Demosaicking



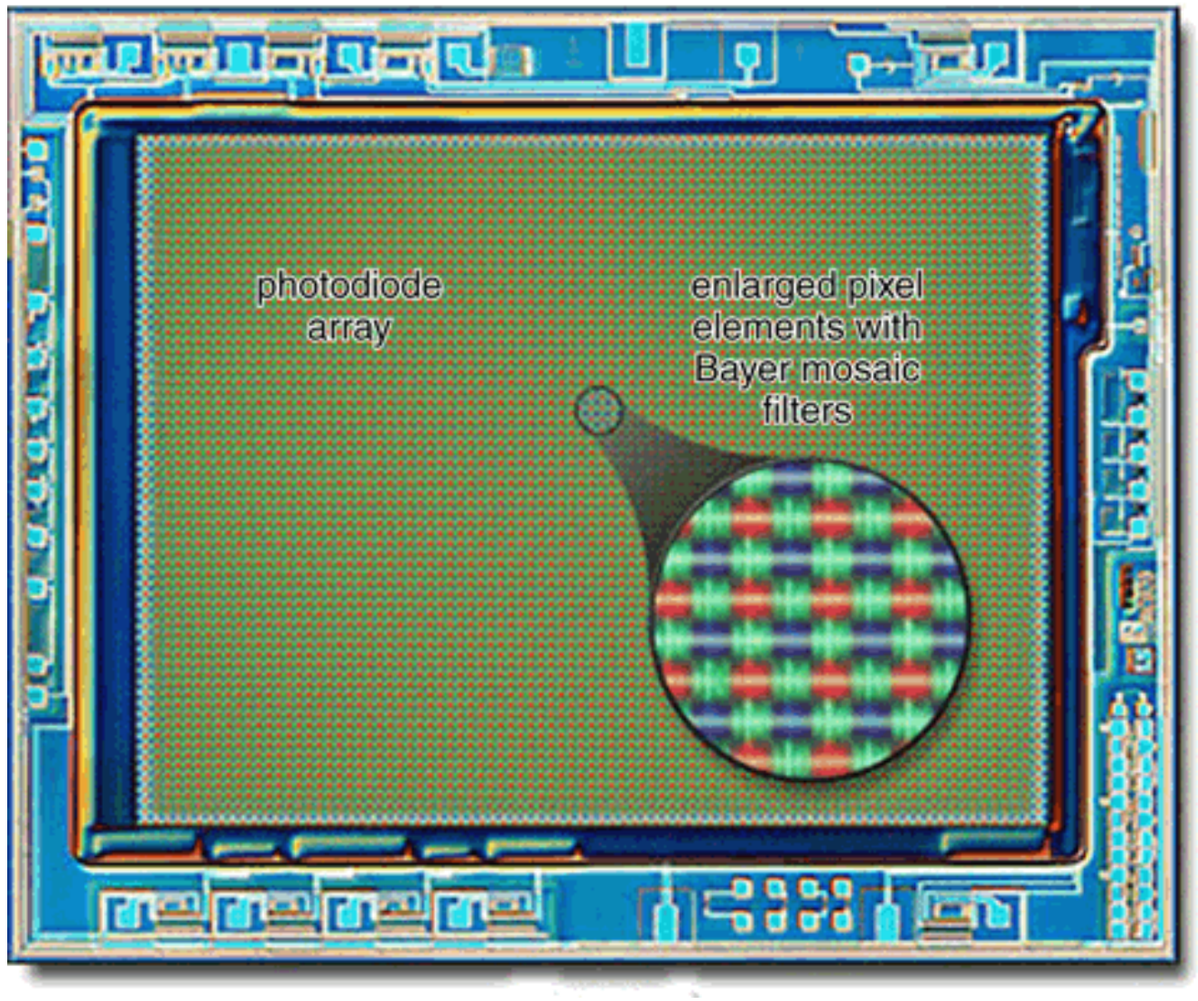
Demosaicking



Fredo Durand
MIT EECS 6.8370/1

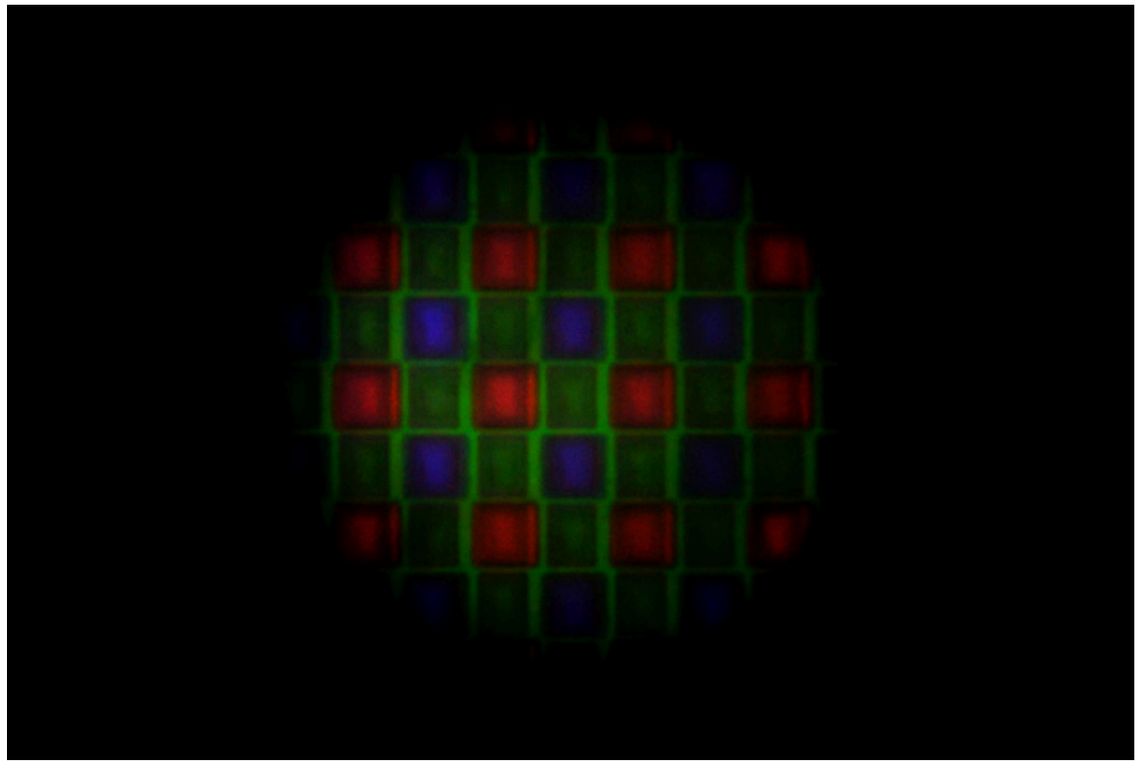
Bayer mosaic

Sensor



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

Microscope view of a CCD

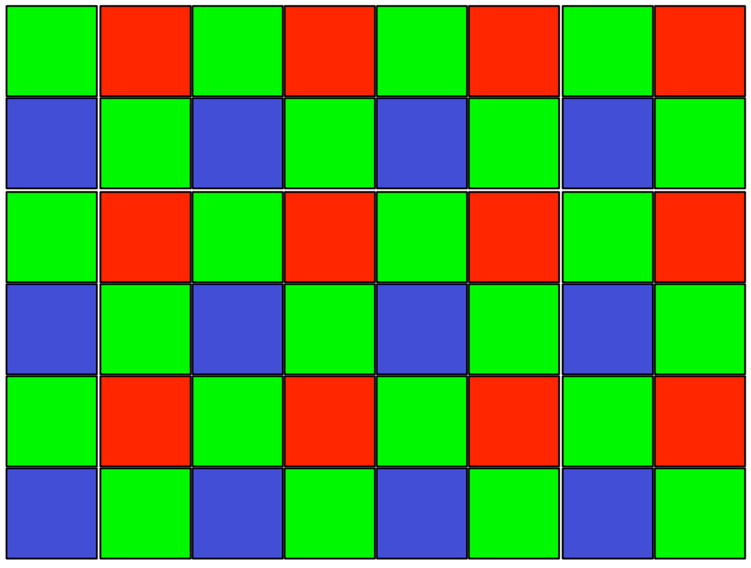


- By **kevincollins123**

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

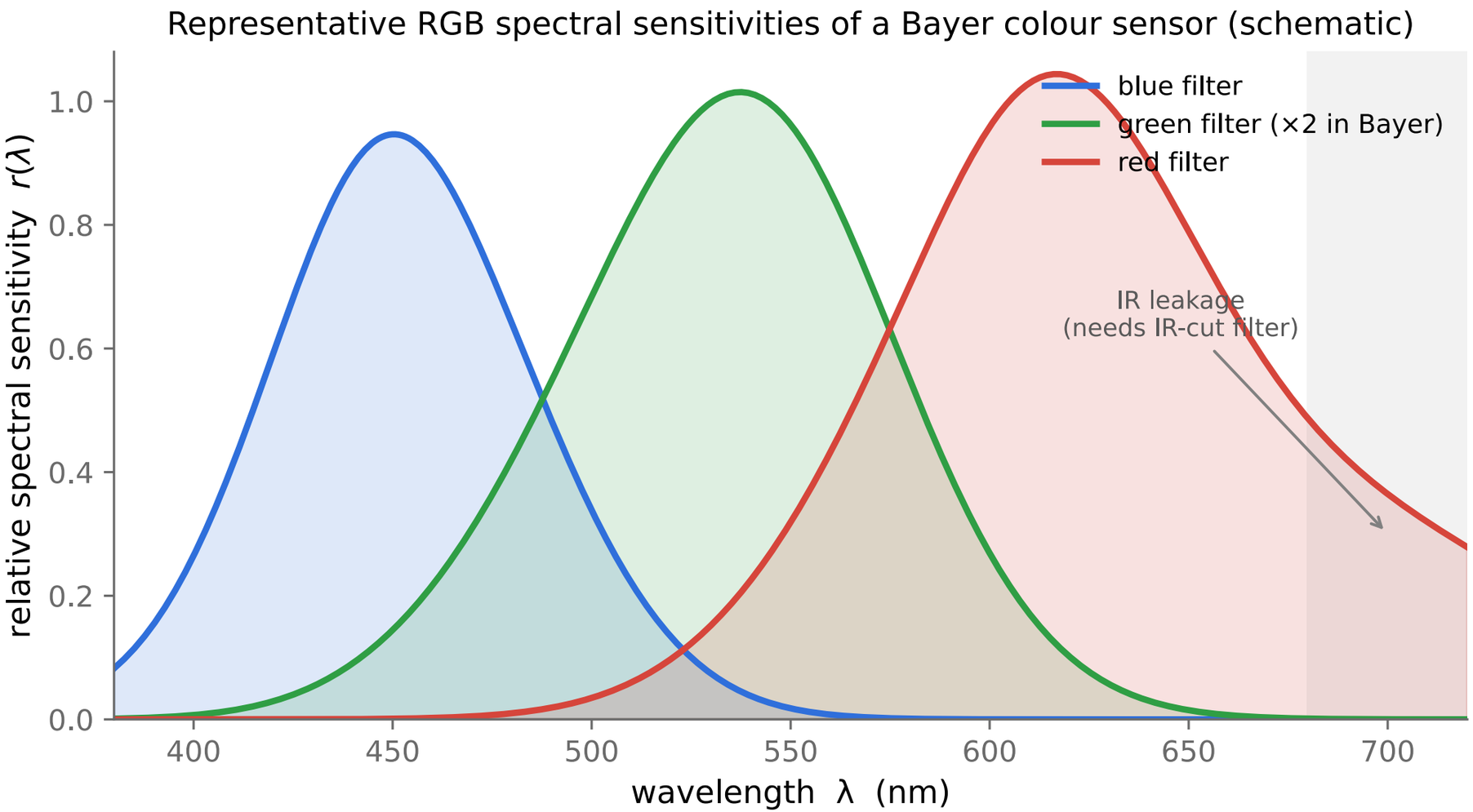
Bayer RGB mosaic

- Each photosite has a different color filter



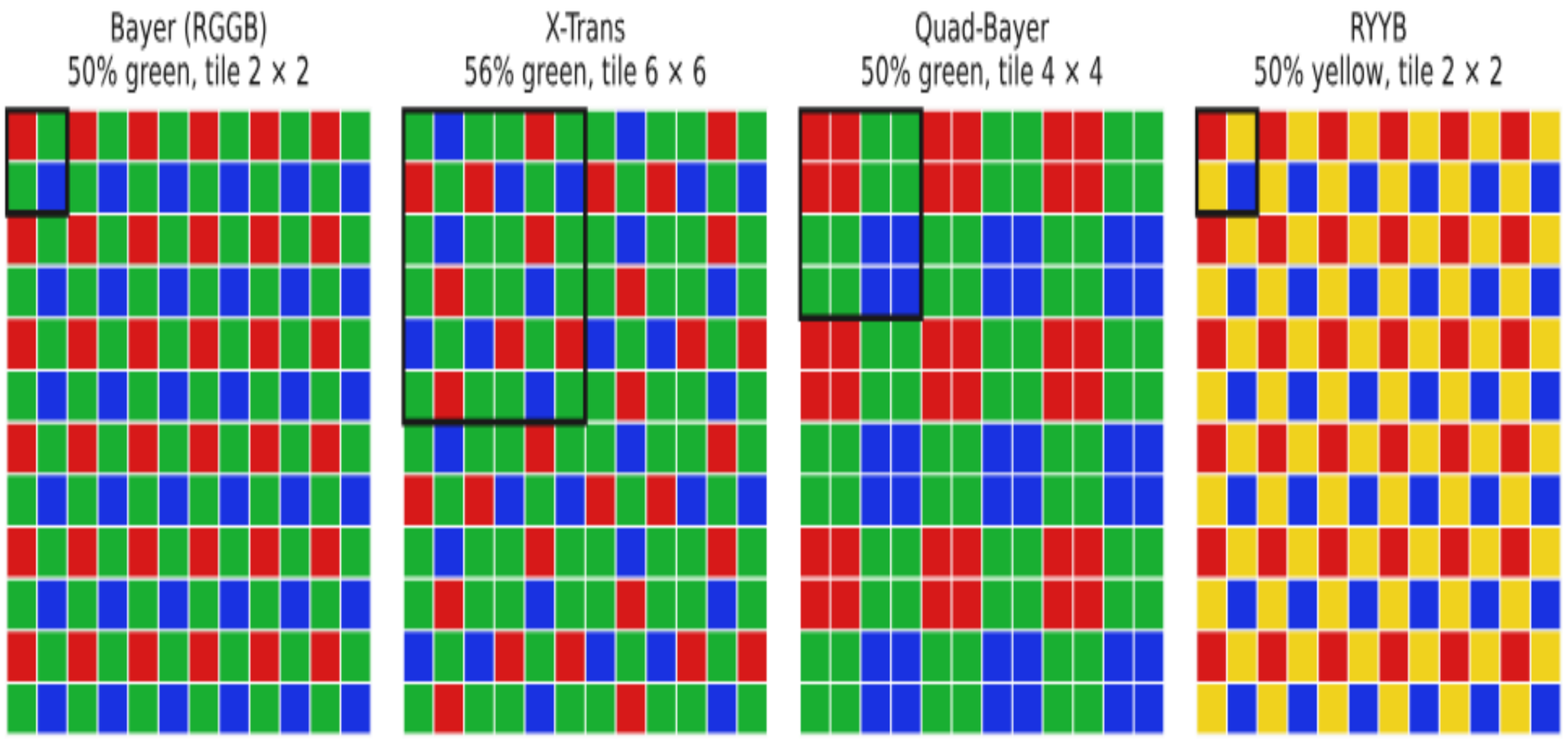
Note that which one is the upper left color is arbitrary and depends on the camera

Figure 2.7.5: Representative RGB spectral sensitivities of a digital color sensor



Each curve is the probability that a photon of a given wavelength, passing the red, green, or blue Bayer filter, is counted by the photosite beneath, the r(λ) of…

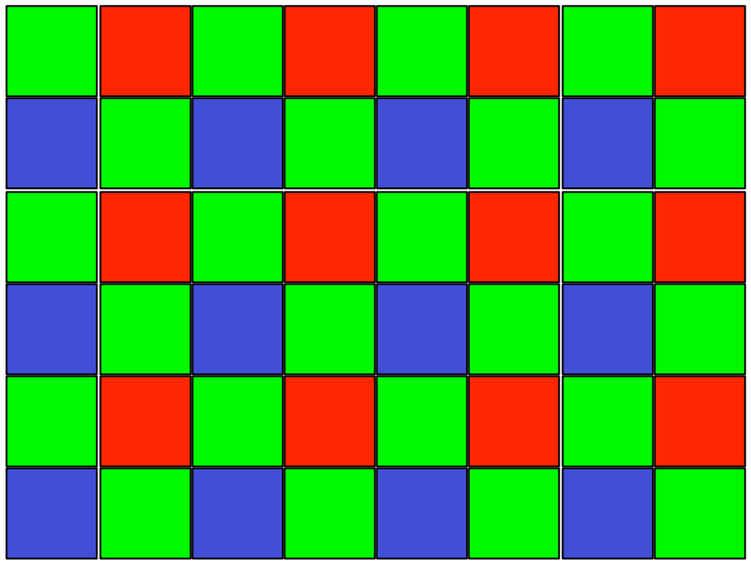
Figure 15.0.1: **The Bayer color filter array (left) and the RYYB variant (right)**



The classic Bayer tile repeats a 2×2 pattern of two green, one red, and one blue filter (RGGB), so each pixel measures only one color and the other two must be…

Bayer RGB mosaic

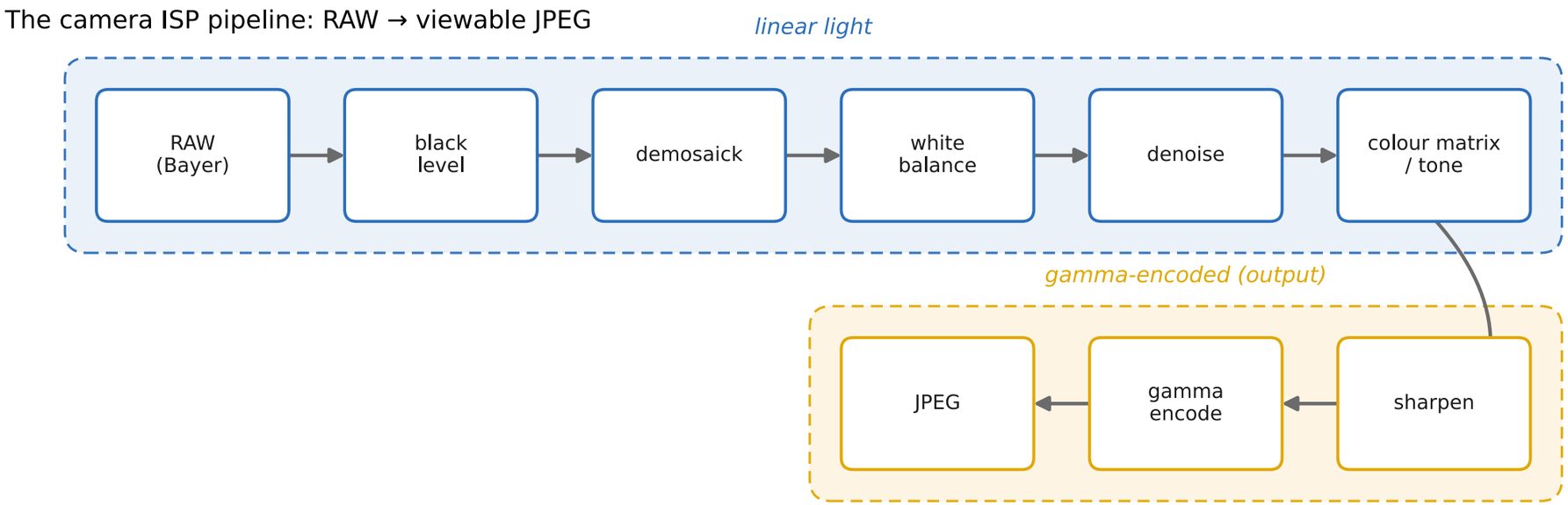
- Why more green?
  - We have 3 channels and square lattices don't like odd numbers
  - It's the spectrum "in the middle"
  - More important to human perception of luminance



RAW files

- **Straight measurement from sensor**
  - right after A/D conversion(well, kind of)
- **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
  - **http://www.mit.edu/~kimo/blog/linear.html**

