodding" your digital camera ($450 + shipping)

anti-aliasing filter removed

normal

Nikon D800 (aa-filter)

Nikon D800E (no aa-filter)

Recap

photons strike a sensor and are converted to electrons

- performance factors include quantum efficiency and pixel size

sensors are typically CCD or CMOS

- both can suffer blooming; only CCDs can suffer smearing

integrating light over an area serves two functions

- capturing more photons, to improve dynamic range
- convolving the image with a prefilter, to avoid aliasing
- to ensure that the area spans pixel spacing, use microlenses
- to improve further on the prefilter, use an antialiasing filter

integrating light over time serves the same two functions

- captures more photons, but may produce motion blur

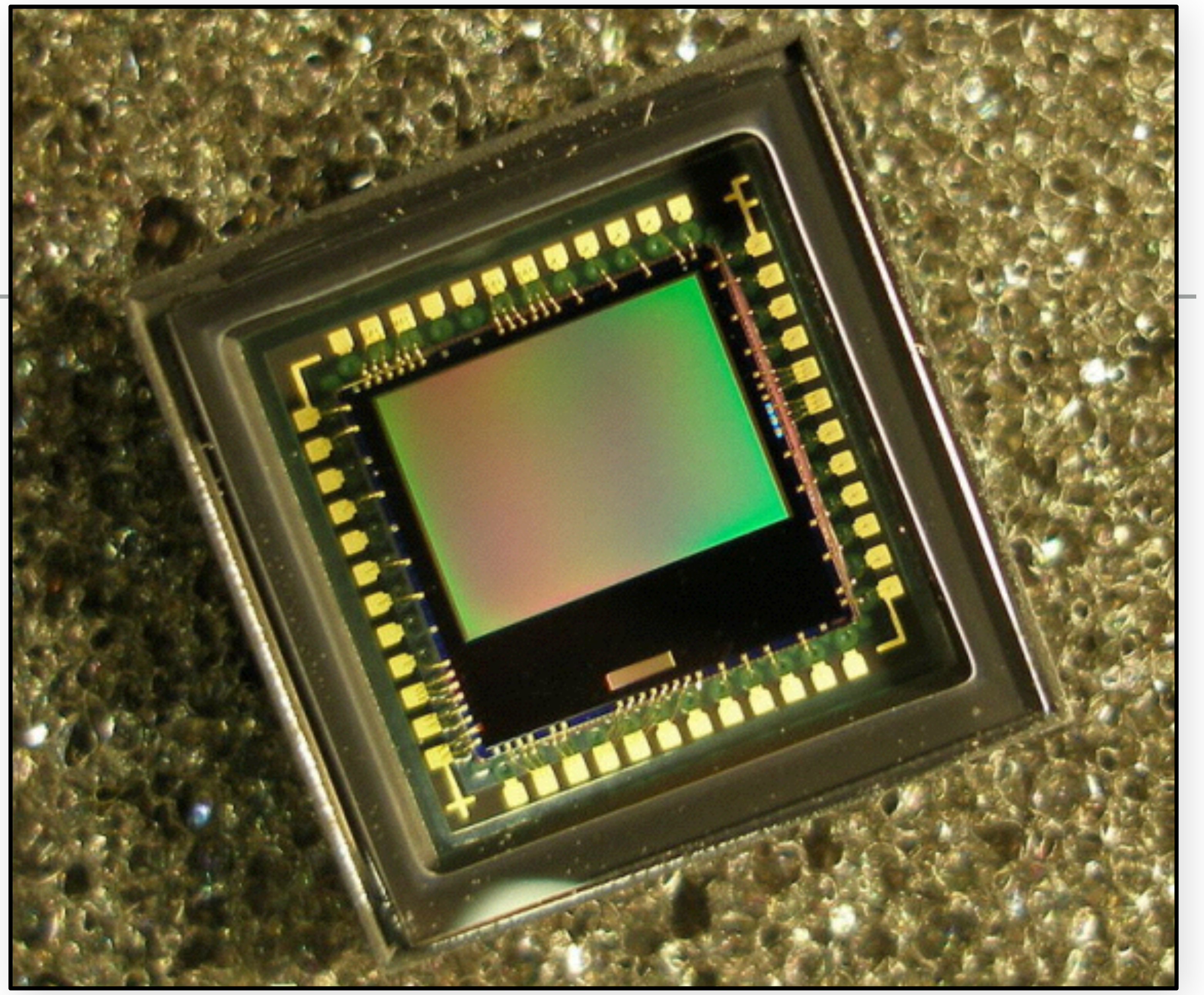
Color acquisition

Sensing color images

Problem: a photosite can record only one number

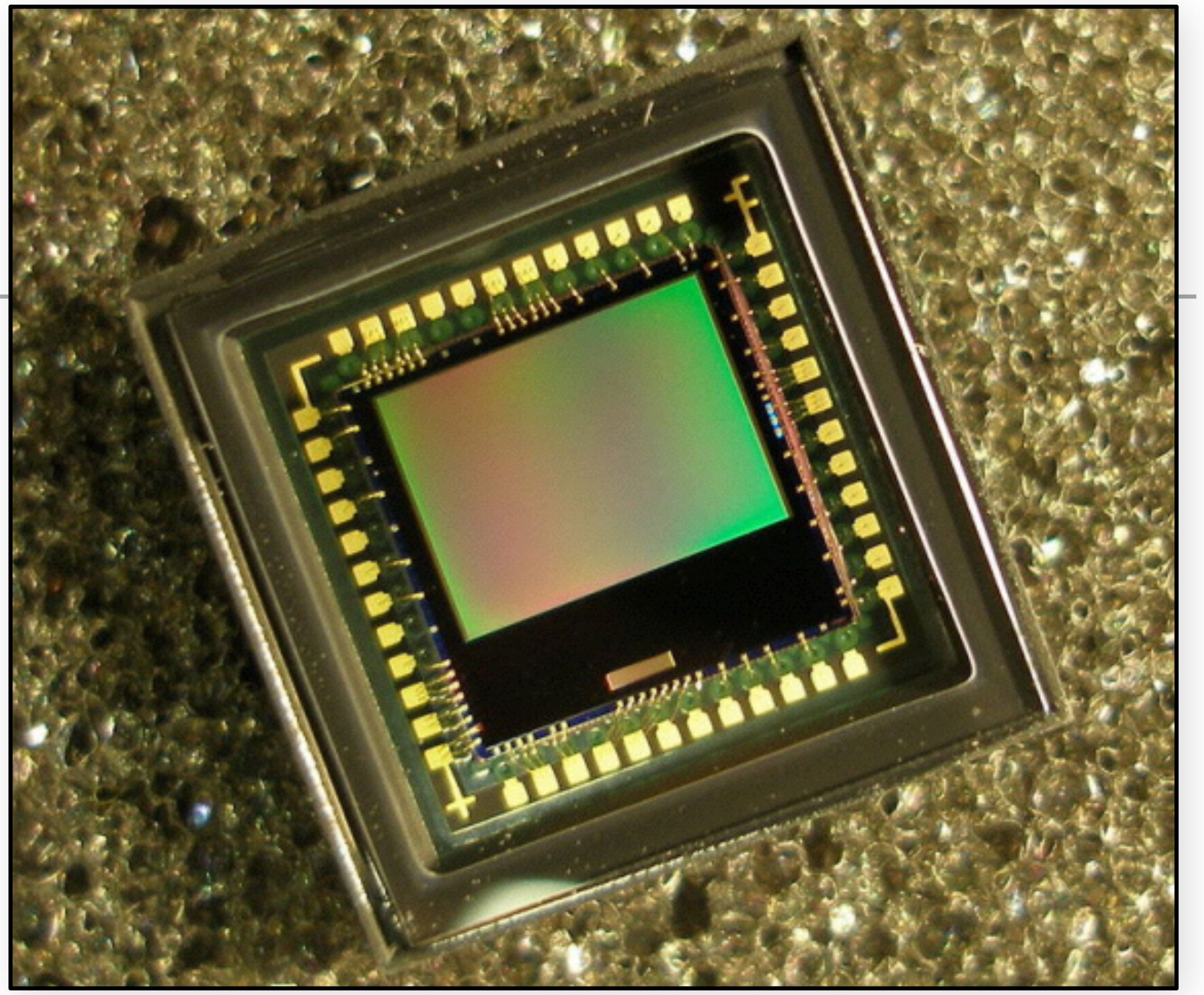
We need 3 numbers for color

What can we do?



CMOS sensor

Sensing color images


CMOS sensor

Digital sensors are sensitive to all (visible) wavelengths

- For details see:
  http://en.wikipedia.org/wiki/Image_sensor
  http://en.wikipedia.org/wiki/Active_pixel_sensor
  http://en.wikipedia.org/wiki/Charge-coupled_device

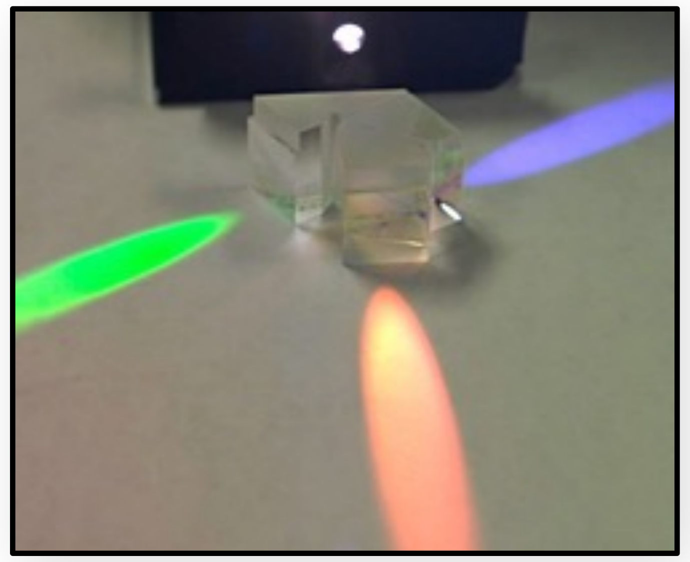
Obtain color measurement using different color filters

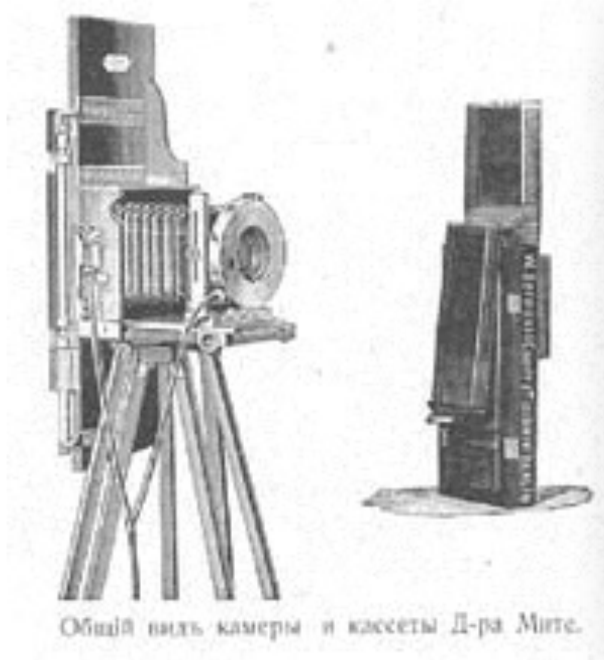
- Color filters play same role as response curves of photoreceptors
- Absorb part of the spectrum

Infrared capture demo

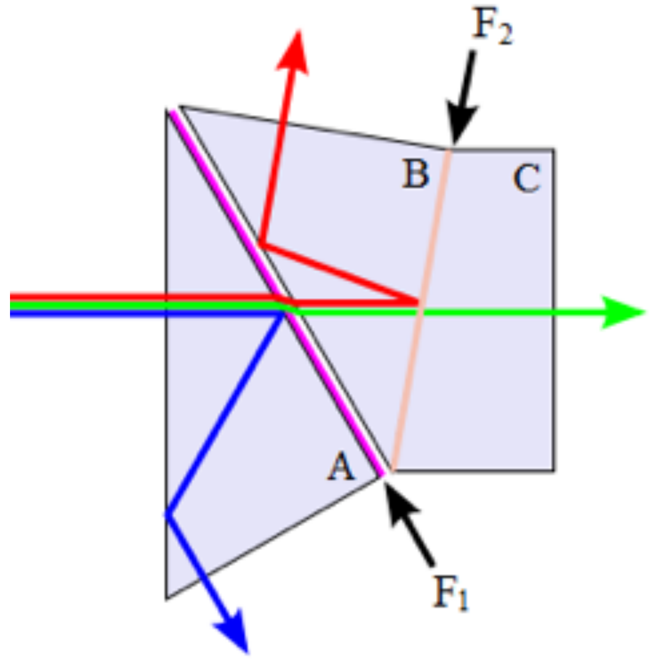
Approaches to sensing color

Scan 3 times (temporal multiplexing)

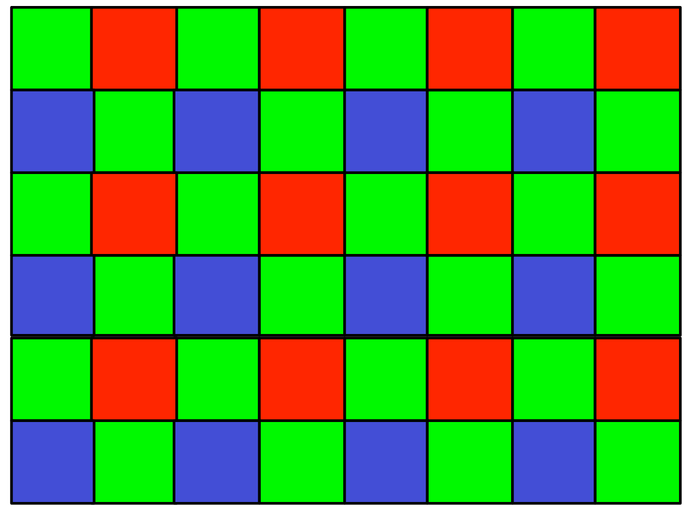




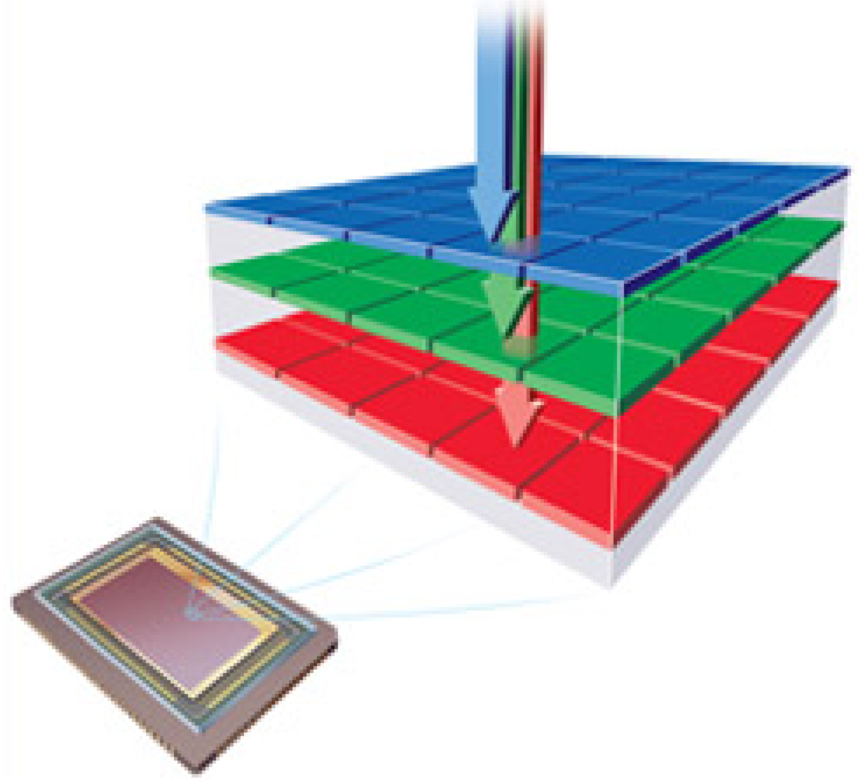
Use 3 detectors
(3-ccd camera)



Use offset color samples (spatial multiplexing)



Multiplex in depth (Tripack film, Foveon)



Interferences (Lipmann)

Temporal multiplexing

Examples:

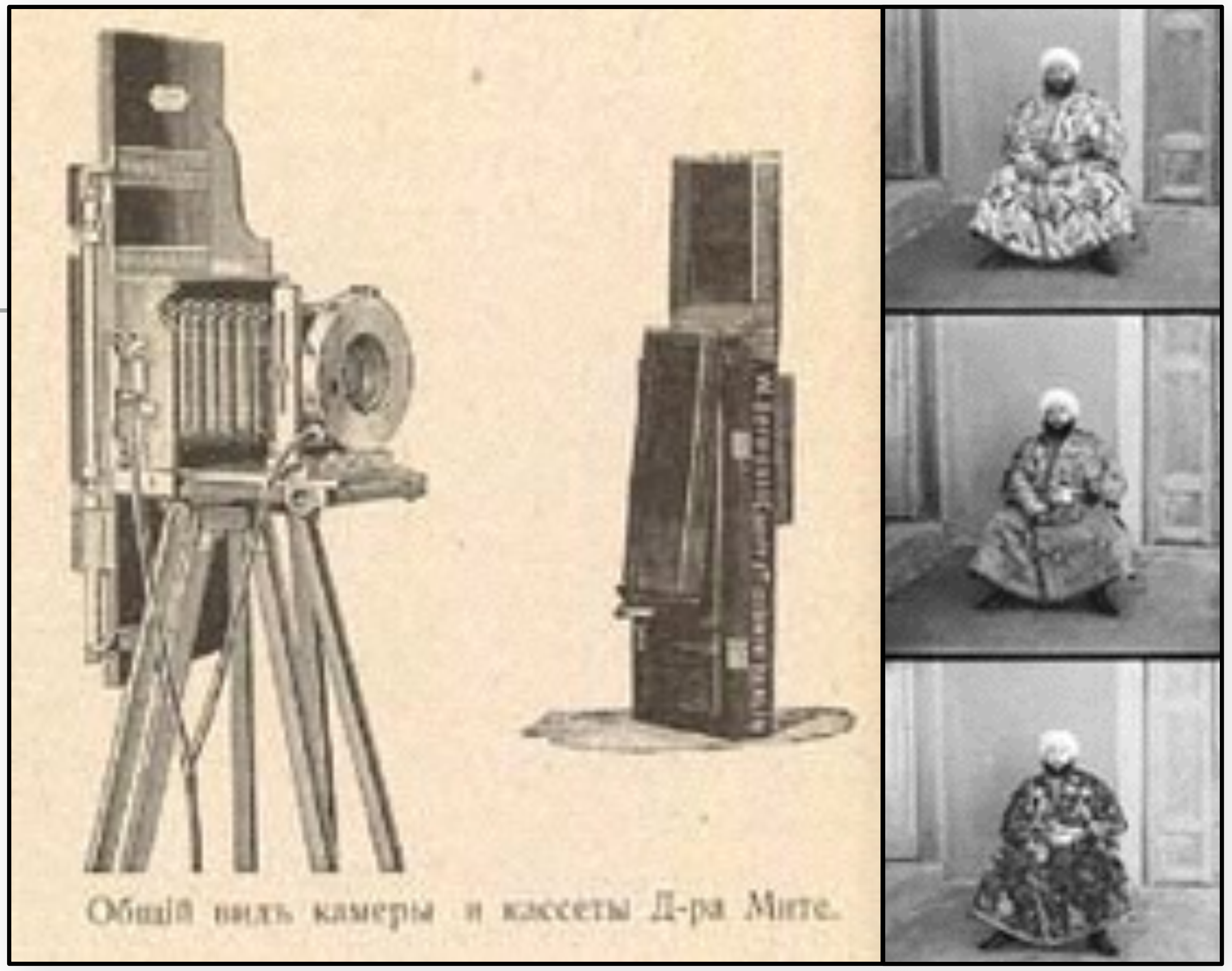
- Drum scanners
- Flat-bed scanners
- Maxwell, Russian photographs from 1900's

Pros:

- 3 real values per pixel
- Can use a single sensor

Cons

- Only for static scenes, slow



Sergey Prokudin-Gorsky

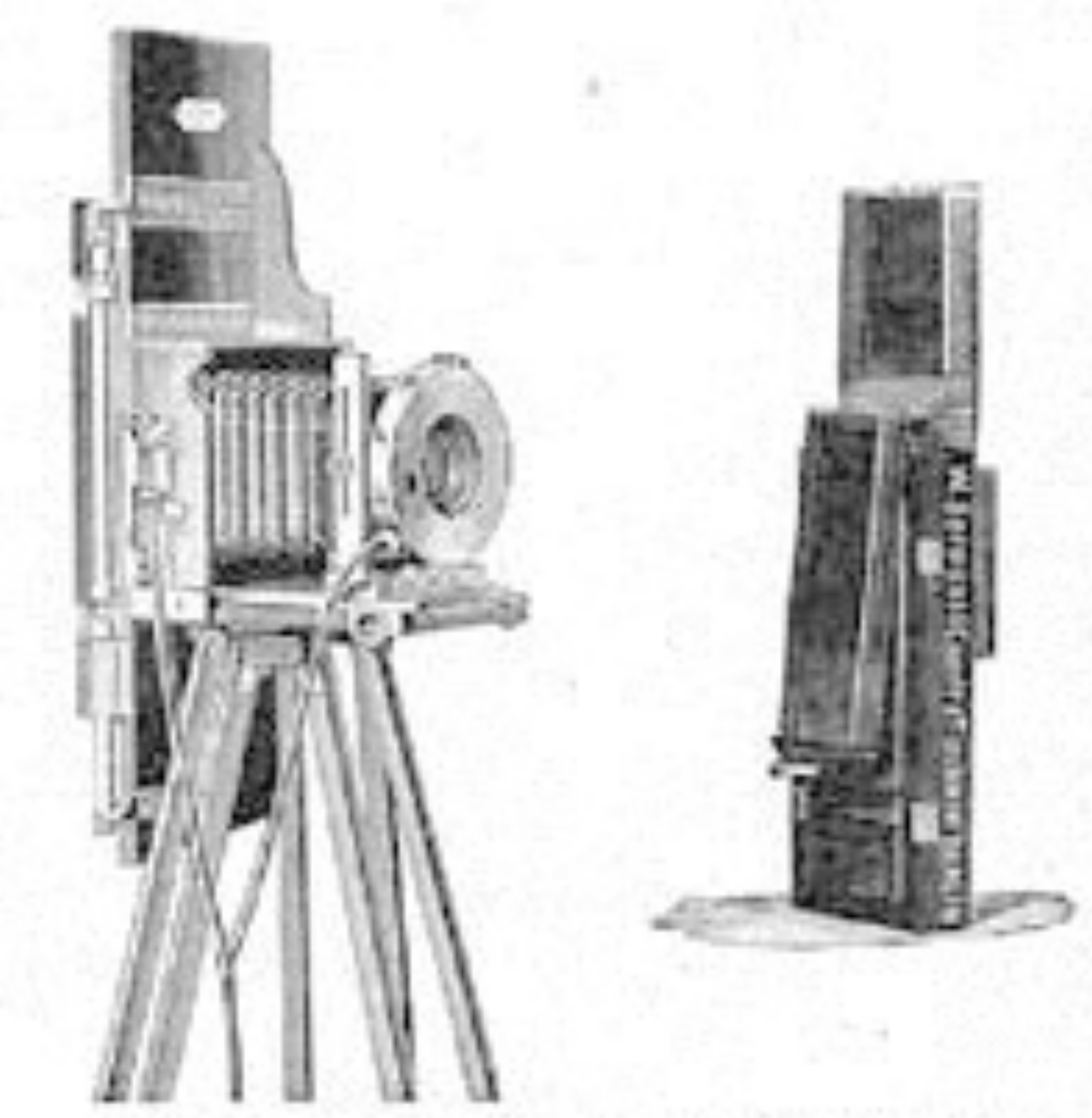
Photographer to the Tzar, 1863–1944

Shot sequentially through R, G, B filters

Printing technology not available, but could project w/ RGB filters!

Entire collection available: http://www.loc.gov/exhibits/empire/

Assignment 3



Sergey Prokudin-Gorsky,
Alim Khan,
emir of Bukhara (1911)

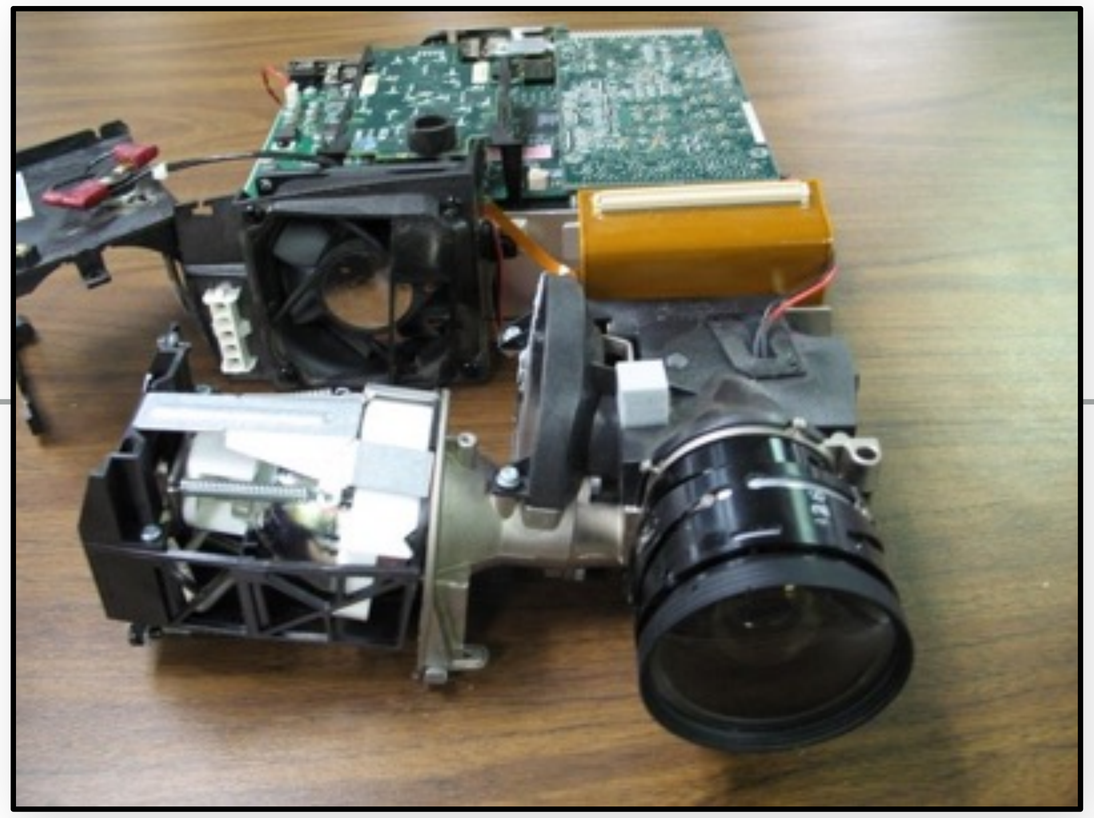
Sergey Prokudin-Gorsky, Pinkhus Karlinskii, Supervisor of the Chernigov Floodgate (1919)

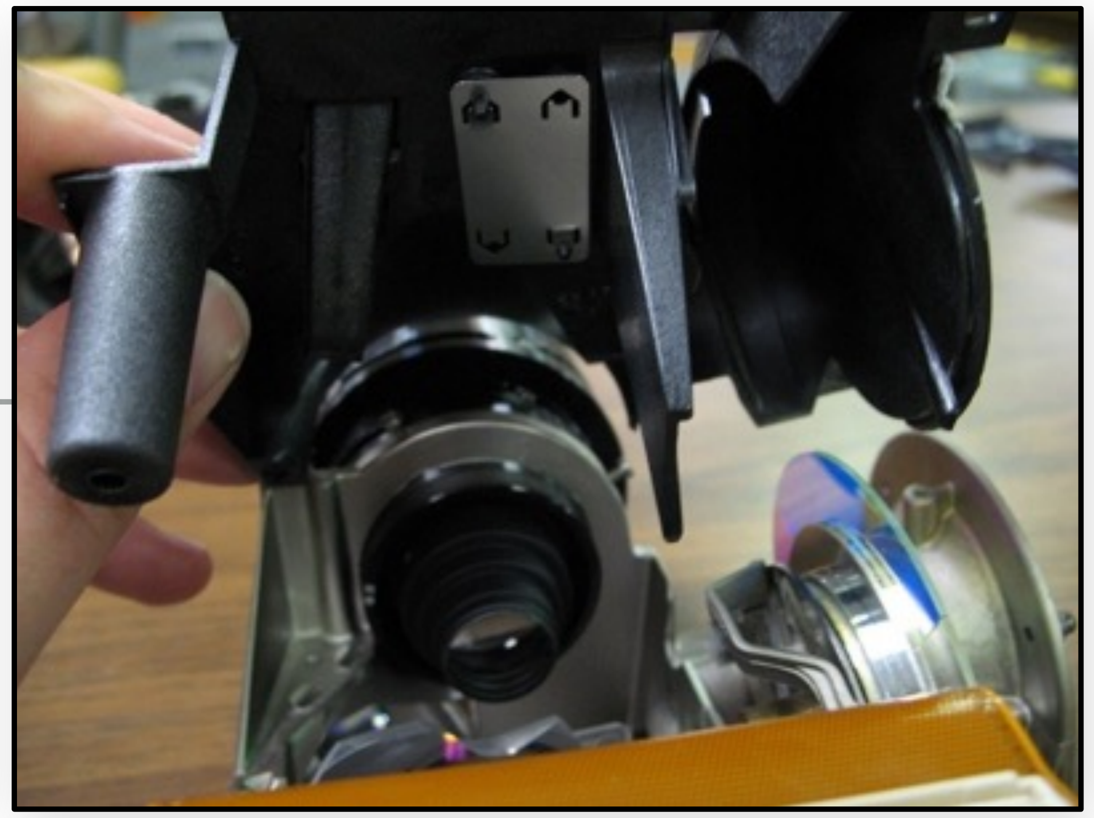
Color displays

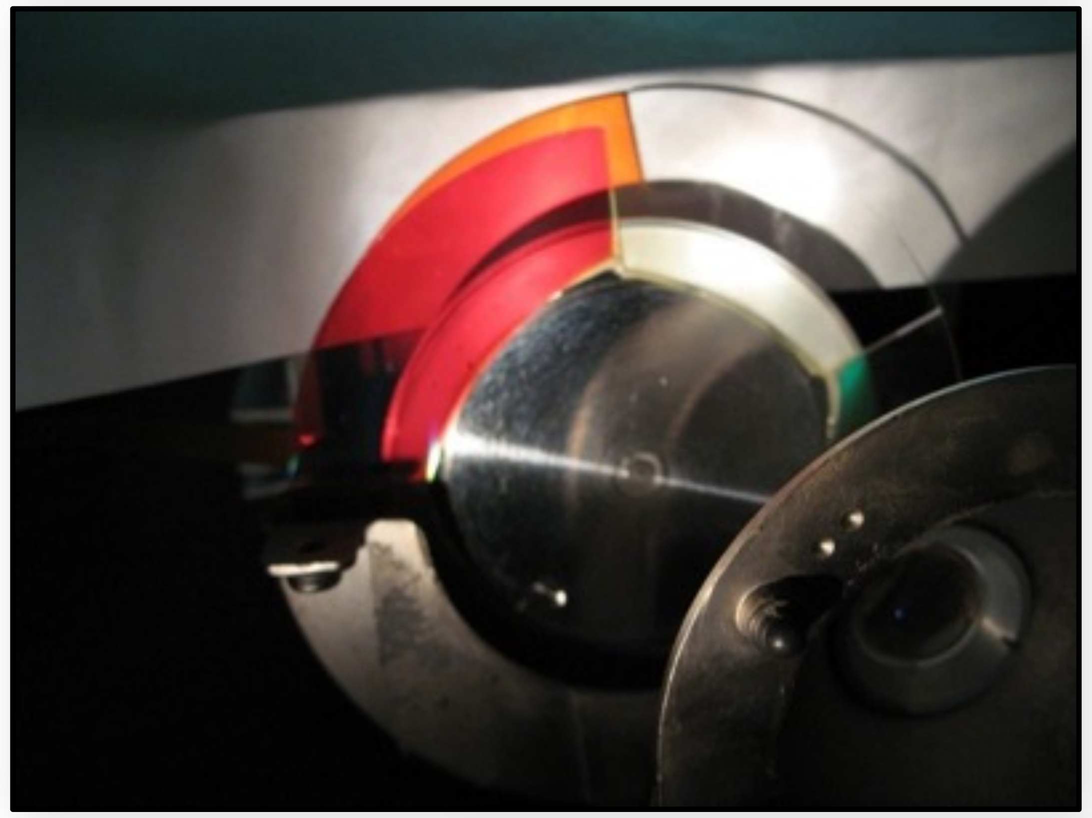
Temporal multiplexing

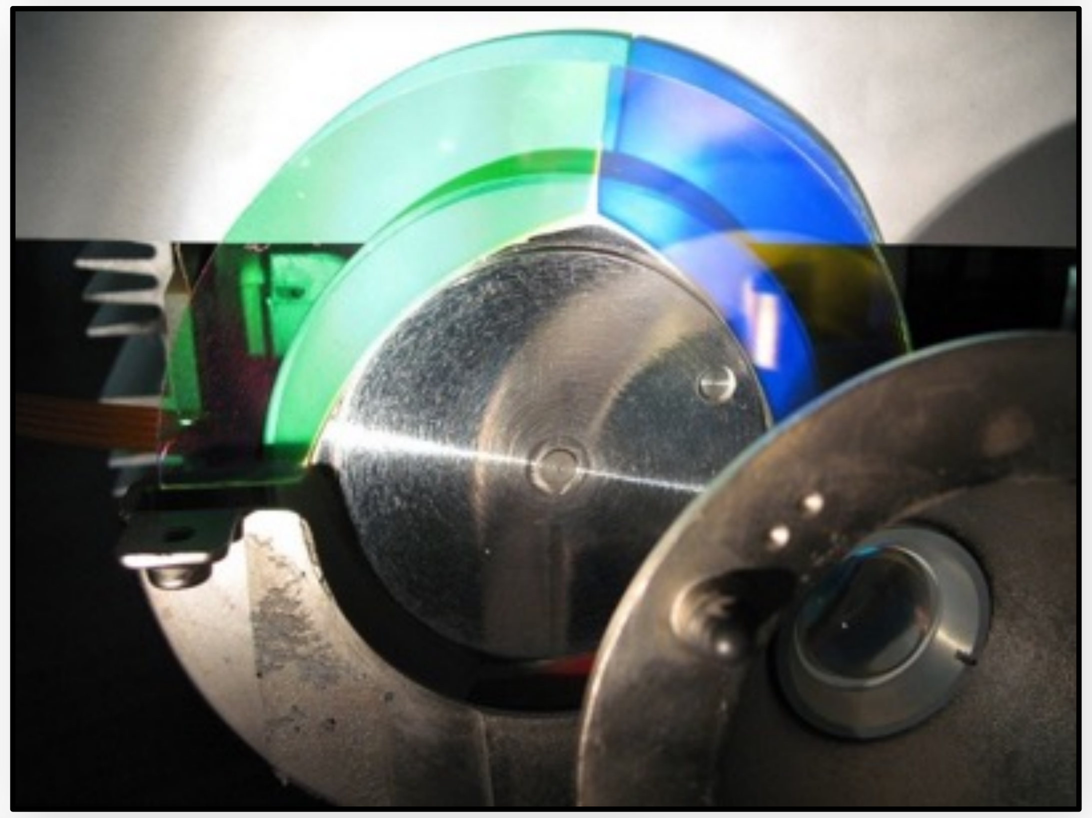
DLP projector

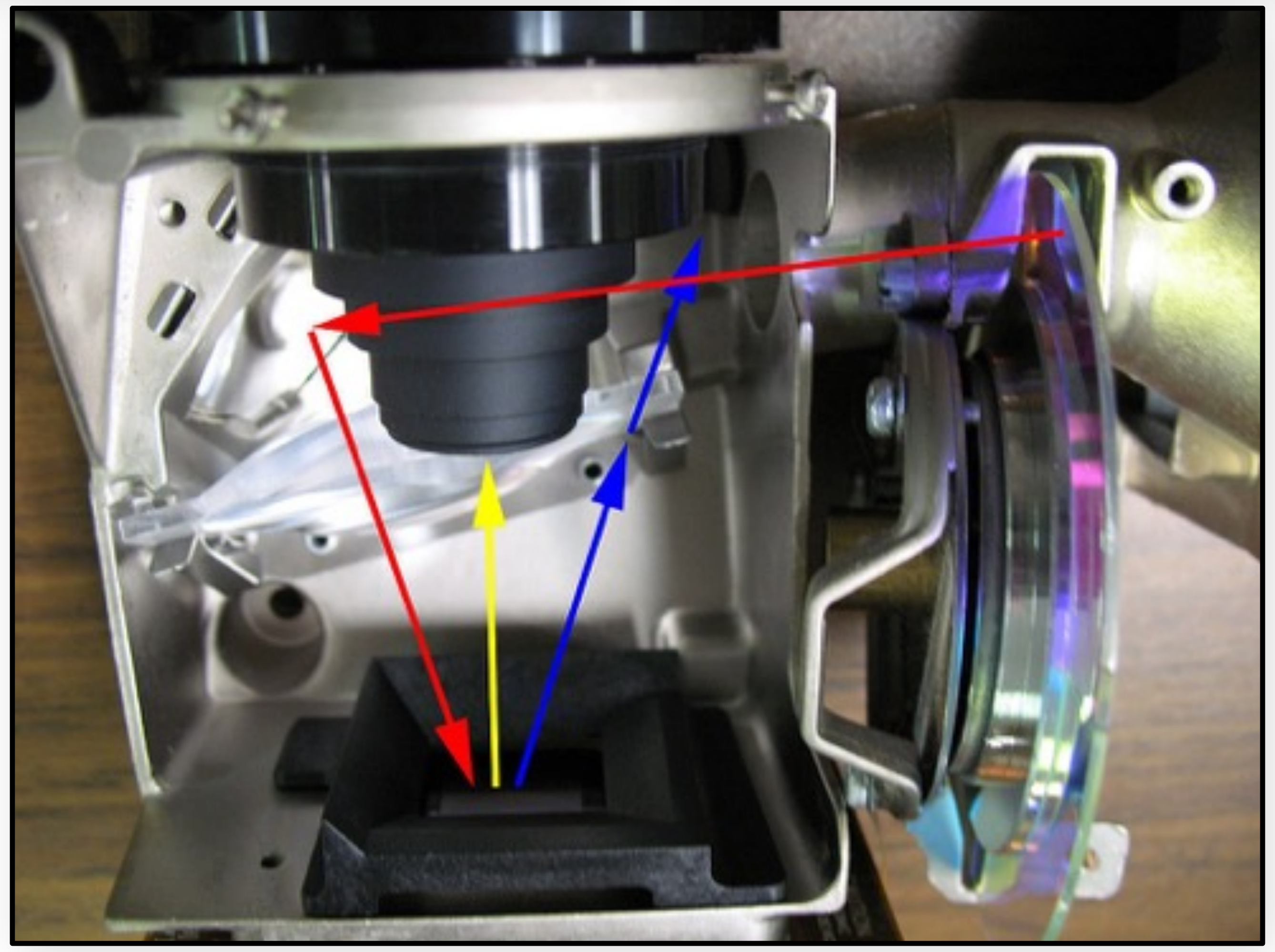
- http://en.wikipedia.org/wiki/Digital_Light_Processing



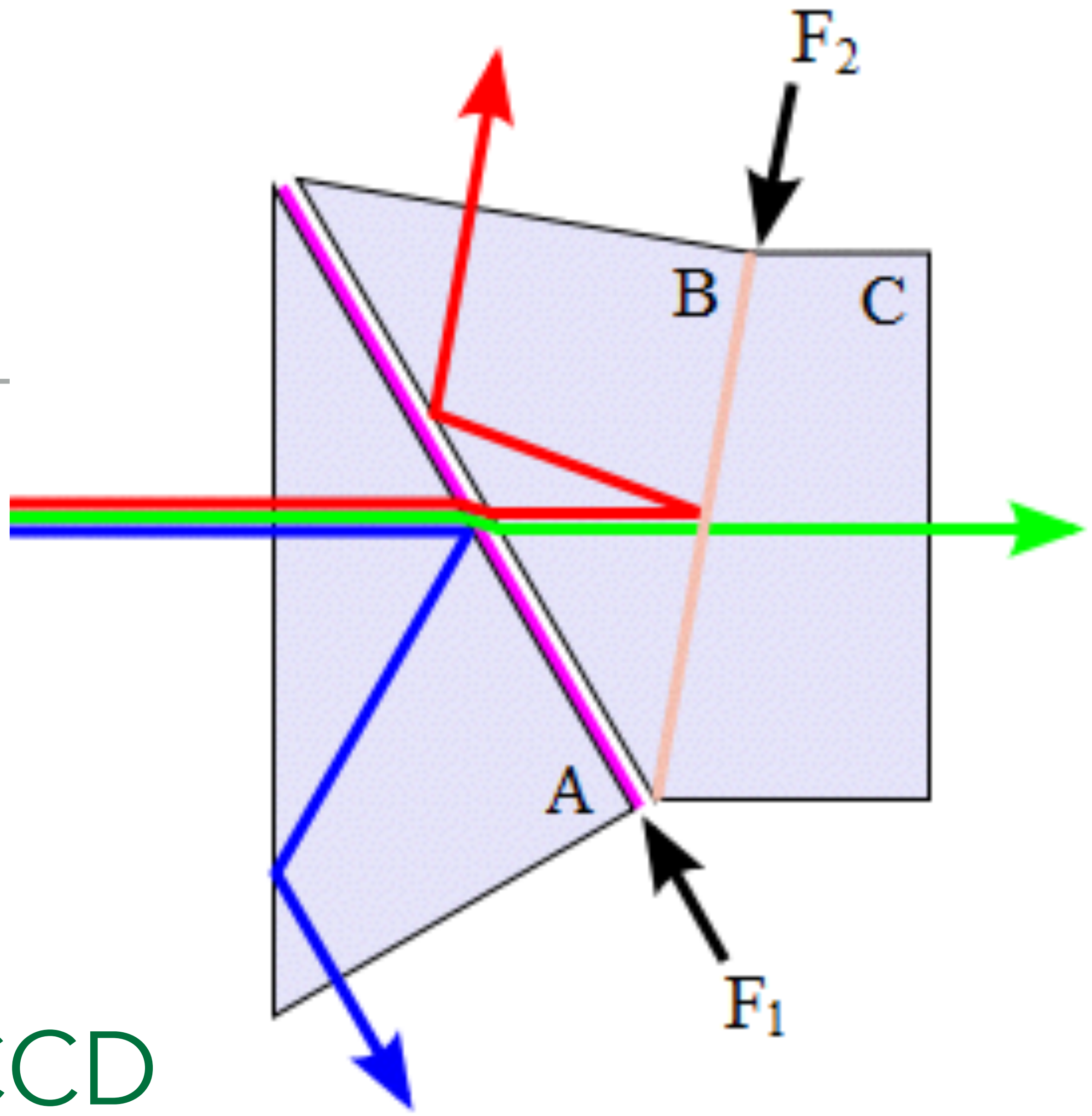








3 sensors + beam splitter



High-end 3-CCD video cameras

Use separation prisms

- prisms that split wavelengths

Pros

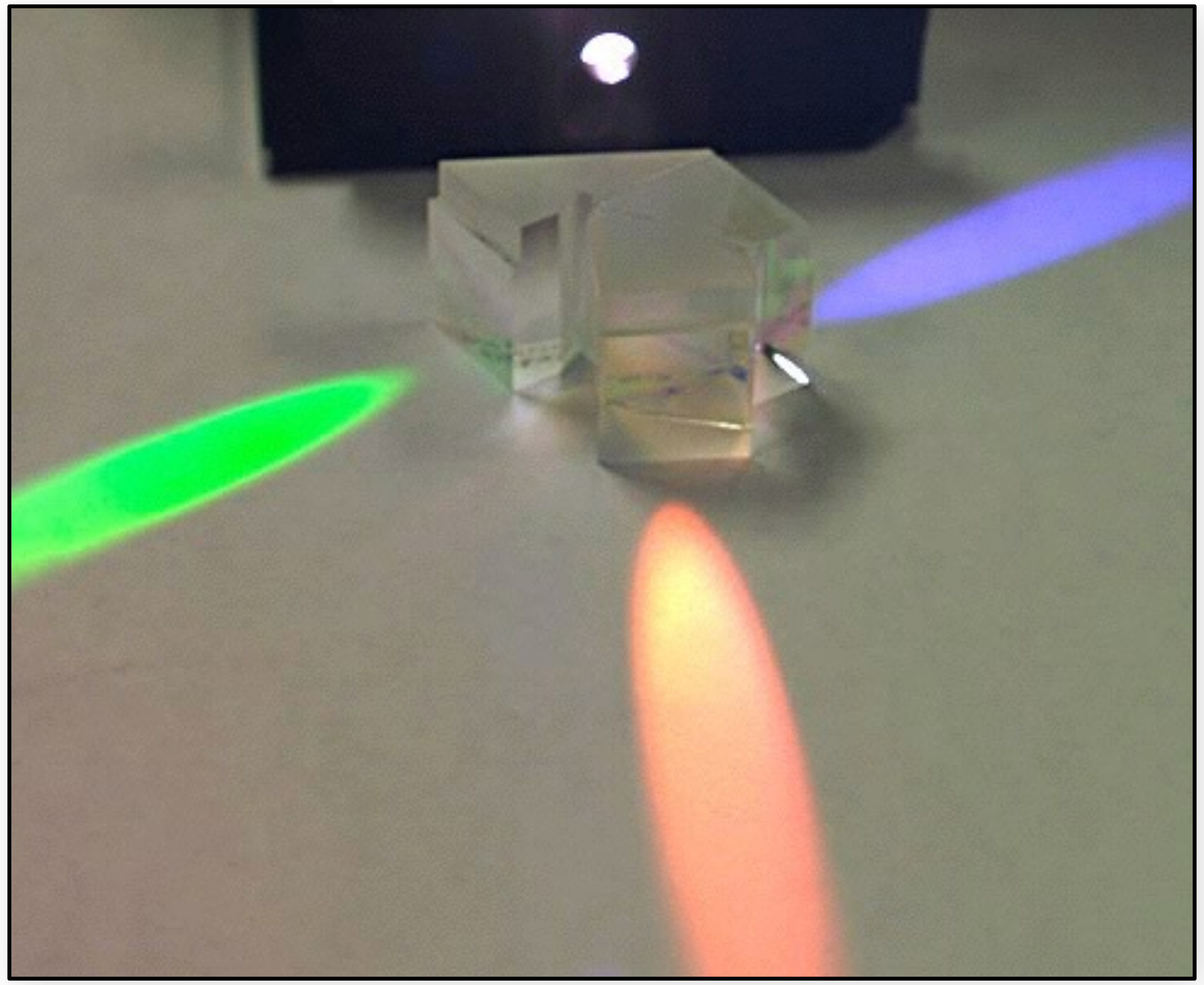
- 3 real values per pixel
- Little photon loss

Cons

- costly (needs 3 sensors)
- space

Trichroic beam splitter prism

http://en.wikipedia.org/wiki/3CCD



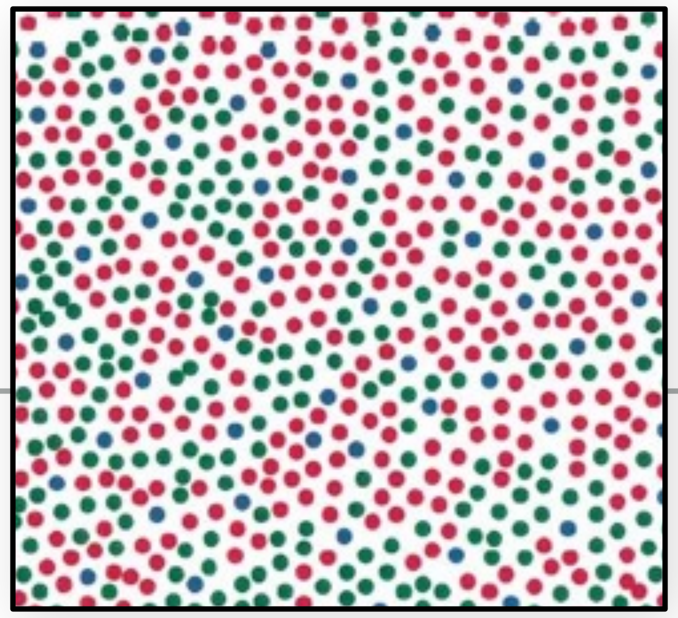
3 sensors + beam splitter

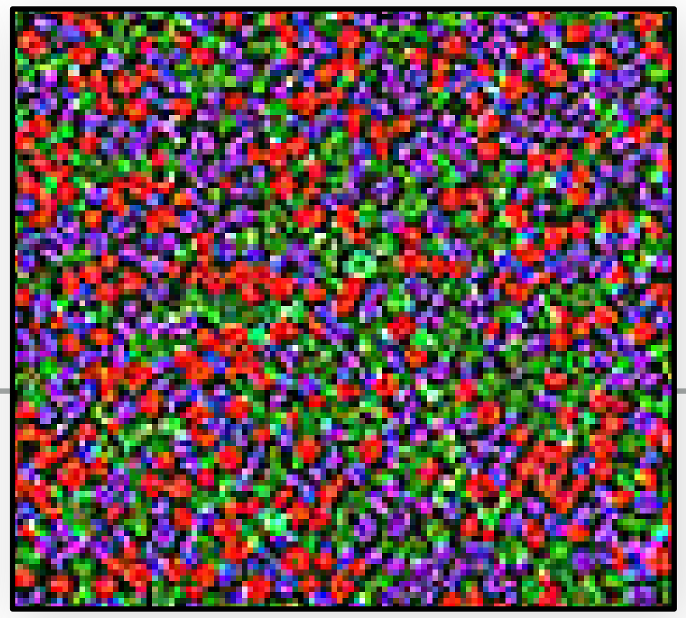
Technicolor

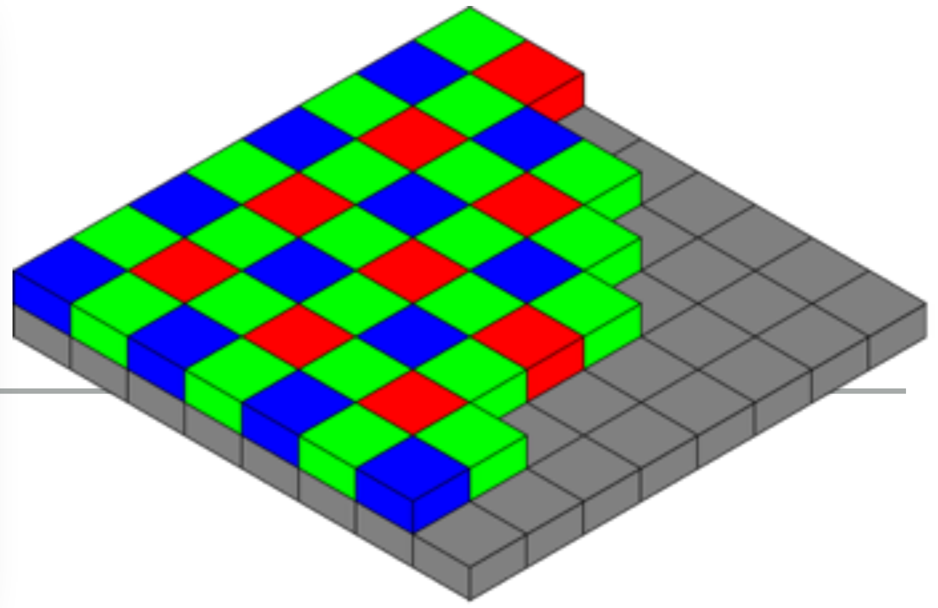
- 3 negatives exposed at once
- via beam splitter and filters
- large, heavy cameras;
  cumbersome printing process

Wizard of Oz (1939)

Spatial multiplexing







Human eye (cone mosaic), older color film, bayer mosaic/CFA (color filter array)

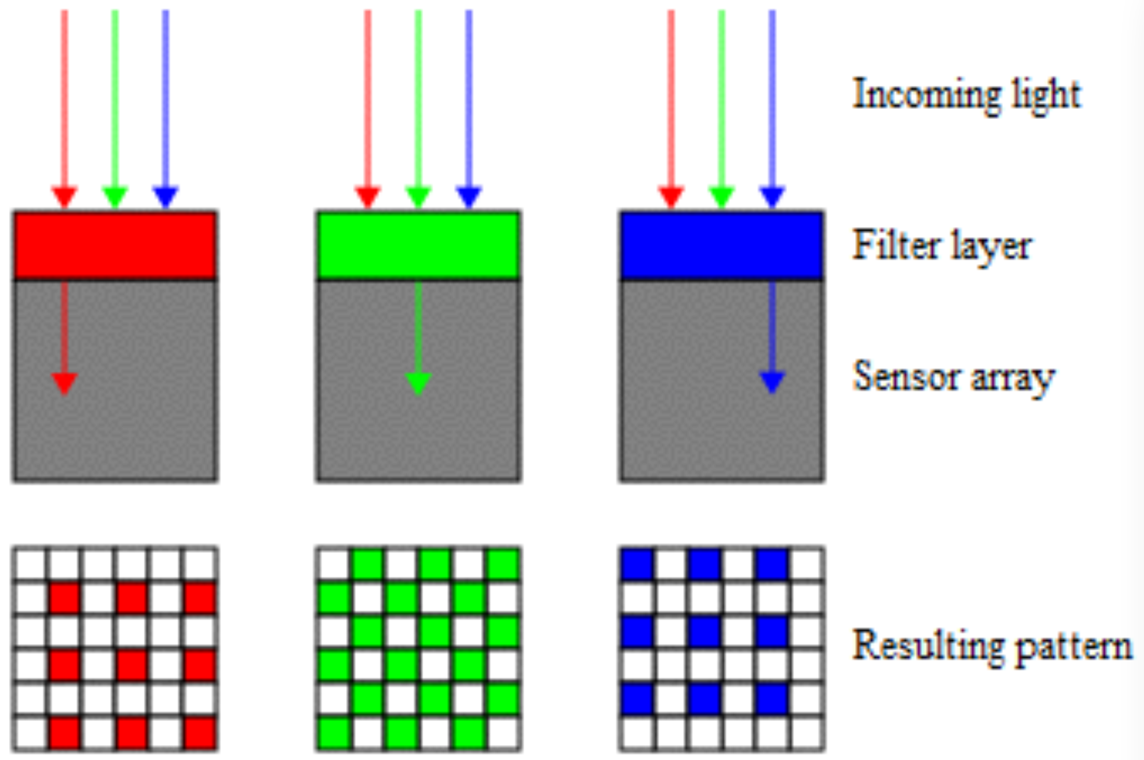
Most still cameras, most cheap camcorder, some high-end video cameras (RED)

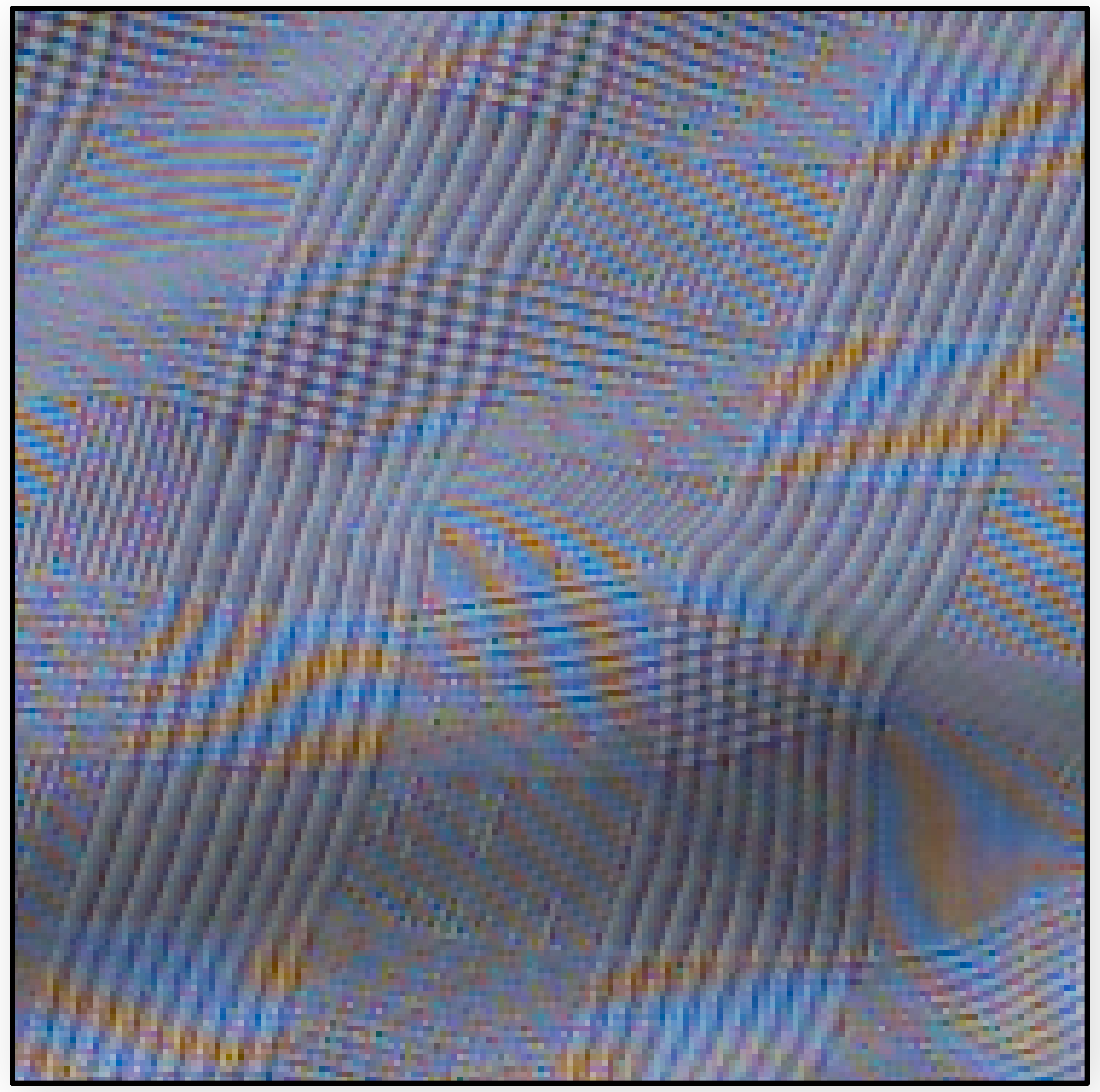
Pros

- single sensor
- well mastered technology, high resolution

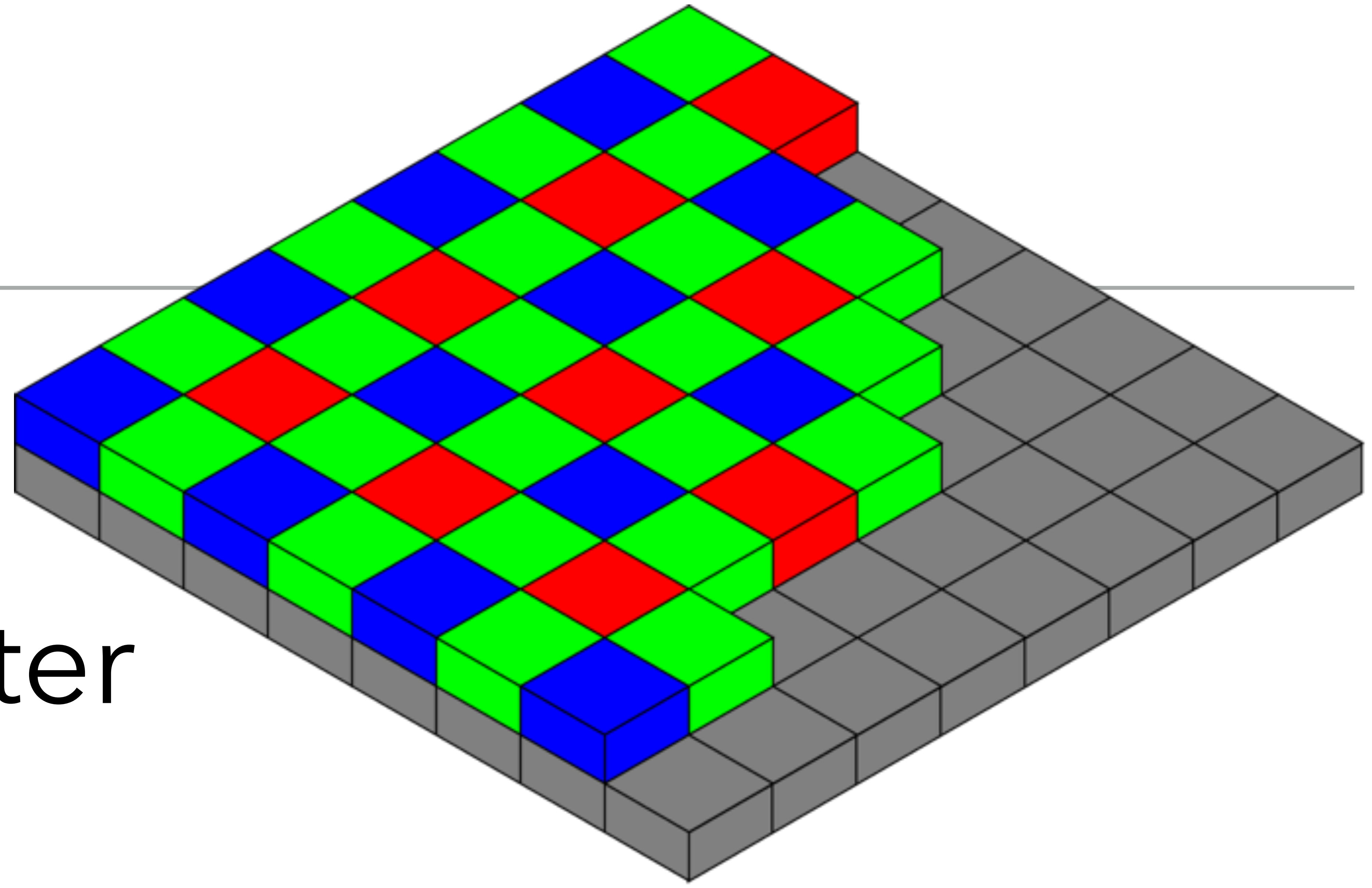
Cons

- needs interpolation, color jaggies
- requires antialiasing filter (reduces sharpness)
- loss of light