Figure 8.12.1: The image signal processor (ISP) pipeline as a block diagram

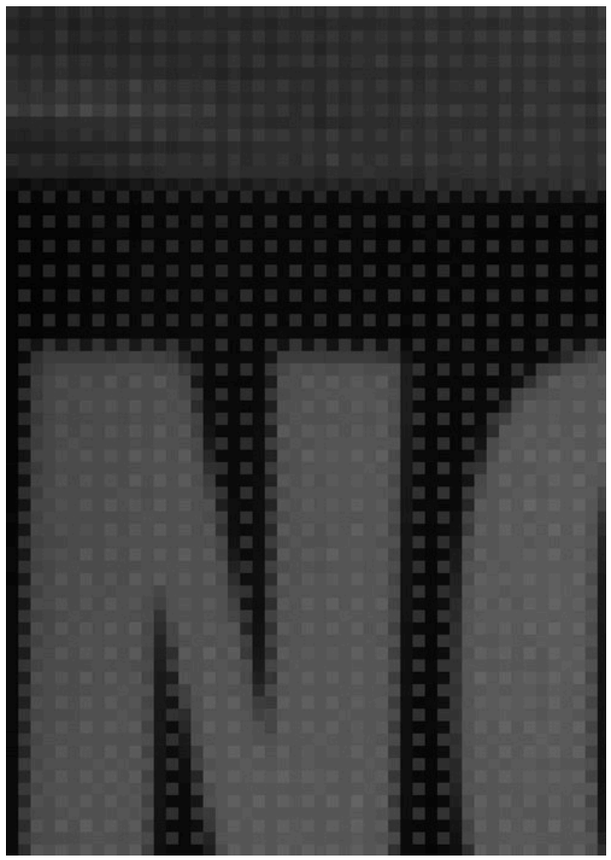
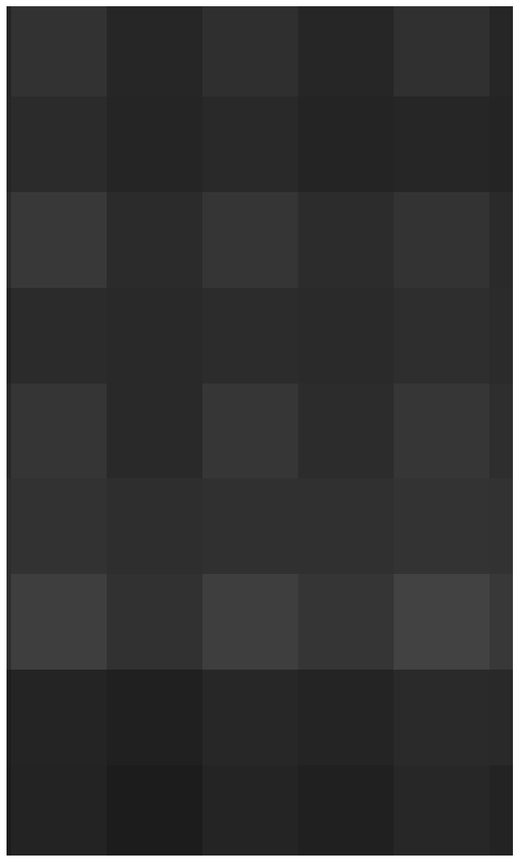


A photograph flows left to right through fixed stages: RAW (linear Bayer mosaic, 12–14 bits) → black level subtraction (with hot-pixel repair) → demosaick (recover full…

A RAW file

- From a Nikon D70



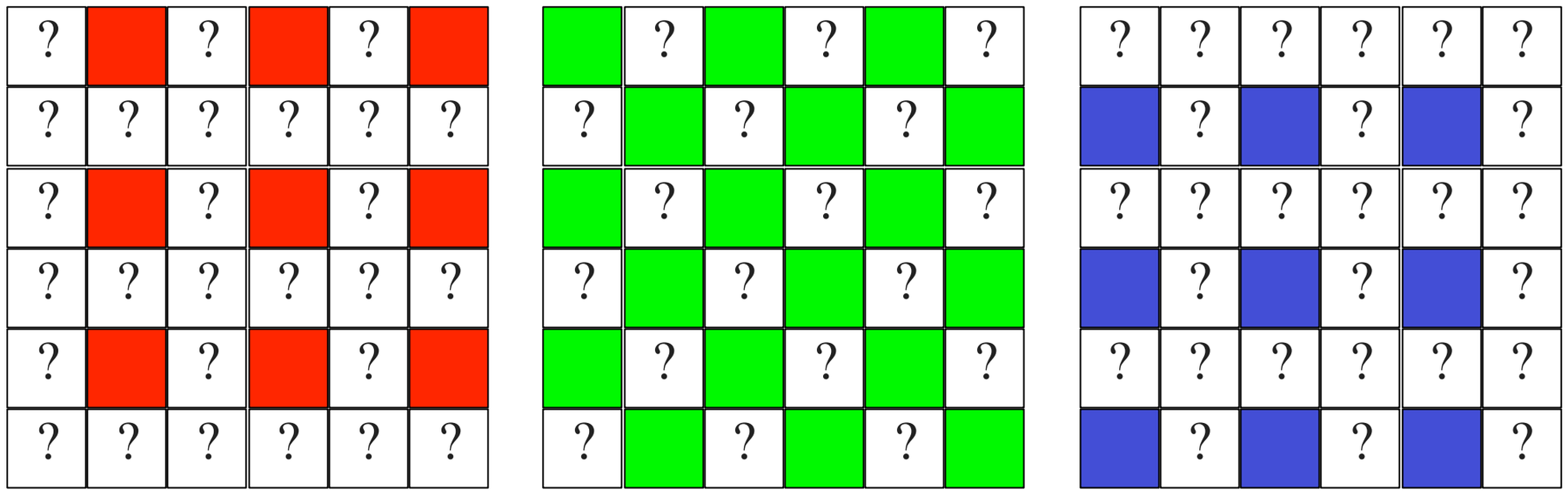
Questions?



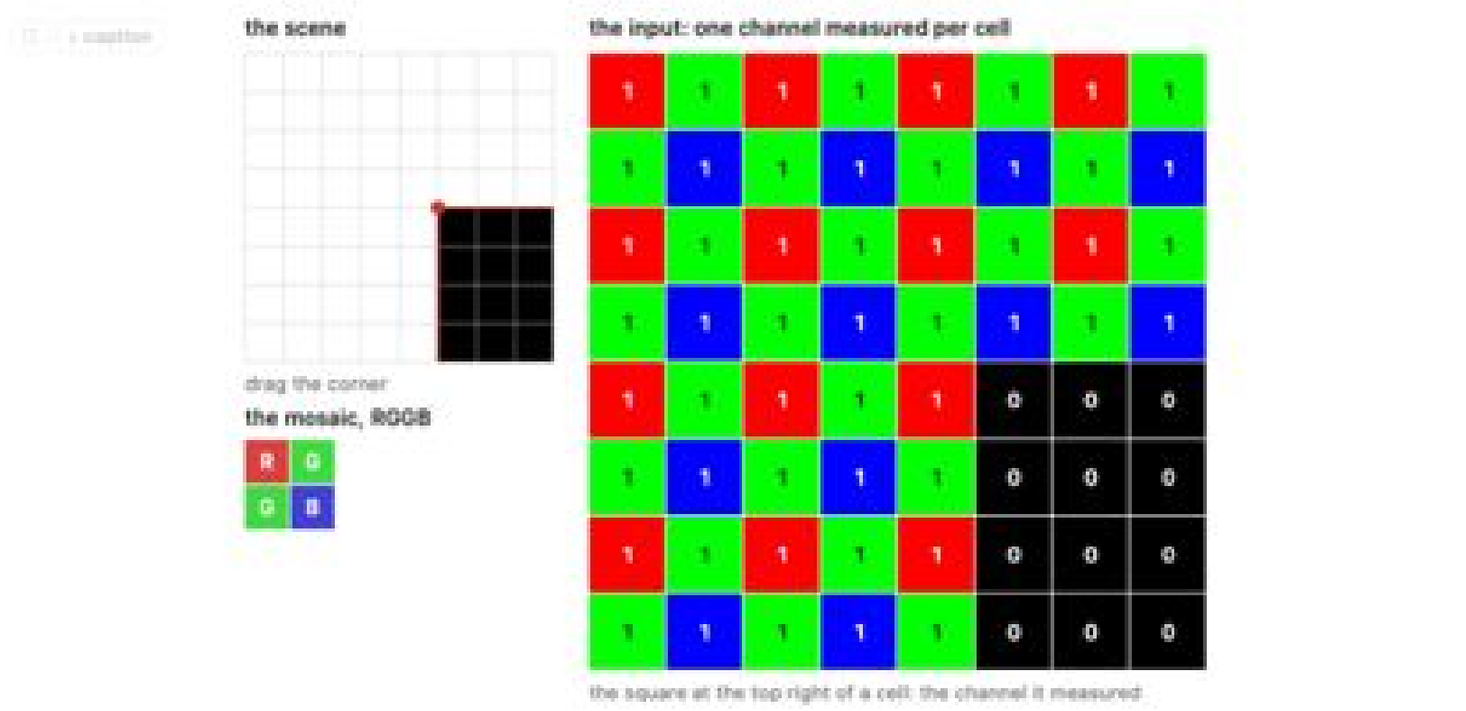
Demosaicking

Demosaicing

- Interpolate missing values



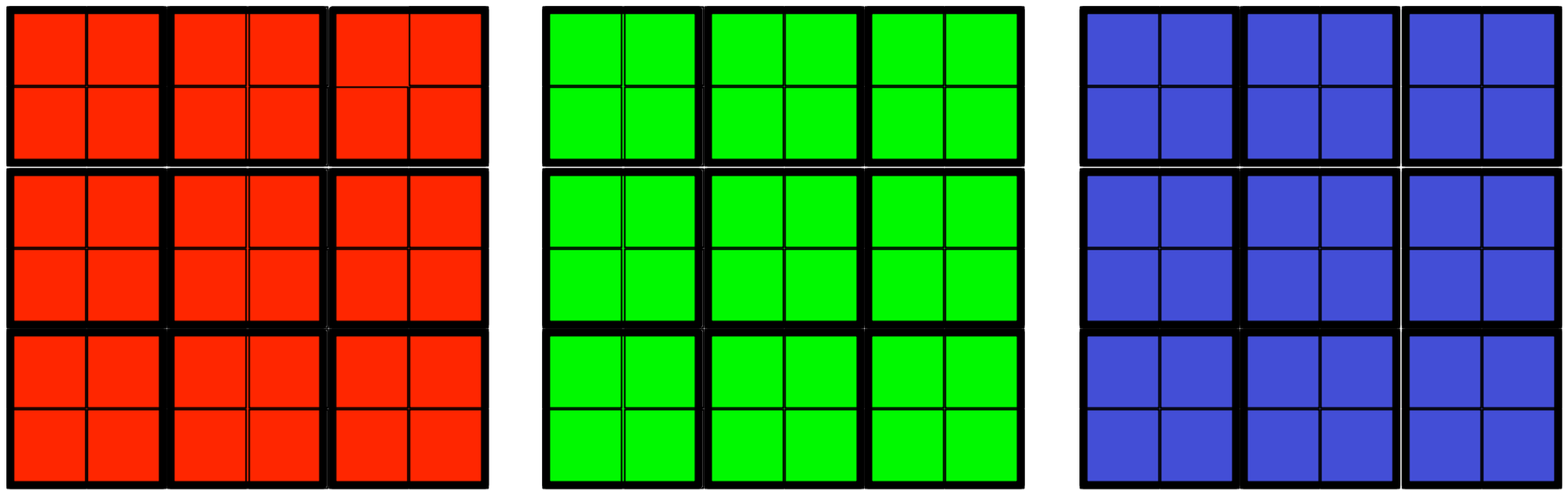
Demo: The corner, step by step: input: mosaic



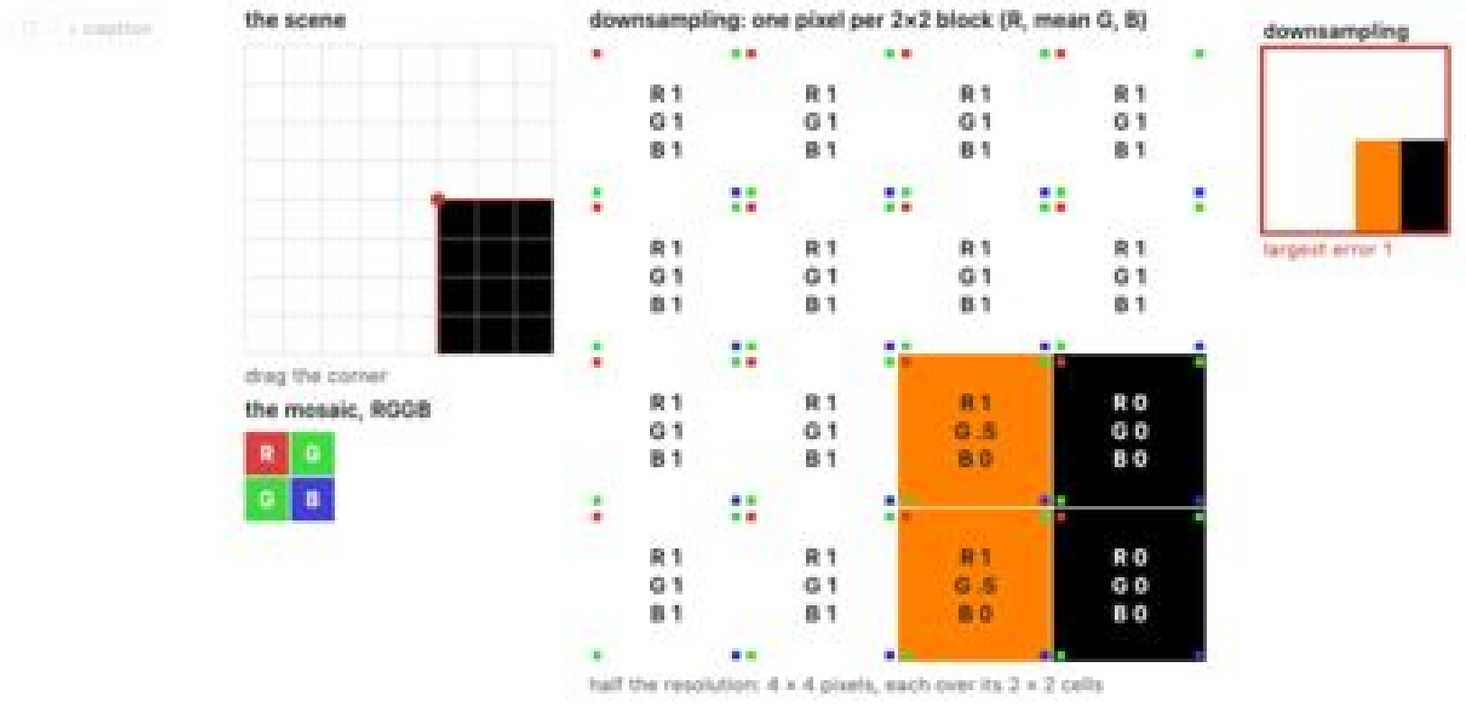
Demo courtesy of Frédo Durand, MIT.

Demosaicing

- Simplest solution: downsample!
  - Nearest-neighbor reconstruction
  - Problems:
    - resolution loss
    - color artifacts at edges



Demo: The corner, step by step: downsample

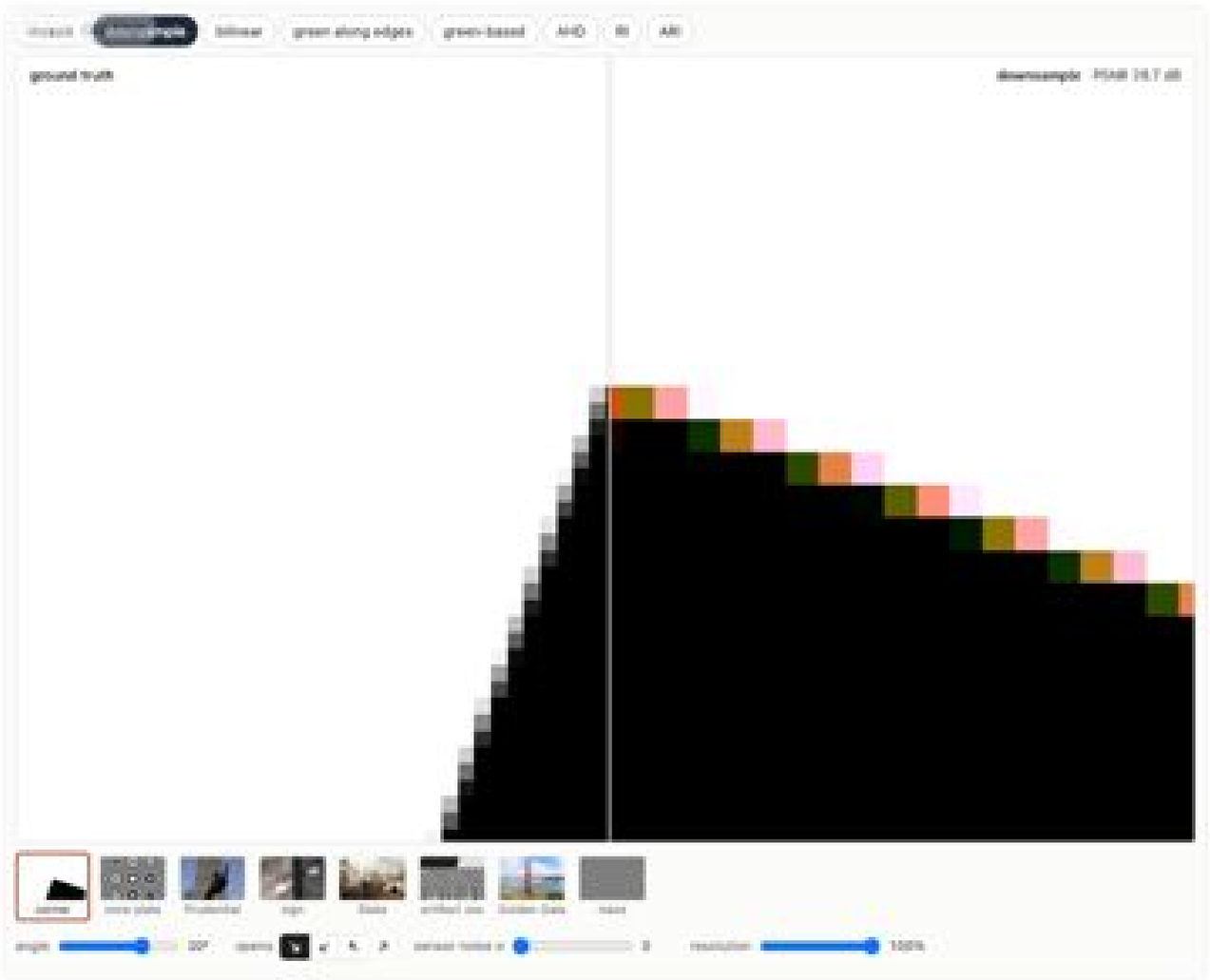


1 input: mosaic | 2 downsample | 3 bilinear: G | 4 bilinear: R, B | 5 green-based: G along edges | 6 green-based: R−G, B−G | 7 green-based: interpolate | 8 green-based: add G back

scene: black on white | corner angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

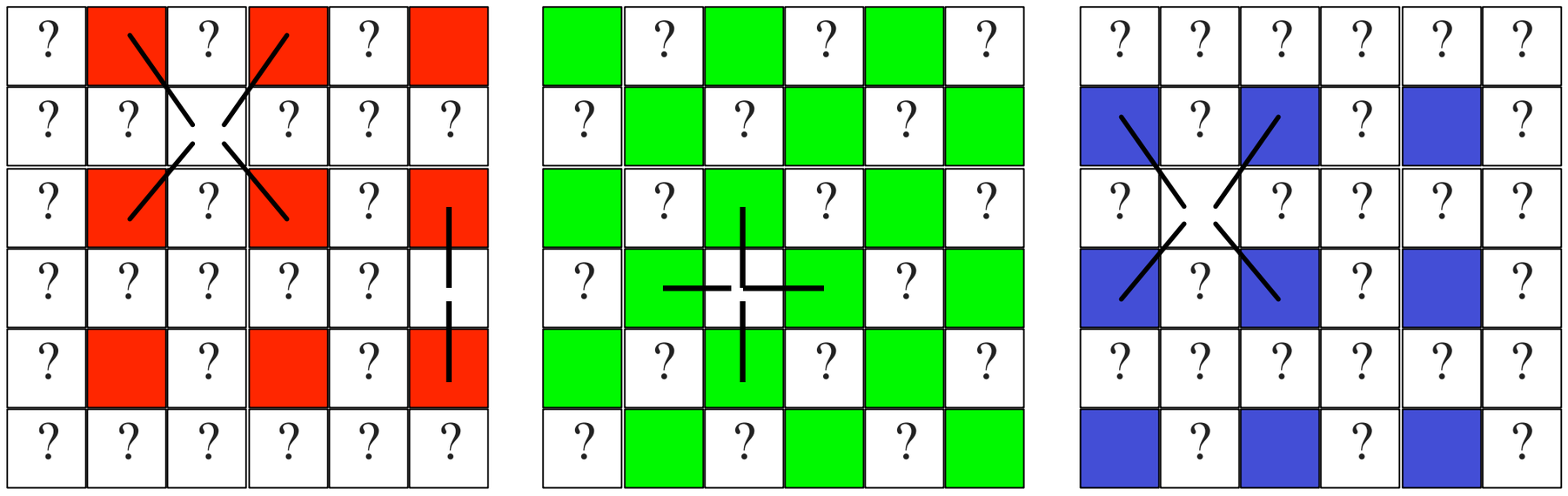
Demo: One method against the truth: downsample



Demo courtesy of Frédo Durand, MIT.