Spatial multiplexing

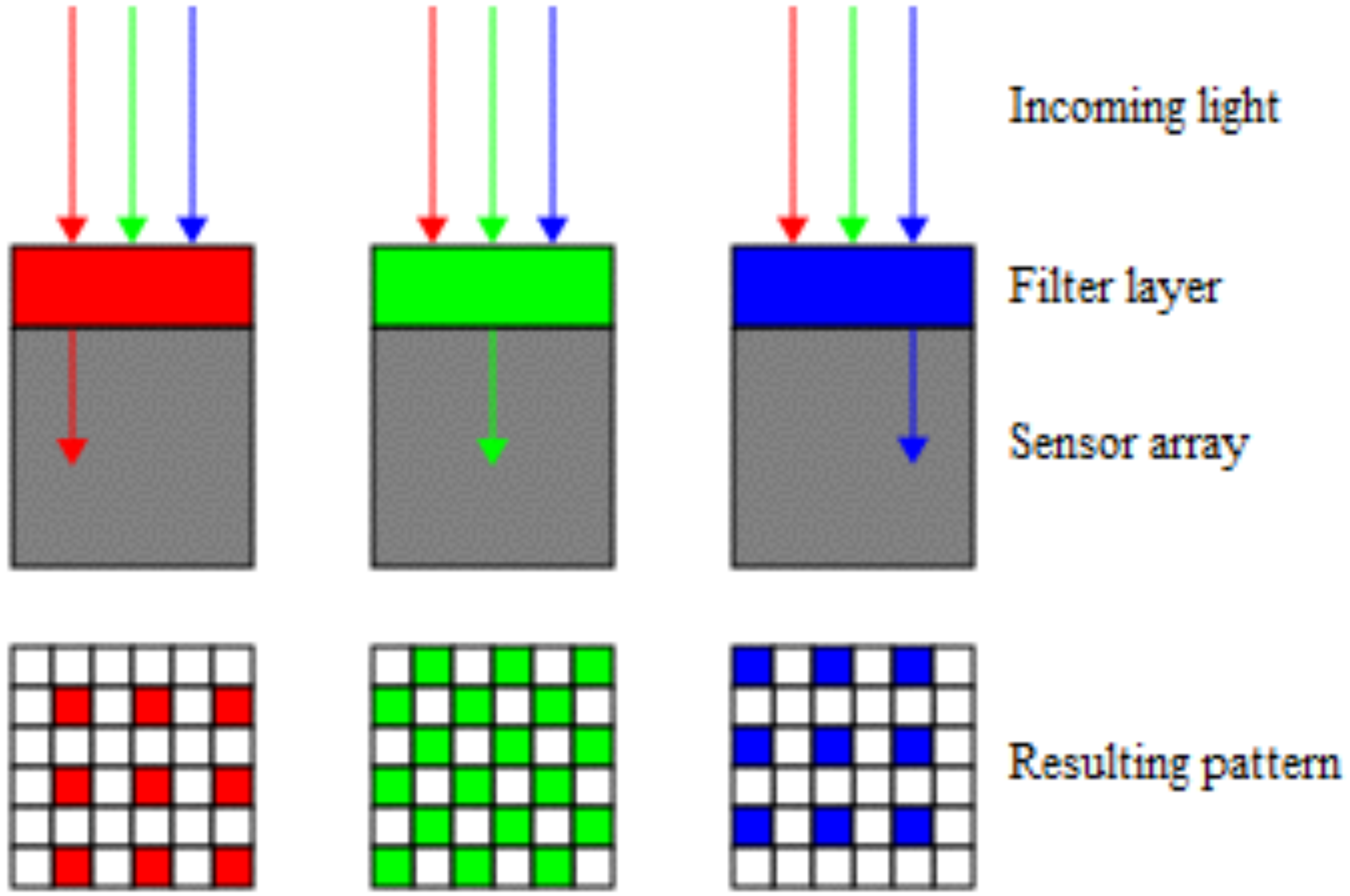


Bayer filter

- http://en.wikipedia.org/wiki/Bayer_filter
- Most common in digital cameras

2x2 pattern

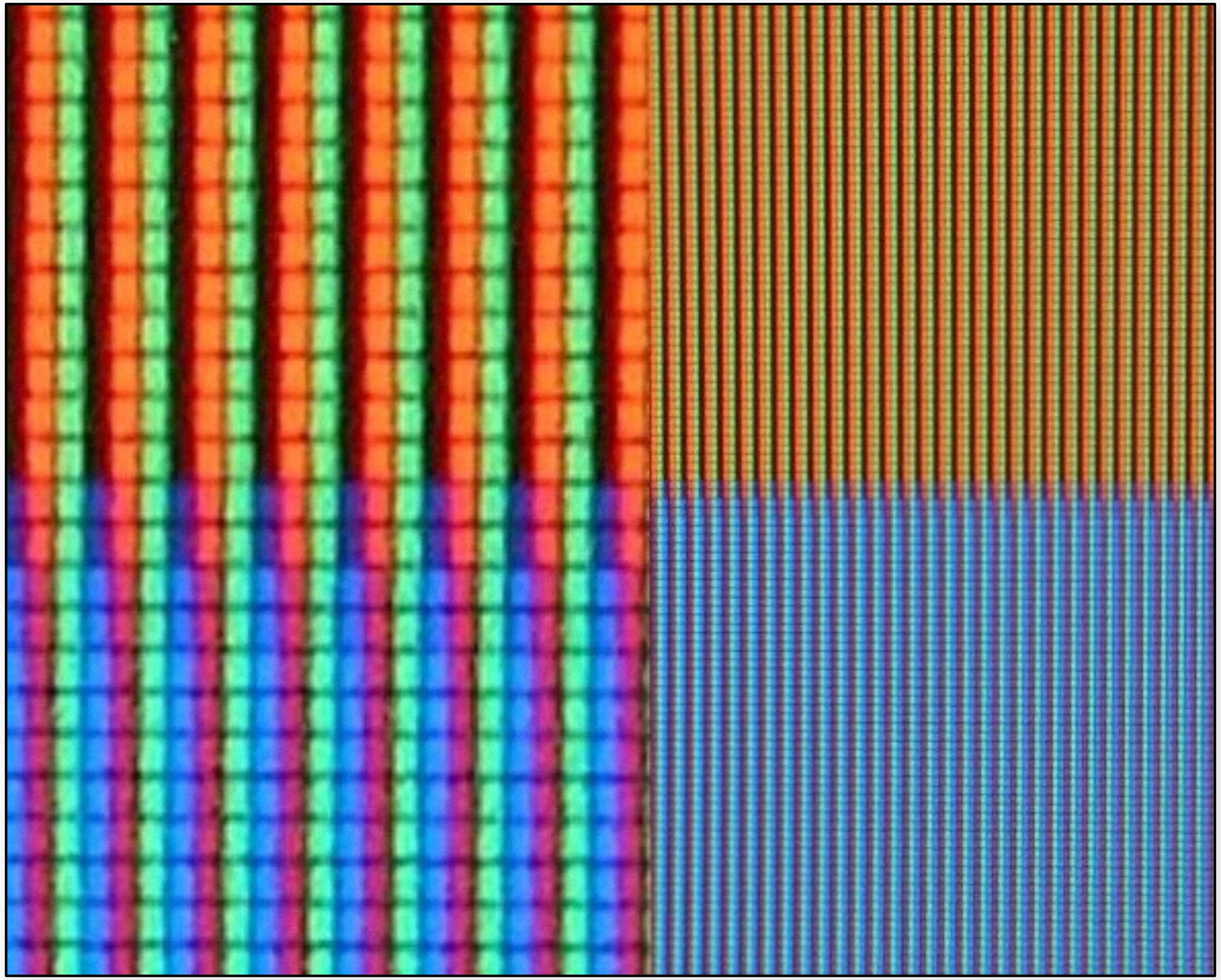
- 2 green, 1 red, 1 blue



Other mosaics exist, but not as widespread

Color displays

Spatial multiplexing



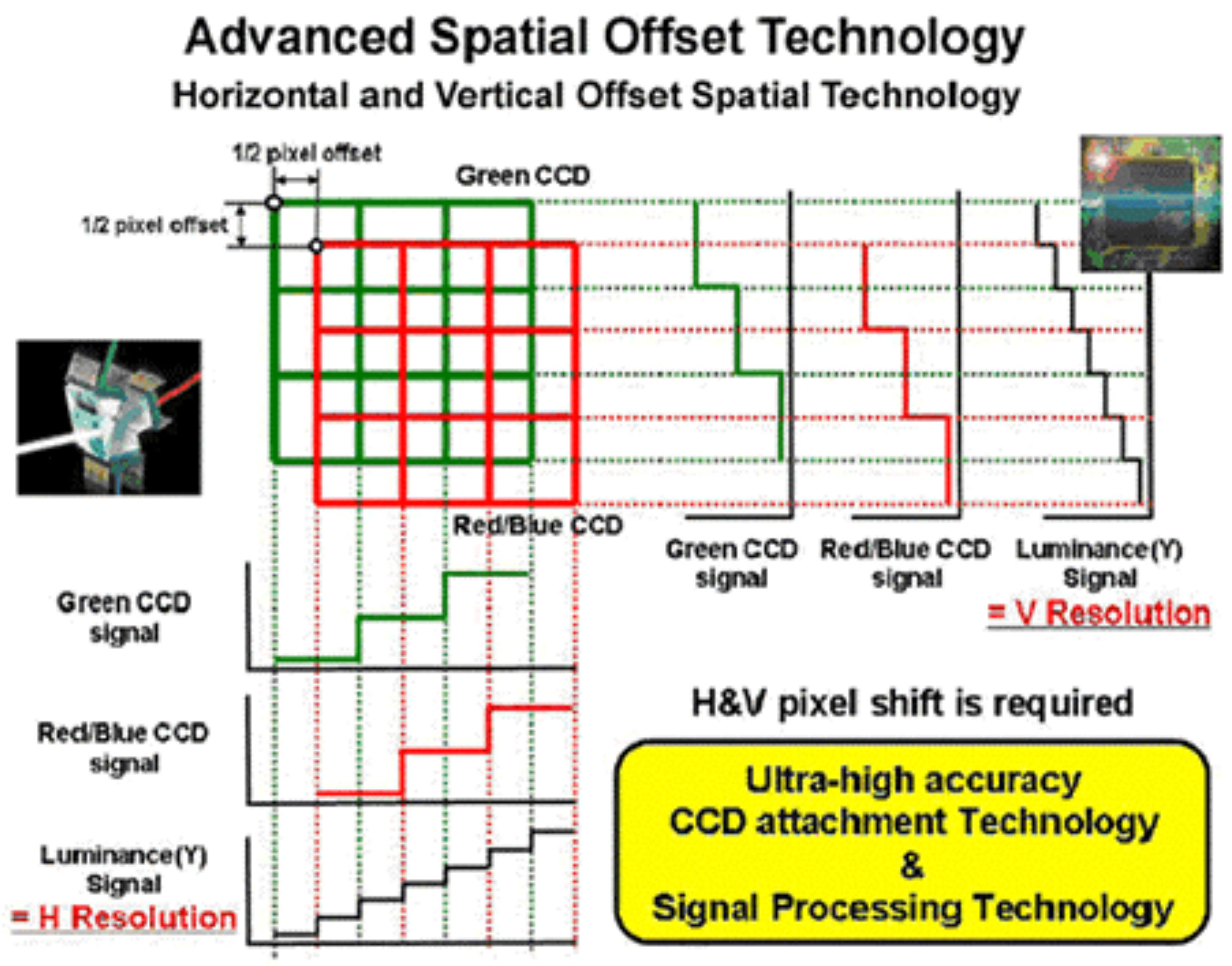
http://en.wikipedia.org/wiki/RGB_color_model

Combination: pixel shift

3-ccd with prisms + spatial multiplexing

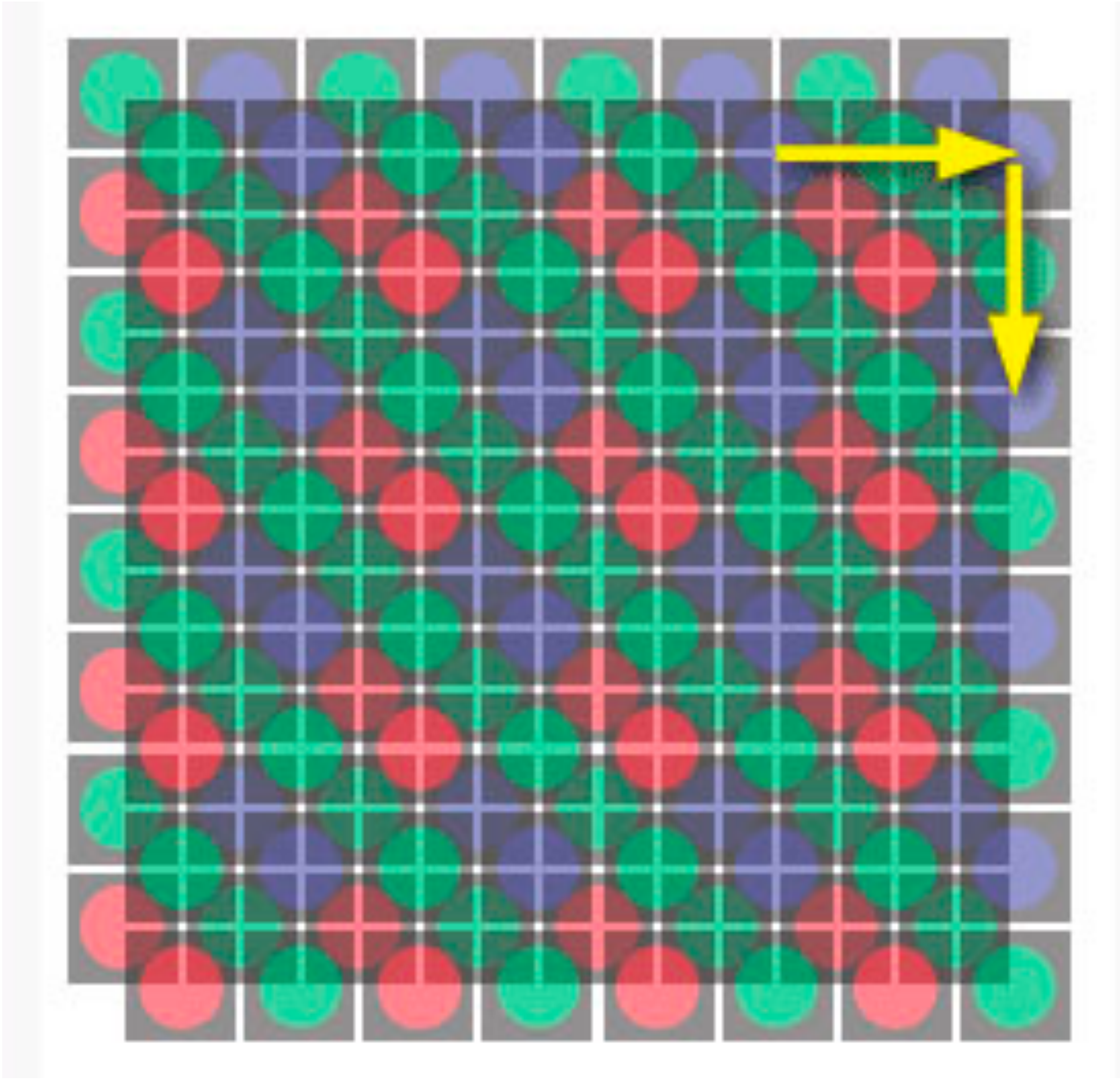
The 3 ccds are shifted by 1/2 pixel to provided resolution increase

- usually selectable (not shifted for lower-res, shifted to get HD)
- Often horizontal only



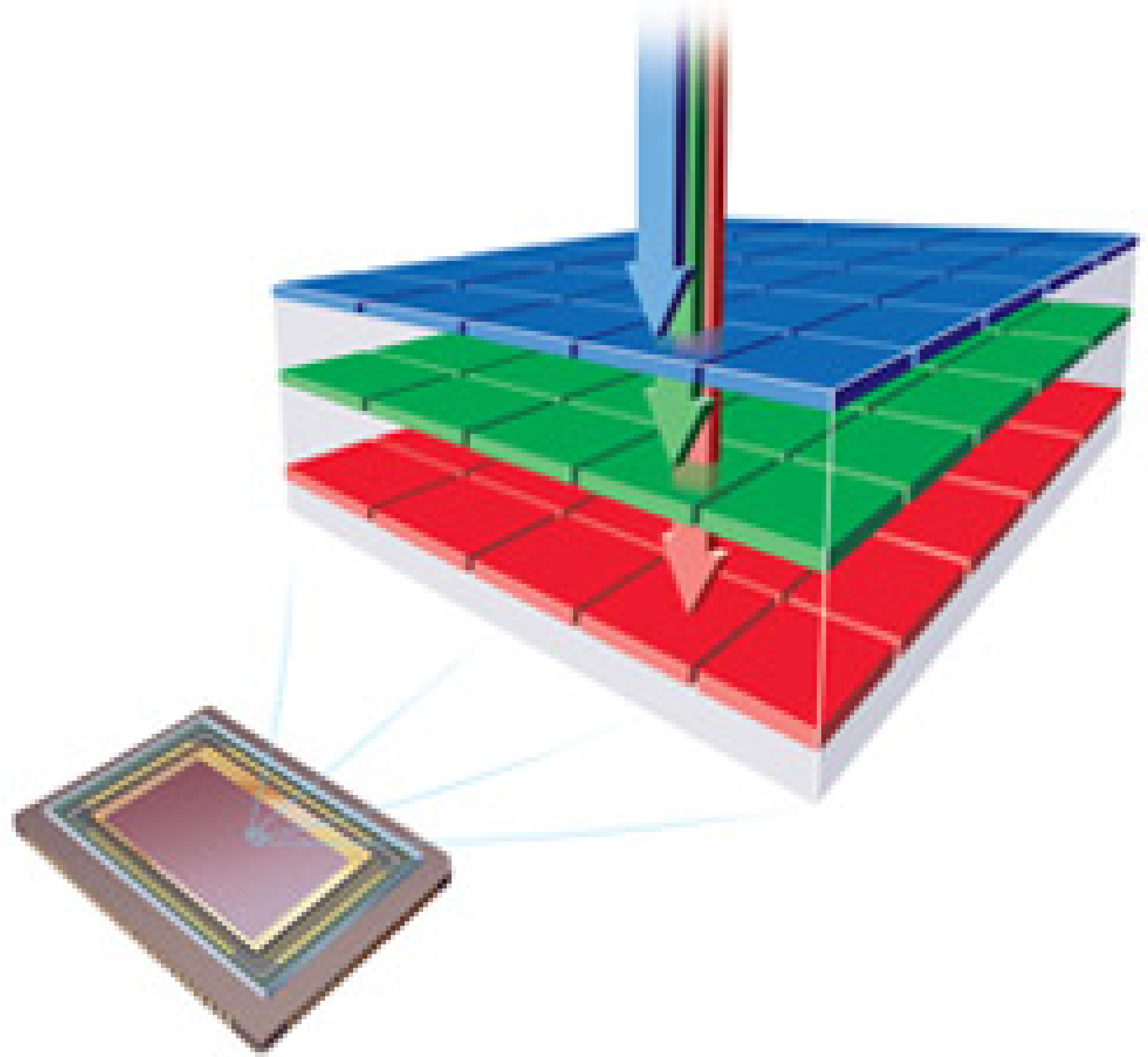
From Panasonic

http://www.petapixel.com/2011/05/26/hasselblad-h4d-200ms-shoots-200mp-photos-with-a-50mp-sensor/



Depth multiplexing (Foveon X3 sensor)

Leverage difference in absorption per wavelength

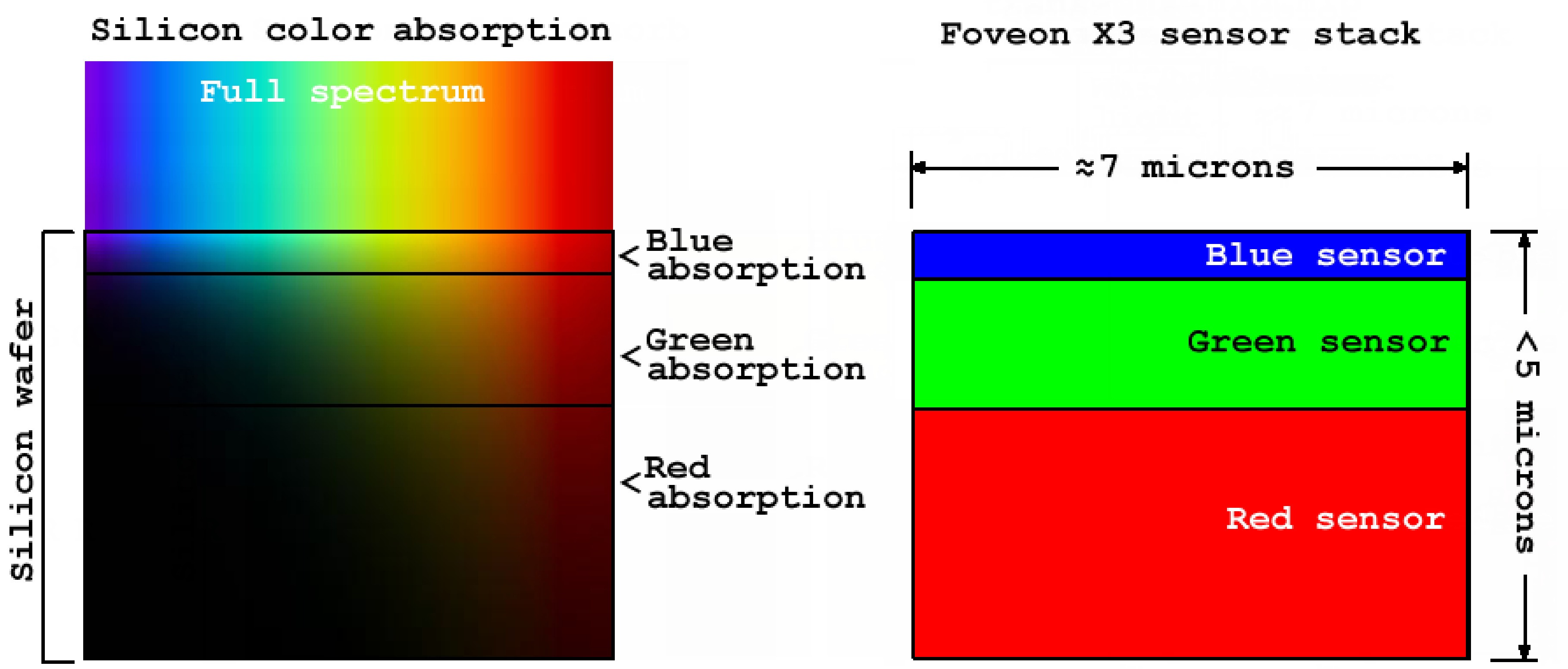


Pros

- 3 real numbers per pixel
- Less light loss

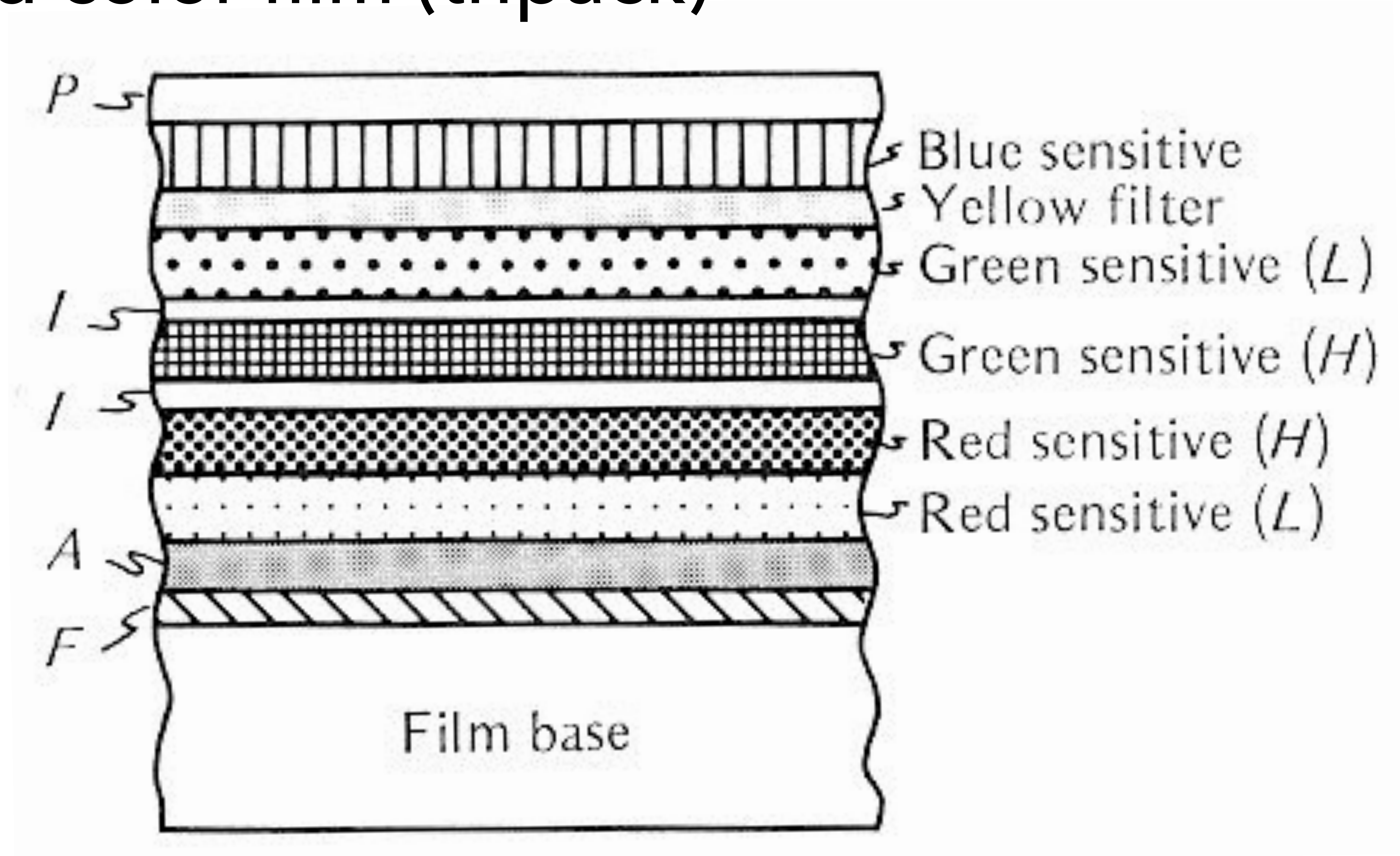
Cons

- Requires more color processing (3 numbers must be multiplied by matrix to get RGB)
- Tends to be noisier (because color processing and because shallow blue layer)
- Lower resolution these days

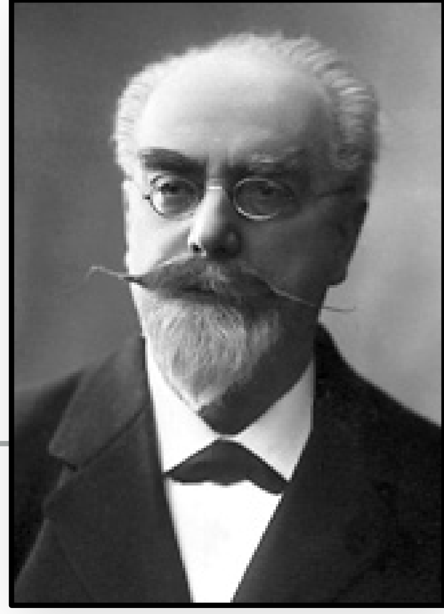


Depth multiplexing

Good old color film (tripack)

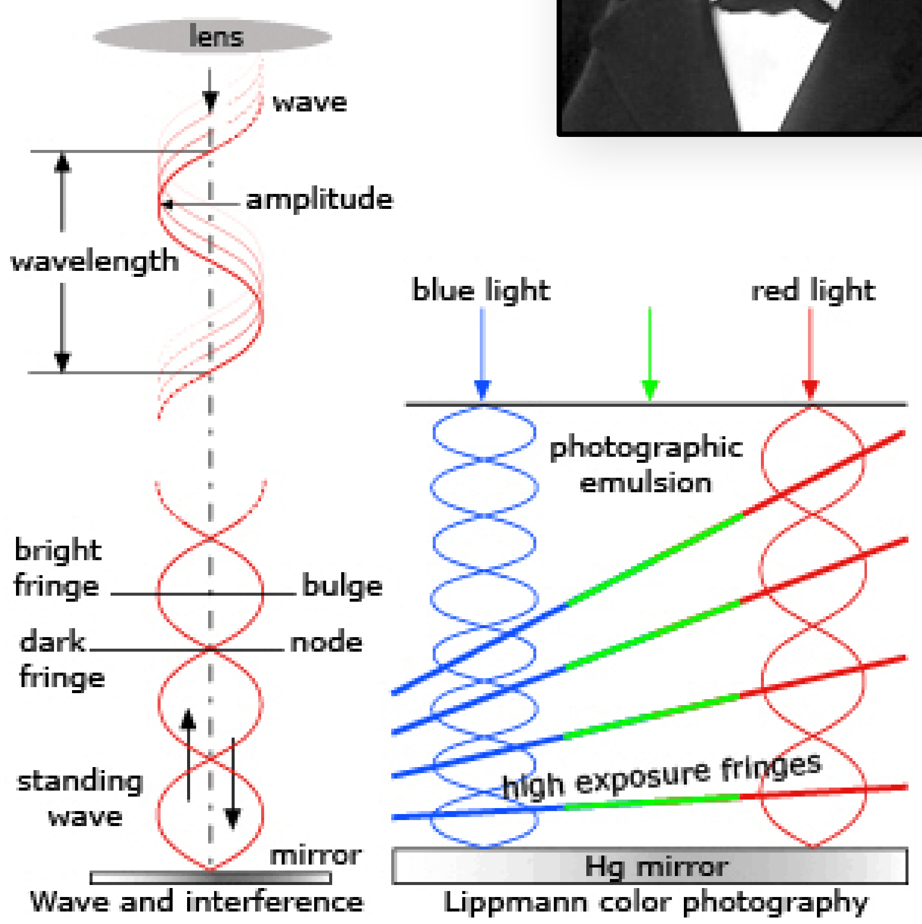


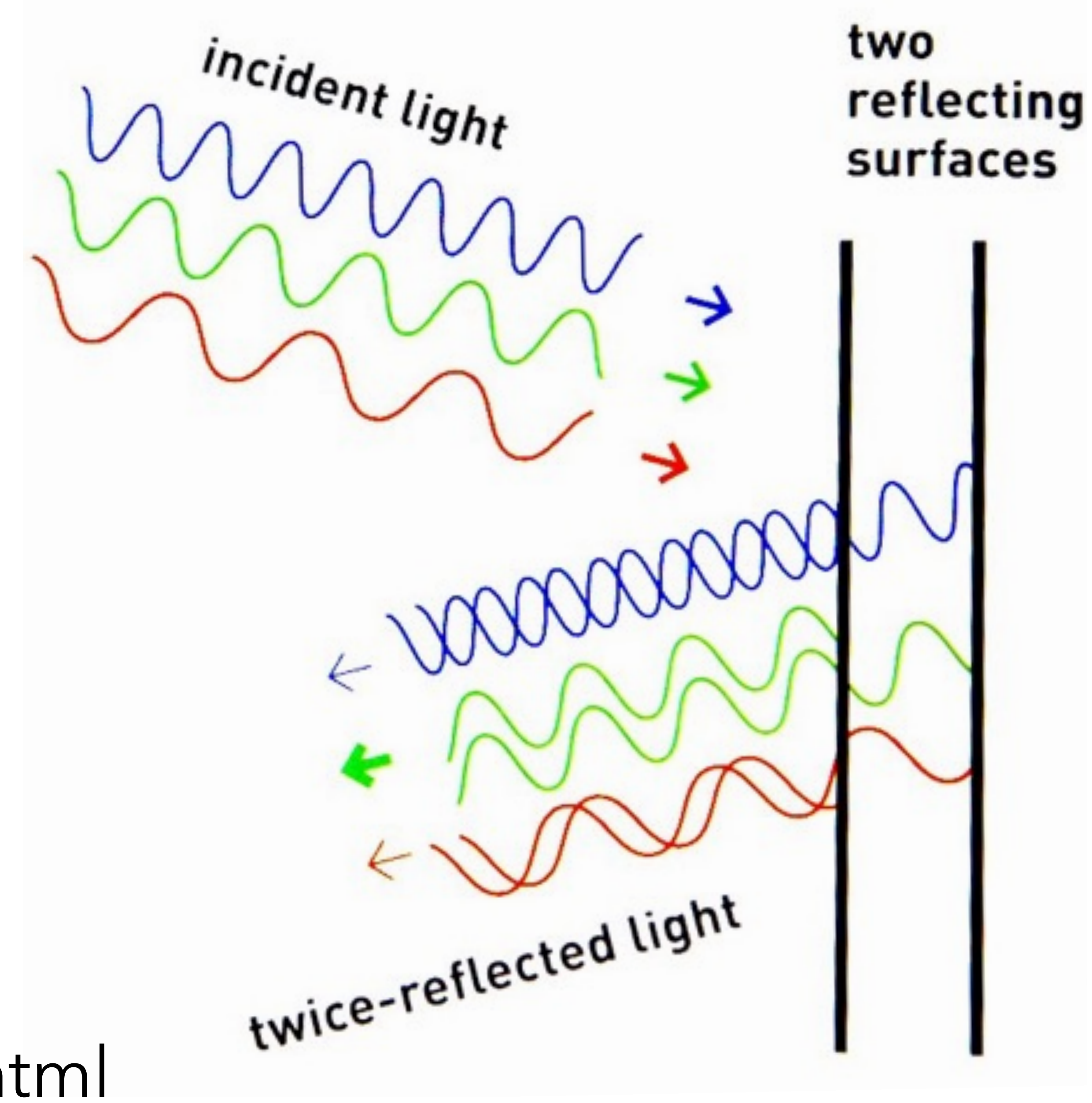
Interferences (Lippmann process)