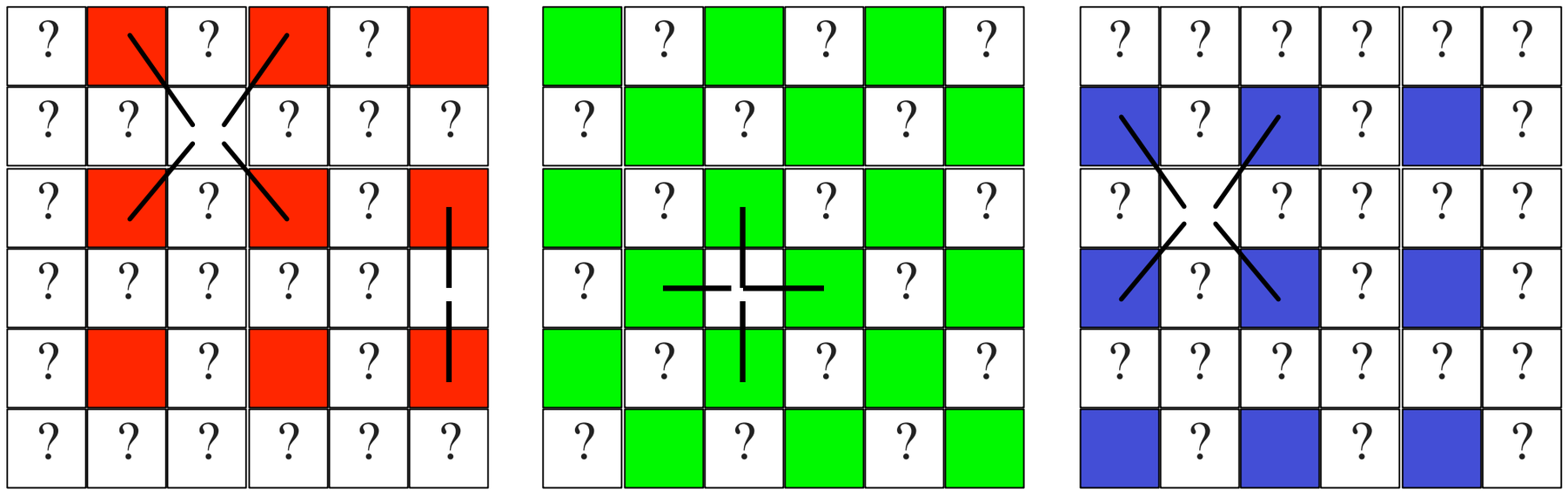
Linear interpolation

- Average of the 4 or 2 nearest neighbors
  - Linear (tent) kernel
- For example:
  newgreen=



Linear interpolation

- Average of the 4 or 2 nearest neighbors
  - Linear (tent) kernel
- For example:
  newgreen=0.25(up+left+right+down)



Better

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

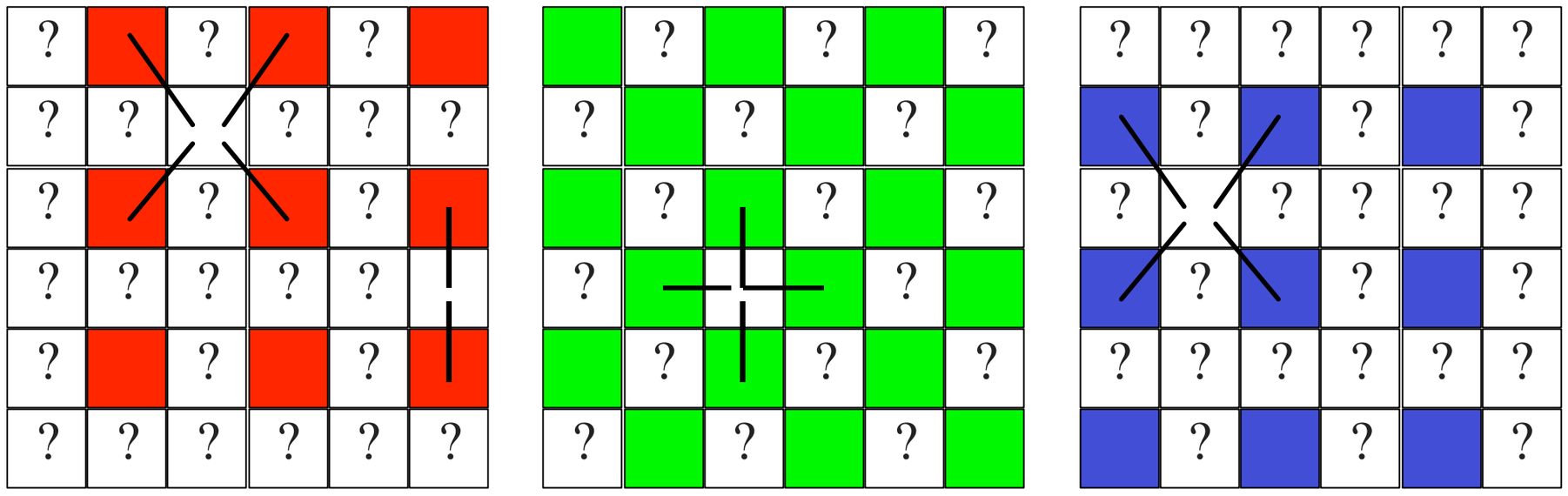
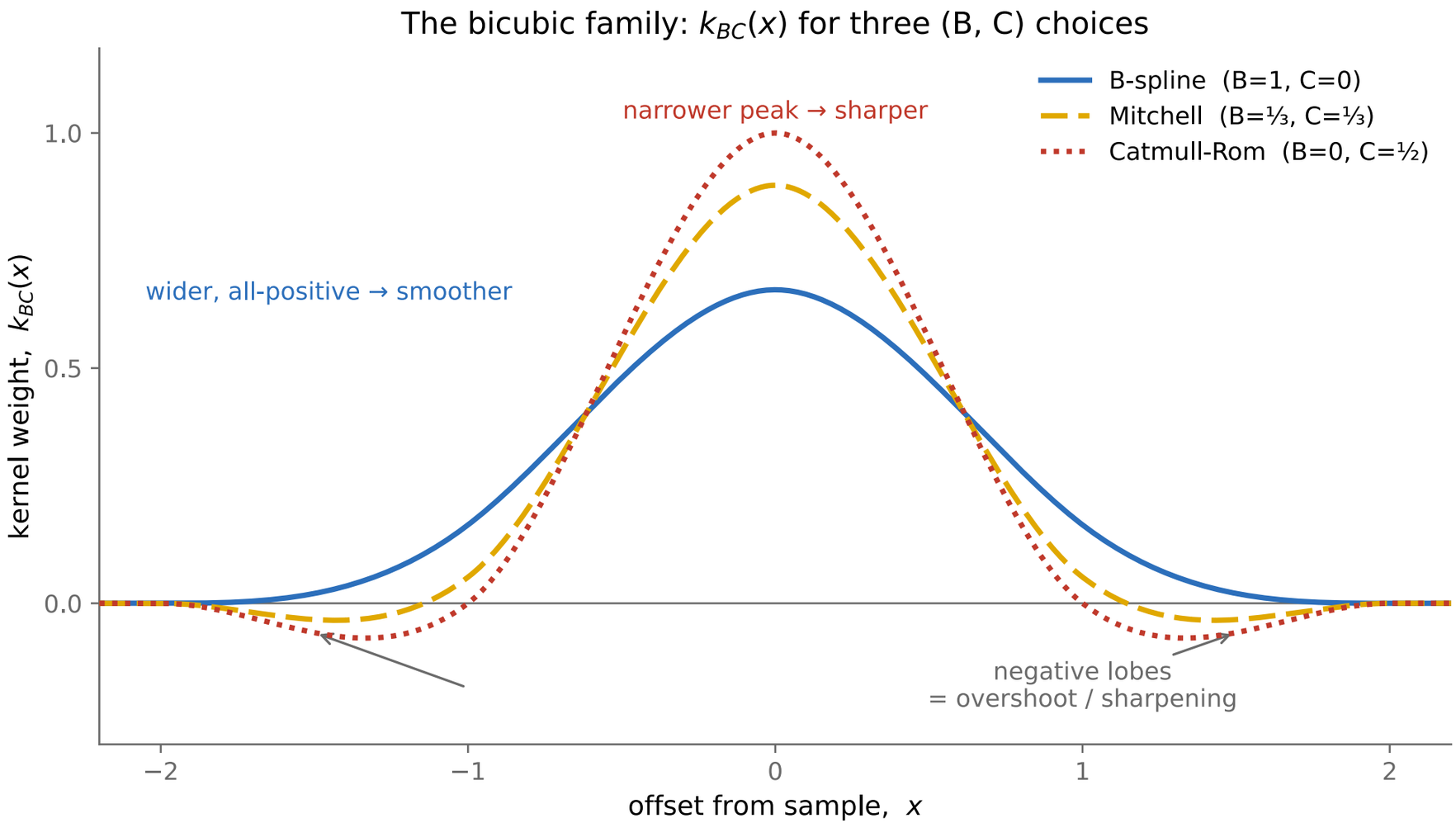


Figure 6.5.8: The cubic kernel k_BC(x) as a one-parameter-pair family, plotted…



The cubic kernel k_BC(x) as a one-parameter-pair family, plotted for the three standard members: cubic B-spline (B,C)=(1,0), all-positive and smooth; Catmull–Rom…

Results of simple linear

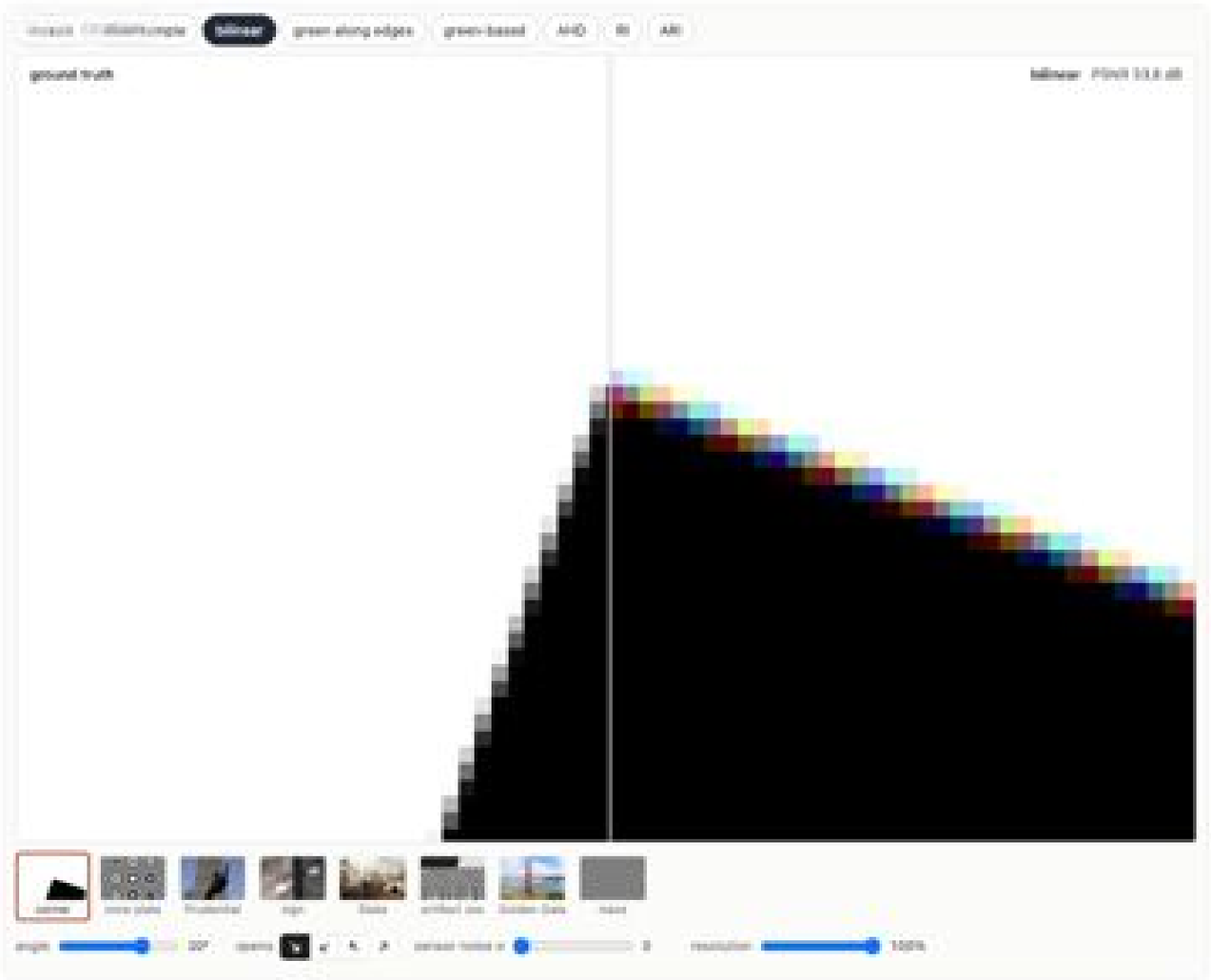


Results - not perfect





Demo: One method against the truth: bilinear



Demo courtesy of Frédo Durand, MIT.

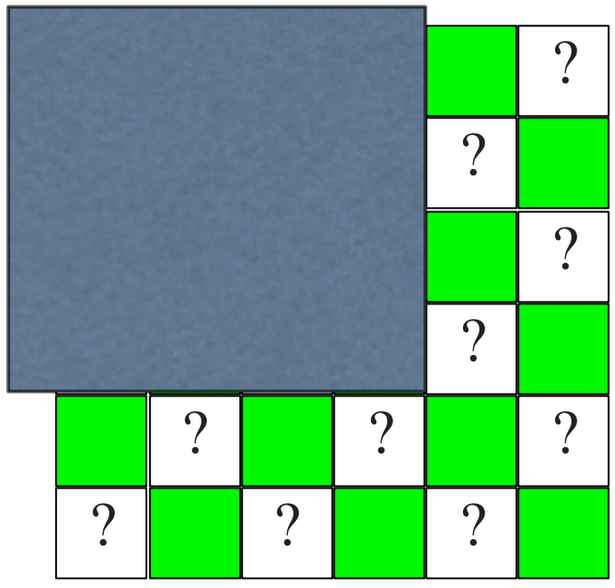
Questions?