Metal mirror to create interferences

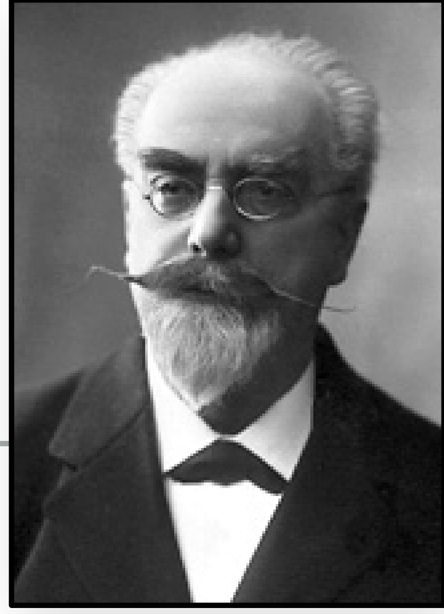
- ancestor of holography
- similar to colors in thin oil film





http://nobelprize.org/nobel_prizes/physics/articles/biedermann/index.html

Interferences (Lippmann process)



Metal mirror to create interferences

- ancestor of holography
- similar to colors in thin oil film

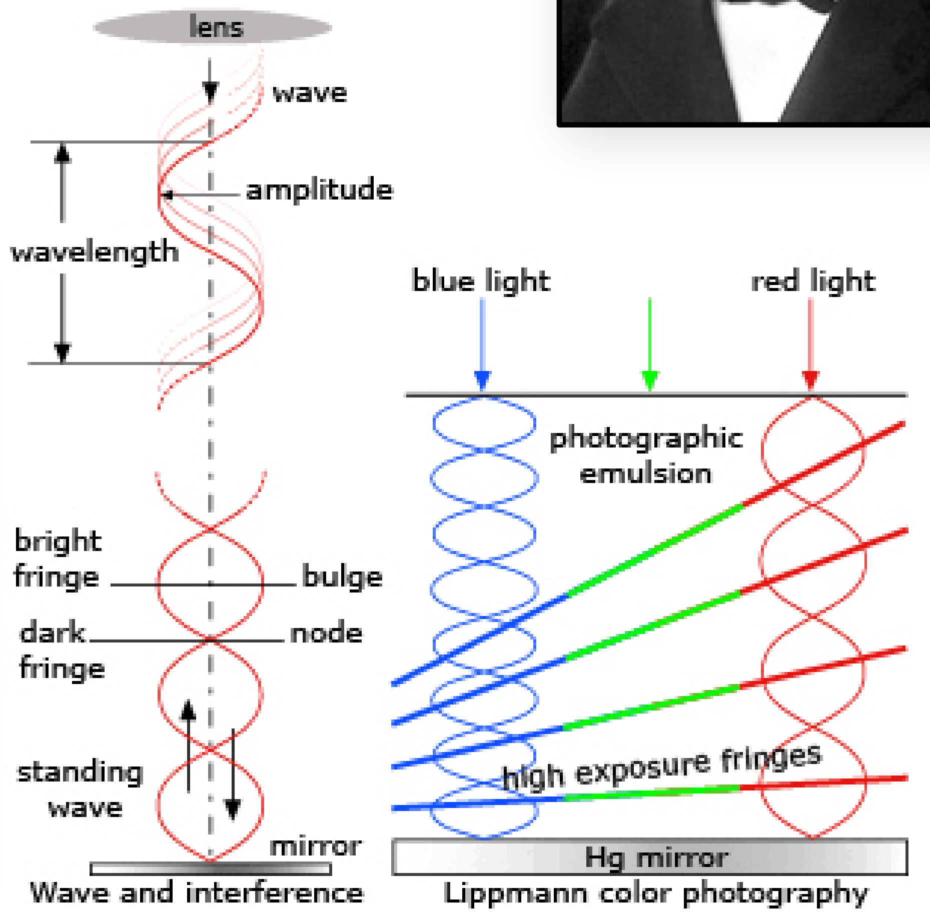
Pros

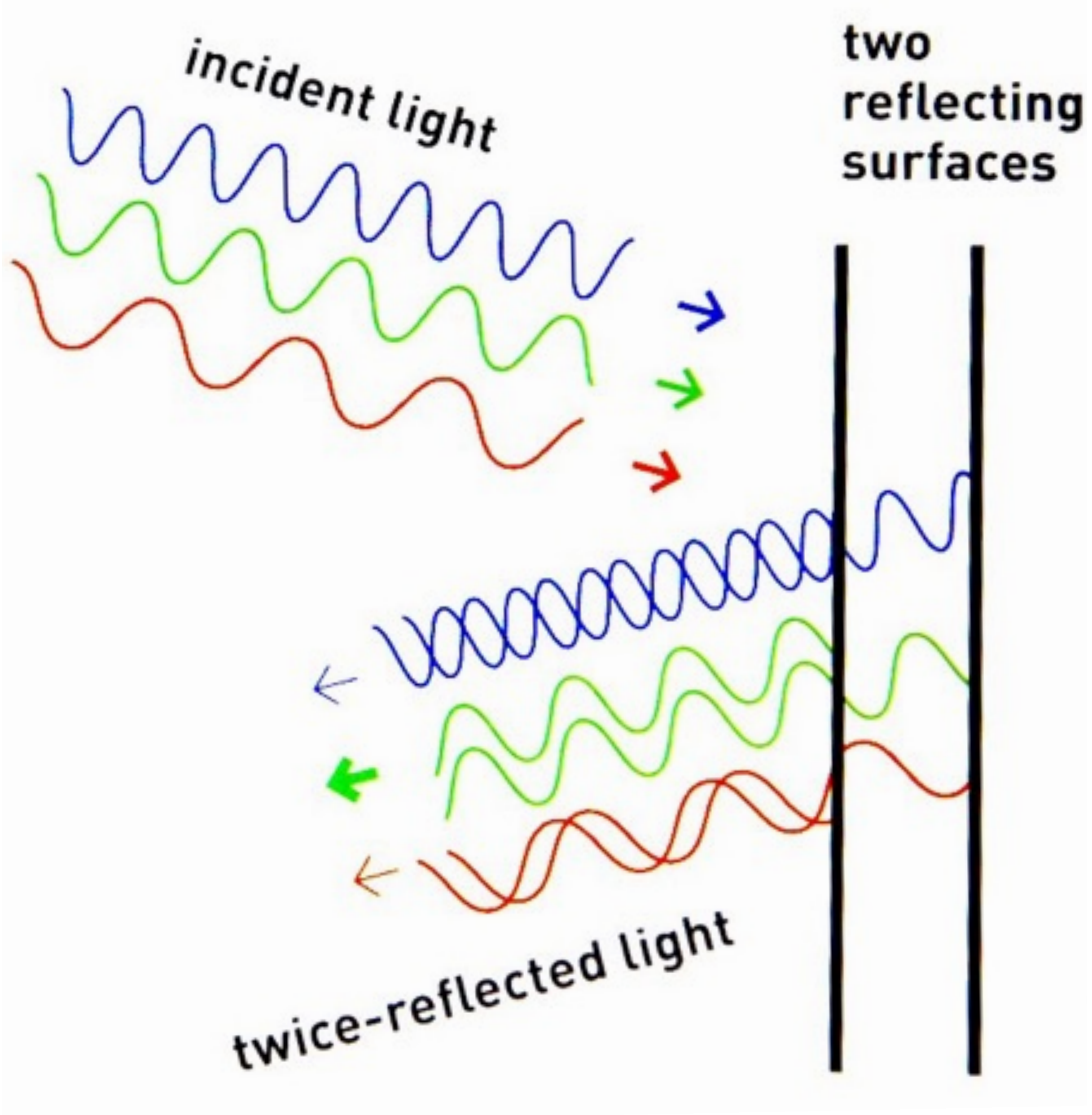
- Full spectrum!!!!!
- Gets you the Nobel if you invent it ;-)

Cons

- Needs high-resolution sensor/film
- limited field of view for display

'Saint-Maxime'', 1891-1899
Photographer: Gabriel Lippmann

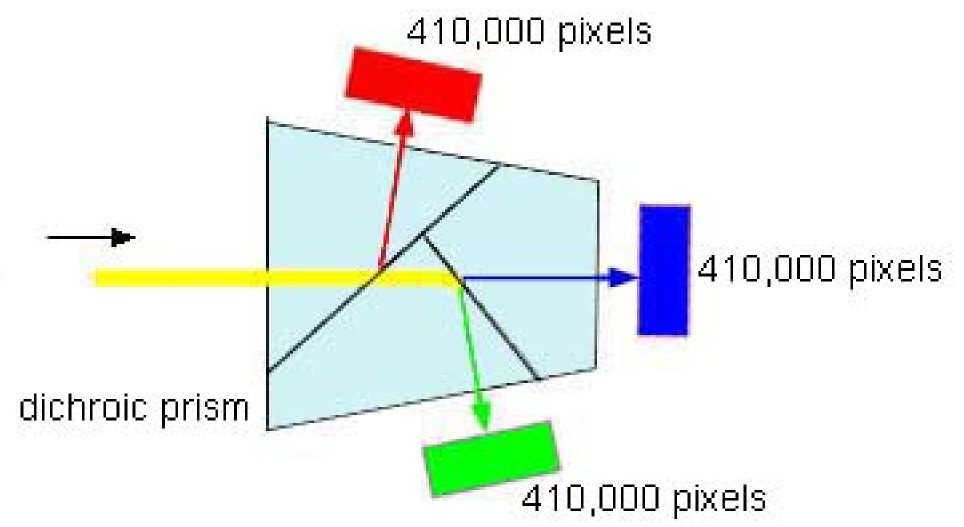


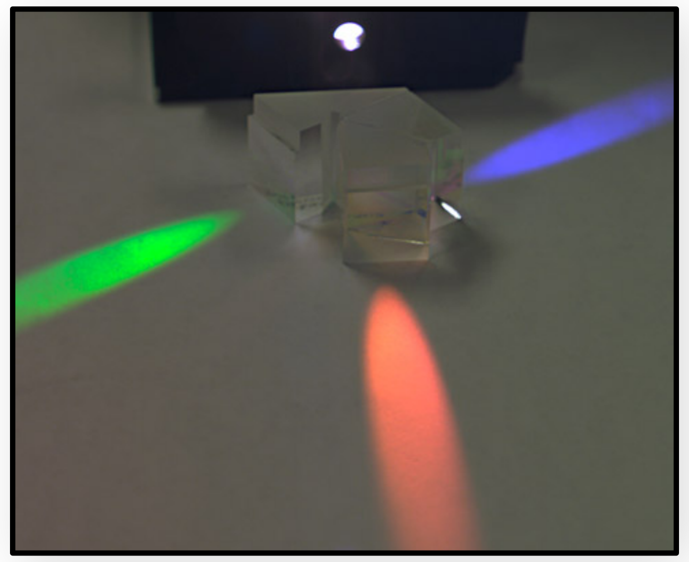


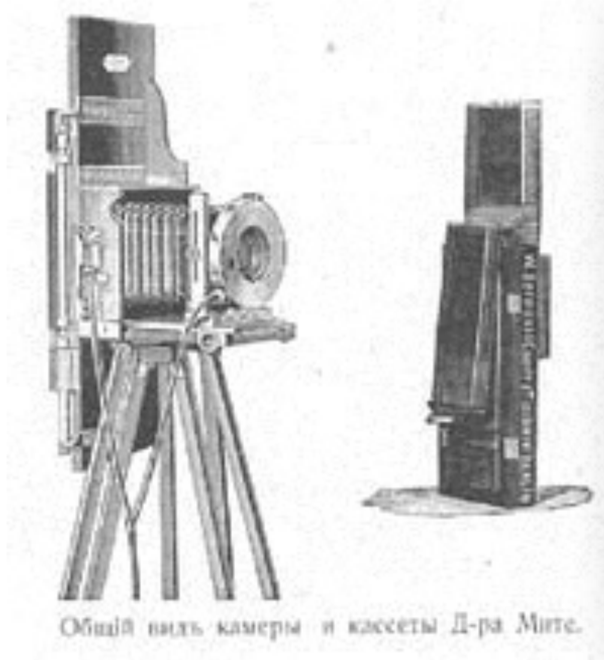
Recap & Questions?

Scan 3 times (temporal multiplexing)

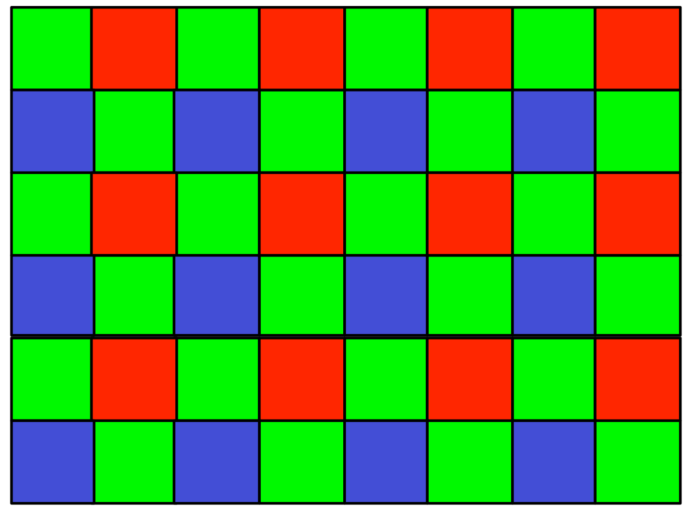
Use 3 detectors
(3-ccd camera)



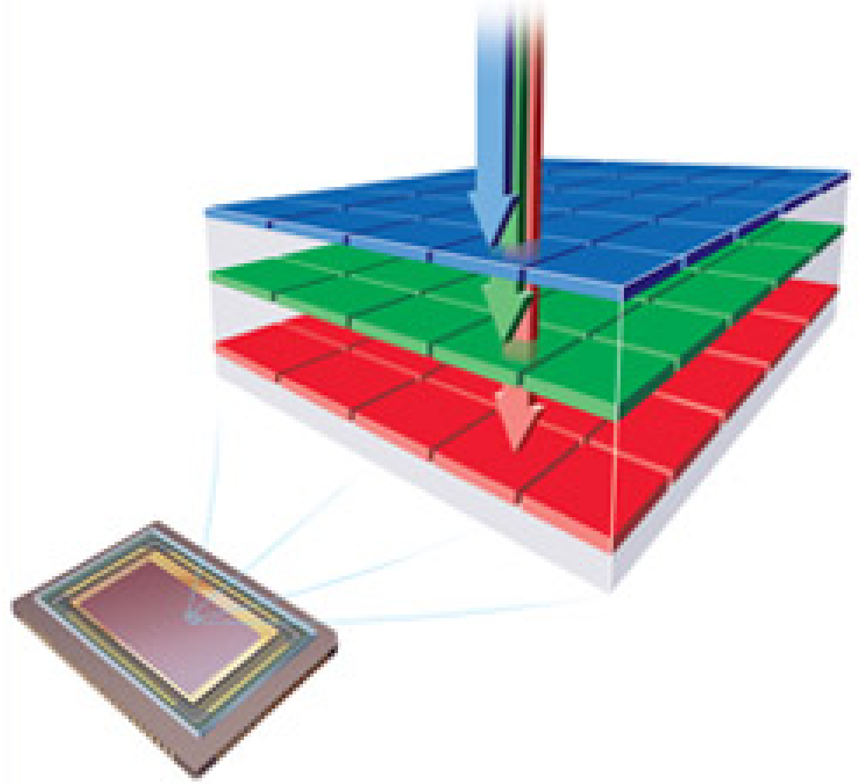




Use offset color samples (spatial multiplexing)



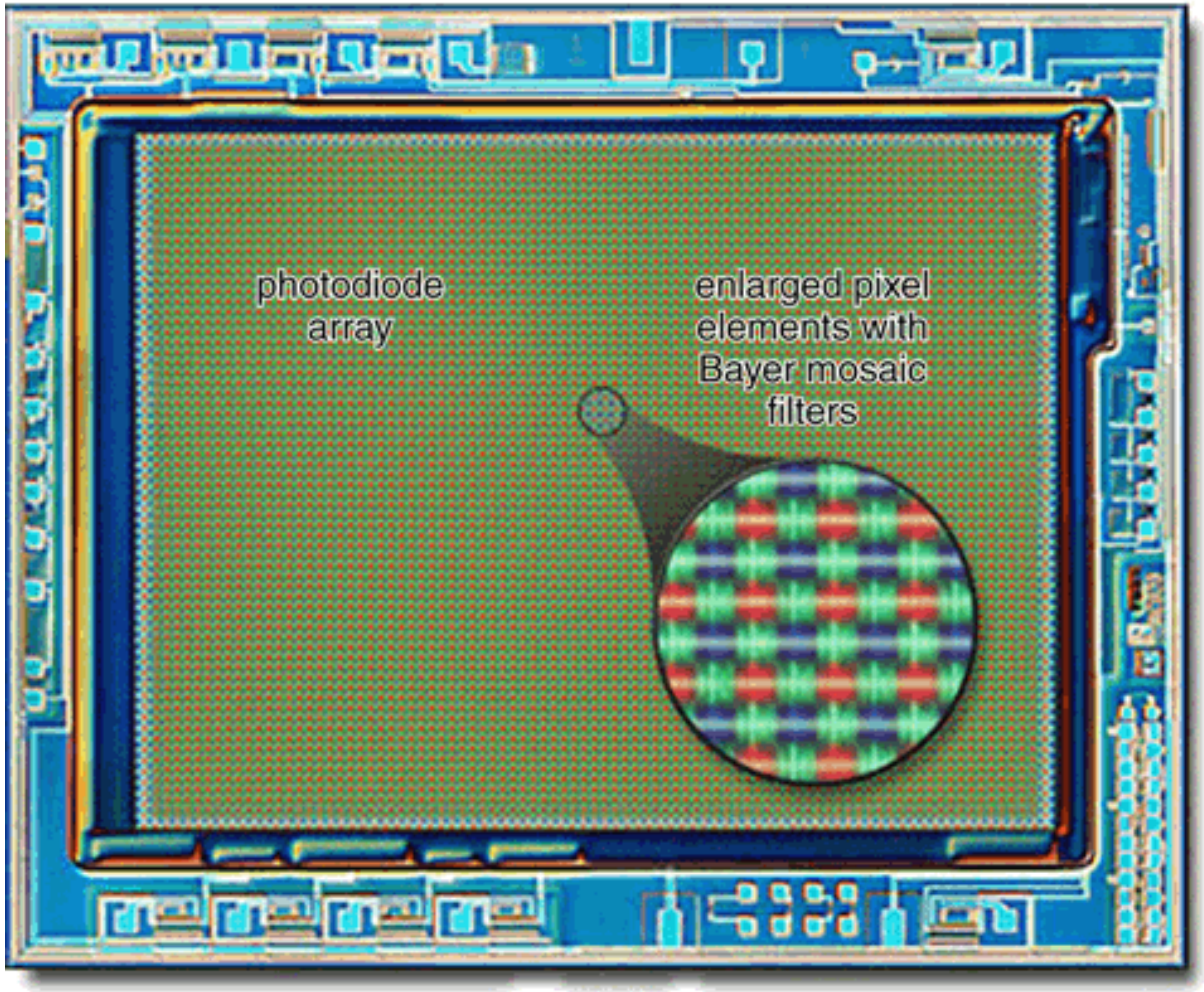
Multiplex in depth (Tripack film, Foveon)



Interferences (Lipmann)

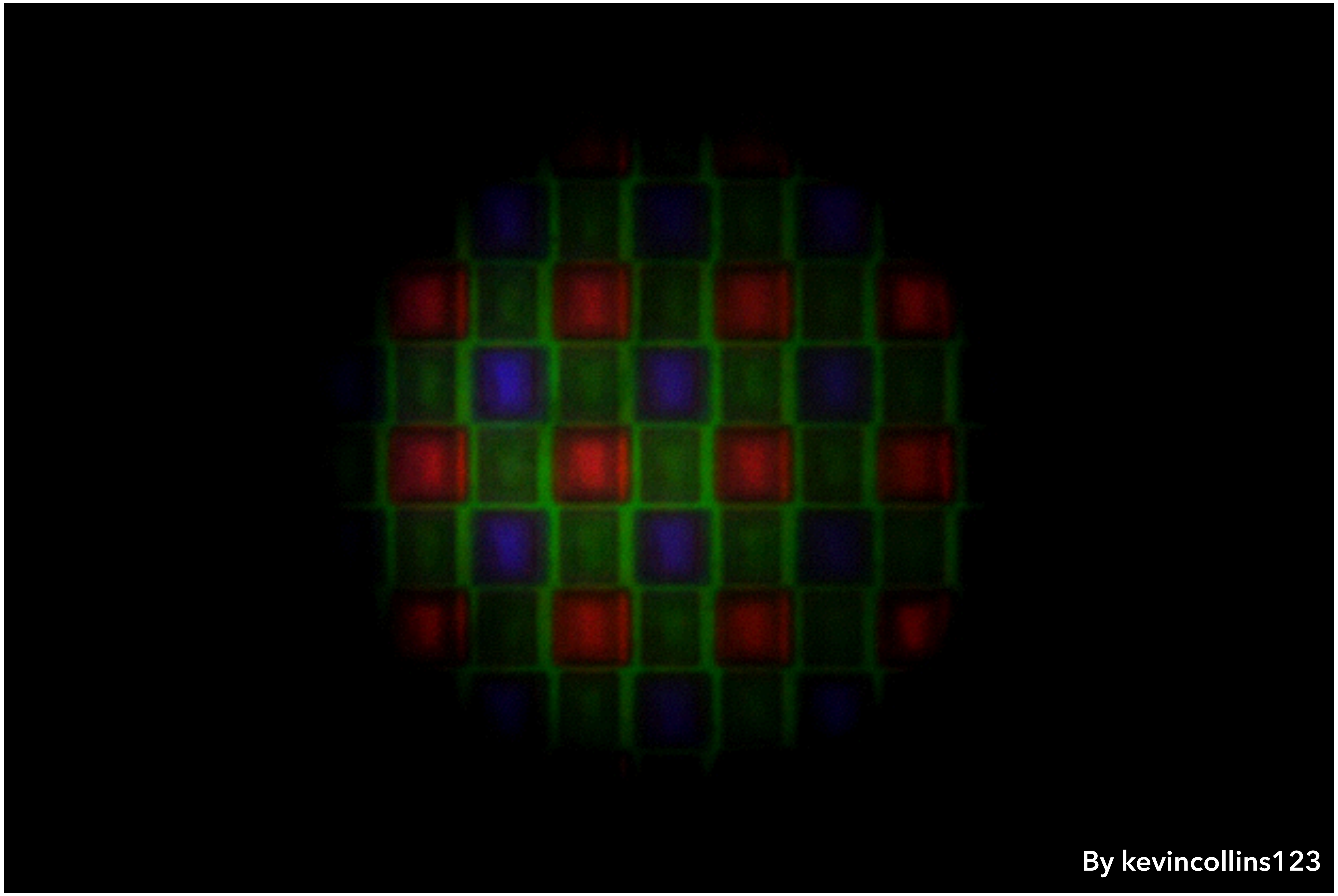
Bayer mosaic

Sensor



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

Microscope view of a CCD

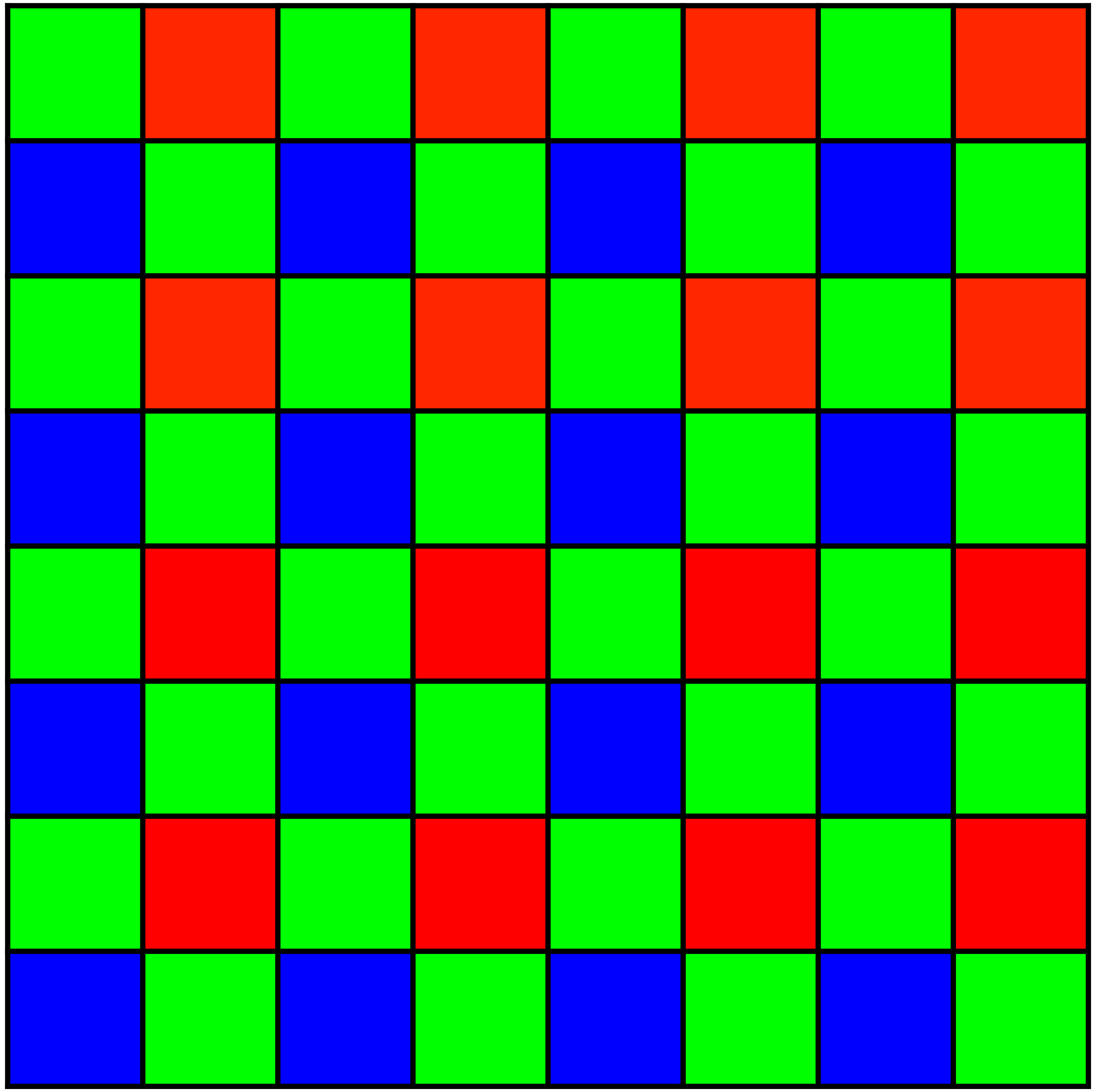


http://www.flickr.com/photos/kevincollins123/4584180753/

Bayer RGB mosaic

Which one is the upper left color is arbitrary and depends on the camera

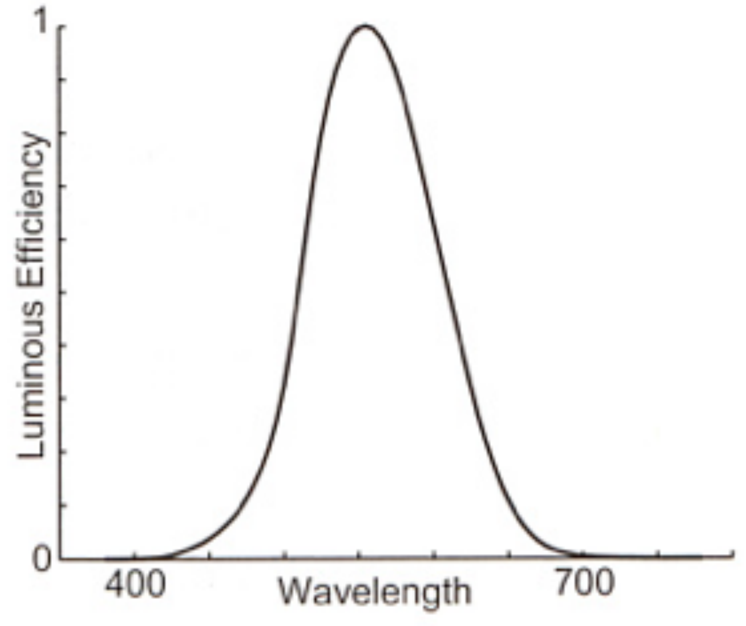
Each photosite has a different color filter

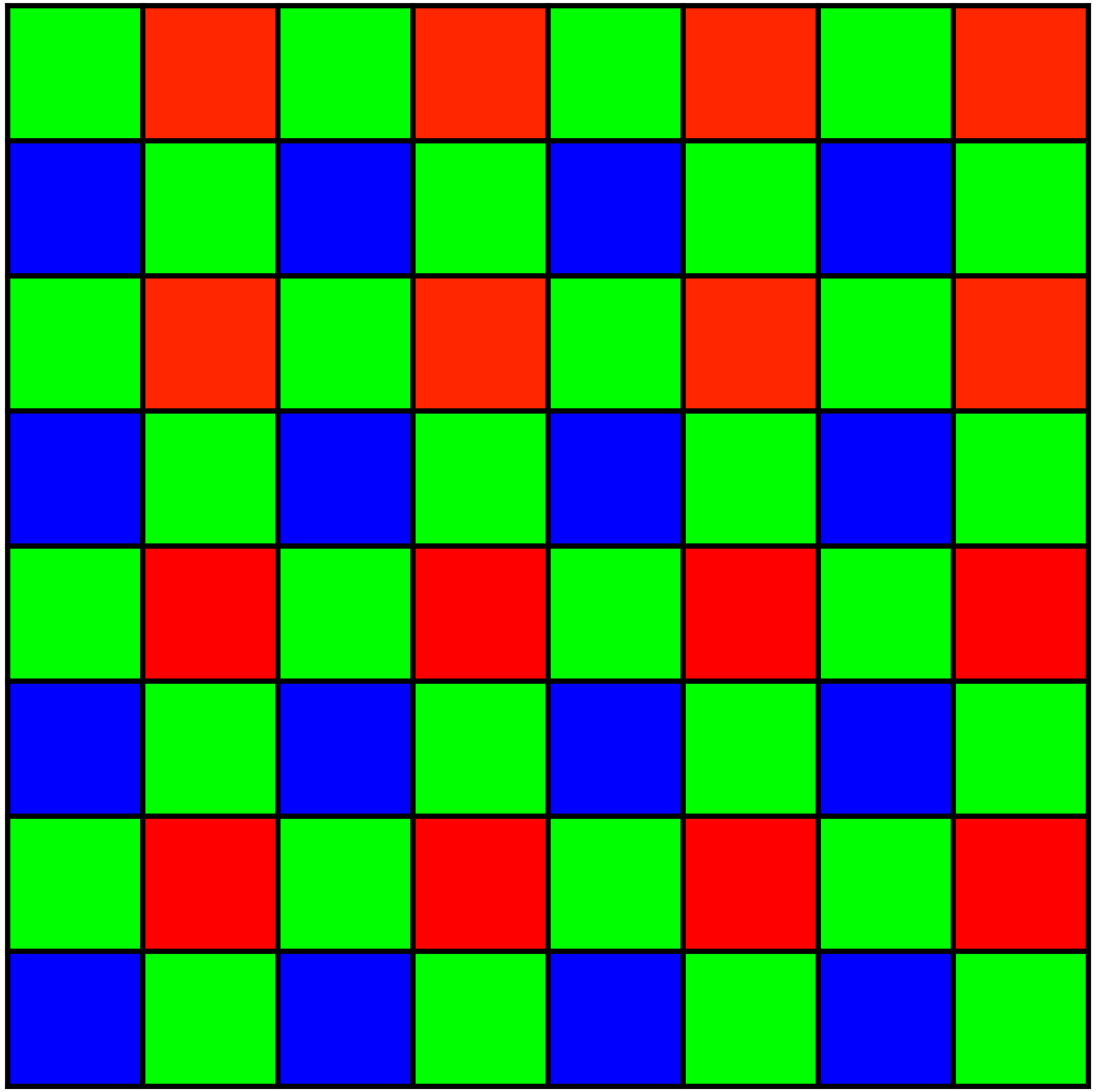


Bayer RGB mosaic

Why more green?