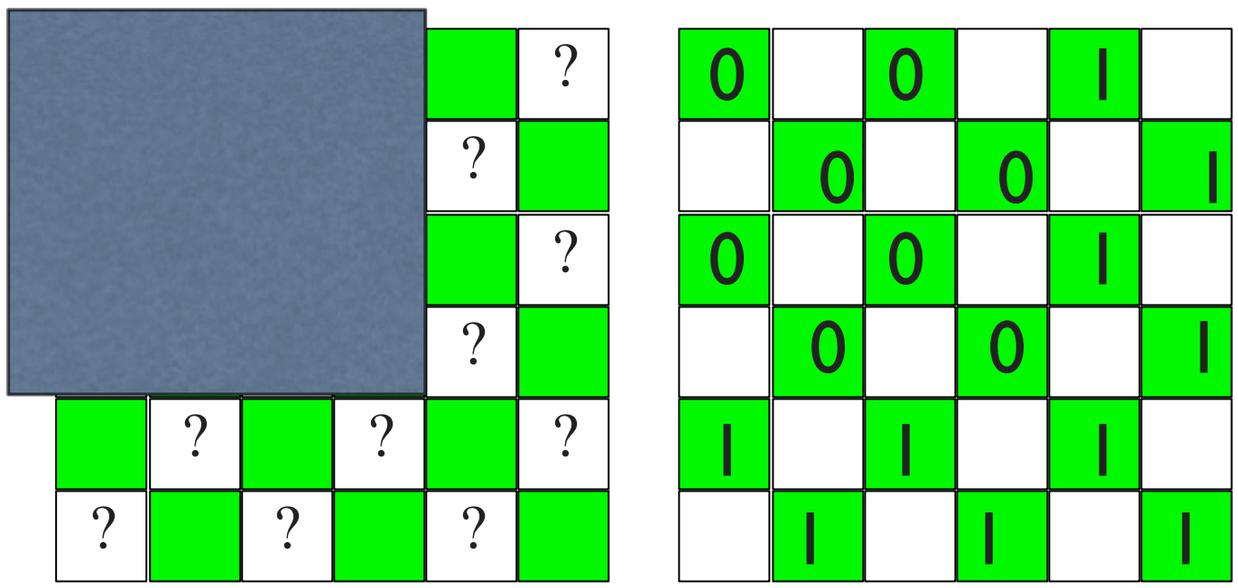
The problem

- Imagine a black-on-white corner
- Let’s focus on the green channel for now



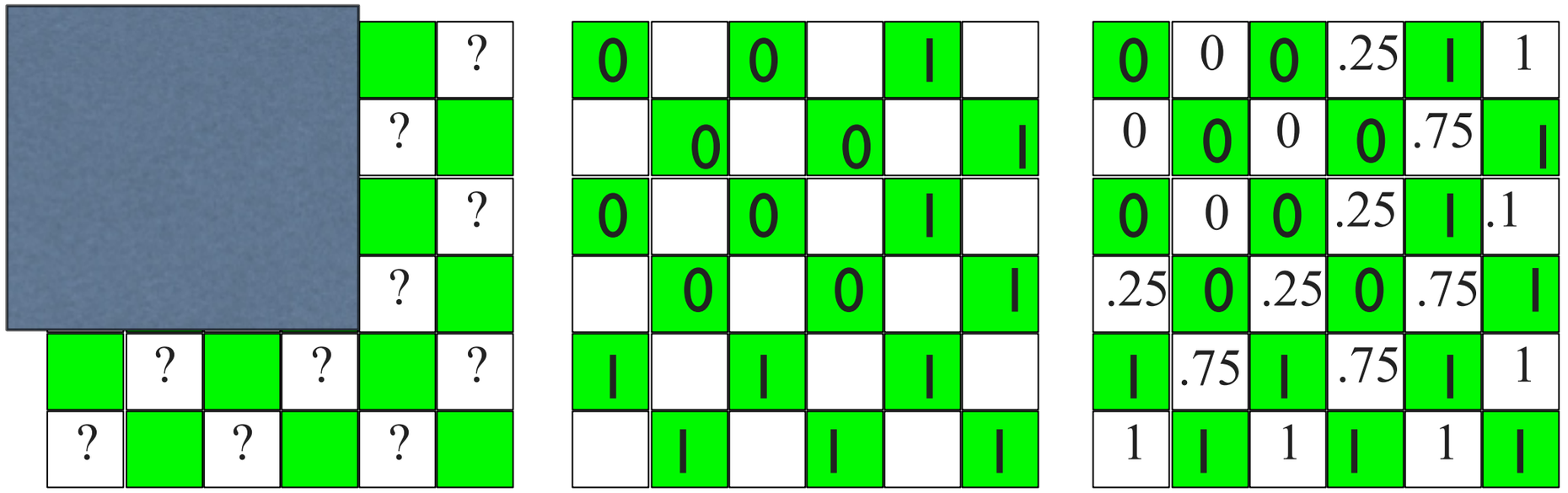
The problem

- Imagine a black-on-white corner



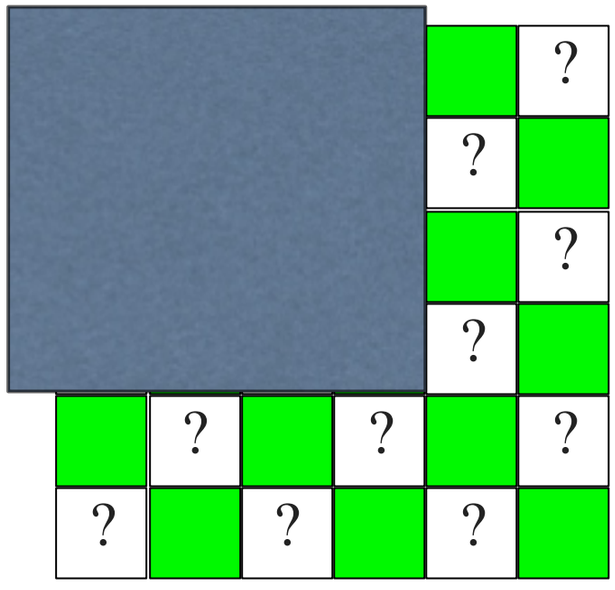
The problem

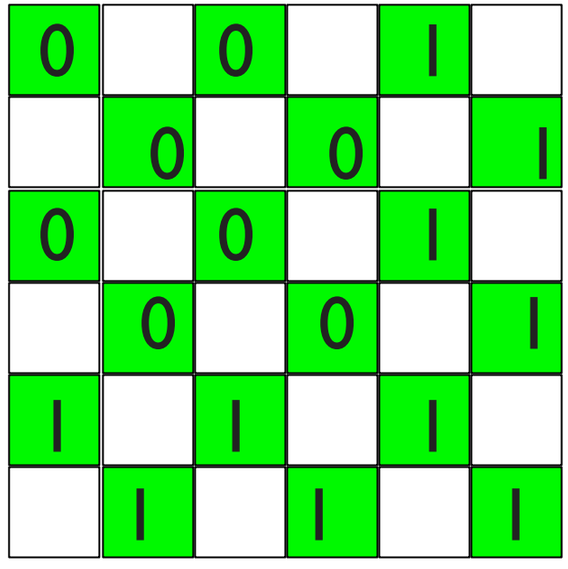
- Imagine a black-on-white corner

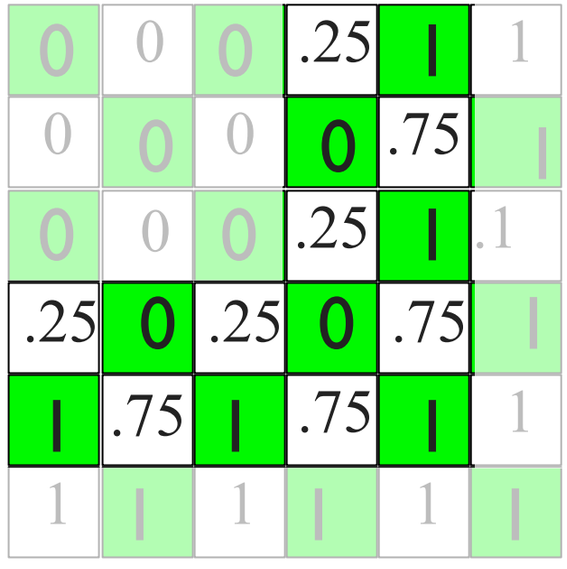


The problem

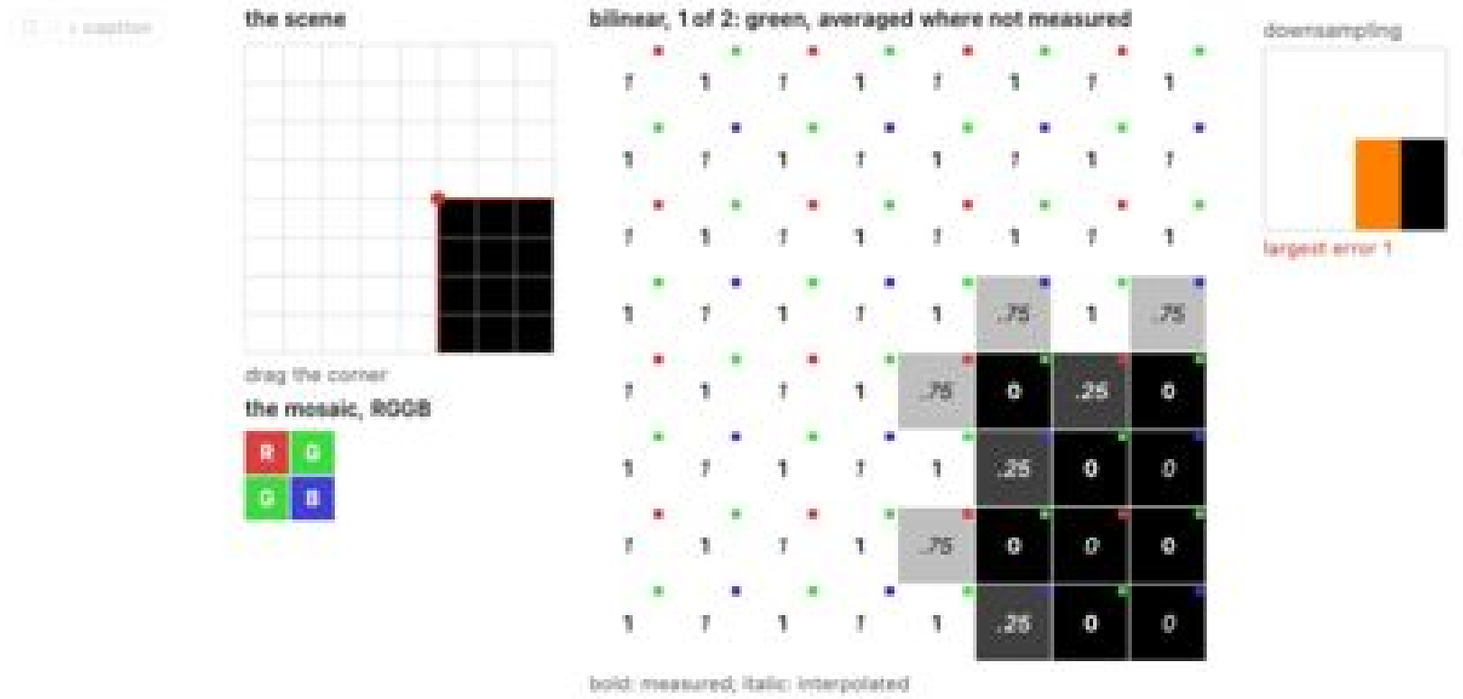
- Imagine a black-on-white corner







Demo: The corner, step by step: bilinear: G



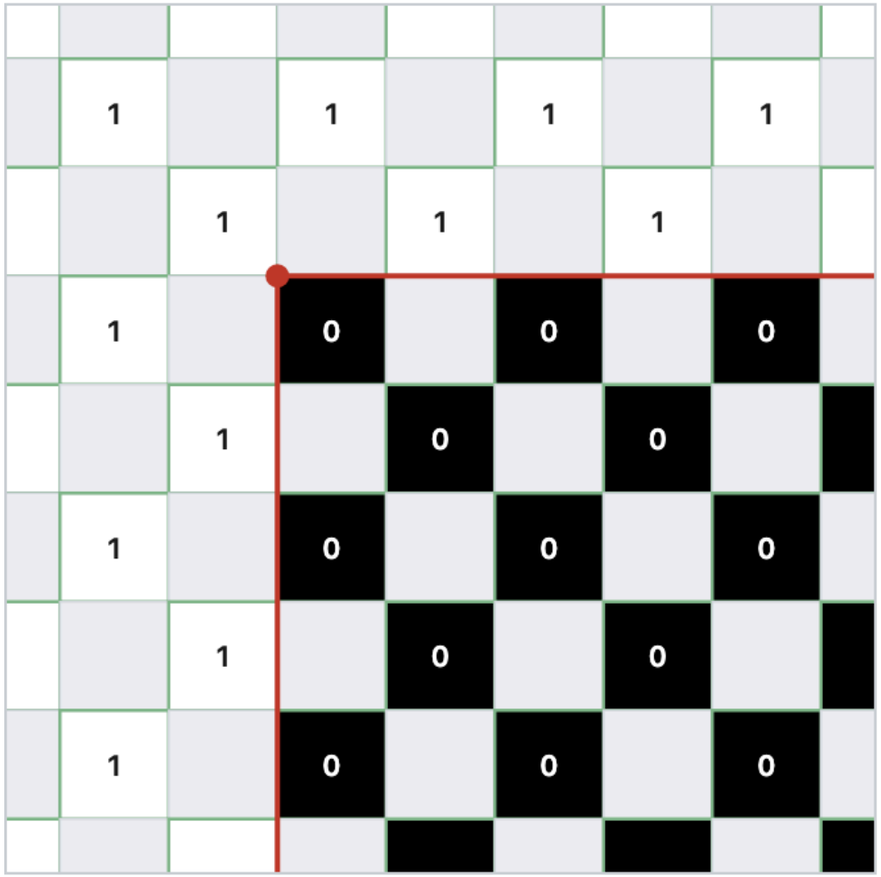
1 input: mosaic | 2 downsample | 3 bilinear: G | 4 bilinear: R, B | 5 green-based: G along edges | 6 green-based: R−G, B−G | 7 green-based: interpolate | 8 green-based: add G back

scene: black on white | corner angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

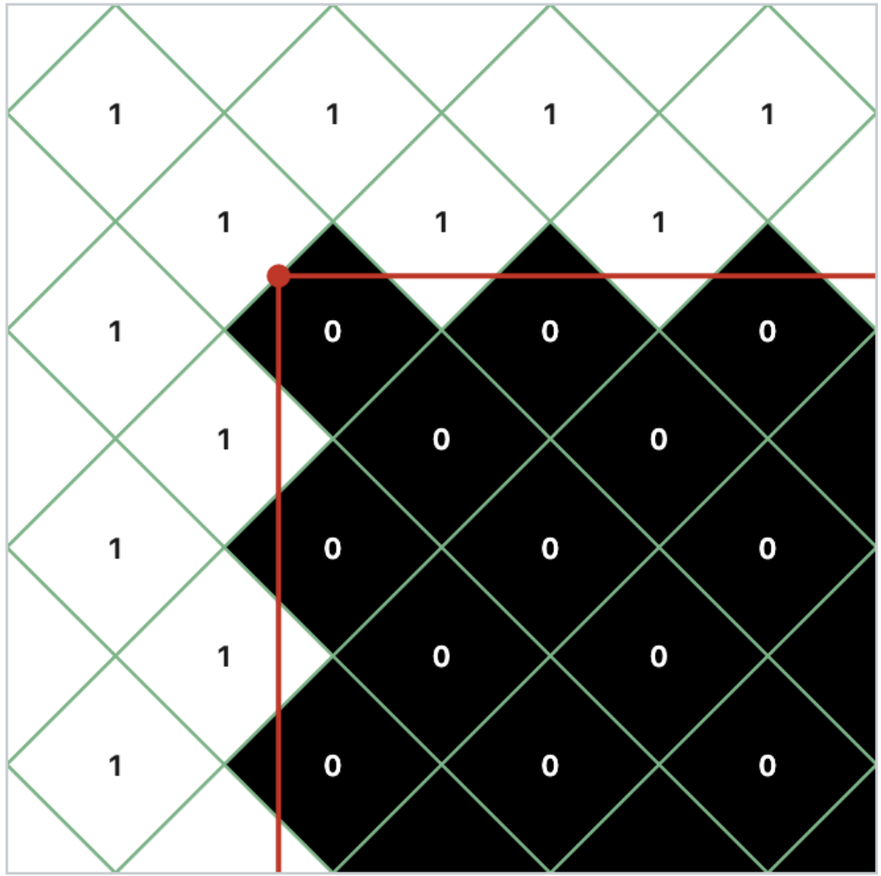
Demo: Green is a square grid turned 45 degrees

green on the Bayer grid



grey: the red and blue sites, no green measured
drag the corner

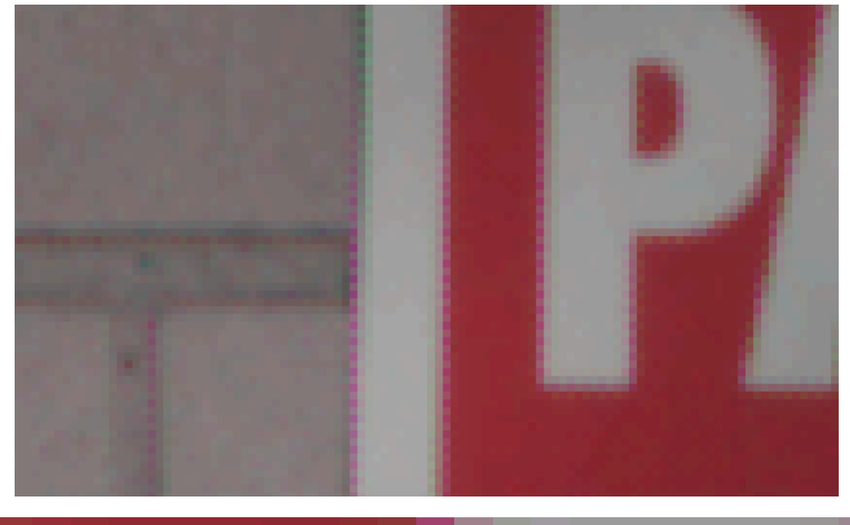
the same samples as a grid of their own

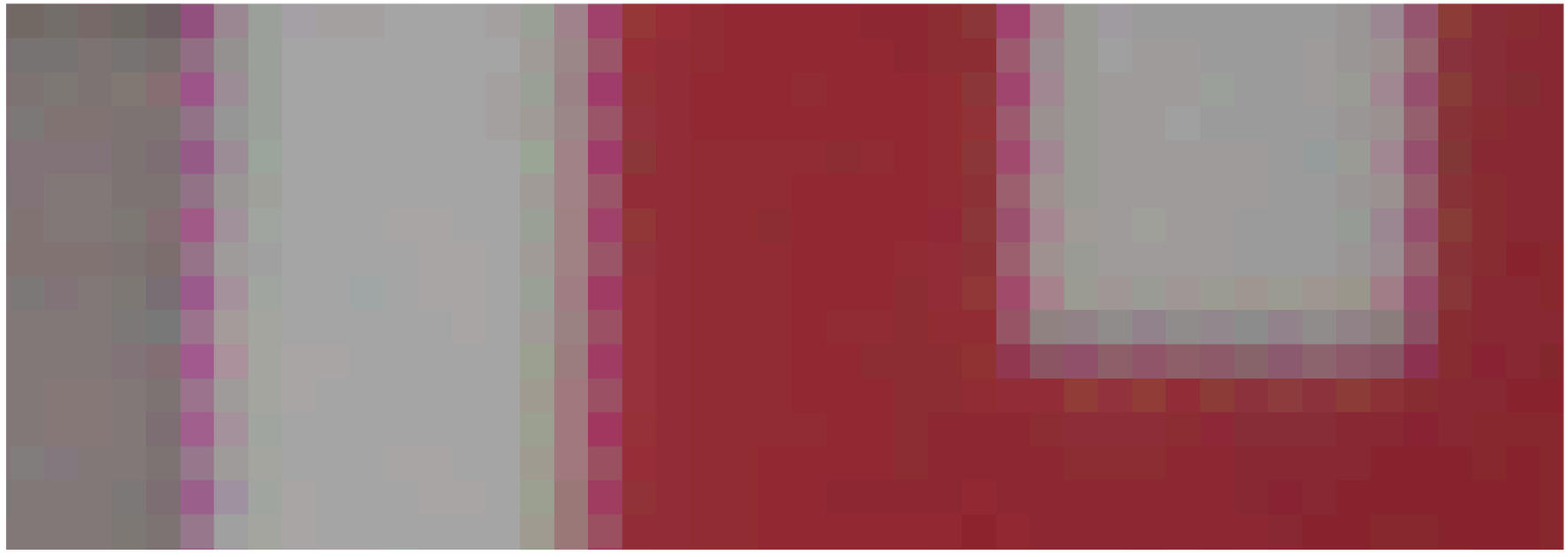


one diamond per green sample

Yep, that’s what we saw







Green channel



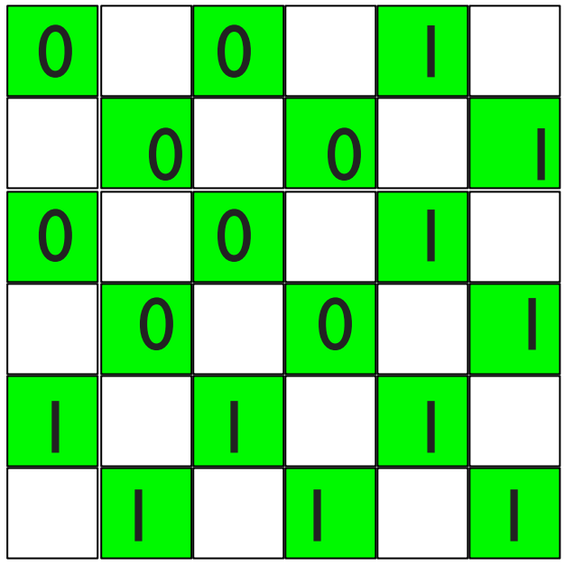
Questions?



Edge-based Demosaicking

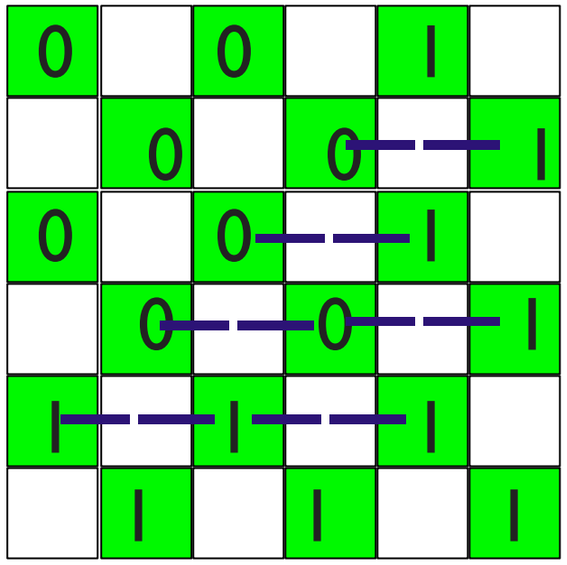
Idea

- **Take into account structure in image**
  - Here, 1D edges
- **Interpolate along preferred direction**
  - In our case, only use 2 neighbors



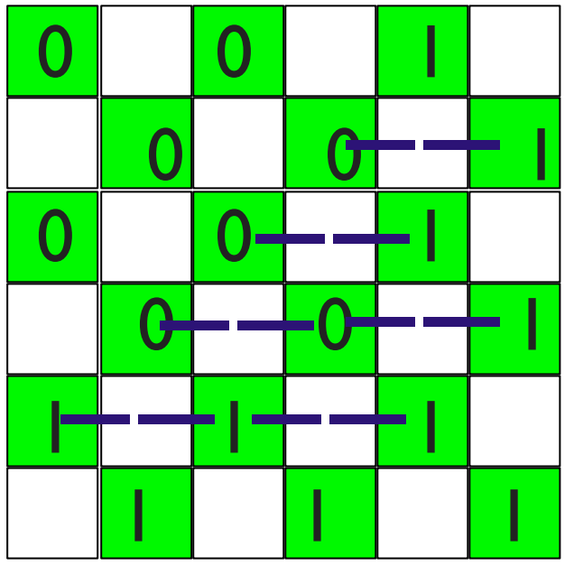
Idea

- **Take into account structure in image**
  - Here, 1D edges
- **Interpolate along preferred direction**
  - In our case, only use 2 neighbors



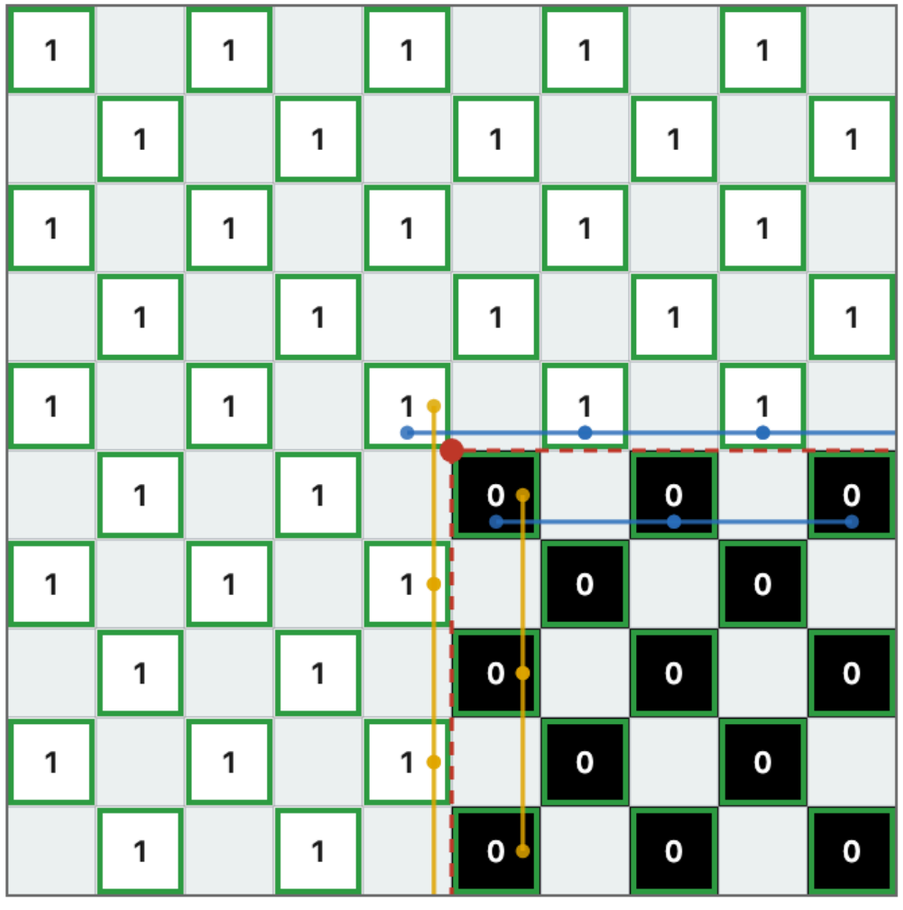
How do we decide

- **Look at the similarity of recorded neighbors**
  - Compare |up-down| and |right-left|
  - Be smart
  - See pset 3
- **Called edge-based demosaicking**