- We have 3 channels and square lattices don't like odd numbers
- It's the spectrum "in the middle"
- More important to human perception of luminance





RAW files

Straight measurement from sensor

- right after A/D conversion

Each photosite has only one value

- Filtered by R, G or B

Usually 12-14 bits per pixel

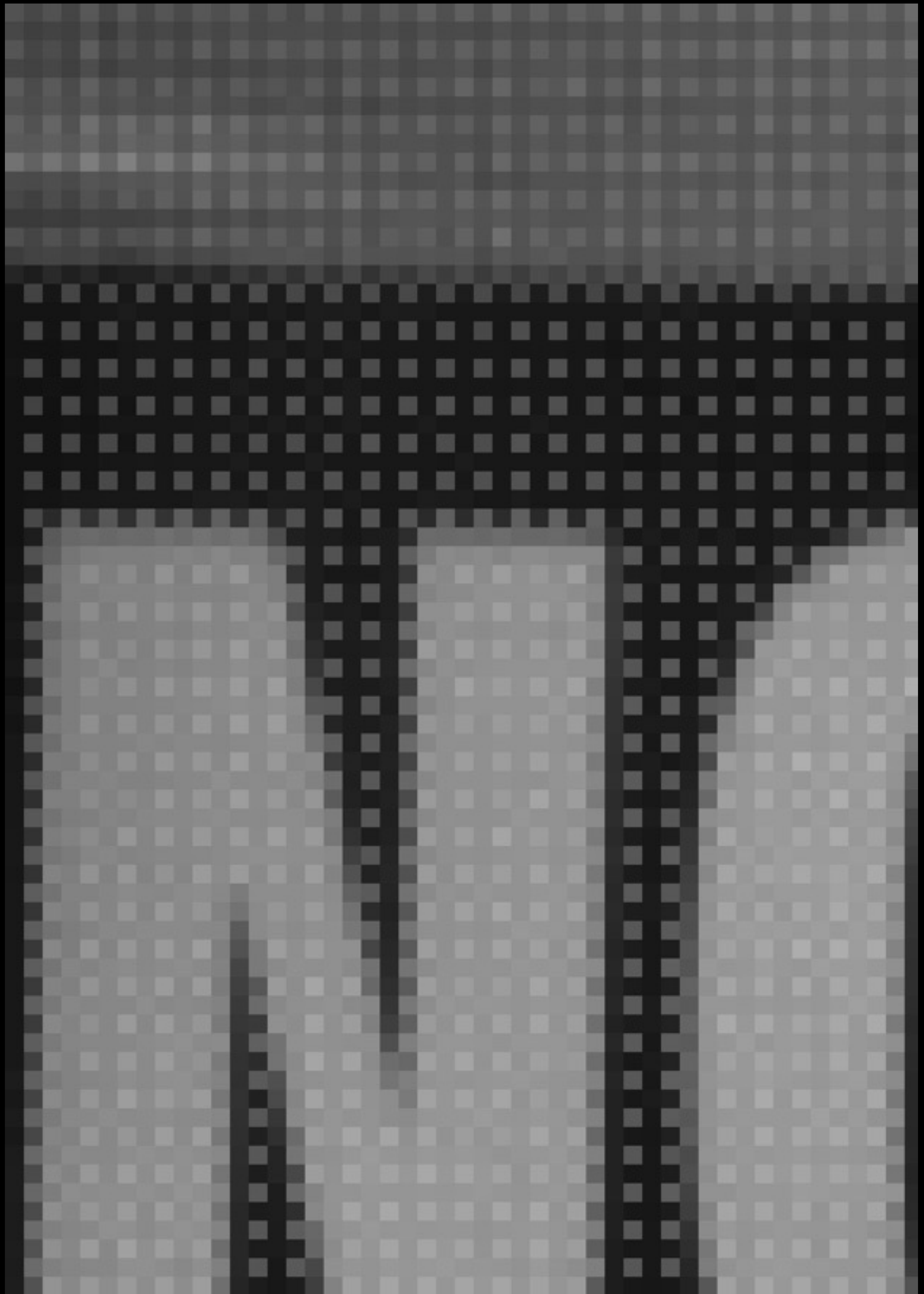
Linear encoding

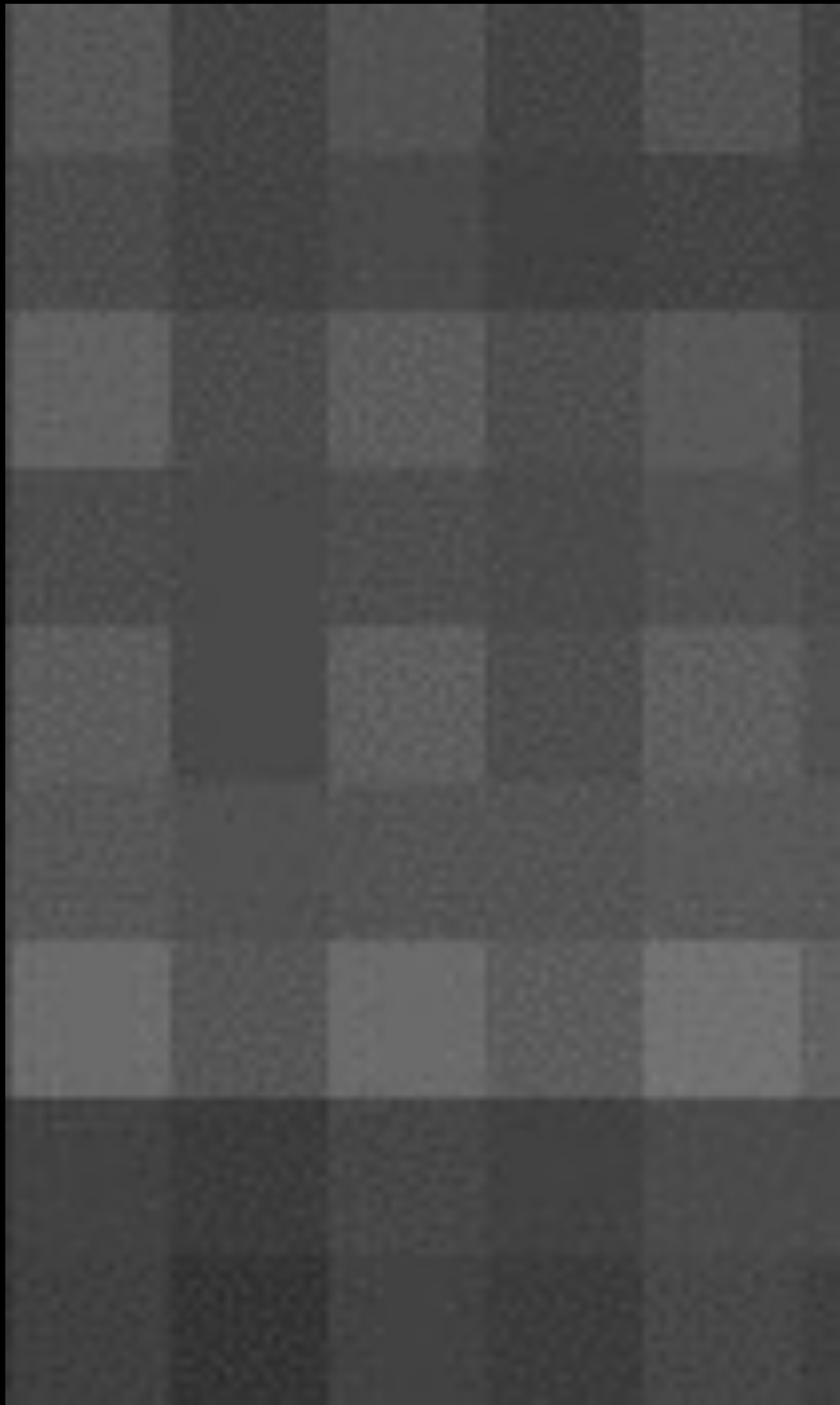
- No gamma!

Can be read and converted using dcraw

- ./dcrawx86 -v -d pics/DSC_8274.nef

A RAW file from a Nikon D70

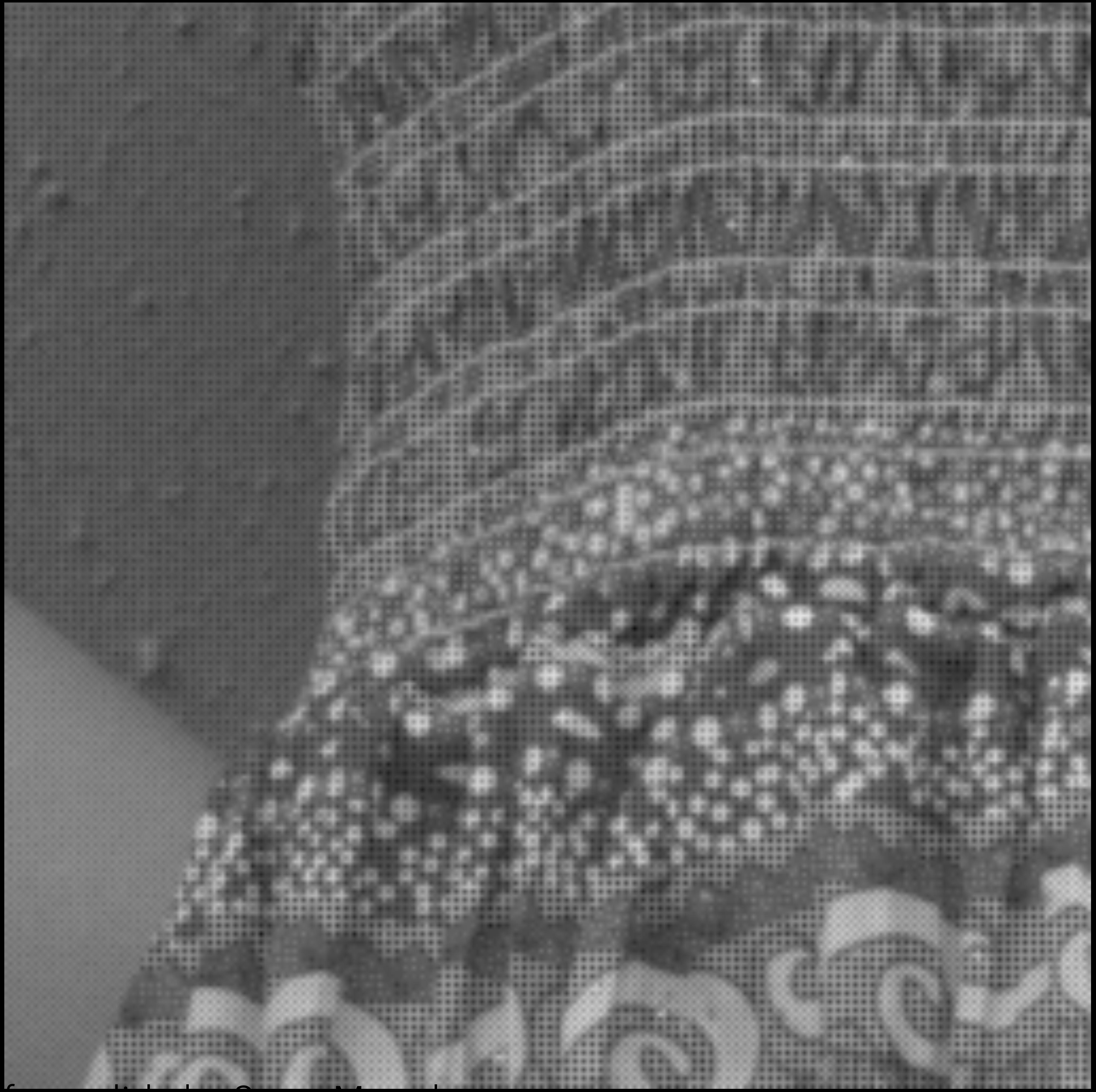




After a slide by Frédo Durand

After a slide by Steve Marschner

RAW Bayer data



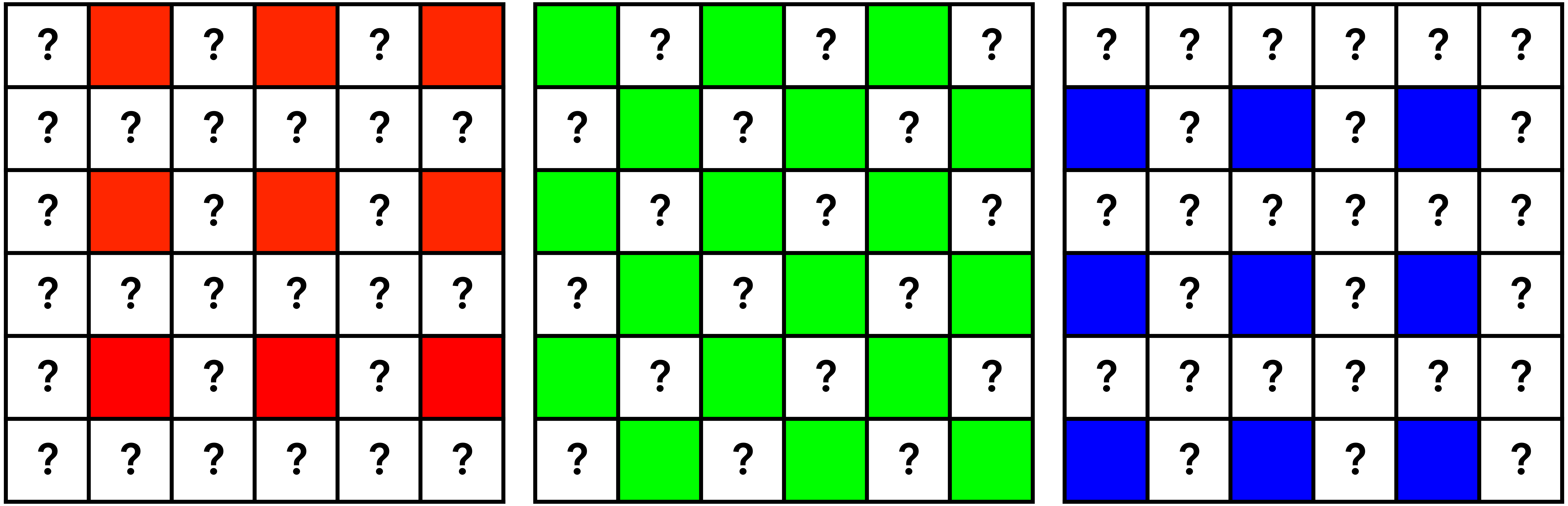
After a slide by Steve Marschner

Demosaicing

Demosaicing

Interpolate missing values

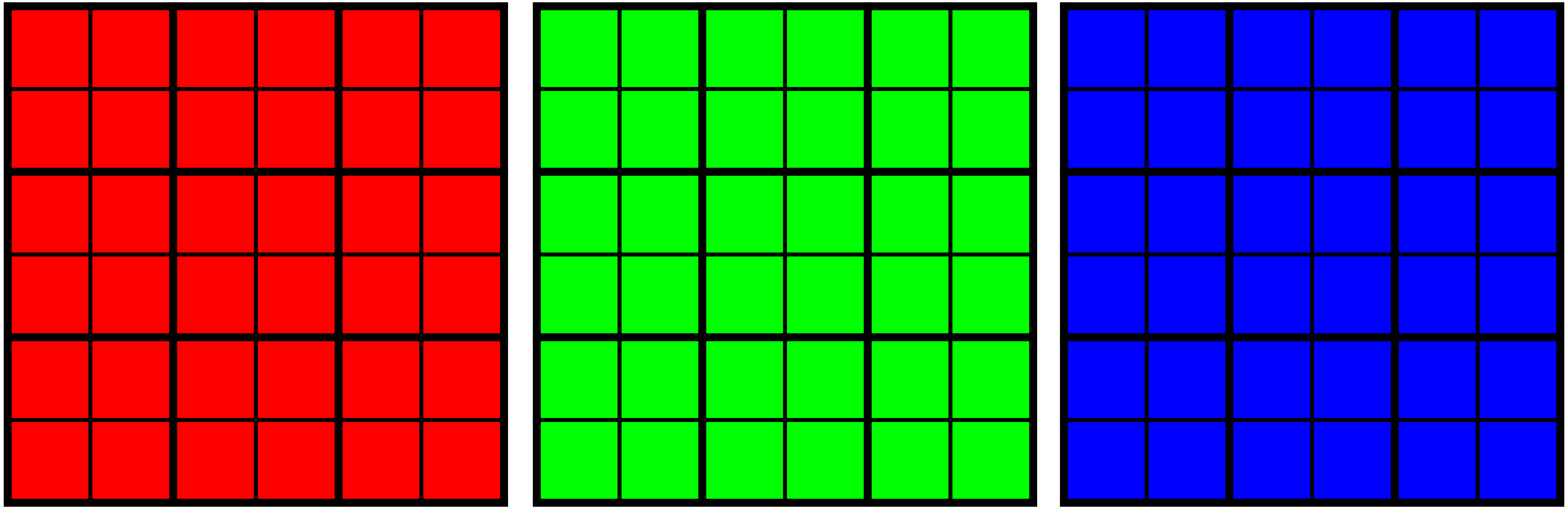
- 2/3 of the full-resolution data will be made up!



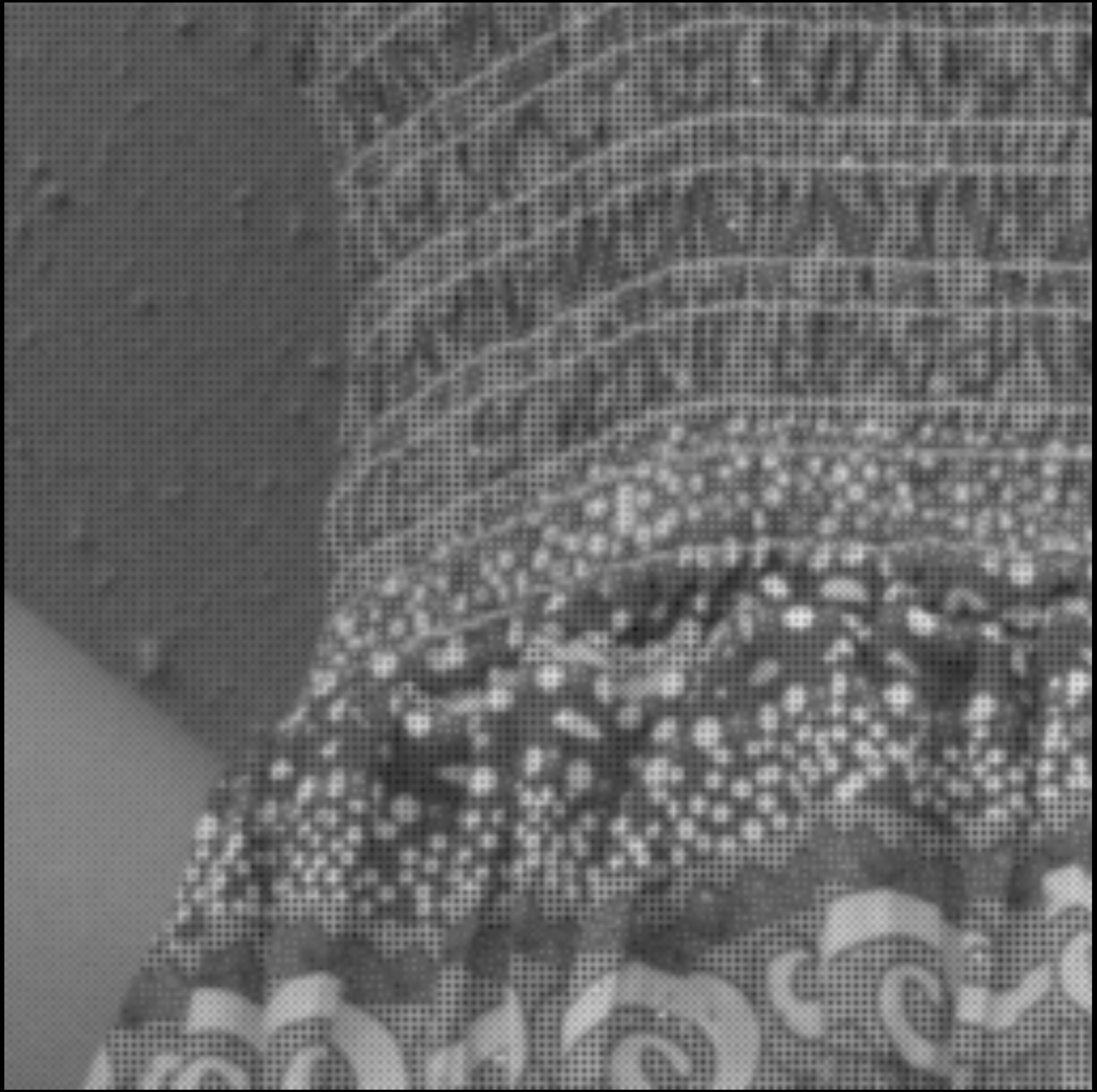
Half-resolution demosaicing

Simplest solution: treat each block of 2x2 as a pixel

- **Problem 1:** resolution loss (megapixels so important for marketing!)
- **Problem 2**: produces subpixel shifts in color planes!



RAW bayer data



After a slide by Steve Marschner

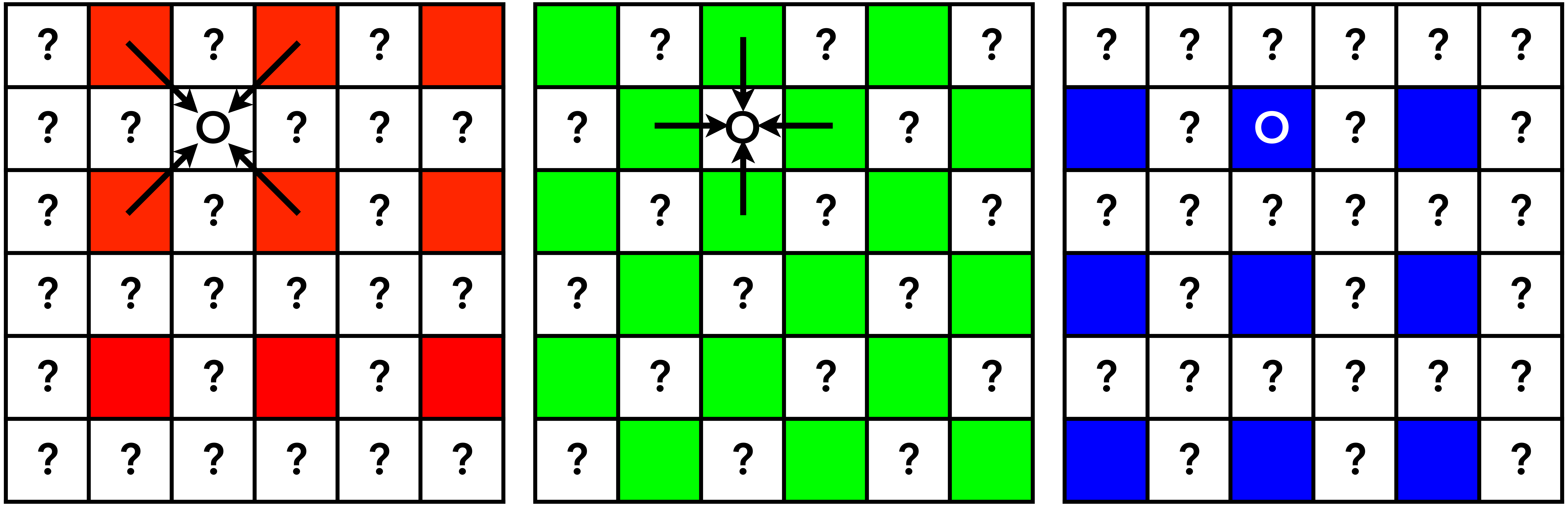
2x2 bayer block

After a slide by Steve Marschner

Centered half-resolution

Average pixels in groups that all have the same "center of gravity"

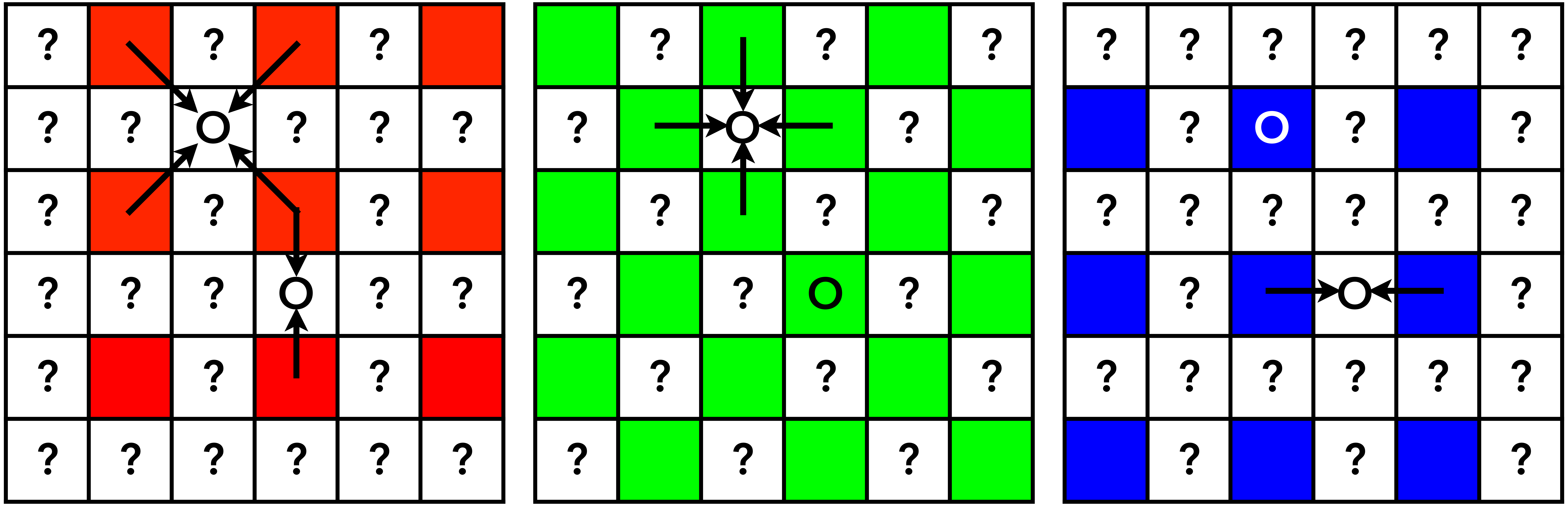
- avoids major color fringing



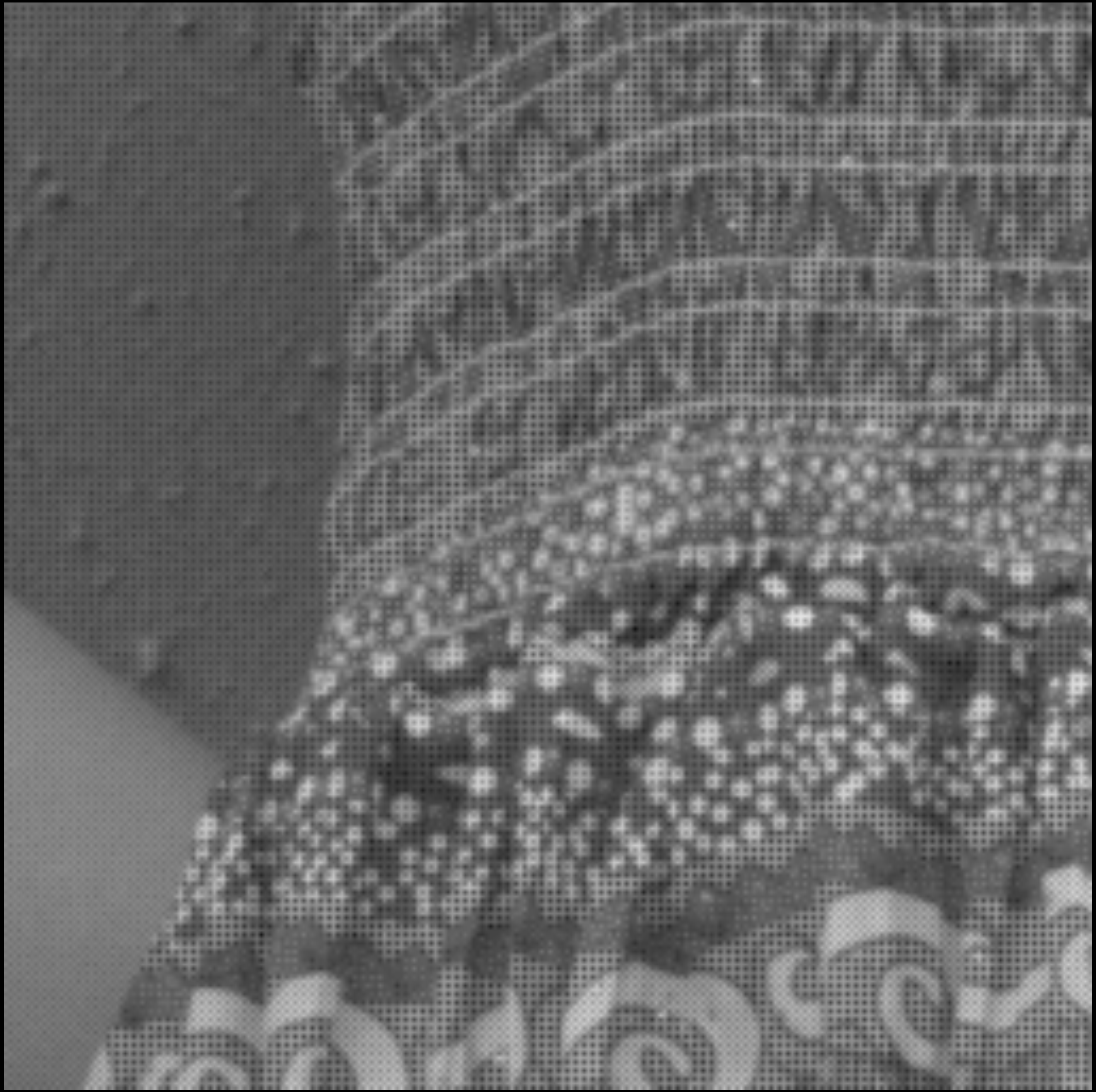
Centered half-resolution

Average pixels in groups that all have the same "center of gravity"

- avoids major color fringing



RAW bayer data



After a slide by Steve Marschner

2x2 bayer block

After a slide by Steve Marschner

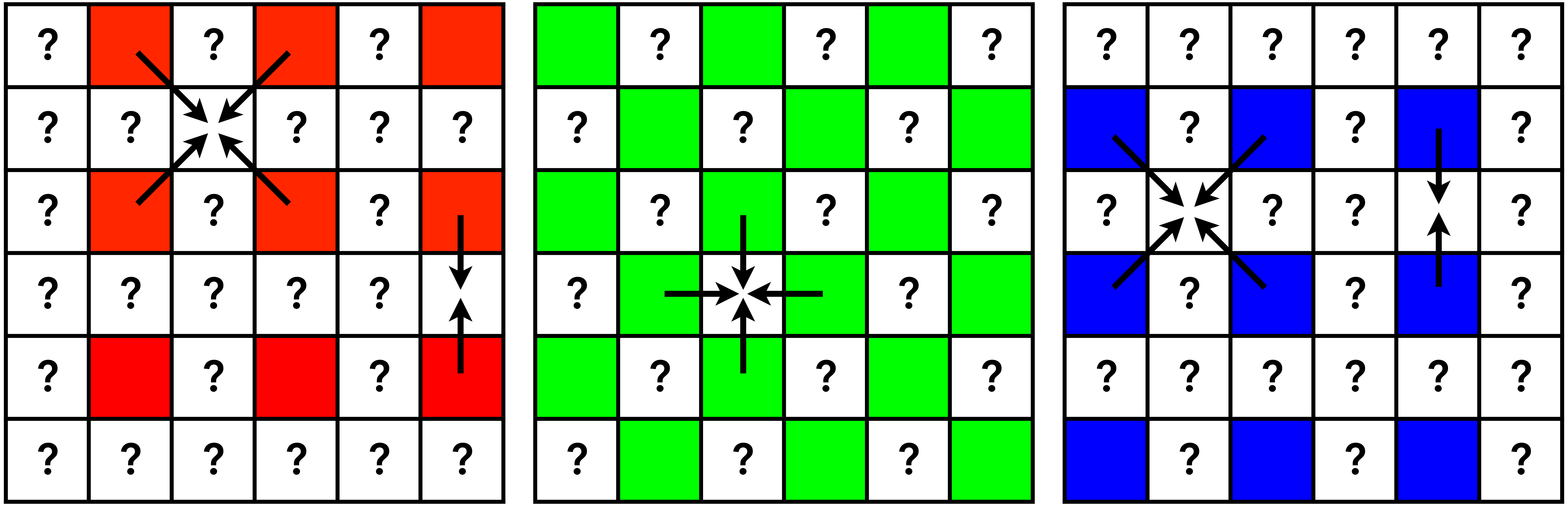
centered

After a slide by Steve Marschner

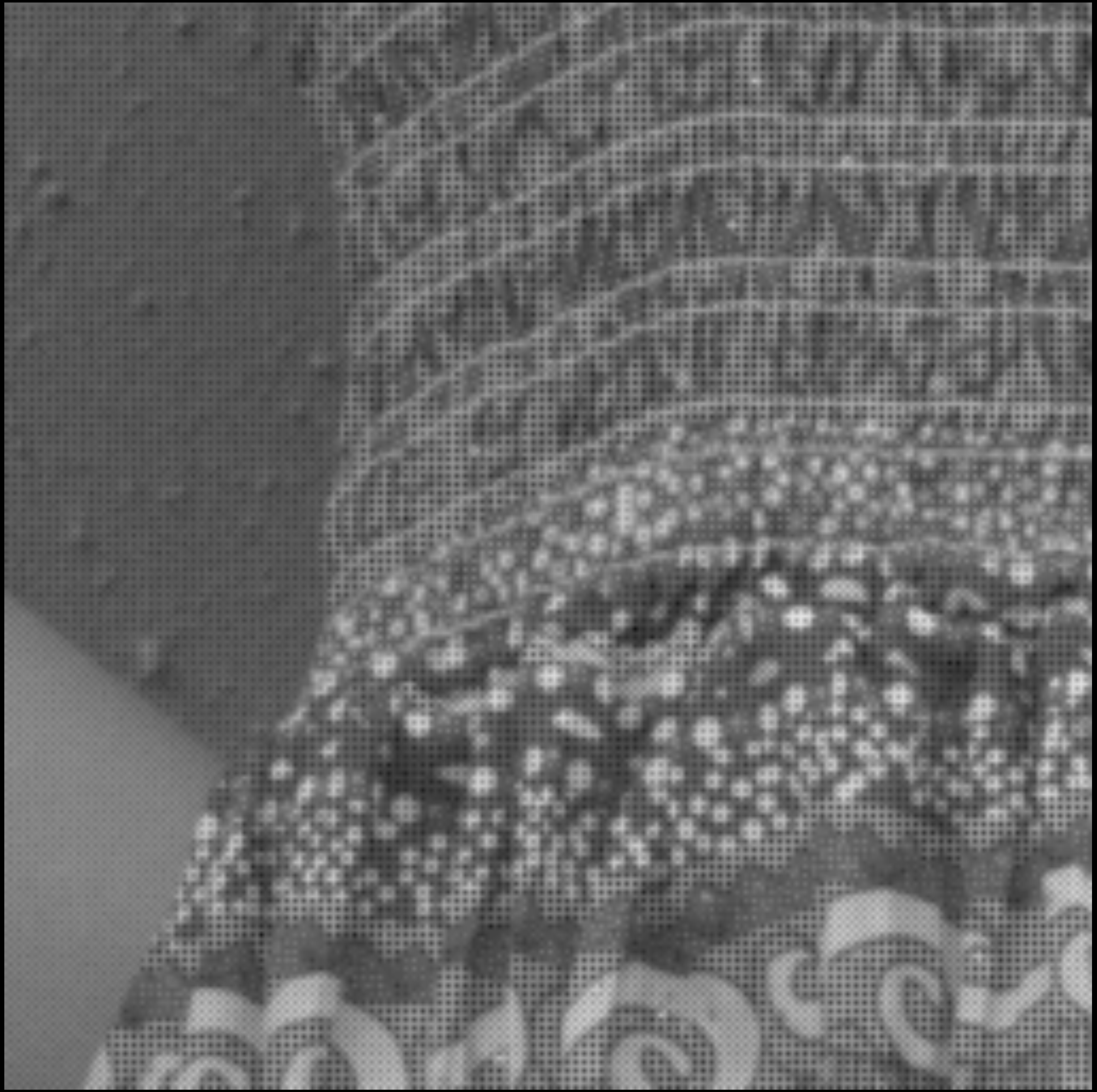
Linear interpolation

Average the 4 or 2 nearest neighbors (linear/tent kernel)

- e.g. `newgreen = 0.25 * (up+left+right+down)`



RAW Bayer data



After a slide by Steve Marschner

2x2 Bayer block

After a slide by Steve Marschner

centered

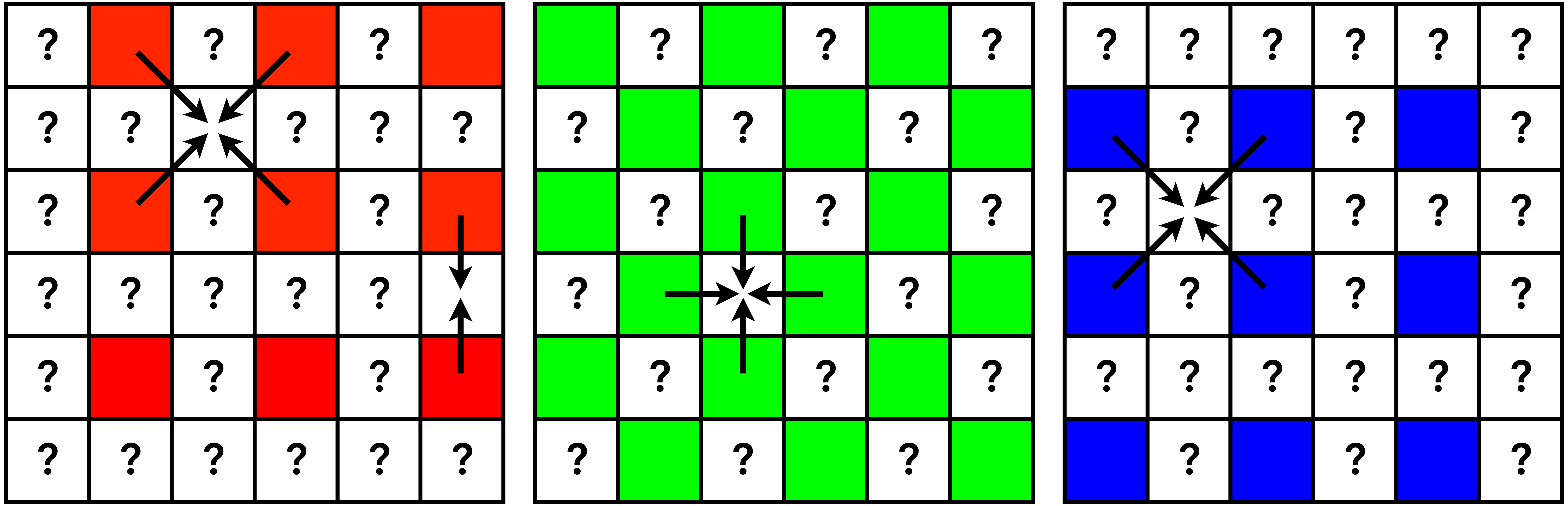
After a slide by Steve Marschner

linear

After a slide by Steve Marschner

Better

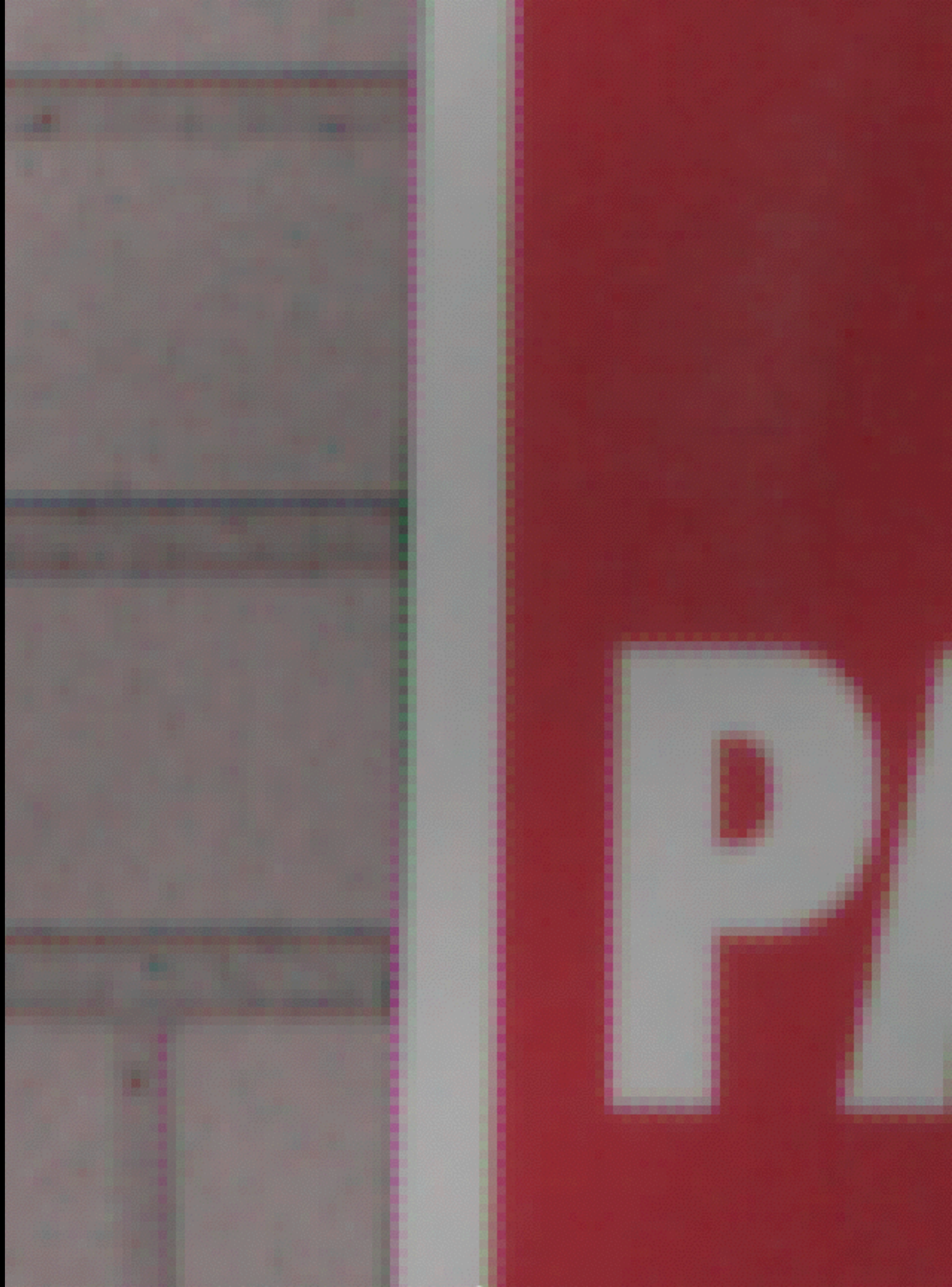
Smoother kernels can also be used (e.g. bicubic) but need wider support



Results of simple linear

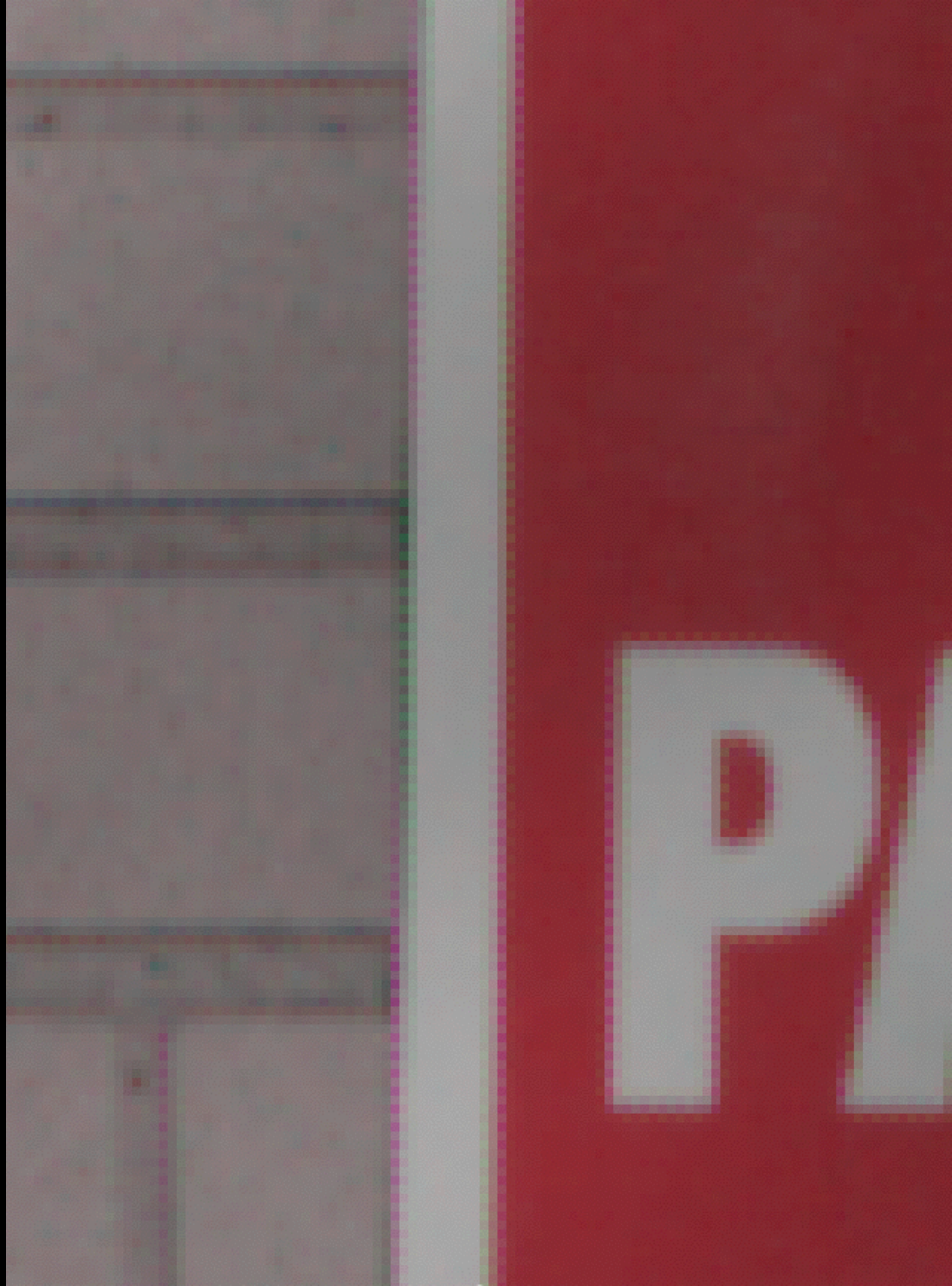
After a slide by Frédo Durand

Results - not perfect



After a slide by Frédo Durand

Questions?

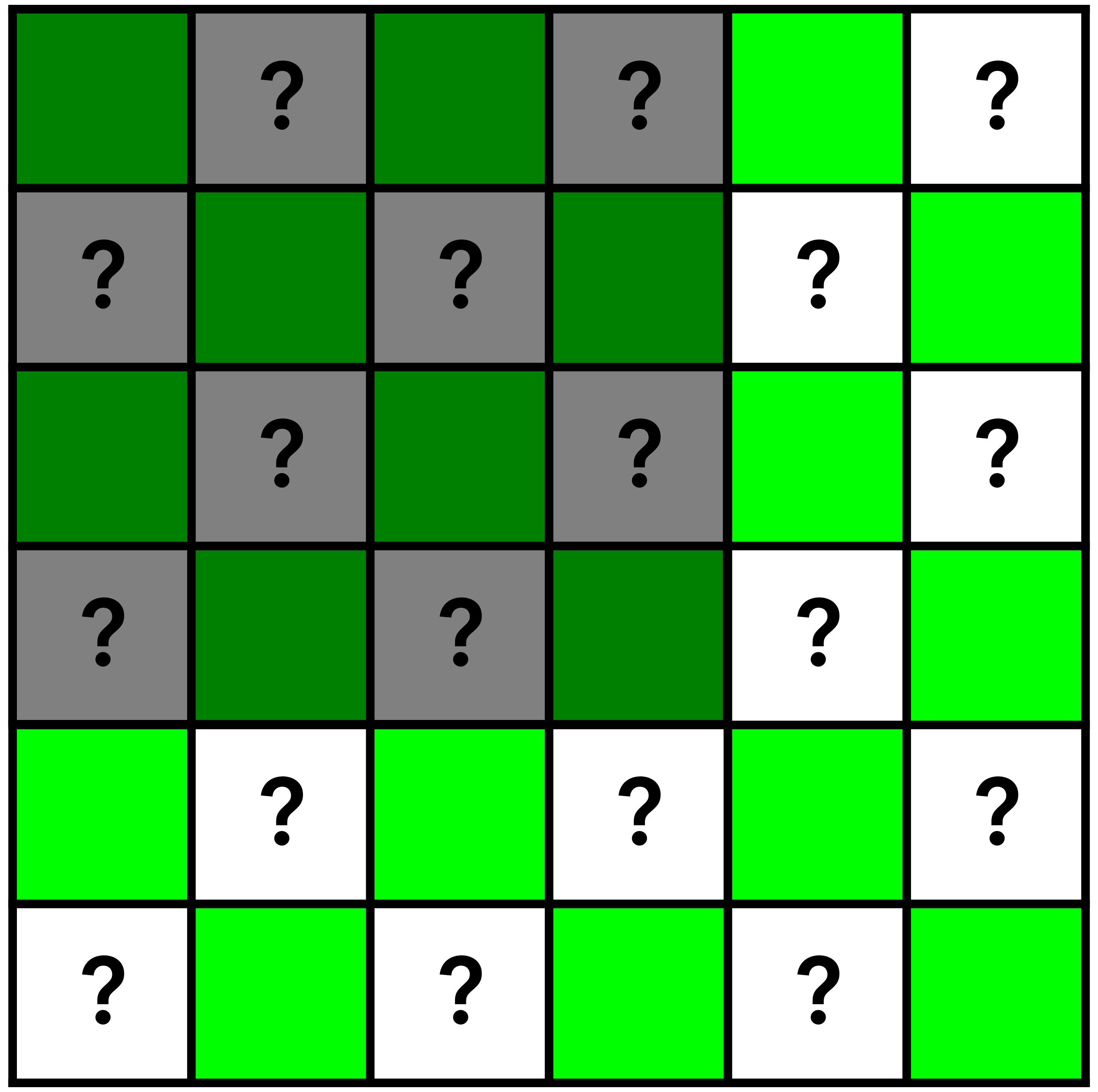


After a slide by Frédo Durand

The problem

Imagine a black-on-white corner

Let's focus on the green channel for now



The problem

Imagine a black-on-white corner

Let's focus on the green channel for now

	?		?		?
?		?		?	
	?		?		?
?		?		?	
	?		?		?
?		?		?	

0		0		1	
	0		0		1
0		0		1	
	0		0		1
1		1		1	
	1		1		1

The problem

Imagine a black-on-white corner

Let's focus on the green channel for now

	?		?		?
?		?		?	
	?		?		?
?		?		?	
	?		?		?
?		?		?	

0		0		1	
	0		0		1
0		0		1	
	0		0		1
1		1		1	
	1		1		1

0	0	0	.25	1	1
0	0	0	0	.75	1
0	0	0	.25	1	1
.25	0	.25	0	.75	1
1	.75	1	.75	1	1
1	1	1	1	1	1

The problem

Imagine a black-on-white corner

Let's focus on the green channel for now

	?		?		?
?		?		?	
	?		?		?
?		?		?	
	?		?		?
?		?		?	

0		0		1	
	0		0		1
0		0		1	
	0		0		1
1		1		1	
	1		1		1

0	0	0	.25	1	1
0	0	0	0	.75	1
0	0	0	.25	1	1
.25	0	.25	0	.75	1
1	.75	1	.75	1	1
1	1	1	1	1	1

Yep, that’s what we saw

After a slide by Frédo Durand

Green channel

After a slide by Frédo Durand

Edge-based Demosaicing

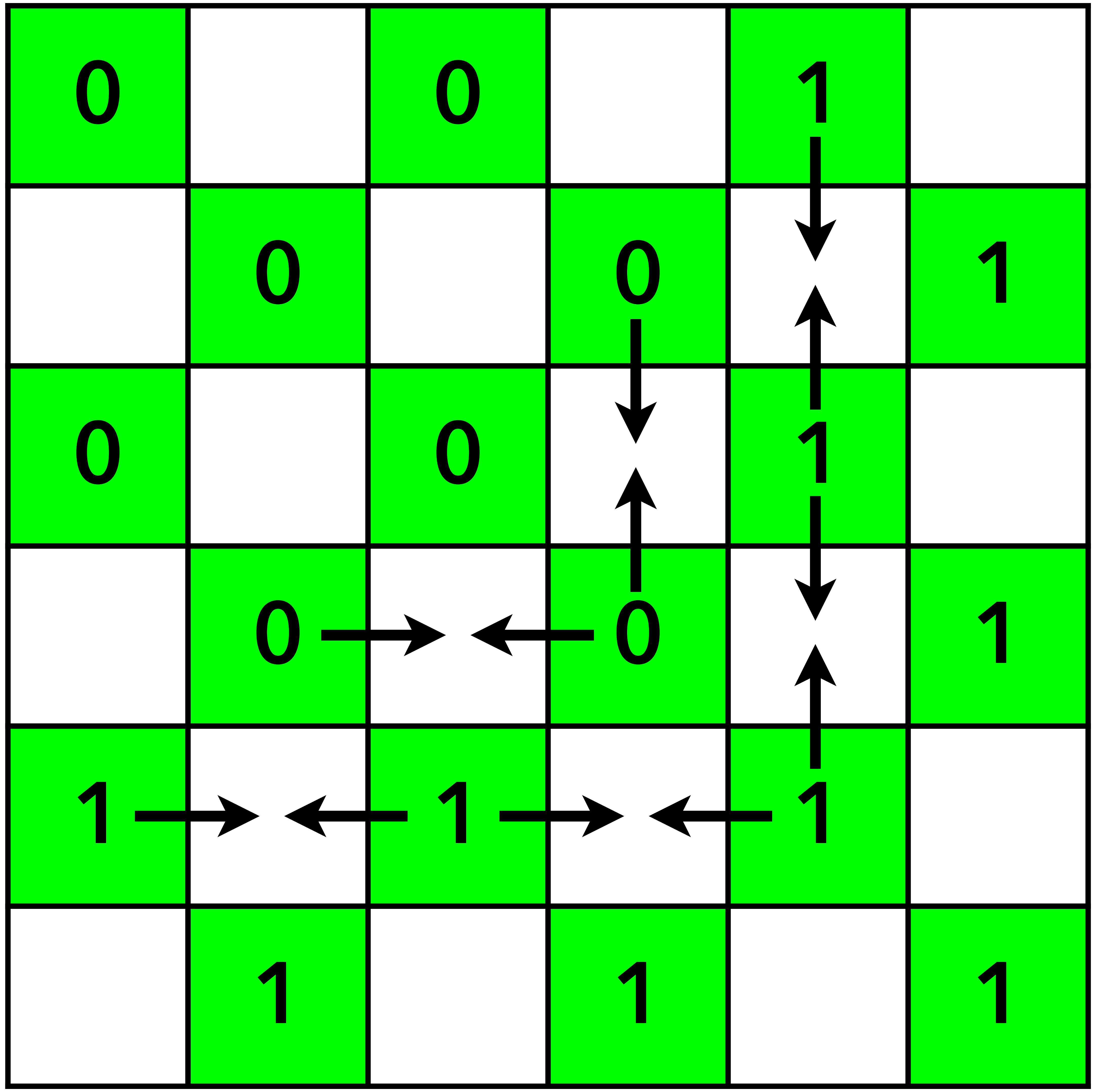
Idea

Take into account structure in image

- Here, 1D edges

Interpolate along preferred direction

- In our case, only use 2 neighbors

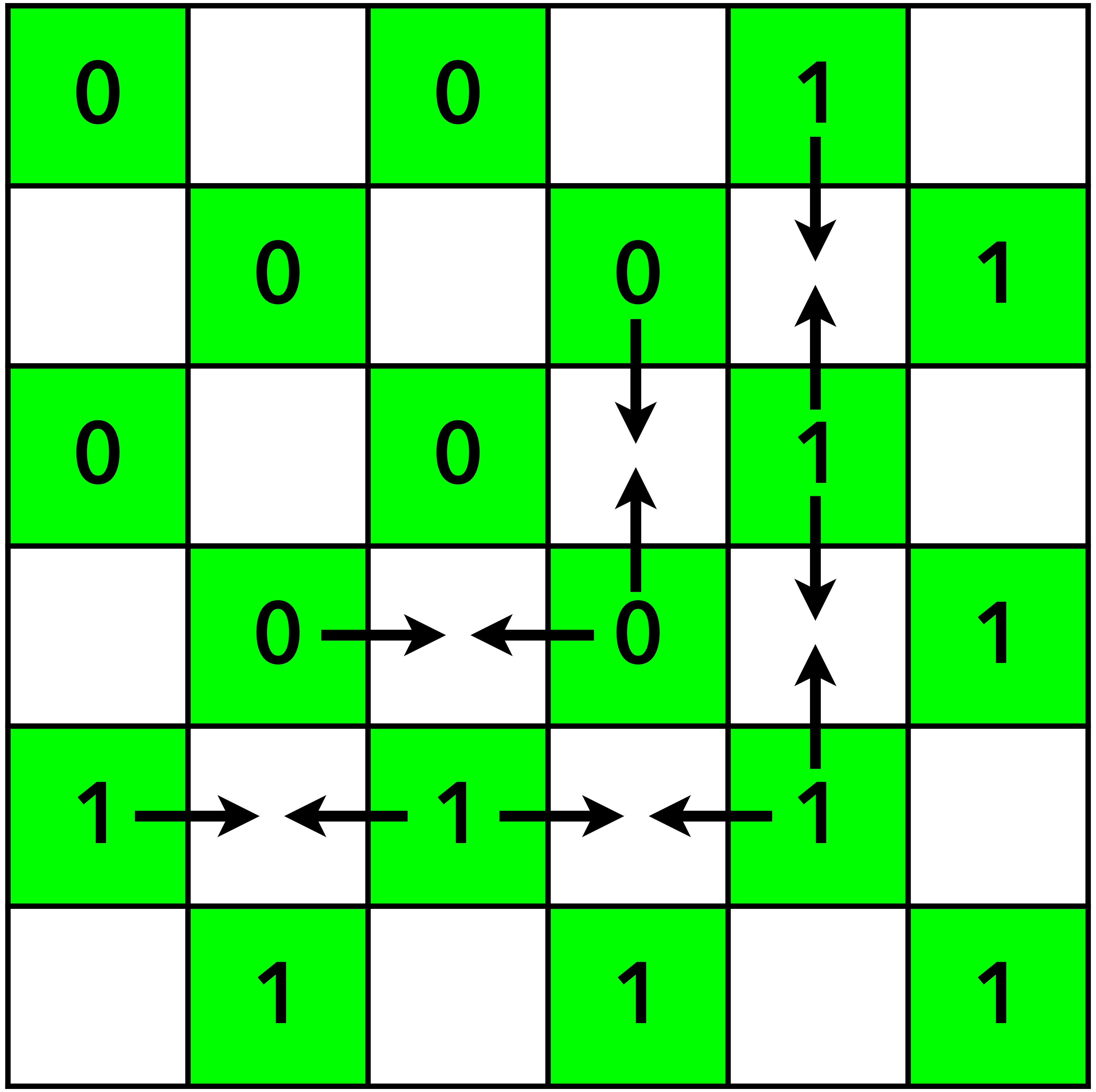


How do we decide?