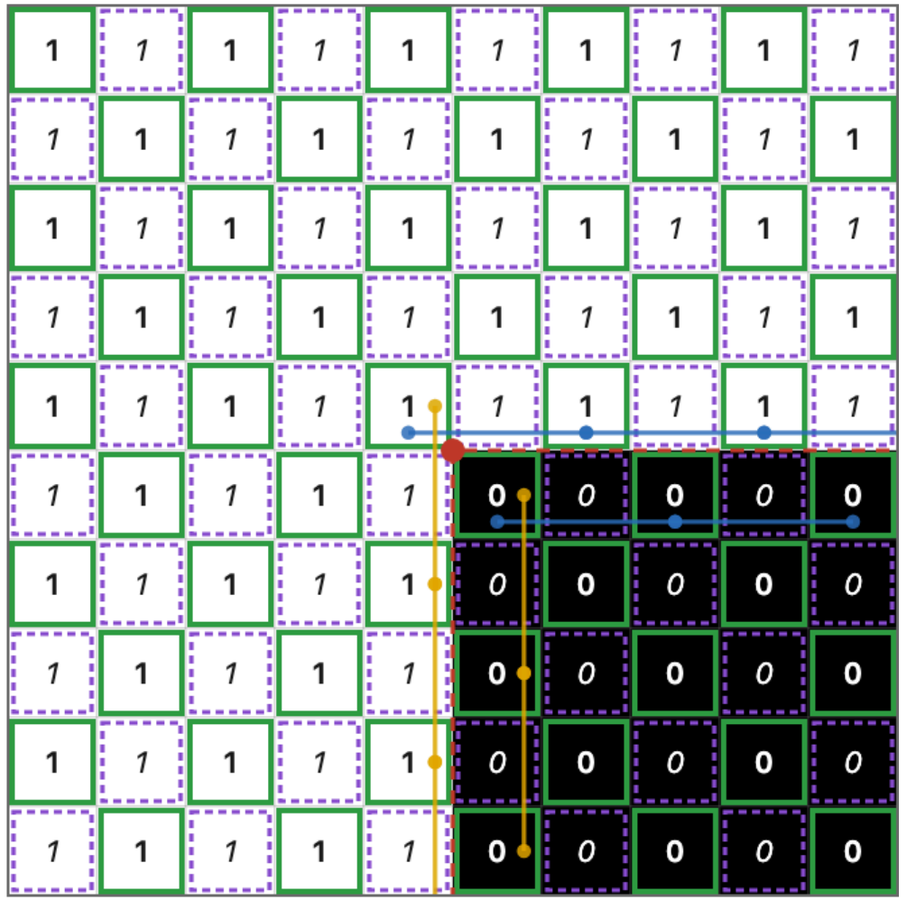


Demo: Edge-directed green

input: green measured on the checkerboard



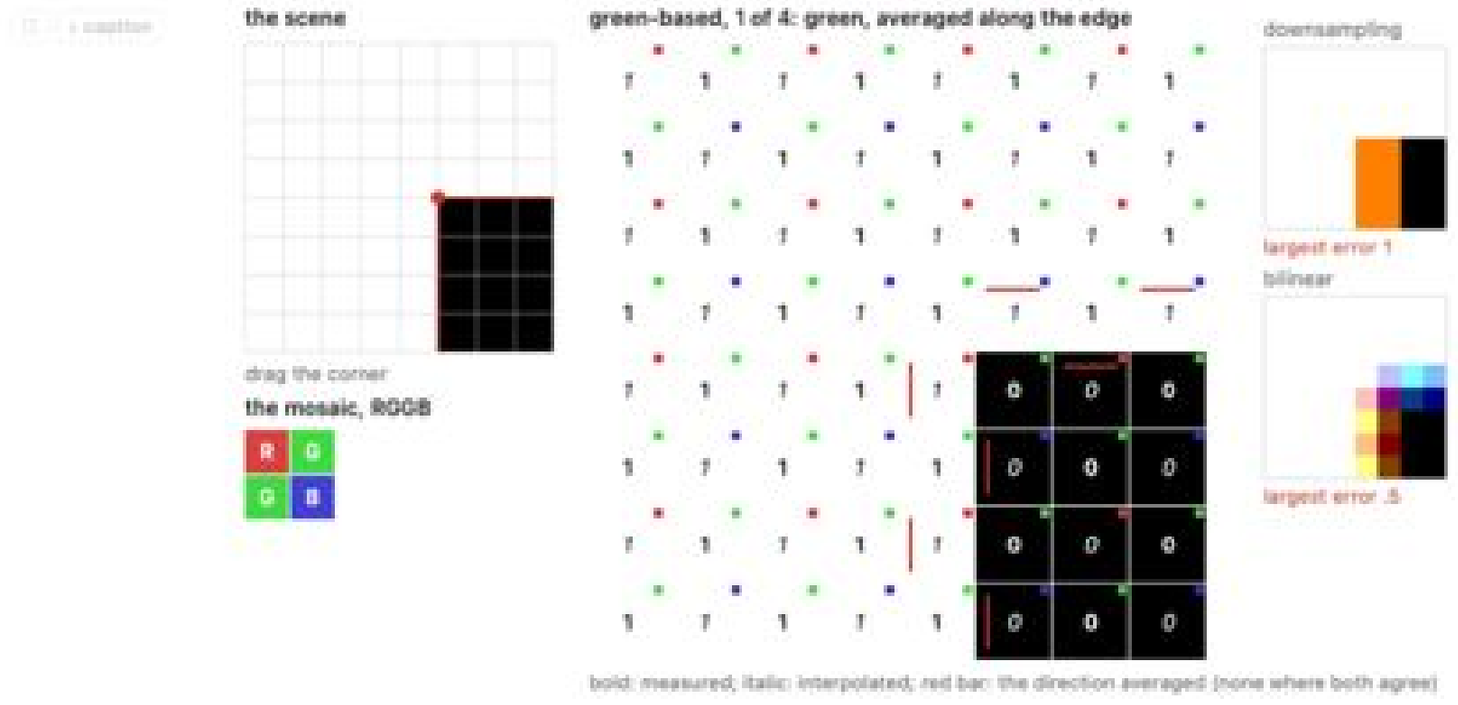
output: green everywhere



measured green · missing (a red or blue site) · interpolated · horizontal pair · vertical pair

no link: a tie, all four averaged · drag the corner

Demo: The corner, step by step: green-based: G along edges



scene: black on white | corner angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

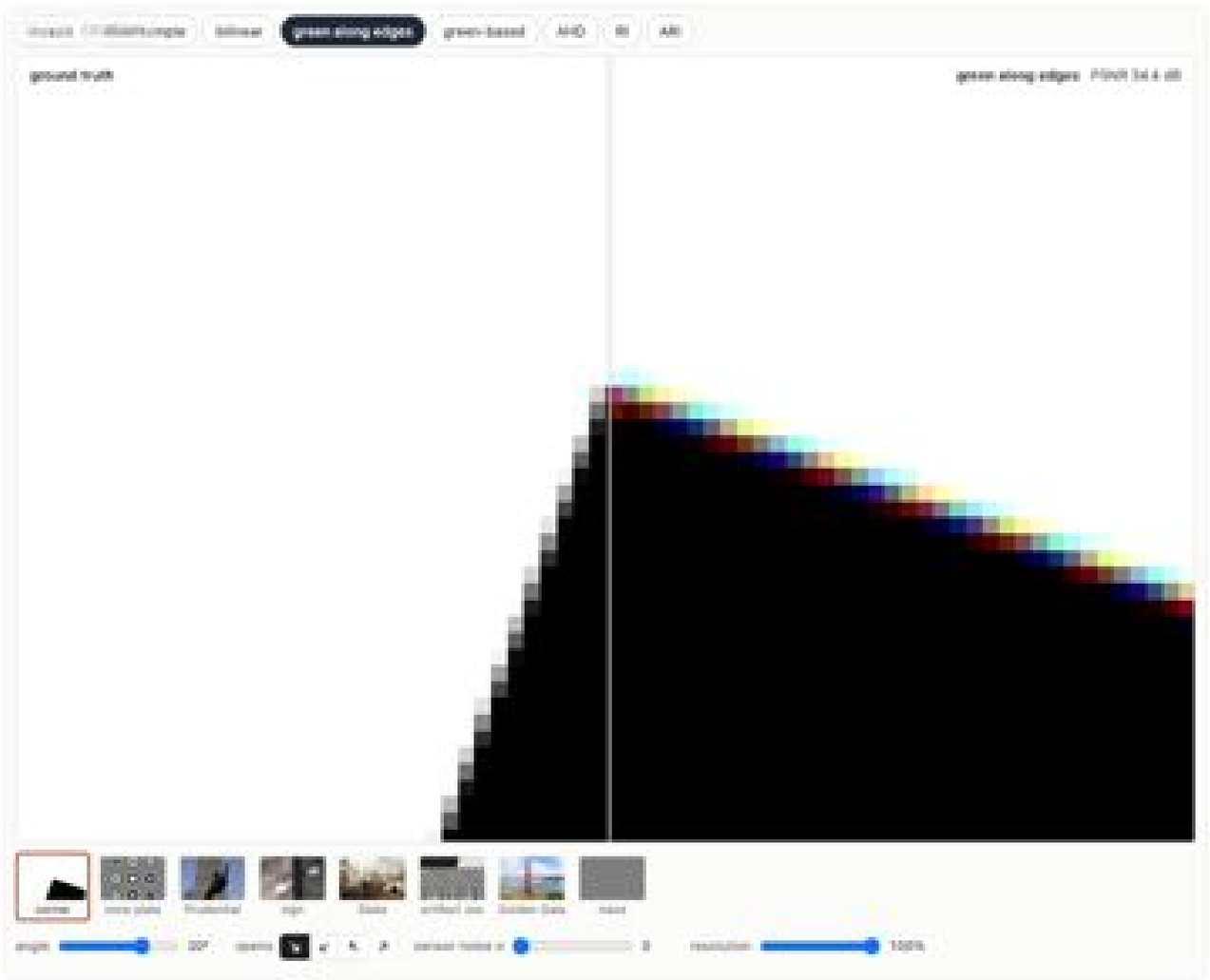
Green channel -- naive



Green channel -- edge-based



Demo: One method against the truth: green along edges



Demo courtesy of Frédo Durand, MIT.


100 PUZZLE

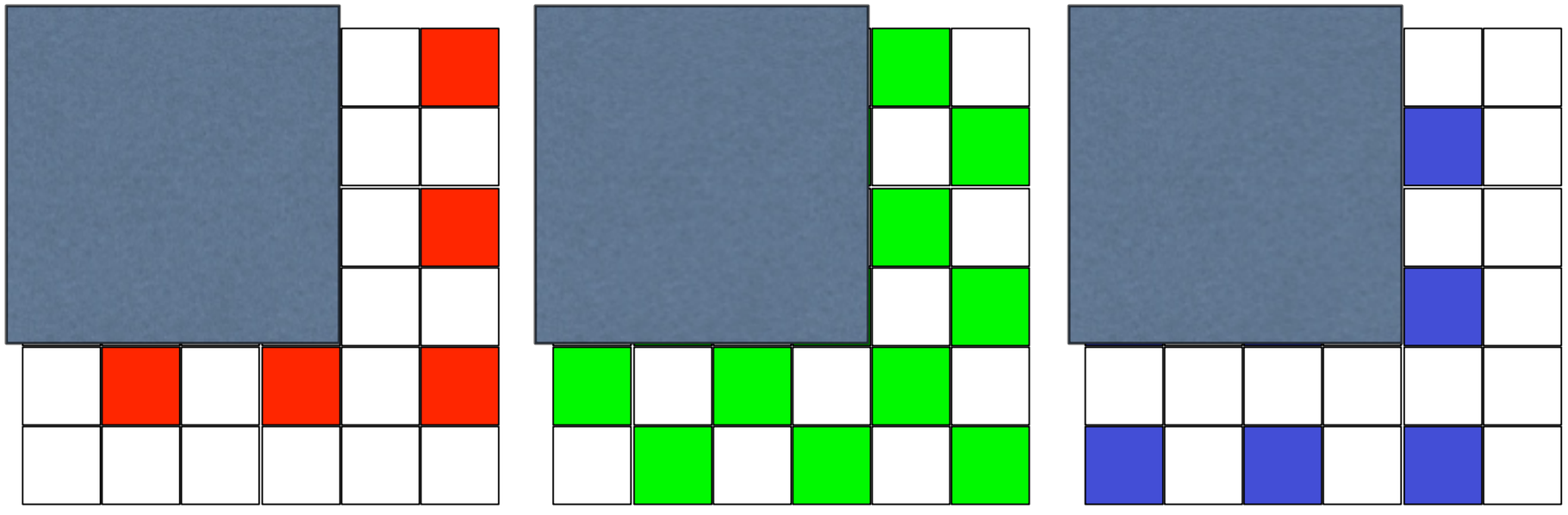
Challenge with other channels

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

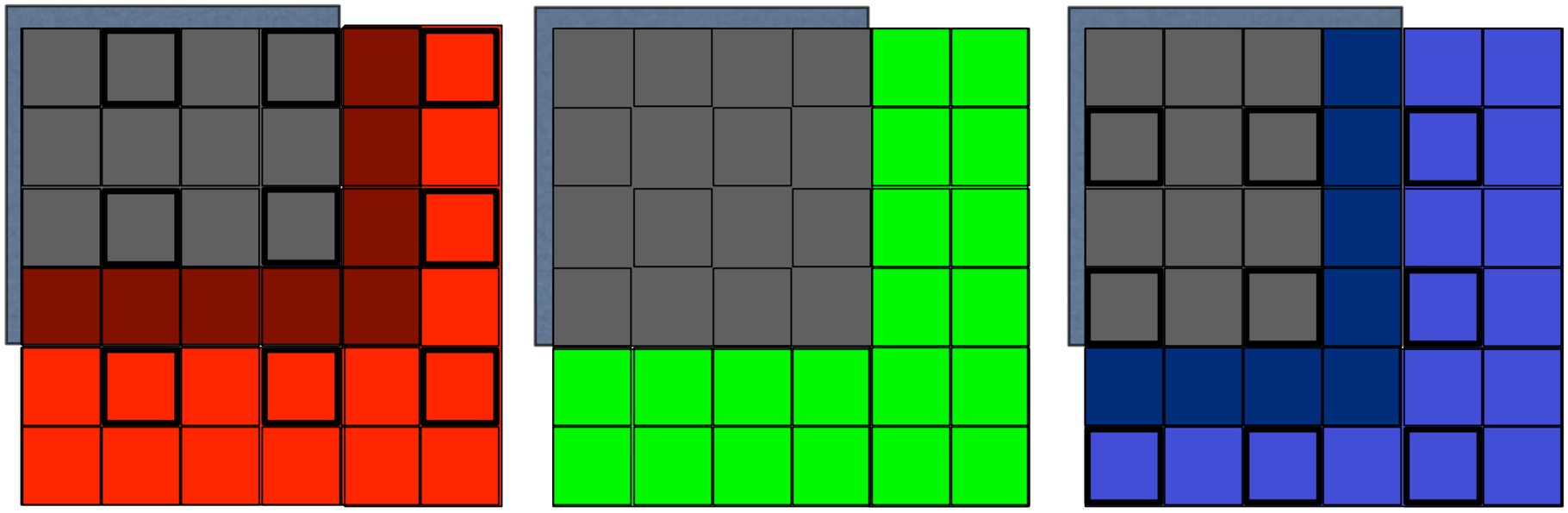
Example

- Black on white corner
- Even if we imagine we can do some decent job for each channel



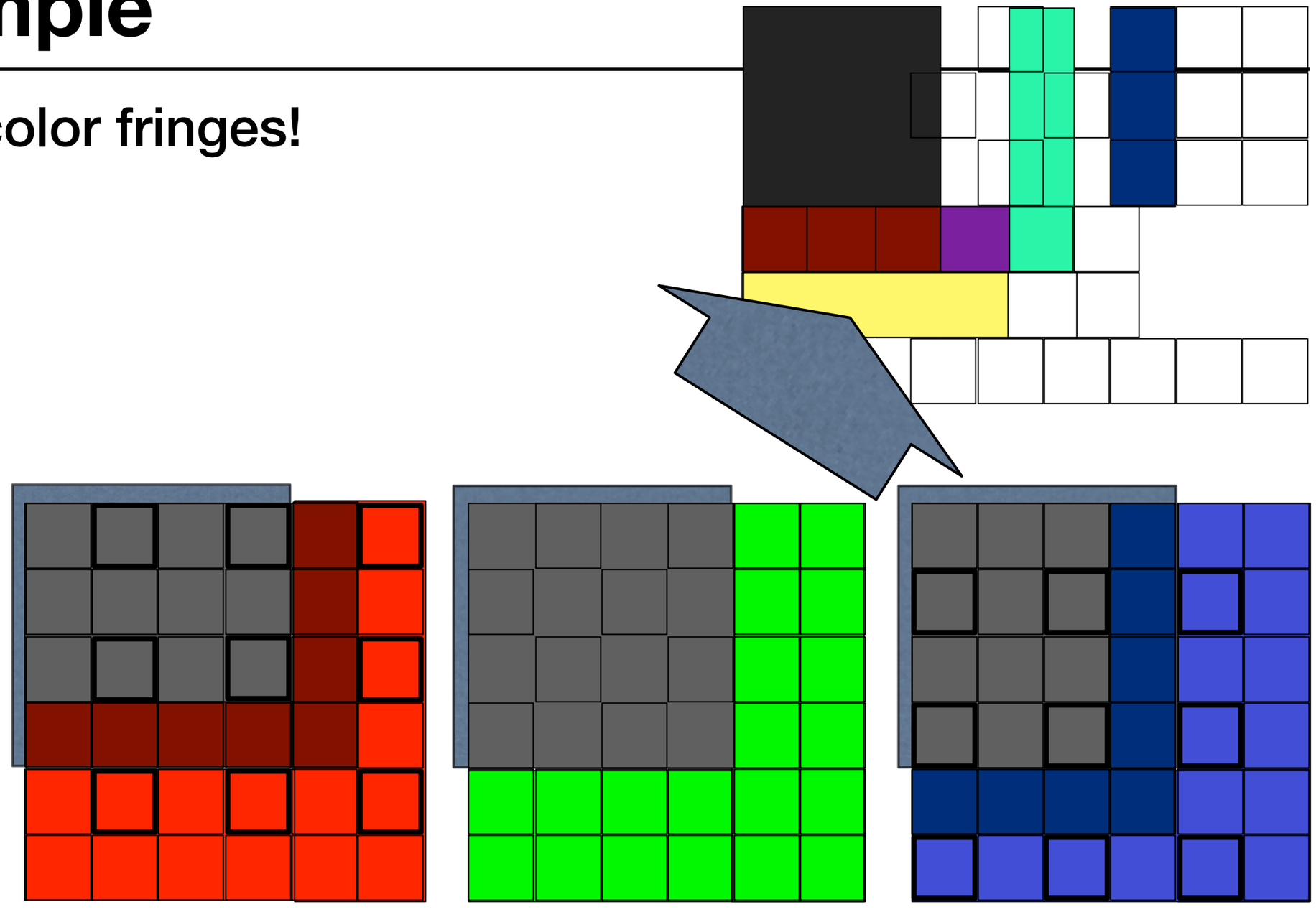
Example

- Black on white corner
- Even if we imagine we can do some decent job for each channel
- The channels don’t line up
  - Because they are not recorded at the same location

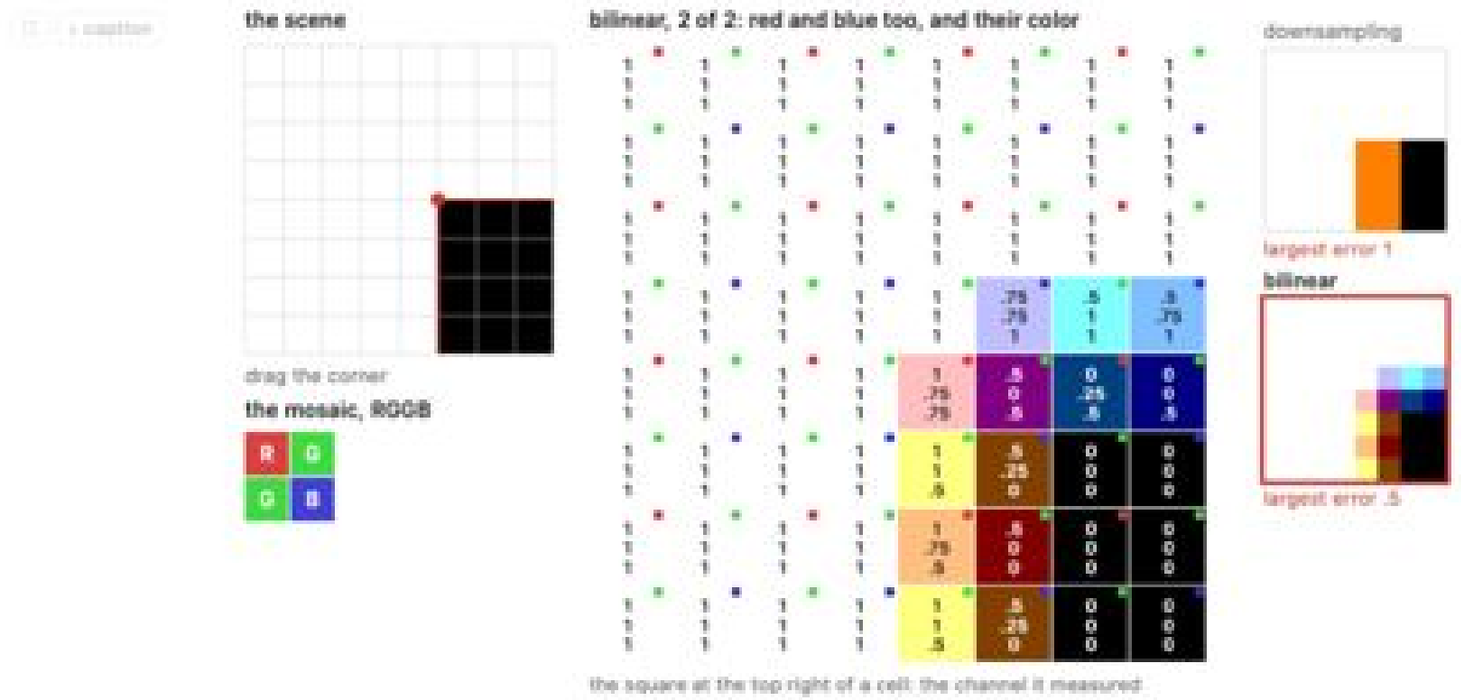


Example

- Bad color fringes!