Look at the similarity of recorded neighbors

- Compare |up-down| to |right-left|
- Be smart
- See Assignment 3

Called edge-based demosaicing



Green channel – naïve

After a slide by Frédo Durand

Green channel – edge-based

After a slide by Frédo Durand

Challenge with other channels

Problem

What do we do with red and blue?

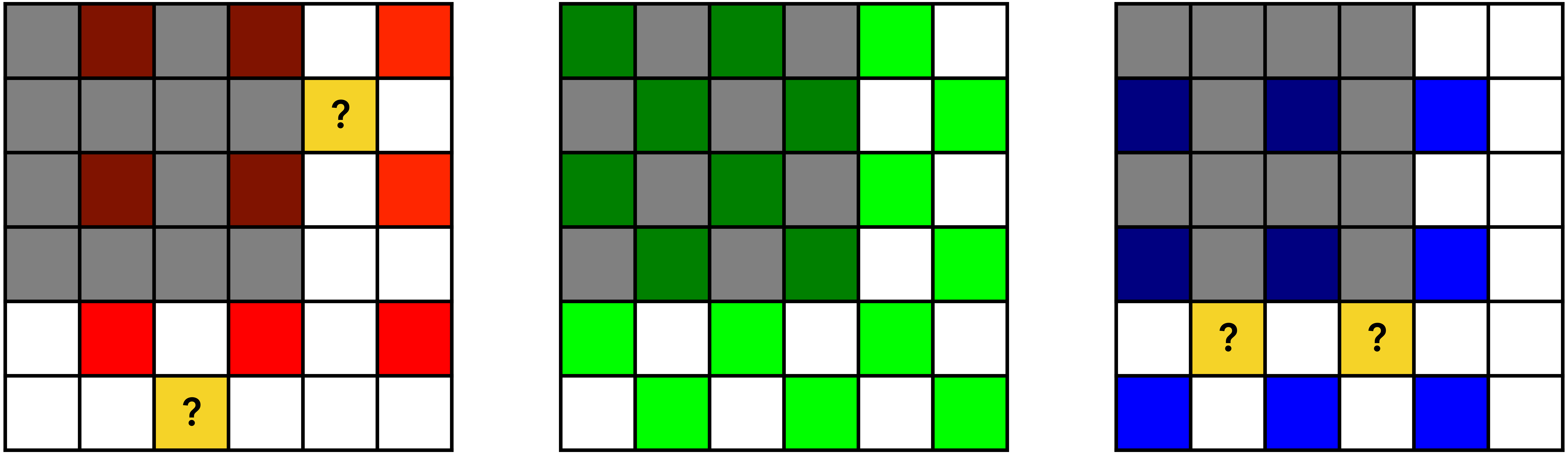
We could apply the edge-based principle

But we're missing more information

But color transitions might be shifted

Example (black-on-white corner)

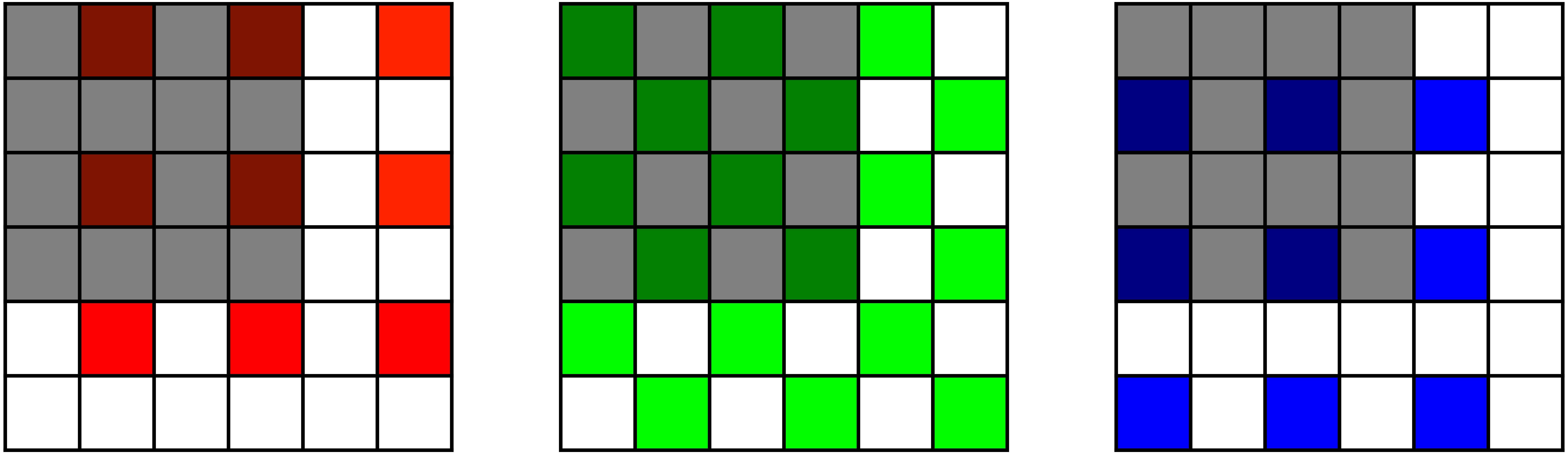
Notion of edges unclear for pixels in empty rows/columns



Example (black-on-white corner)

Even if we could do a decent job for each channel, the channels don't line up

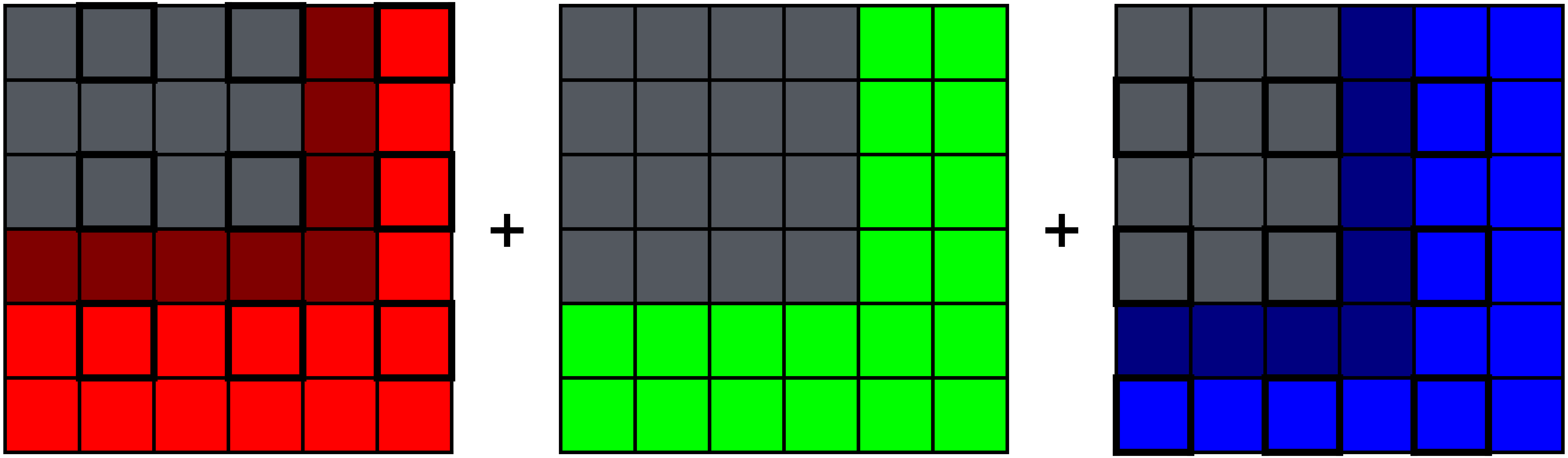
- because they are not recorded at the same location



Example (black-on-white corner)

Even if we could do a decent job for each channel, the channels don't line up

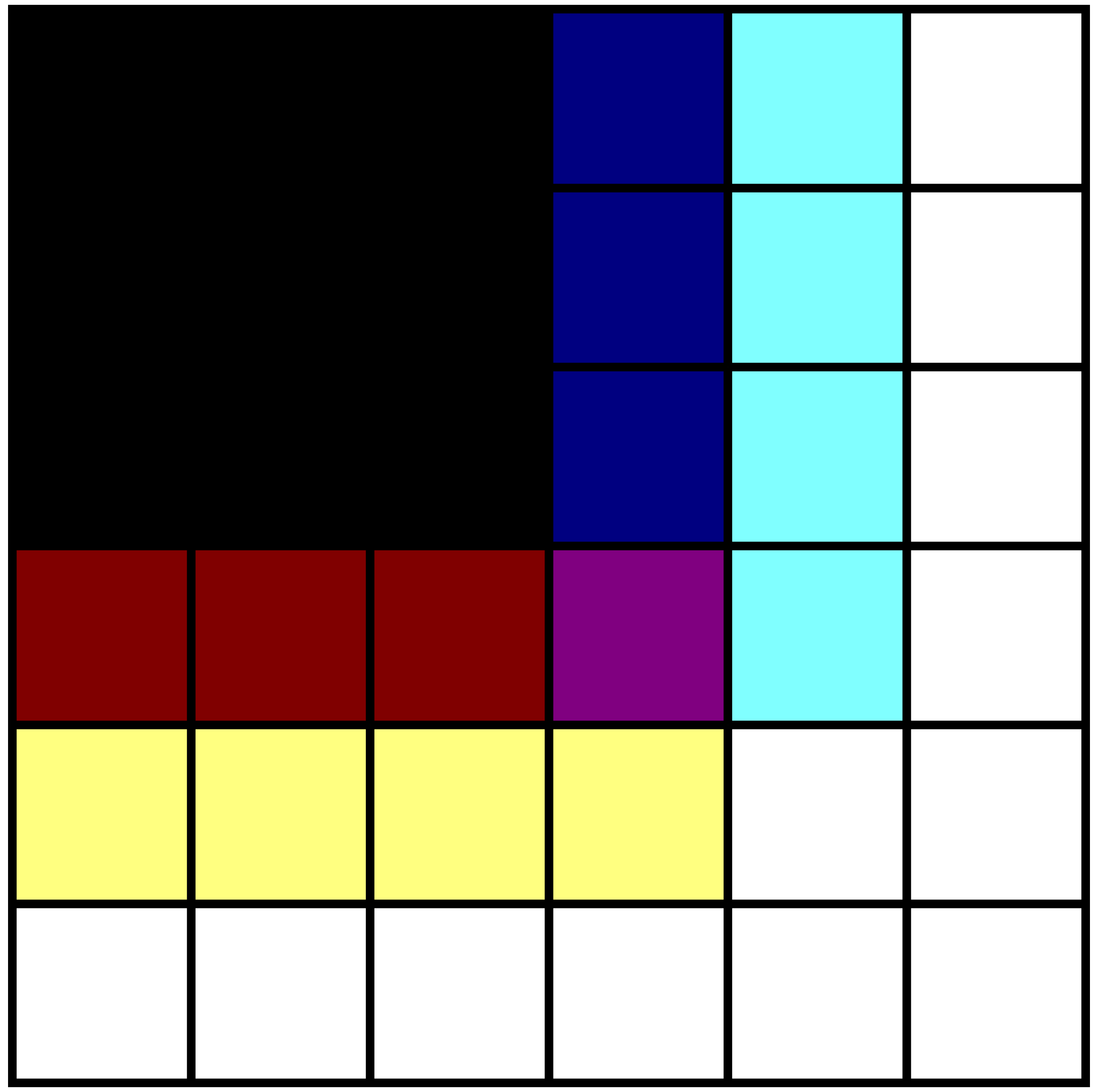
- because they are not recorded at the same location



Example (black-on-white corner)

Even if we could do a decent job for each channel, the channels don't line up

- because they are not recorded at the same location



Bad color fringes!

Recall color artifacts

After a slide by Frédo Durand

Green-based Demosaicing

Green-based demosaicing

Green is a better color channel

- Twice as many pixels
- Often better SNR
- We know how to do edge-based green interpolation

Do the best job you can and get high resolution from green

Then use green to guide red & blue interpolation

Interpolate difference to green

Interpolate green

- using e.g. edge-based

For recorded red pixels

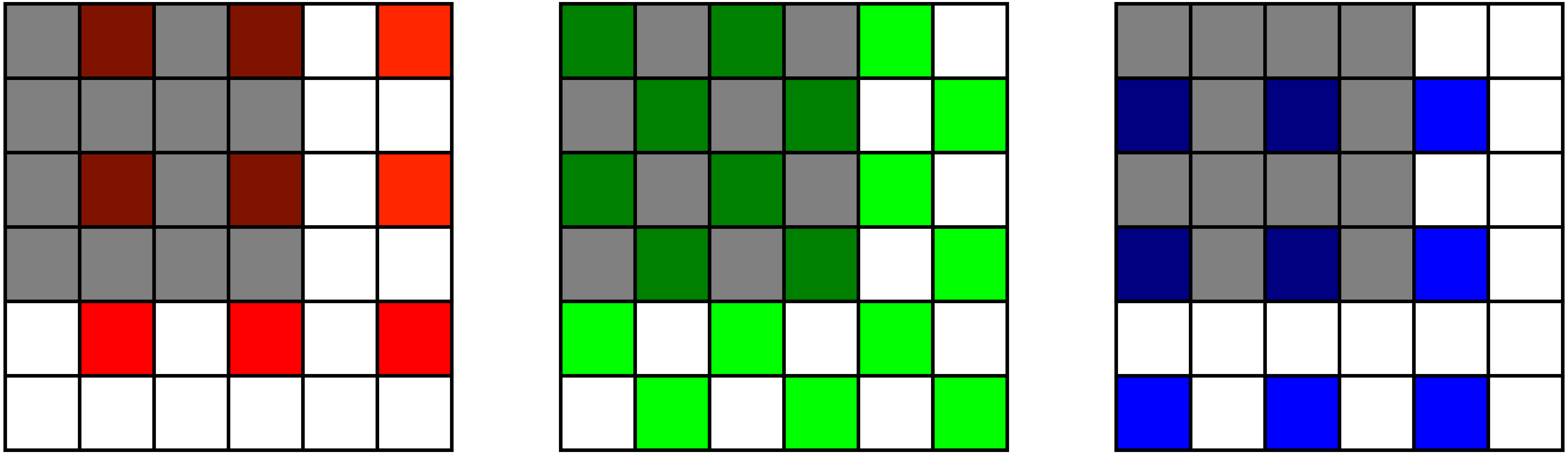
- compute R-G

At empty pixels

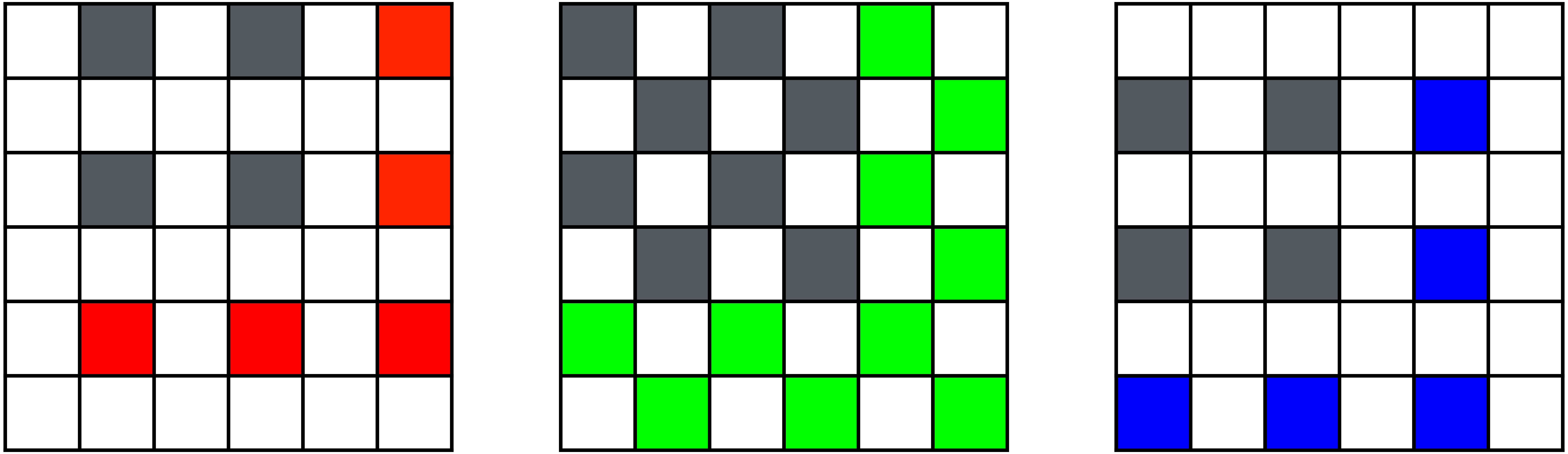
- Interpolate R-G naïvely
- Add G

Same for blue

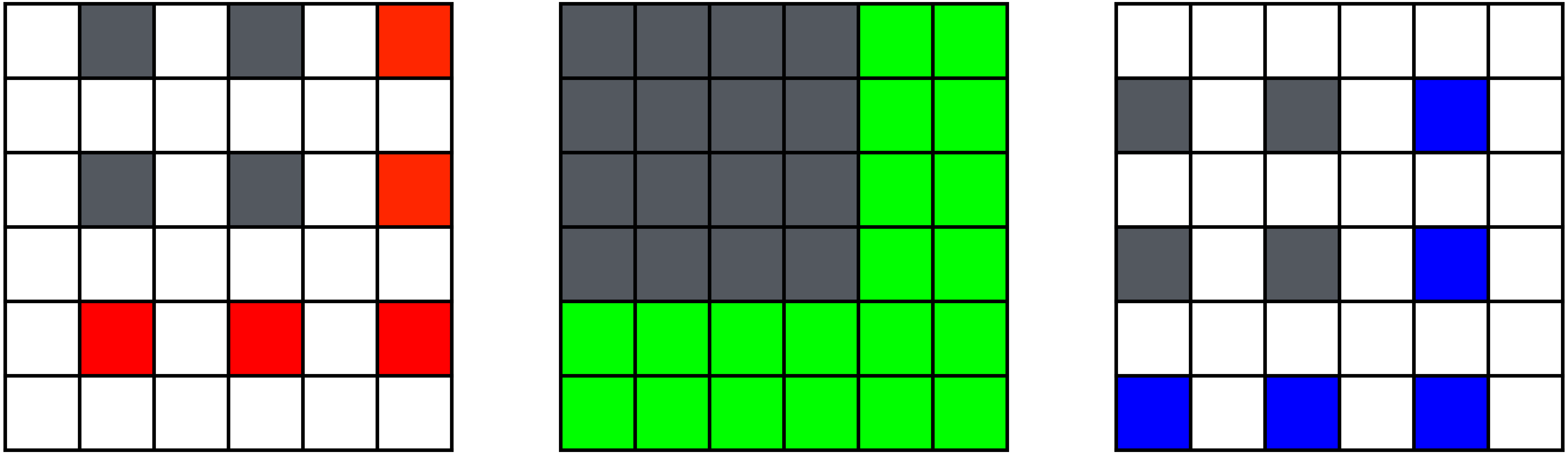
Black-on-white corner



Measurements

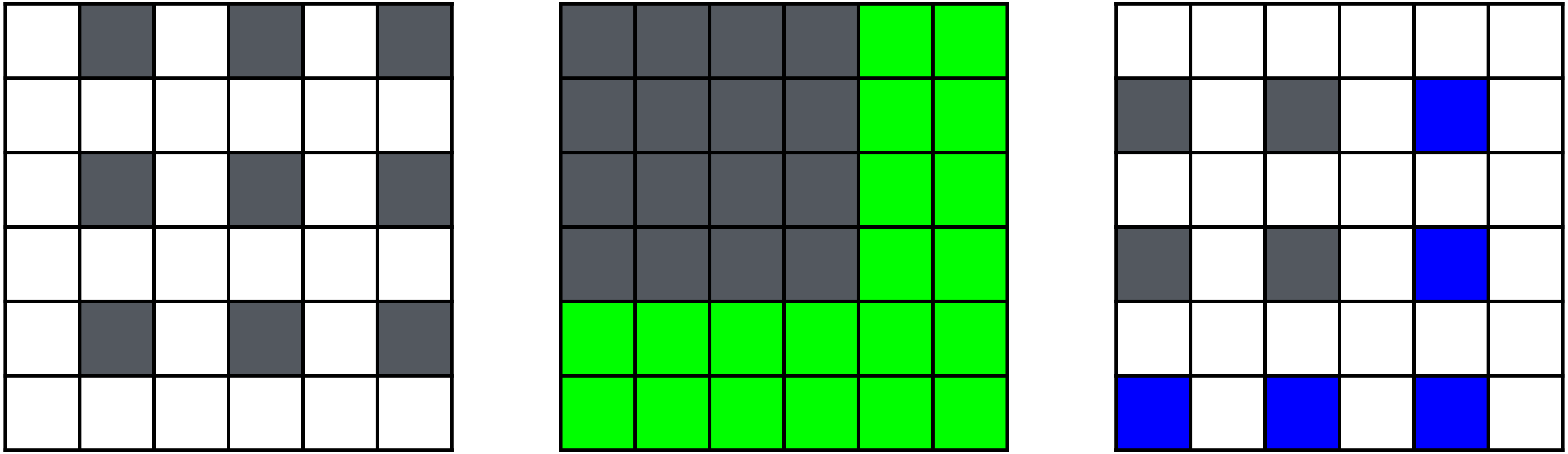


Edge-based green



Red-Green difference

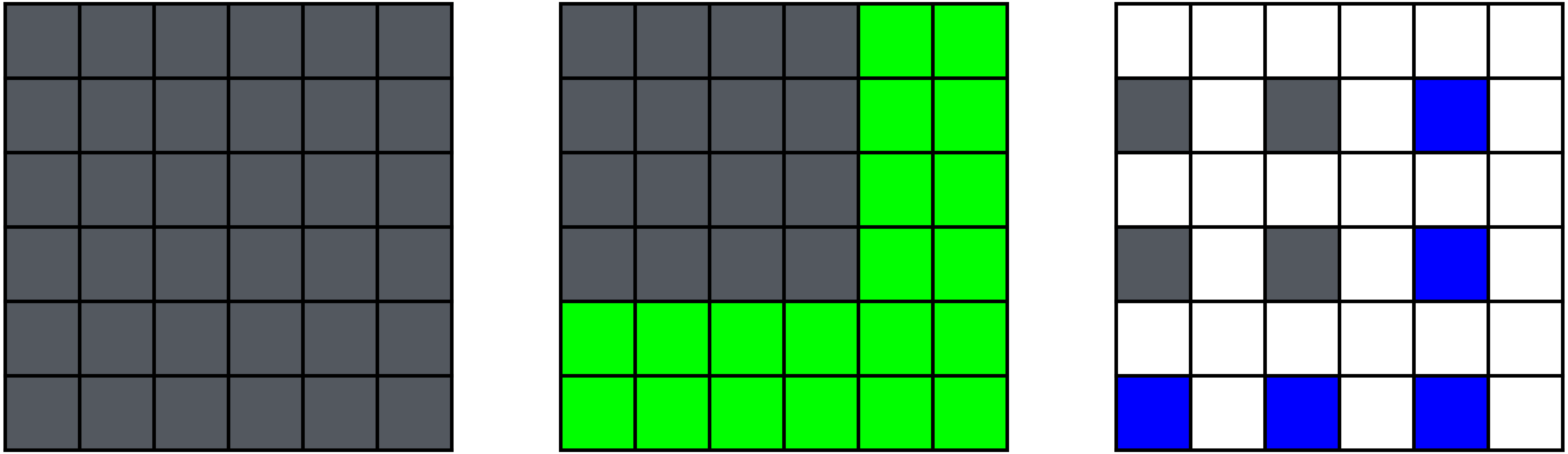
Zero everywhere!



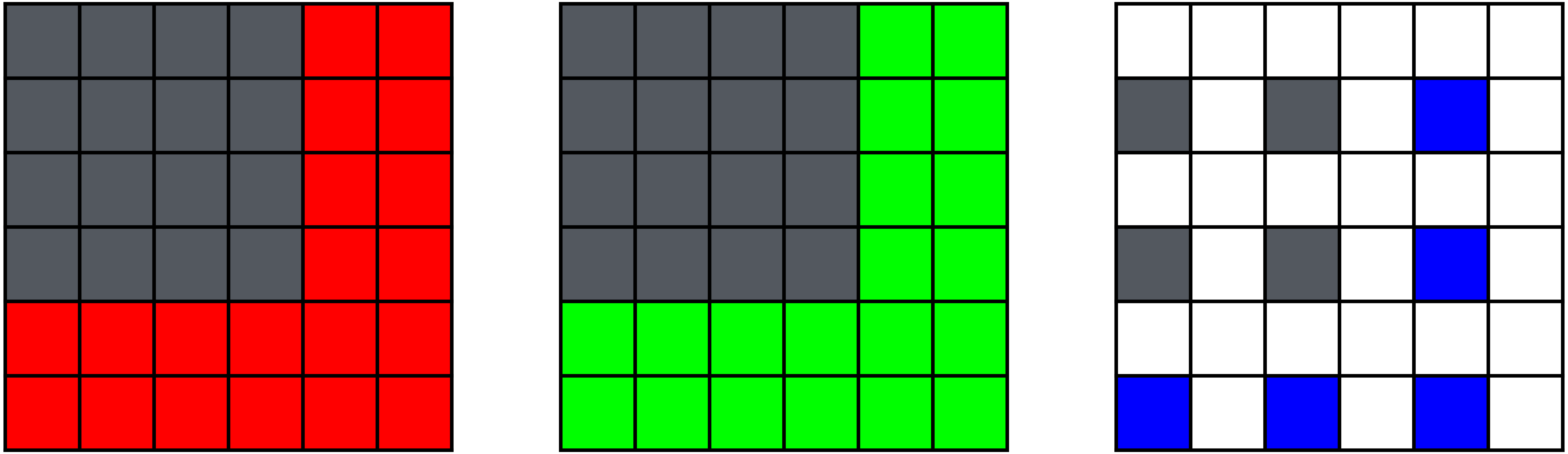
Red-Green difference

Zero everywhere!

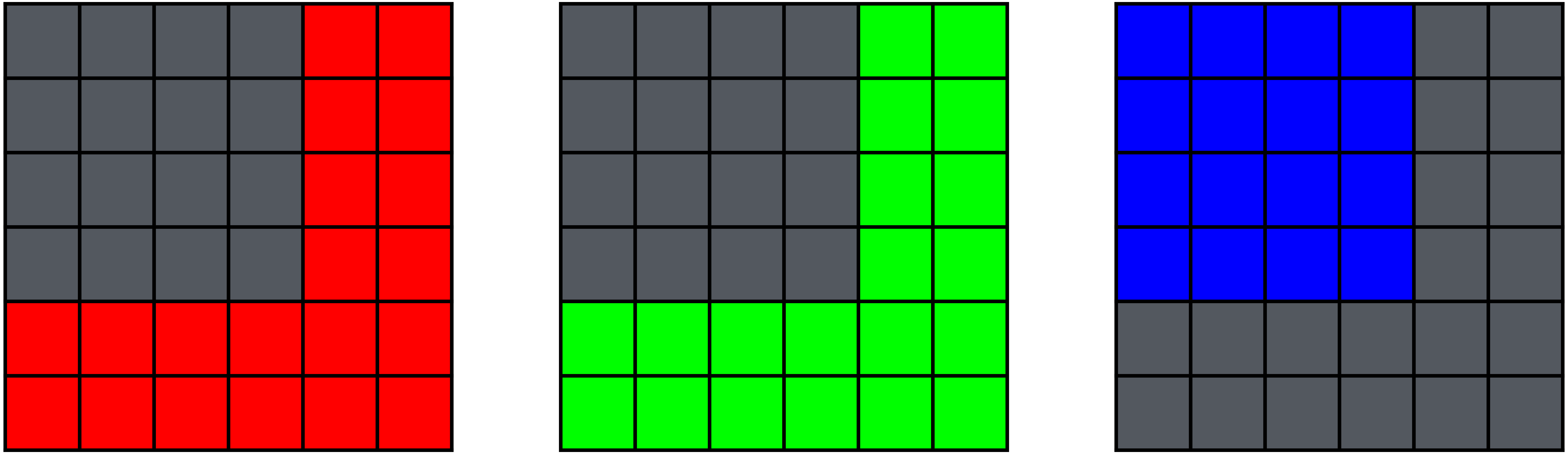
- Easy!