Demo: The corner, step by step: bilinear: R, B

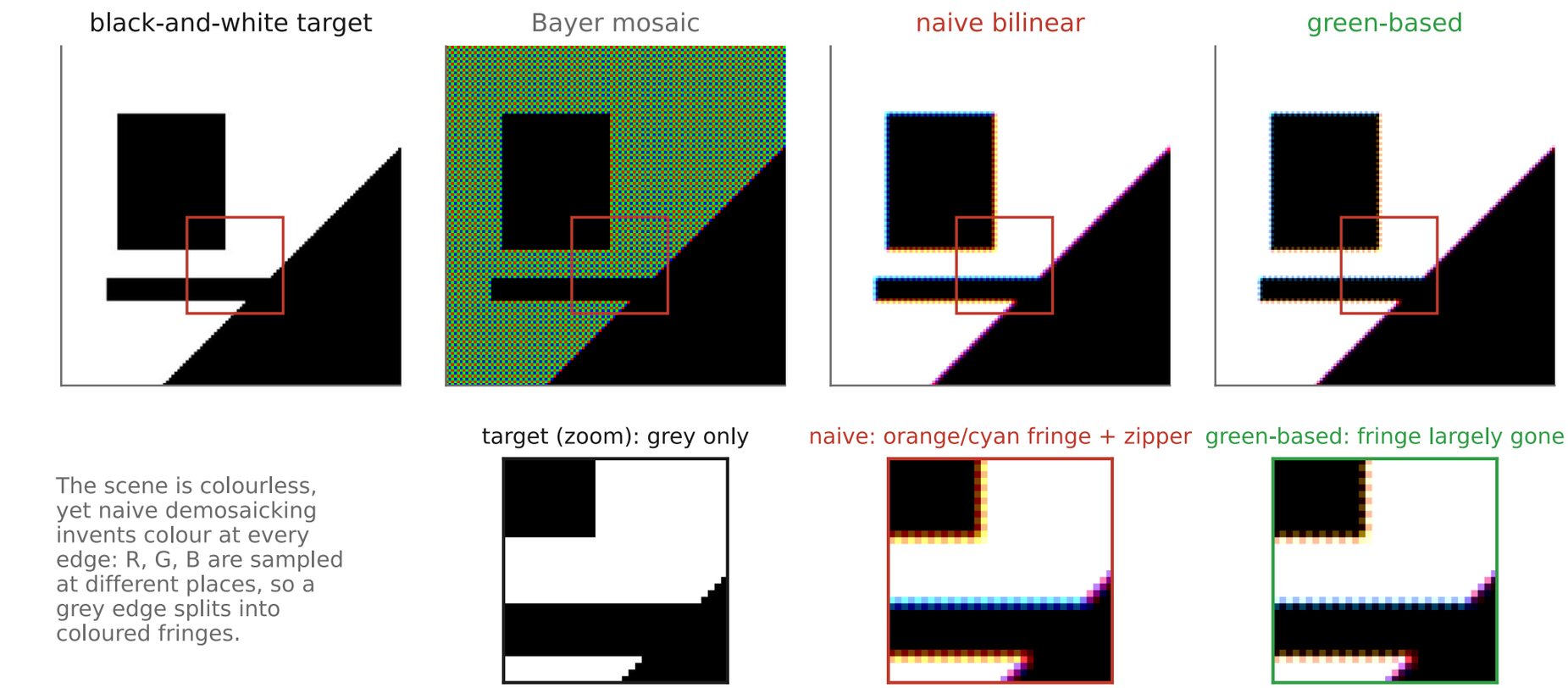


Demo courtesy of Frédo Durand, MIT.

Recall color artifacts



Figure 15.1.8: Color out of a black-and-white scene: the cleanest way to see the…



Color out of a black-and-white scene: the cleanest way to see the problem and the fix.

Questions?



Green-based Demosaicking

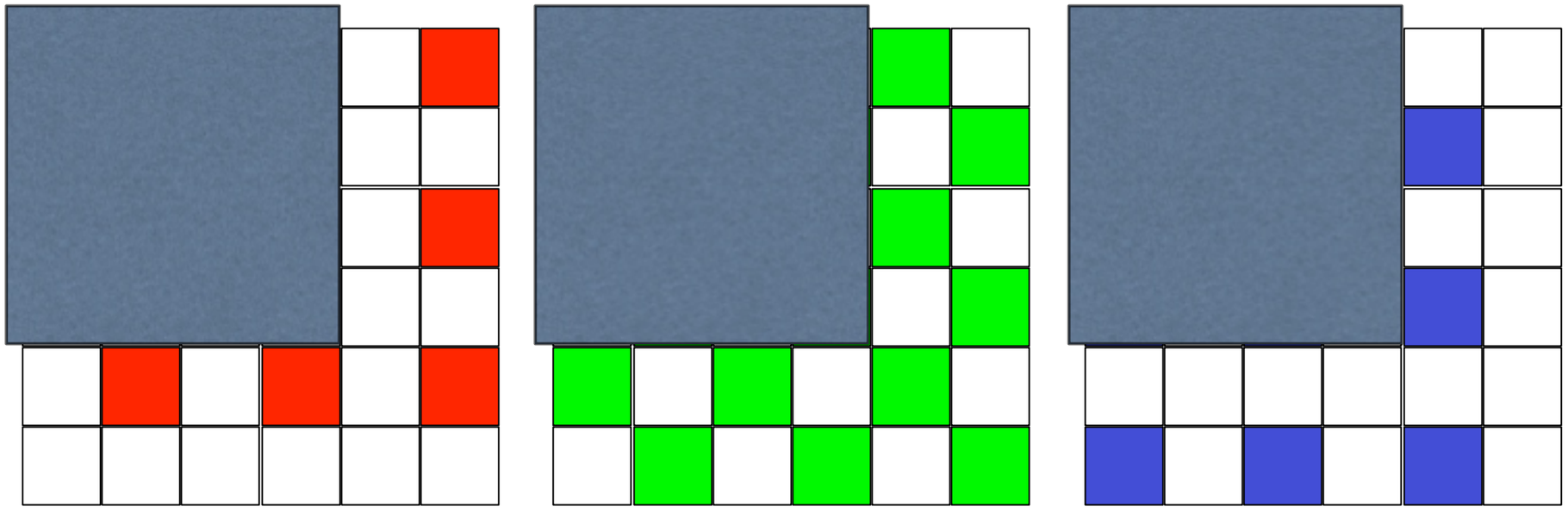
Green-based demosaicking

- 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

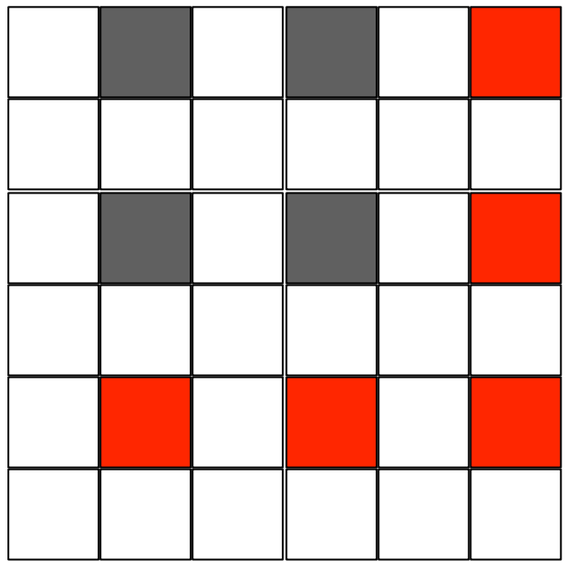
Interpolate difference to green

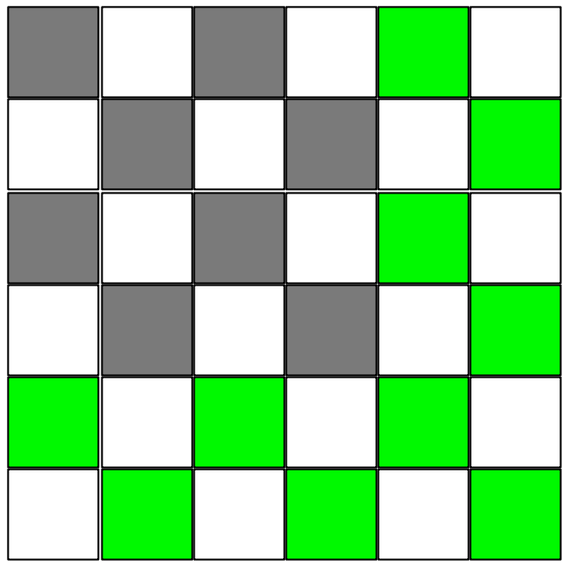
- **Interpolate green**
  - using e.g. edge-based
- **For recorded red pixels**
  - compute R-G
- **At empty pixels**
  - Interpolate R-G naively
  - Add G
- **Same for blue**

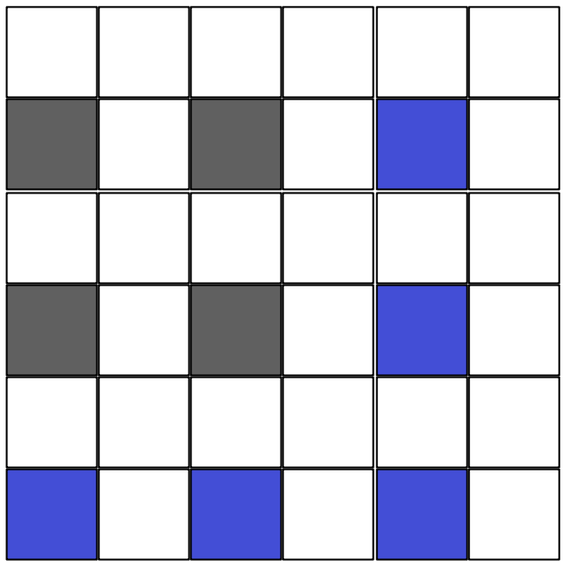
Black on white corner



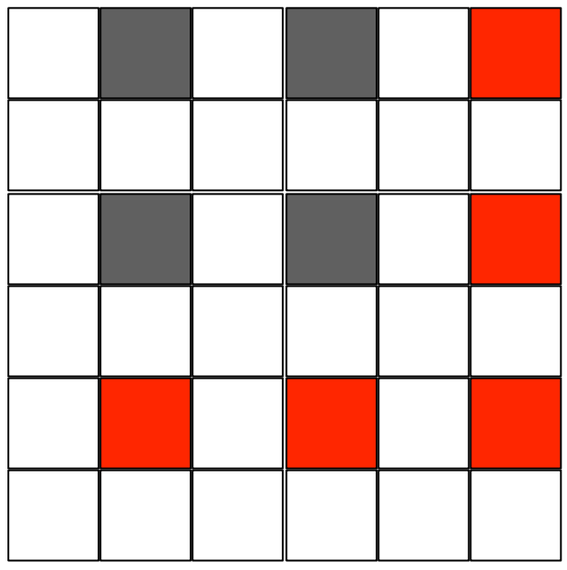
Measurements

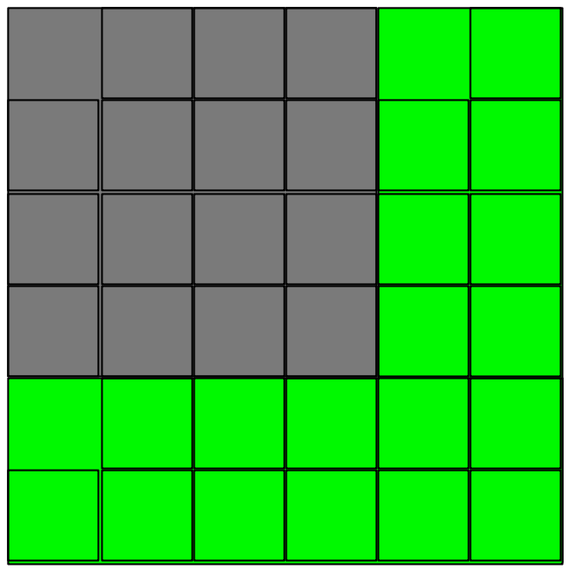


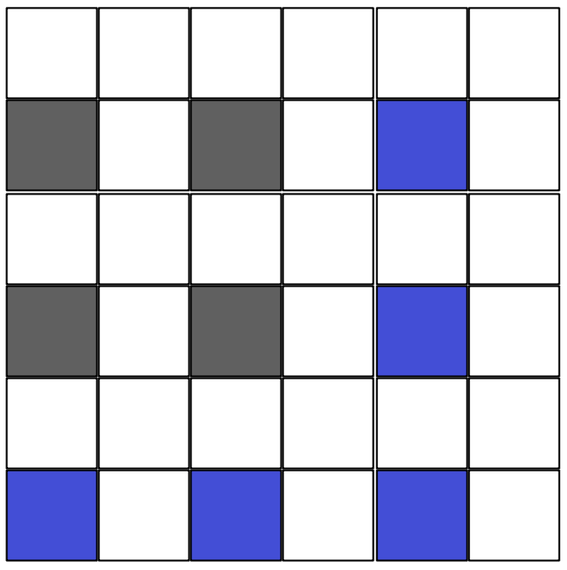




Edge-based green

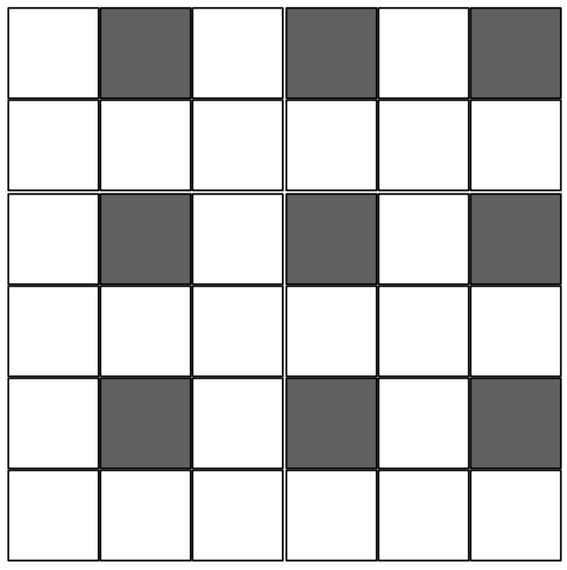


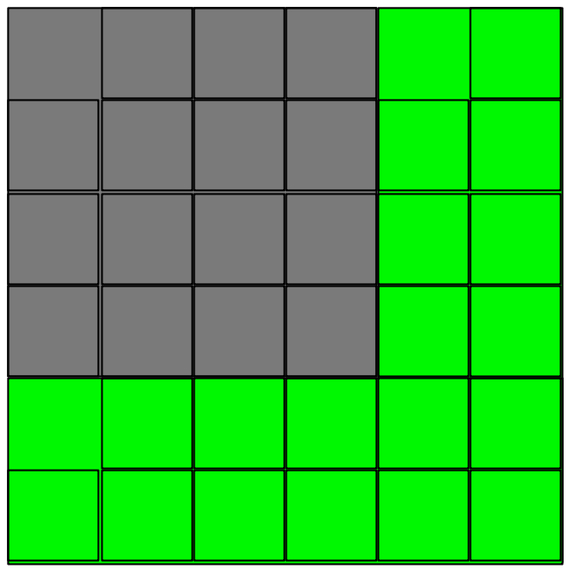


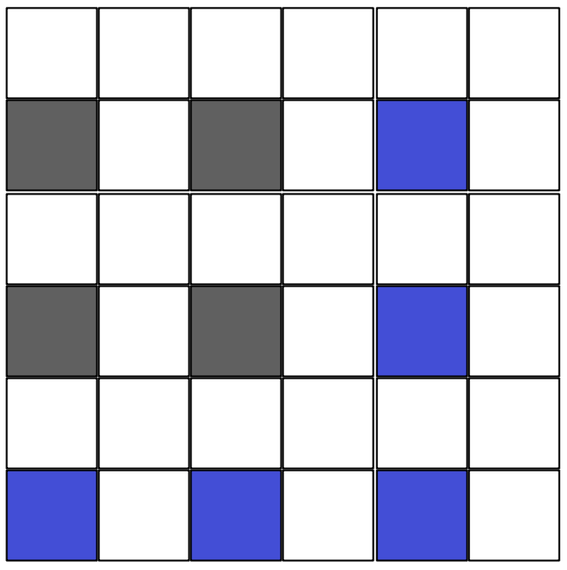


Red-Green difference

- Zero everywhere!



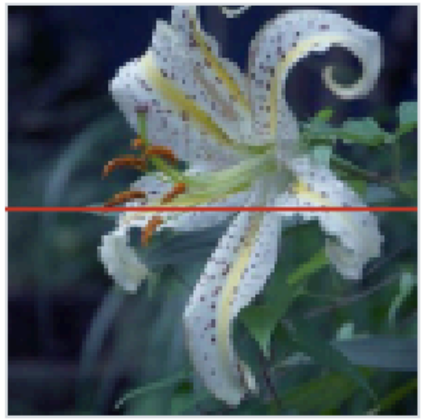


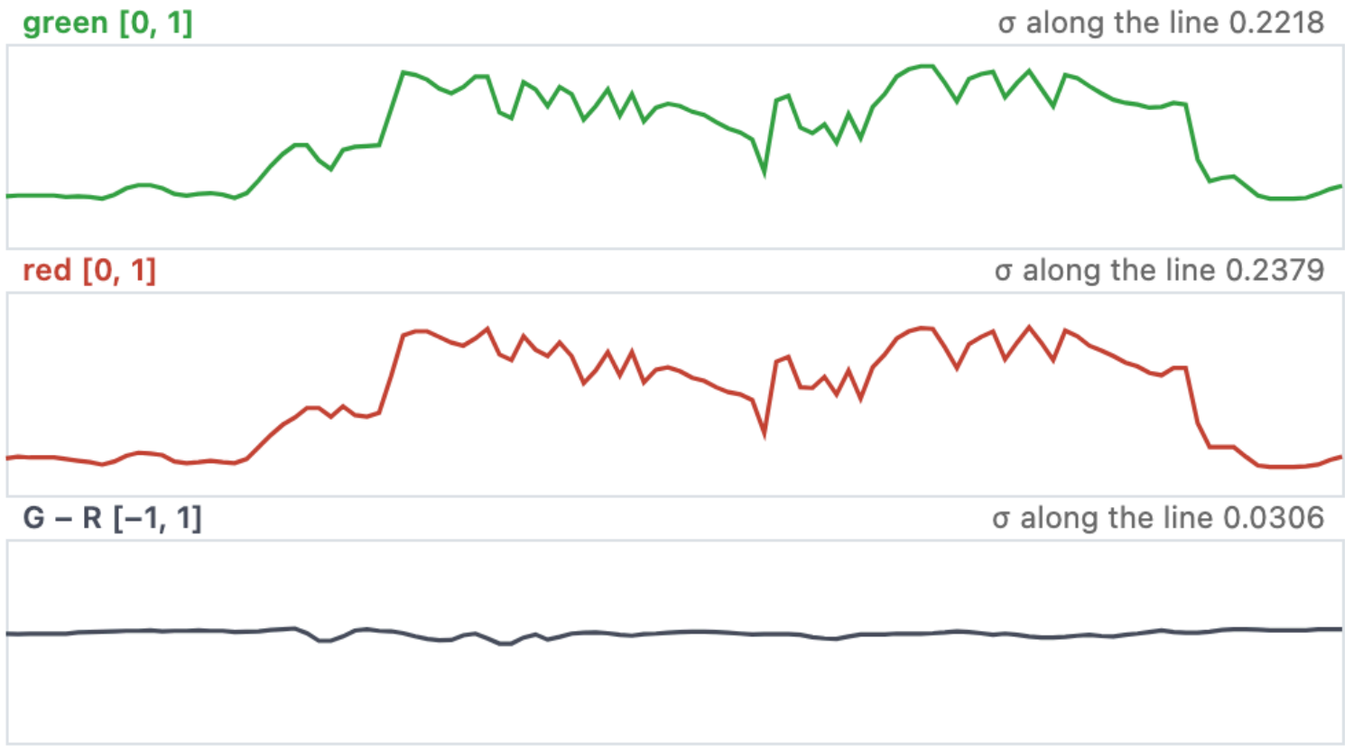


Demo: Colour differences are smooth

inspect how colour channels and their difference vary

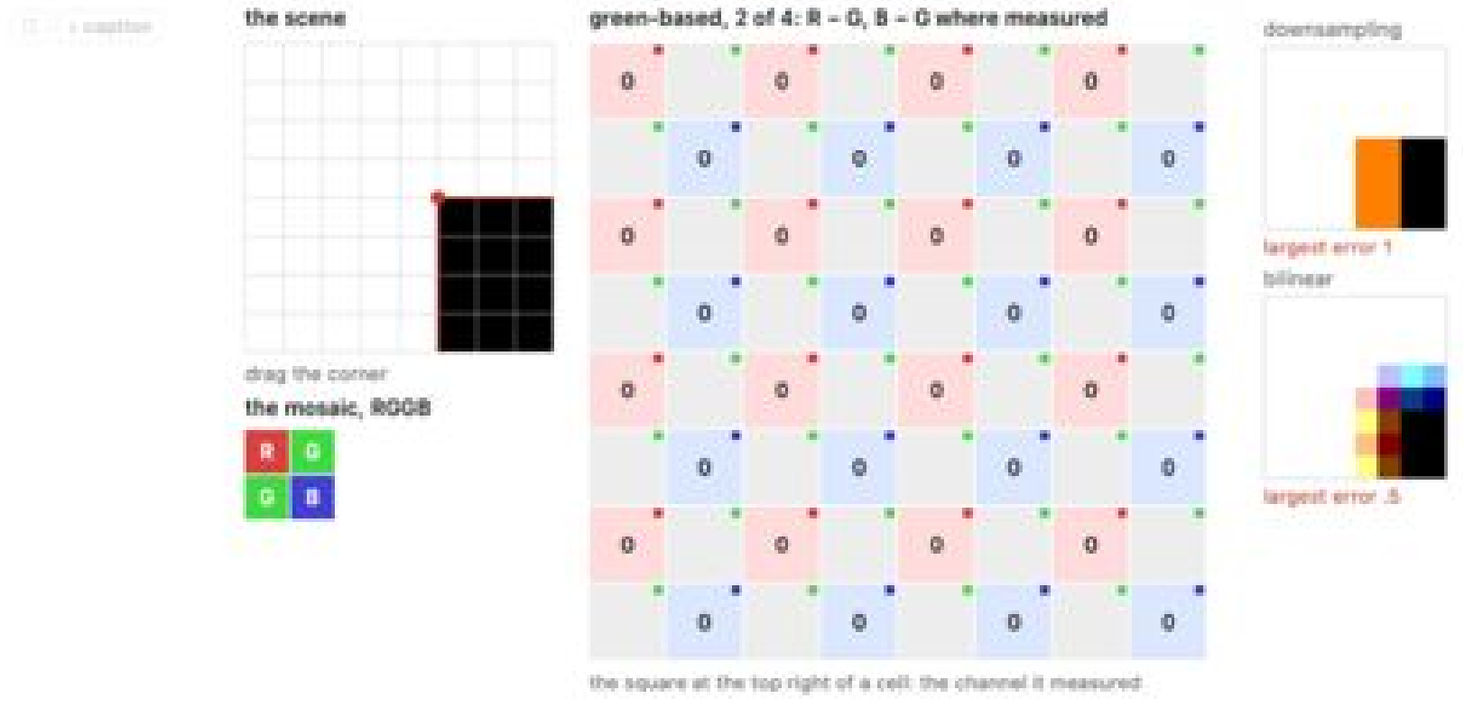
click to move the scanline





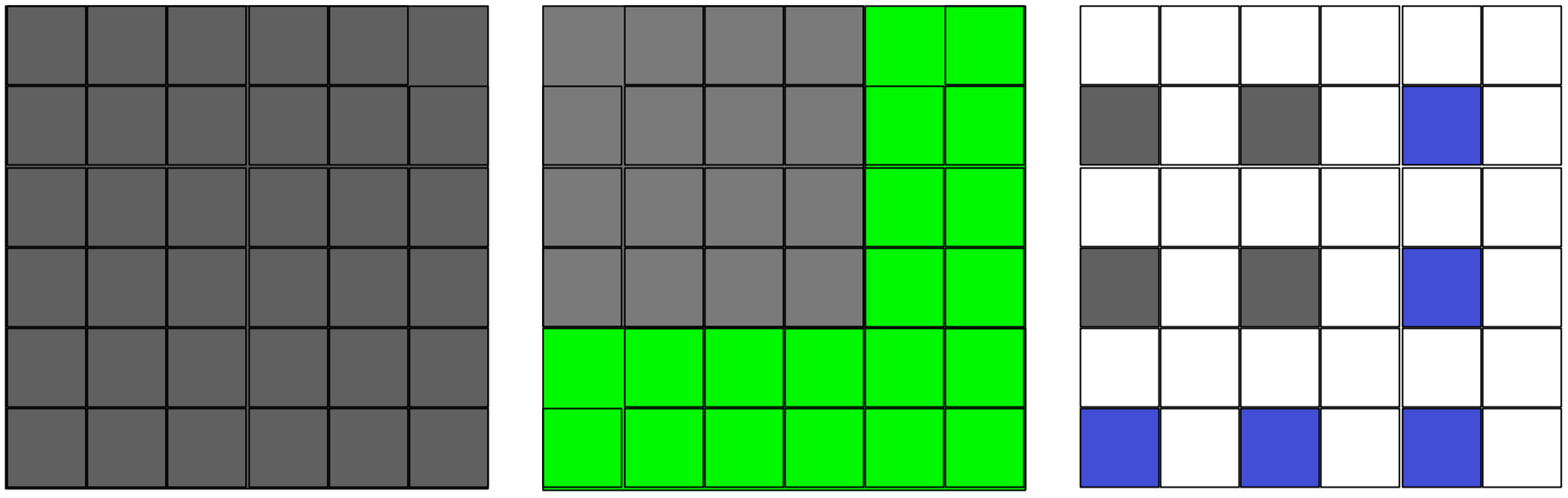
Line SD: green 0.222, difference 0.031. SD measures variation, not spatial smoothness.

Demo: The corner, step by step: green-based: R−G, B−G

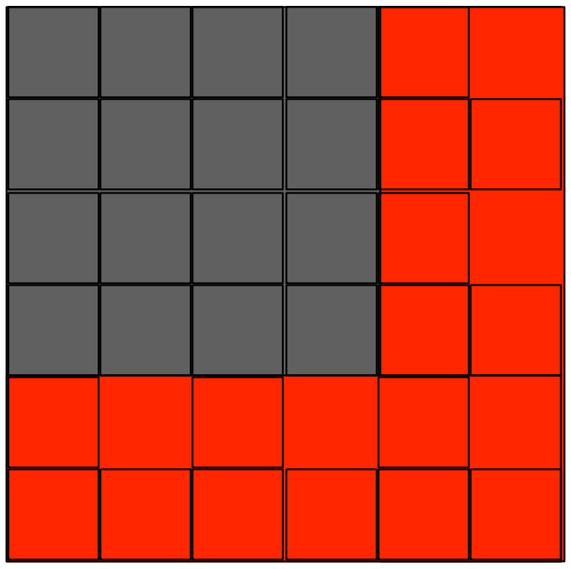


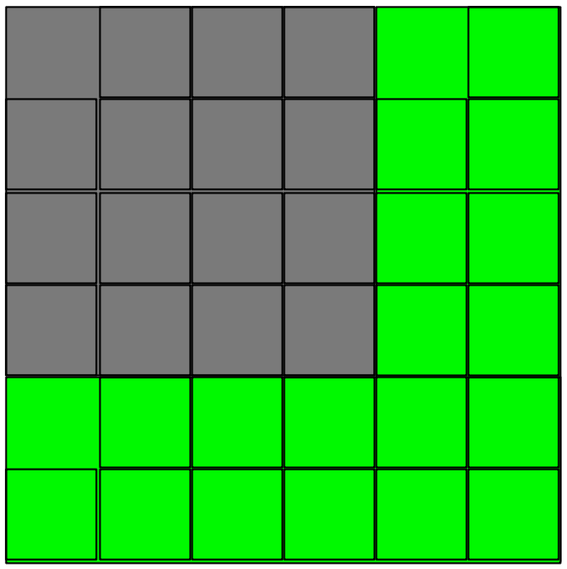
Red-Green interpolation

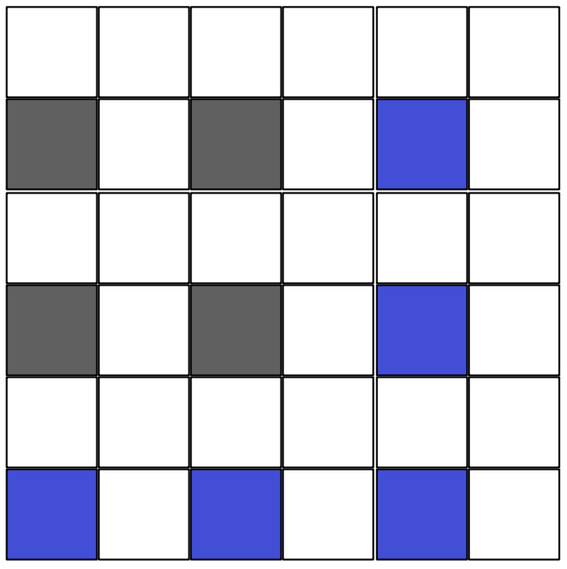
- Easy!



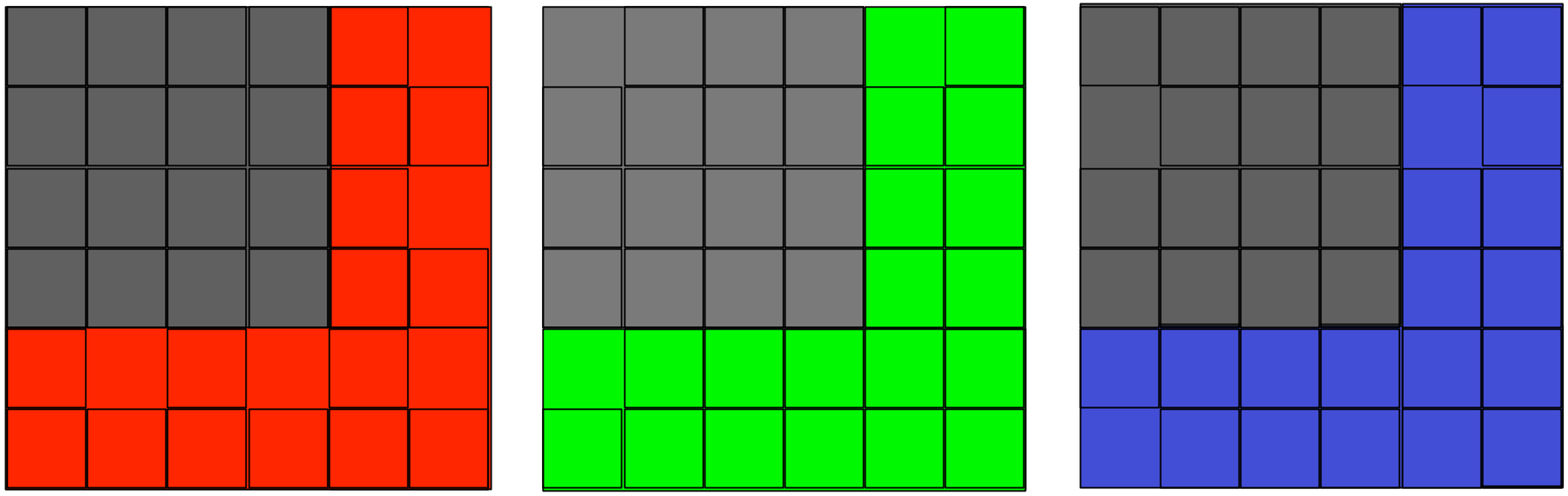
Add back green



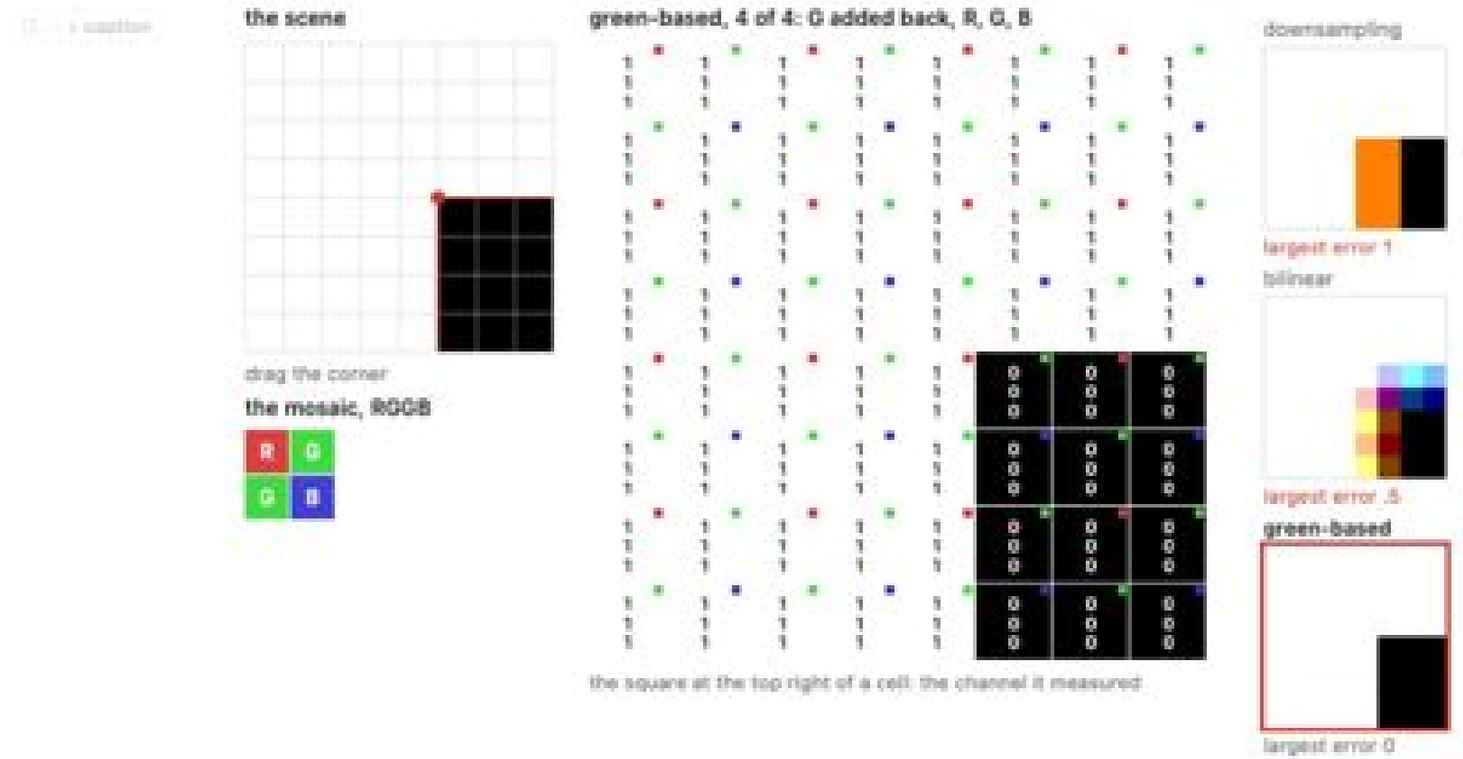




Same for blue



Demo: The corner, step by step: green-based: add G back



Fully naive



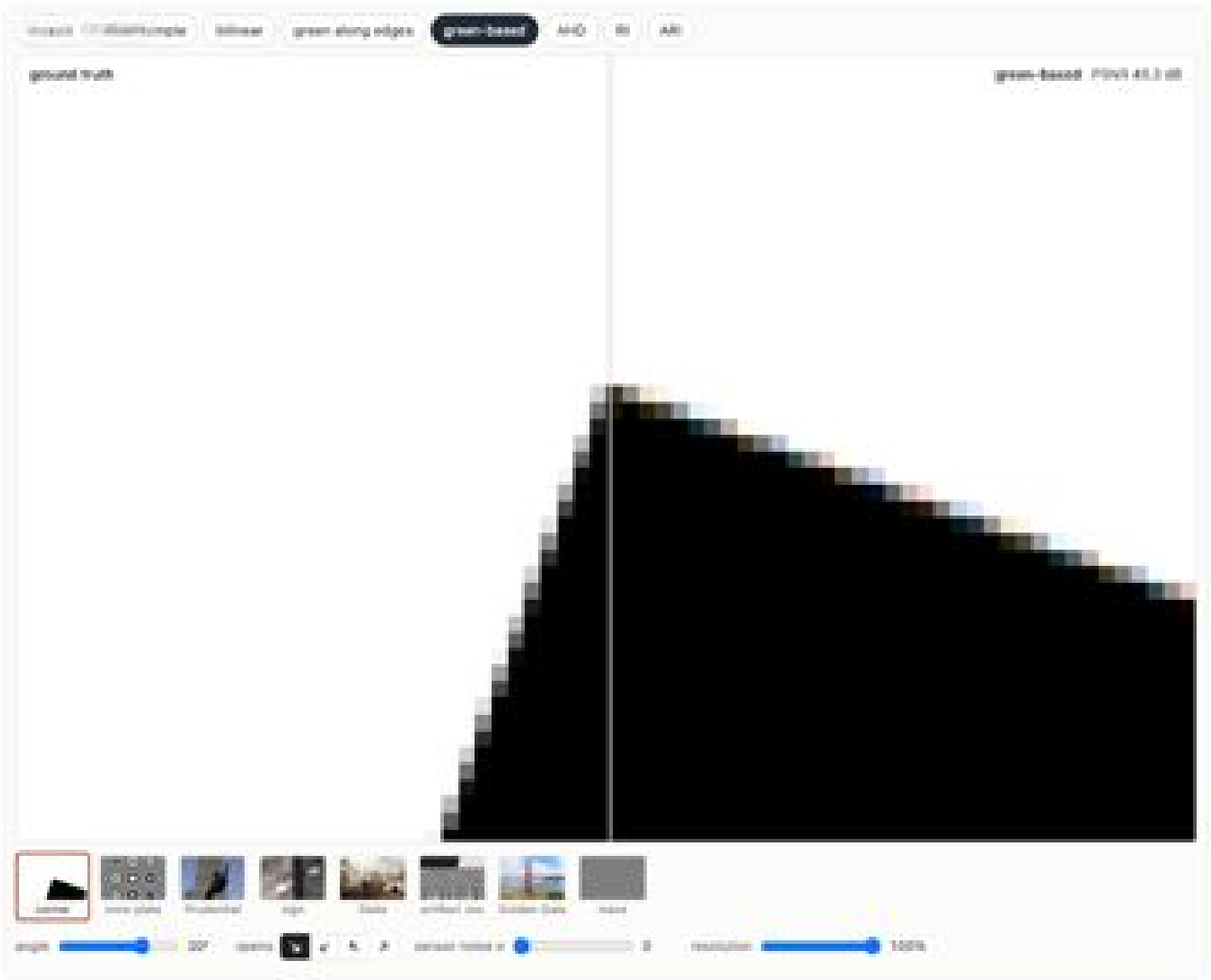
Edge-based green, naive red blue



Green-based blue and red



Demo: One method against the truth: green-based



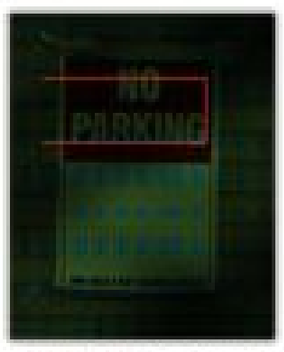
Demo courtesy of Frédo Durand, MIT.

Demo: The problem-set image, demosaicked: and green-based

◀ 1 bilinear 2 and green-based ▶

Click any picture to enlarge it; the arrow keys step through them.

The whole sign



raw Bayer mosaic, one colour per pixel



naive bilinear: false colour



green-based: clean

The lettering, enlarged (the red box)



naive: orange and cyan on the letter edges



green-based: the edges stay grey

Input: problem set 3's raw mosaic signs-small.png (BGGR), © Frédo Durand.

Demo courtesy of Frédo Durand, MIT.