Add back green



Repeat for blue



Fully naïve

After a slide by Frédo Durand

Edge-based green, naïve red blue

After a slide by Frédo Durand

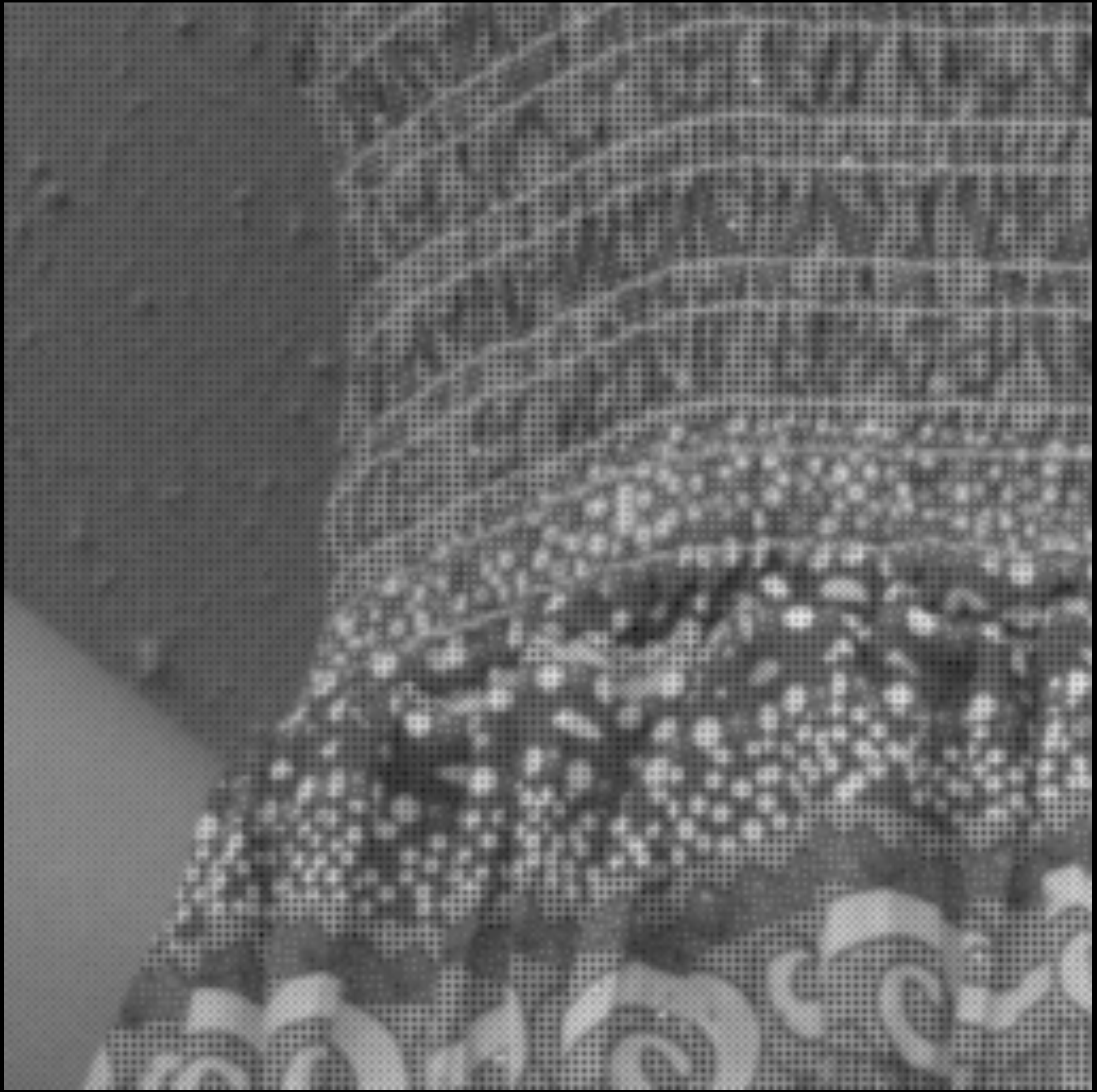
Green-based blue and red

After a slide by Frédo Durand

Still not 100% perfect

After a slide by Frédo Durand

RAW bayer data



After a slide by Steve Marschner

2x2 bayer block

After a slide by Steve Marschner

centered

After a slide by Steve Marschner

linear

After a slide by Steve Marschner

edge-based

After a slide by Steve Marschner

dcraw

After a slide by Steve Marschner

Questions?

Alternative

Interpolate ratio

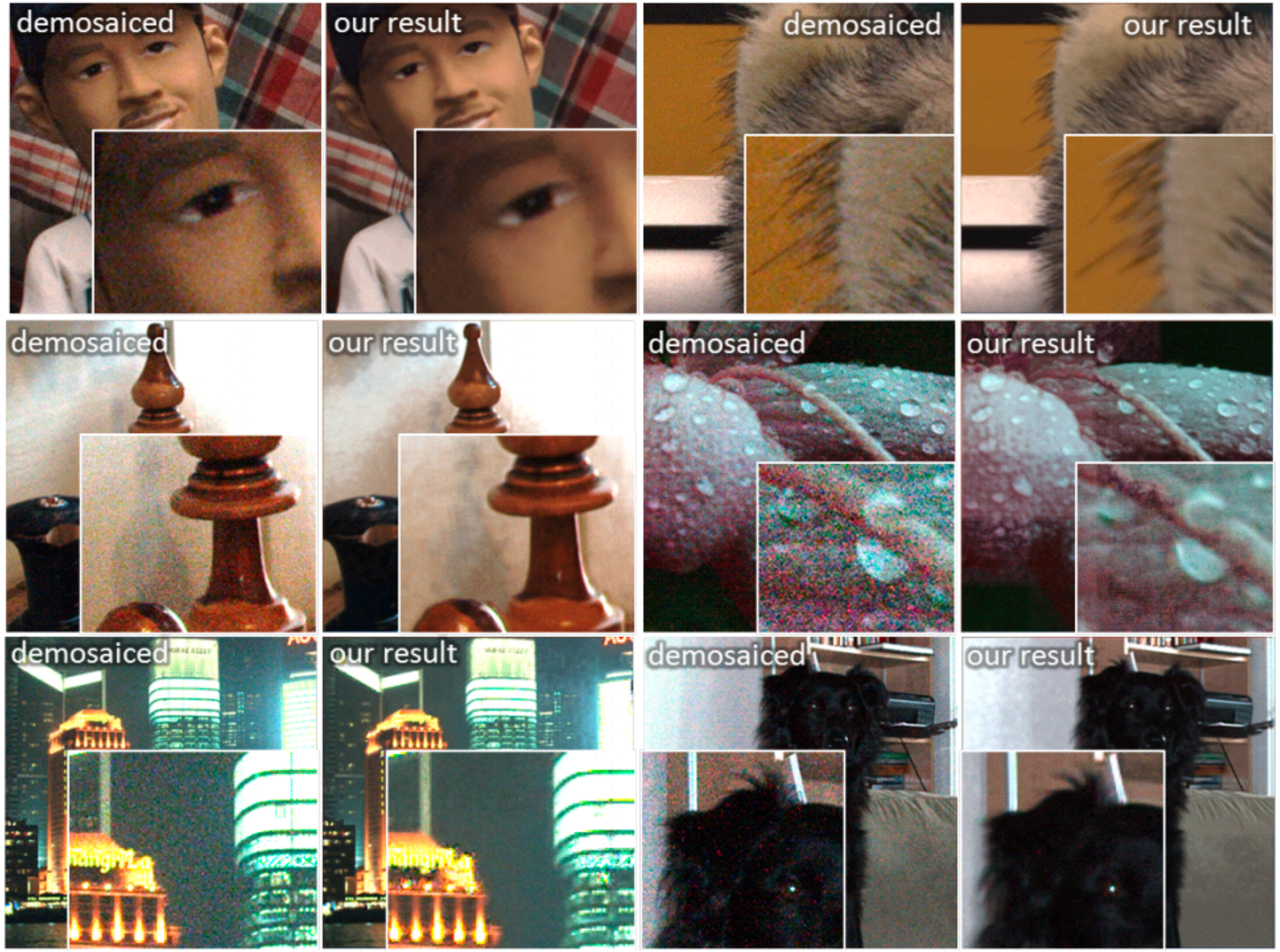
Edge cases

http://www.luminous-landscape.com/contents/DNG-Recover-Edges.shtml

http://www.luminous-landscape.com/forum/index.php?topic=51328.0

Denoising & Demosaicking

http://research.microsoft.com/en-us/UM/people/yasumat/papers/lowlight_CVPR11.pdf



Demosaicing inversion

http://research.microsoft.com/en-us/UM/people/yasumat/papers/cvpr2010_Takamatsu.pdf

Links

http://www.csee.wvu.edu/~xinl/papers/demosaicing_survey.pdf

http://www.unc.edu/~rjean/demosaicing/demosaicing.pdf

http://www.pages.drexel.edu/~par24/rawhistogram/40D_Demosaicing/40D_DemosaicingArtifacts.html

http://www.guillermoluijk.com/tutorial/dcraw/index_en.htm

http://www.cambridgeincolour.com/tutorials/RAW-file-format.htm

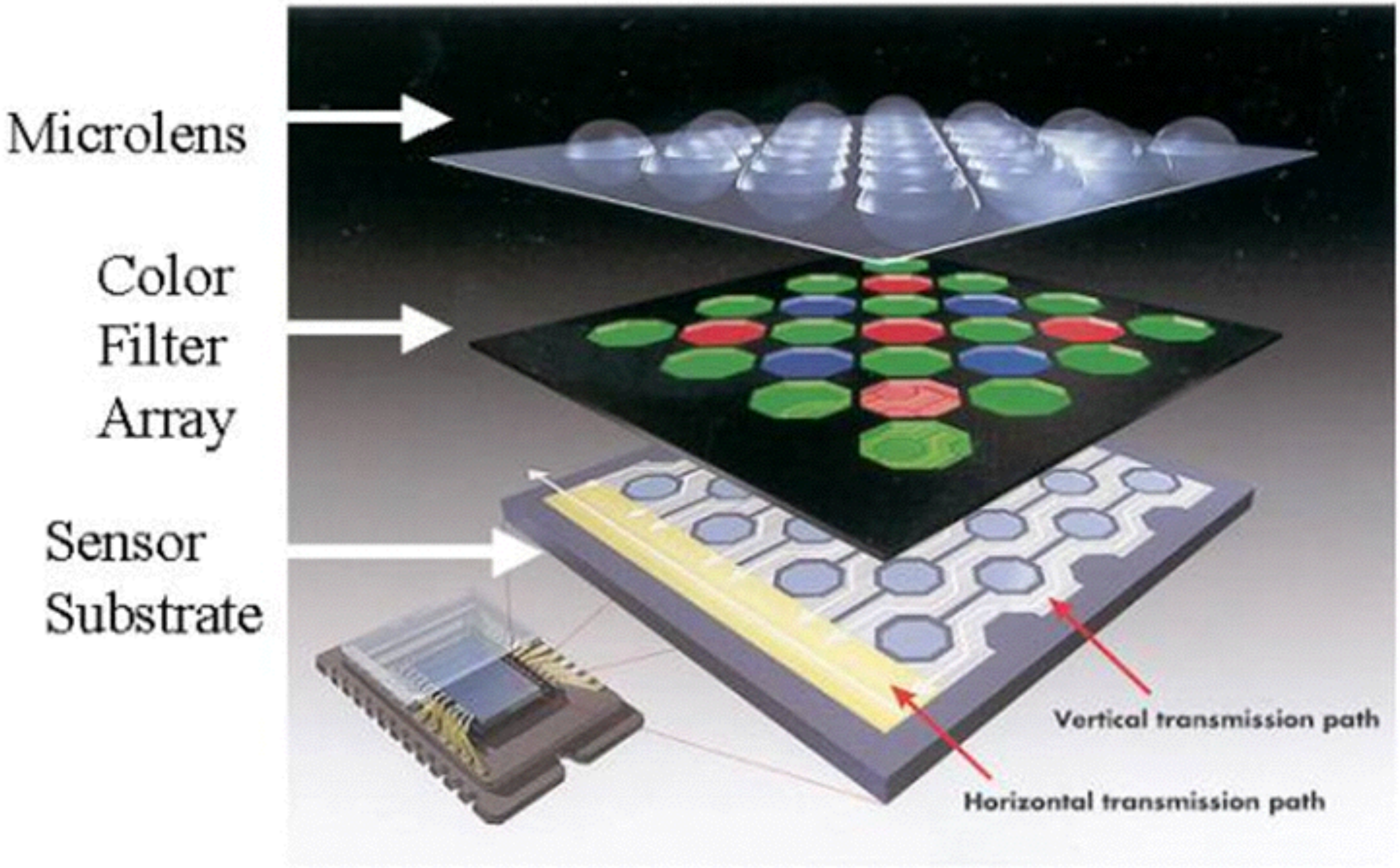
http://www.cambridgeincolour.com/tutorials/camera-sensors.htm

Color correction & calibration

Sensor architecture

Measured pixel values are not *CIE RGB* values!

Remap to appropriate colorspace using transformation derived from response curves of color filters (sensor specific)



Color sensing

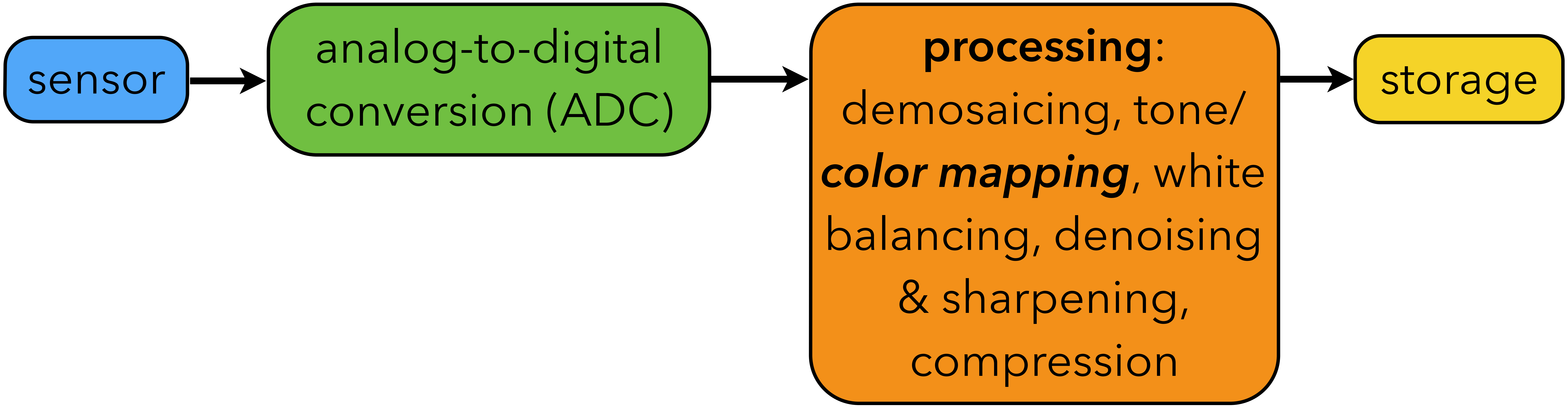
Sensor is like eye

- gives you projection onto a 3D (or >3D) space
- but it is the wrong space!

Problems with measured data

- we have RGB, but not the right RGB
- projection onto sensitivities, not coefficients for primaries (always)
- projection onto wrong space
- results depend strongly on illuminant (help!)

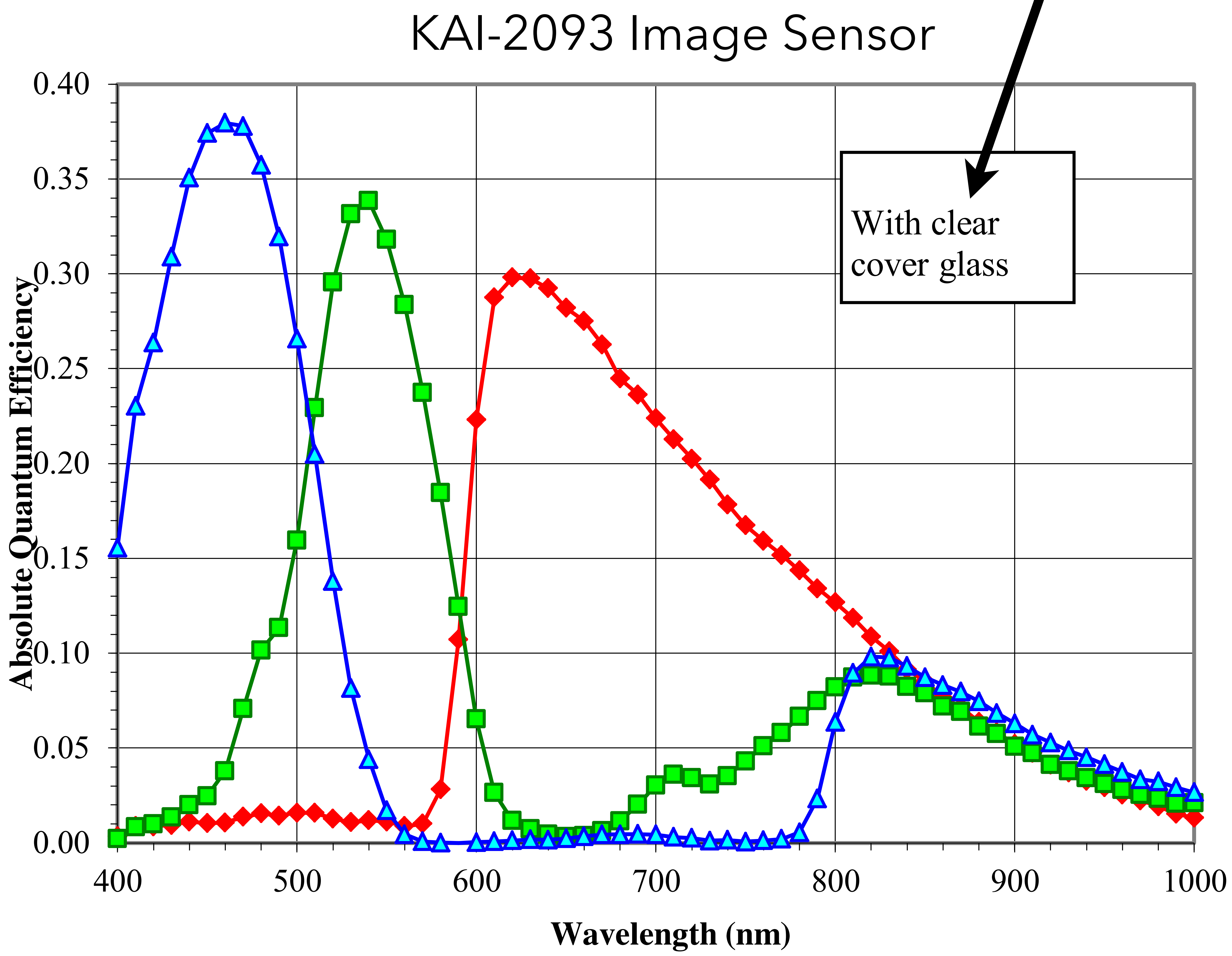
Camera pixel pipeline



Sensor color properties

Like eye, key property is the spectral sensitivity curves

In real cameras there will be a filter to block infrared



Camera color problem

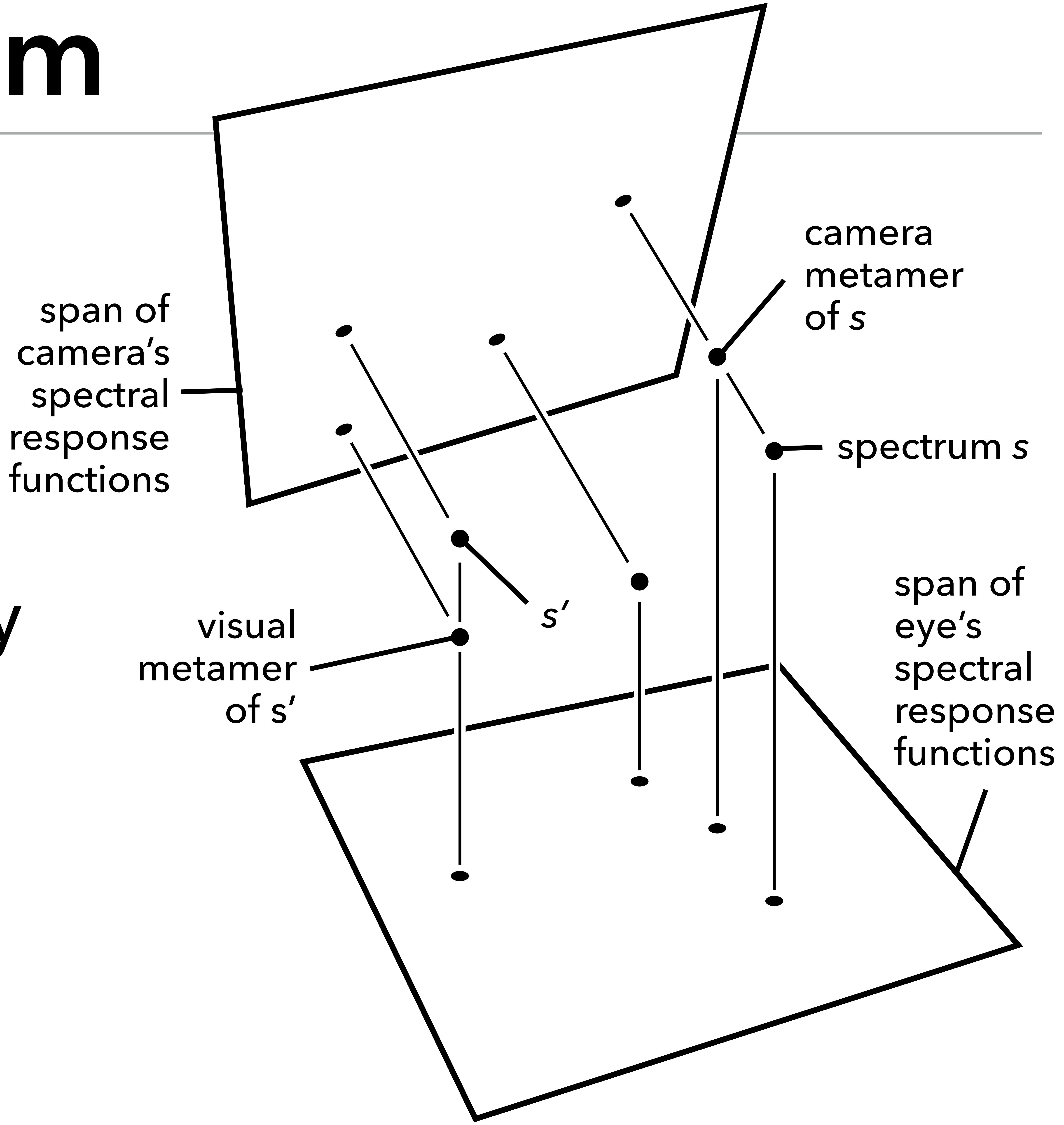
Given camera response, guess corresponding visual response

This guess has to involve assumptions about which reflectance spectra are more likely

Mathematical approach:

- assume spectra in fixed subspace

Or, more often, just derive a transformation empirically



Camera color rendering via subspace

Assume spectrum s is a combination of three spectra

$$\begin{bmatrix} | \\ s \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Work out what combination it is

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \left(\begin{bmatrix} — & r_R & — \\ — & r_G & — \\ — & r_B & — \end{bmatrix} \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \right)^{-1} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- same math as additive color matching

Project that combination onto visual response

Camera color rendering via subspace

Assume spectrum s is a combination of three spectra

$$\begin{bmatrix} | \\ s \\ | \end{bmatrix} = \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Work out what combination it is

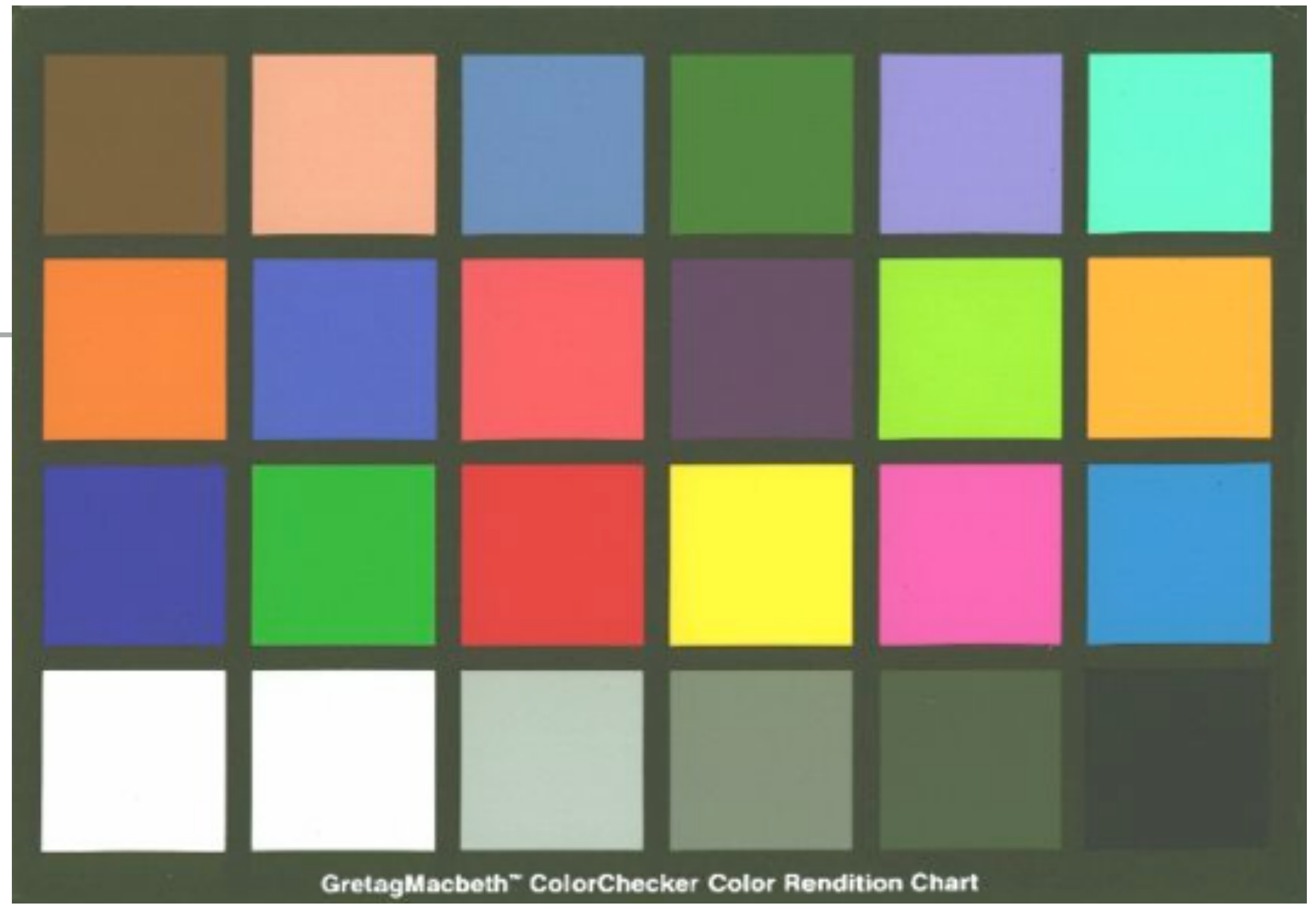
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \left(\begin{bmatrix} — & r_R & — \\ — & r_G & — \\ — & r_B & — \end{bmatrix} \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \right)^{-1} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- same math as additive color matching

Project that combination onto visual response

$$\begin{bmatrix} S \\ M \\ L \end{bmatrix} = \begin{bmatrix} — & r_S & — \\ — & r_M & — \\ — & r_L & — \end{bmatrix} \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \left(\begin{bmatrix} — & r_R & — \\ — & r_G & — \\ — & r_B & — \end{bmatrix} \begin{bmatrix} | & | & | \\ s_1 & s_2 & s_3 \\ | & | & | \end{bmatrix} \right)^{-1} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Empirical color transformation



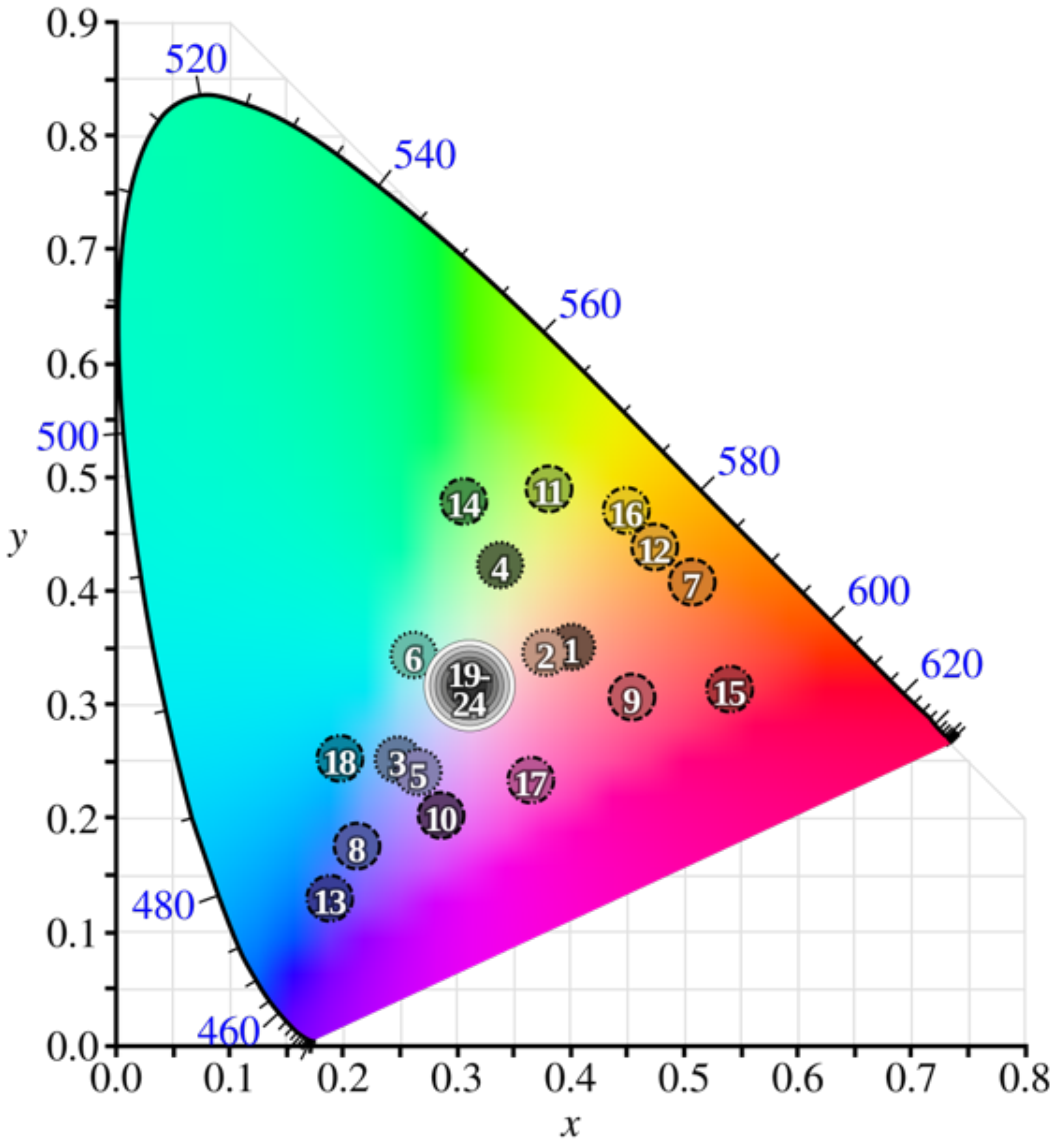
Baseline method: use Macbeth Color Checker

- a set of square patches of known color
  (these days you buy the MCC from X-Rite)

Procedure

1. Photograph the color checker under uniform illumination
2. Measure the camera-RGB values of the 24 squares
3. Look up the XYZ colors of the 24 squares
4. Use linear least squares to find a 3x3 matrix that approximately maps the camera responses to the correct answers

$$\min_{M} \left\| \underset{3\times 24}{C_{\text{macbeth}}} - \underset{3\times 3}{M}\, \underset{3\times 24}{C_{\text{camera}}} \right\|$$



Considering the illuminant

Problem with previous slide

- the camera-RGB colors depend on the illuminant
- the matrix M only works for the illuminant that was used to calibrate!

Solutions?

- calibrate separately for every illuminant?
- correct for illuminant first, then apply matrix!

von Kries hypothesis: eye accounts for illuminant by simply scaling the three cone signals separately

- leads to “von Kries transform”: multiply by a diagonal matrix

Putting it together: color processing

Calibrate your color matrix using a carefully white-balanced image

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

For each photograph:

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

raw sensor color

white balanced raw sensor color

white balanced and matrixed to sRGB

Slide credits

Frédo Durand

Steve Marschner

Matthias Zwicker

Marc Levoy