Still not 100% perfect

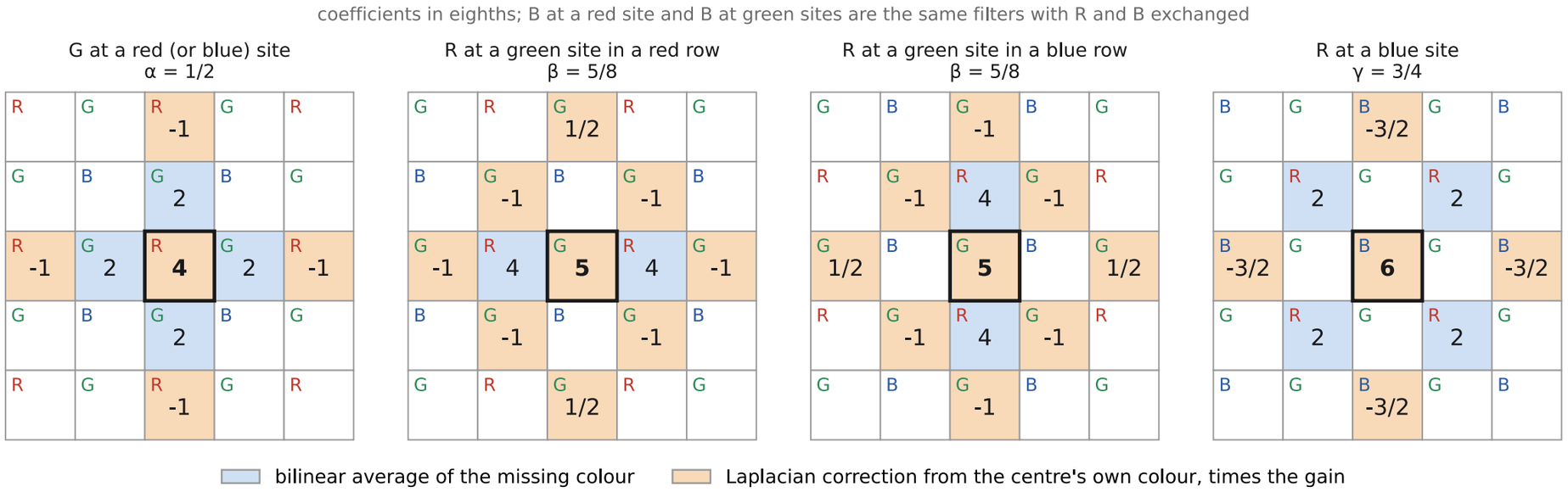
- But will be good enough for pset 3



Alternative

- Interpolate ratio

Figure 15.1.13: The four Malvar–He–Cutler filters, in eighths: green at a red or blue…



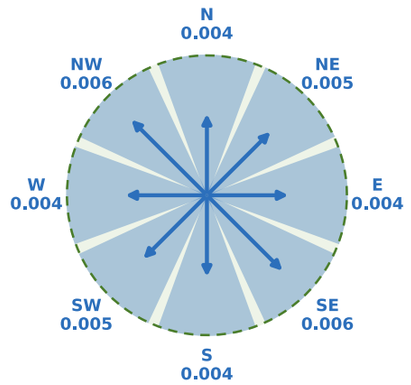
The four Malvar–He–Cutler filters, in eighths: green at a red or blue site, red at a green site in a red row and in a blue row, and red at a blue site (blue at a red…

Figure 15.1.14: Variable number of gradients at one red site of a real mosaic, the NO…

a flat region: the 5 x 5 mosaic around a red site

0.32	0.31	0.31	0.31	0.30
0.30	0.33	0.31	0.34	0.31
0.31	0.31	0.31	0.31	0.31
0.31	0.33	0.31	0.35	0.31
0.32	0.32	0.31	0.31	0.31

gradients in the eight directions (arrow length)

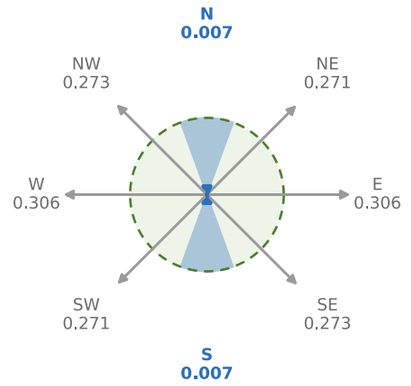


threshold T = 1.5 min + 0.5 (max − min) = 0.007
8 of 8 directions pass

on an edge: the 5 x 5 mosaic around a red site

0.36	0.08	0.44	0.55	0.60
0.09	0.07	0.24	0.57	0.59
0.36	0.07	0.43	0.55	0.60
0.09	0.07	0.24	0.60	0.59
0.36	0.08	0.43	0.55	0.60

gradients in the eight directions (arrow length)



threshold T = 1.5 min + 0.5 (max − min) = 0.160
2 of 8 directions pass

Samples: problem set 3's raw mosaic signs-small.png (BGGR). Dashed circle: the threshold; shaded sectors: the directions below it, which VNG averages over.

Variable number of gradients at one red site of a real mosaic, the NO PARKING sign of problem set 3.

AHD

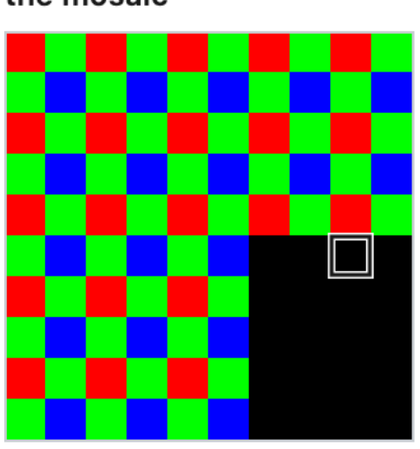
AHD: interpolate first, decide afterwards

- Every method so far picks a direction from gradients in the mosaic, then interpolates
- AHD (Hirakawa & Parks 2005) interpolates the whole image twice
  - once only along rows, once only along columns
- then keeps, at each pixel, the more plausible of the two
- Plausible means locally homogeneous
  - interpolating across an edge invents colors found on neither side of it
- dcraw's default for years; still a strong classical baseline

AHD: the homogeneity criterion

- At each pixel, in each candidate, look at its four neighbors in CIELAB
- A neighbor counts if it is close in lightness (within εL) and in color (within εC)
- Across an edge, the wrong direction invents colors, so fewer neighbors count

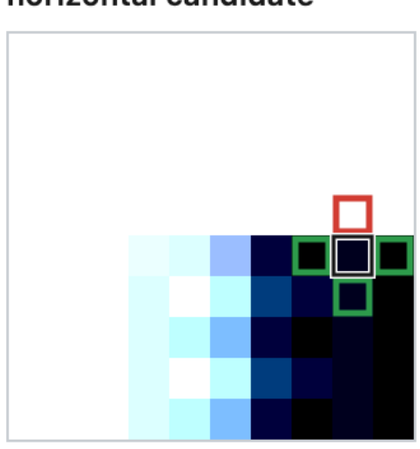
the mosaic



at the ringed pixel
$\varepsilon_L = 0.9$ $\varepsilon_C = 18$
close: $\Delta L \le \varepsilon_L$ and $\Delta C \le \varepsilon_C$
in CIELAB, both thresholds set from the pixel's own neighbours

drag the mosaic to move the corner

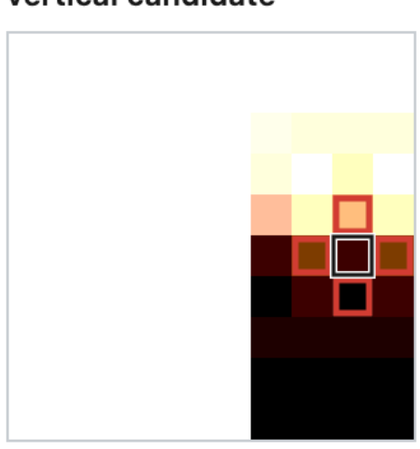
horizontal candidate



neighbour	ΔL	ΔC
left ✓	0.9	18
right ✓	0.9	18
up ✗	99	18
down ✓	0.0	0.0

close: 3 of 4 **3 × 3 sum: 26**

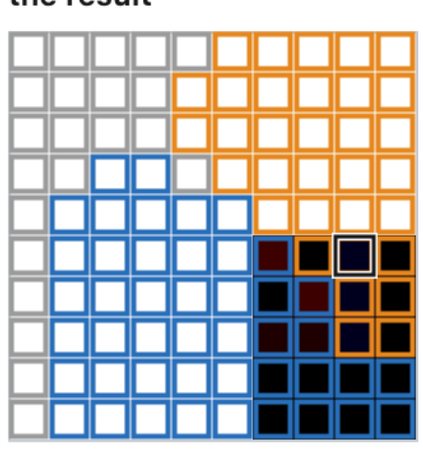
vertical candidate



neighbour	ΔL	ΔC
left ✗	25	30
right ✗	25	30
up ✗	72	29
down ✗	9.7	33

close: 0 of 4 **3 × 3 sum: 1**

the result



ringed: 26 vs 1, horizontal
- horizontal chosen
- vertical chosen
- a tie, the two averaged

click a pixel to select it

AHD in five steps

1. **Two candidates, horizontal and vertical**
   - green along that direction (Hamilton-Adams style), then R-G and B-G along it
2. **Convert both to CIELAB: lightness and color judged apart**
3. **Homogeneity: count neighbors within εL in L and εC in (a, b)**
   - thresholds set per pixel from its nearest neighbors along each direction
4. **Sum the counts over 3x3; keep the larger (tie: average)**
5. **Clean-up: a few median passes on R-G and B-G (Freeman)**

AHD: pseudocode

```
def ahd(mosaic):
  for d in (H, V):                    # 1. two reconstructions
    G = interp_green(mosaic, along=d)   # Hamilton-Adams
    R = G + interp(mosaic.R - G, along=d)
    B = G + interp(mosaic.B - G, along=d)
    rgb[d] = (R, G, B)
    lab[d] = to_lab(rgb[d])           # 2. perceptual space

  for each pixel p:                   # 3a. adaptive thresholds
    epsL[p] = min(max |L_H(p)-L_H(q)| for q left/right of p,
                  max |L_V(p)-L_V(q)| for q above/below p)
    epsC[p] = same, with the color distance |ab(p)-ab(q)|

  for d in (H, V):                    # 3b. homogeneity
    hom[d][p] = count of q in N4(p) with |L_d(p)-L_d(q)| <= epsL[p]
                                    and |ab_d(p)-ab_d(q)| <= epsC[p]
    score[d] = box_sum_3x3(hom[d])  # 4. choose
  out = where(score[H] > score[V], rgb[H],
        where(score[V] > score[H], rgb[V], (rgb[H] + rgb[V]) / 2))

  repeat 3 times:                     # 5. Freeman's clean-up
    out.R = out.G + median3x3(out.R - out.G)
    out.B = out.G + median3x3(out.B - out.G)
```

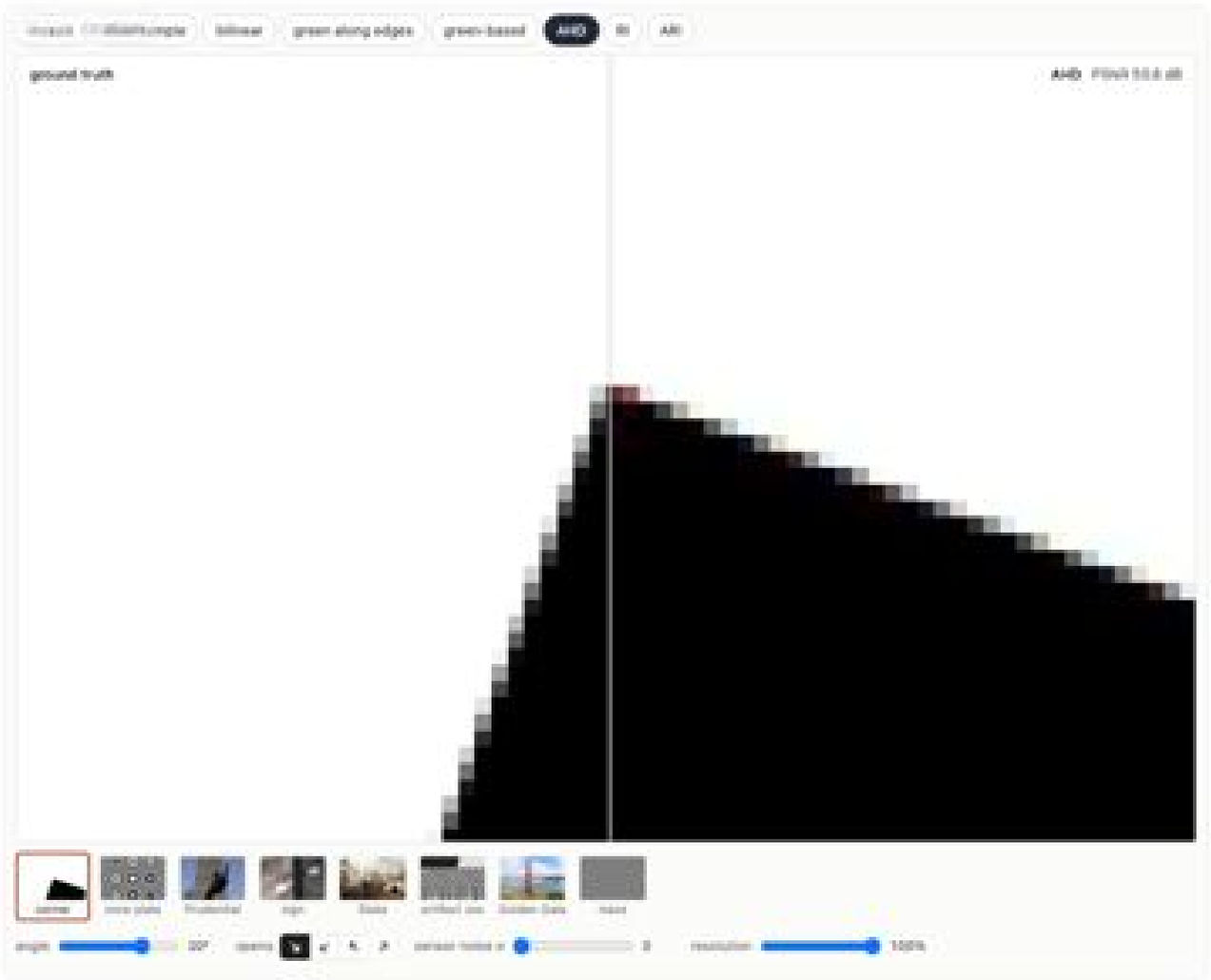
AHD: cost and limits

- **Cost: two reconstructions, two Lab conversions and the counting**
  - several times Hamilton-Adams
- **Still only two directions: diagonal texture fools both candidates**
- **Fine texture near the sampling limit: both are inhomogeneous, the choice turns erratic, and mazes appear**
- **Noise lowers homogeneity everywhere: denoise first, or jointly**
- Book: **Adaptive homogeneity-directed demosaicking (AHD)**

Demo: Demosaicking lab

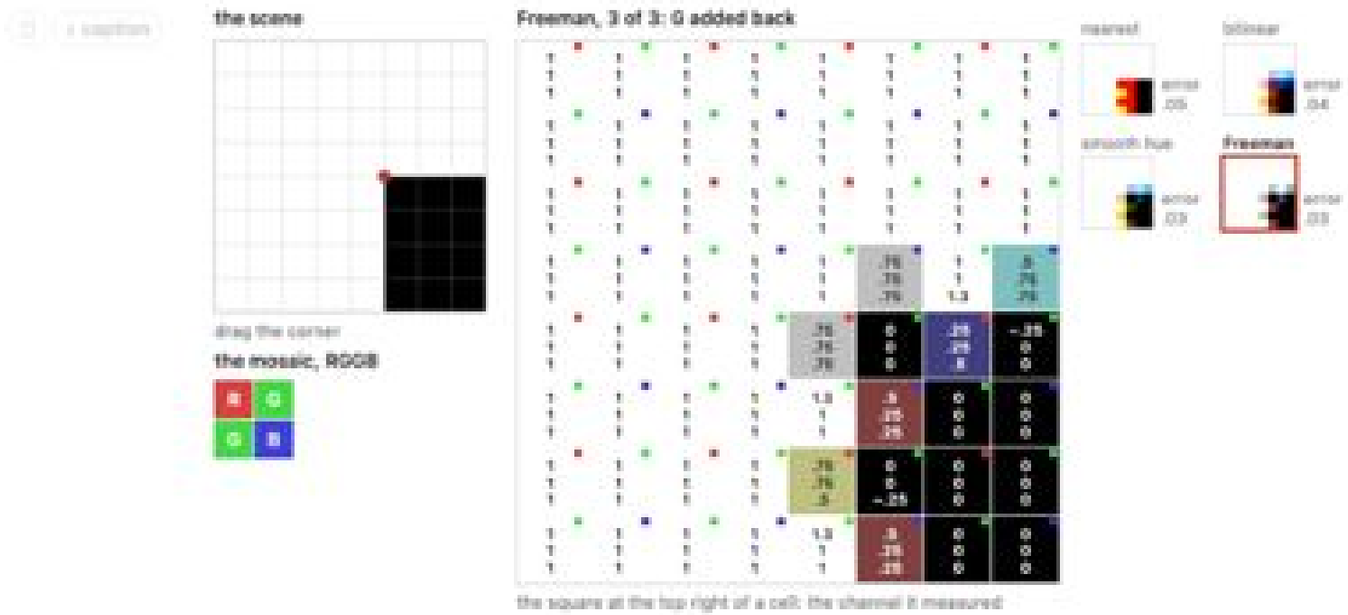


Demo: One method against the truth: AHD



Demo courtesy of Frédo Durand, MIT.

Demo: The classical methods, step by step: Freeman: add G back



1 input: mosaic | 2 nearest | 3 bilinear | 4 smooth hue: R/G, B/G | 5 smooth hue: interpolate | 6 smooth hue: times G | 7 Freeman: R−G, B−G

8 Freeman: median | **9 Freeman: add G back** | 10 Freeman on edges, 5×5: add G back | 11 Malvar: correction | 12 Malvar: add to bilinear

13 Hamilton–Adams: G | 14 Hamilton–Adams: R, B at G | 15 Hamilton–Adams: diagonals | 16 VNG: directions | 17 VNG: average | 18 AHD: horizontal

19 AHD: vertical | 20 AHD: choose | 21 AHD: result | 22 RI: G by residuals | 23 RI: tentative $\hat{R} = aG + b$ | 24 RI: residual vs G − R | 25 RI: add back

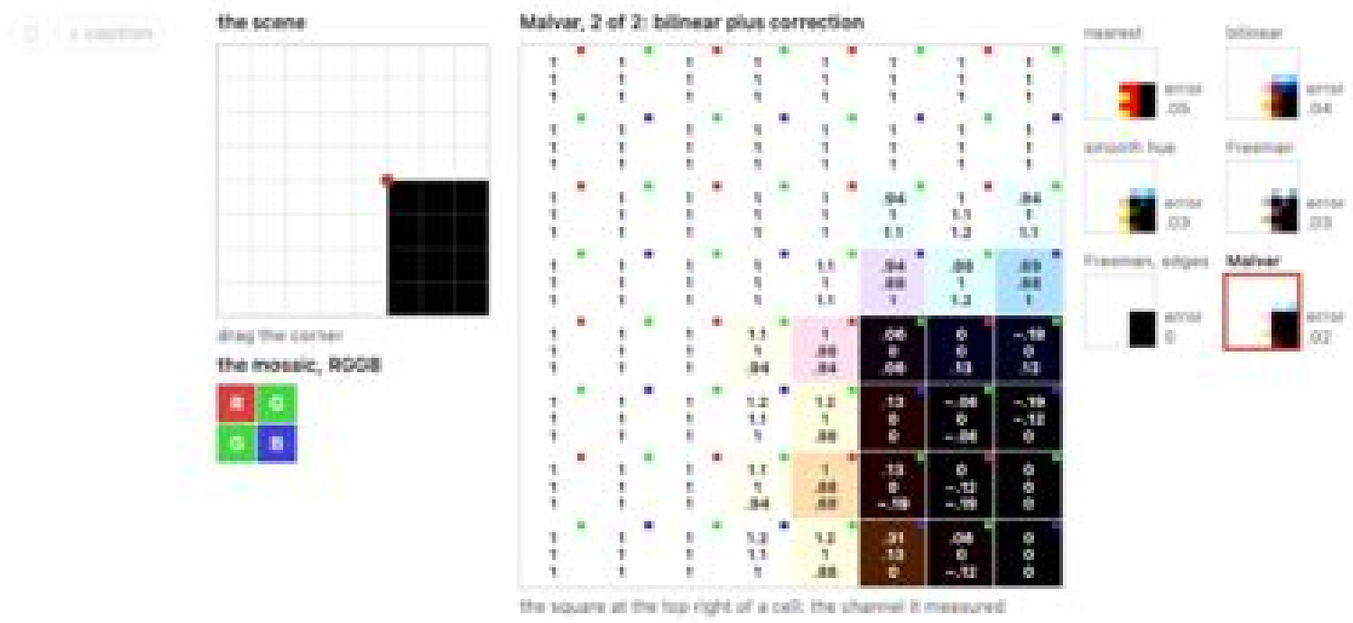
26 AHI: per-pixel choice | 27 frequency: spectrum | 28 frequency: C_g | 29 frequency: C_r, two copies | 30 frequency: luma | 31 frequency: R, G, B ▶

input: corner | colors: black on white | grid: 8 × 8, with values

angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

Demo: The classical methods, step by step: Malvar: add to bilinear



1 input: mosaic | 2 nearest | 3 bilinear | 4 smooth hue: R/G, B/G | 5 smooth hue: interpolate | 6 smooth hue: times G | 7 Freeman: R−G, B−G

8 Freeman: median | 9 Freeman: add G back | 10 Freeman on edges, 5×5: add G back | 11 Malvar: correction | **12 Malvar: add to bilinear**

13 Hamilton–Adams: G | 14 Hamilton–Adams: R, B at G | 15 Hamilton–Adams: diagonals | 16 VNG: directions | 17 VNG: average | 18 AHD: horizontal

19 AHD: vertical | 20 AHD: choose | 21 AHD: result | 22 RI: G by residuals | 23 RI: tentative $\hat{R} = aG + b$ | 24 RI: residual vs G − R | 25 RI: add back

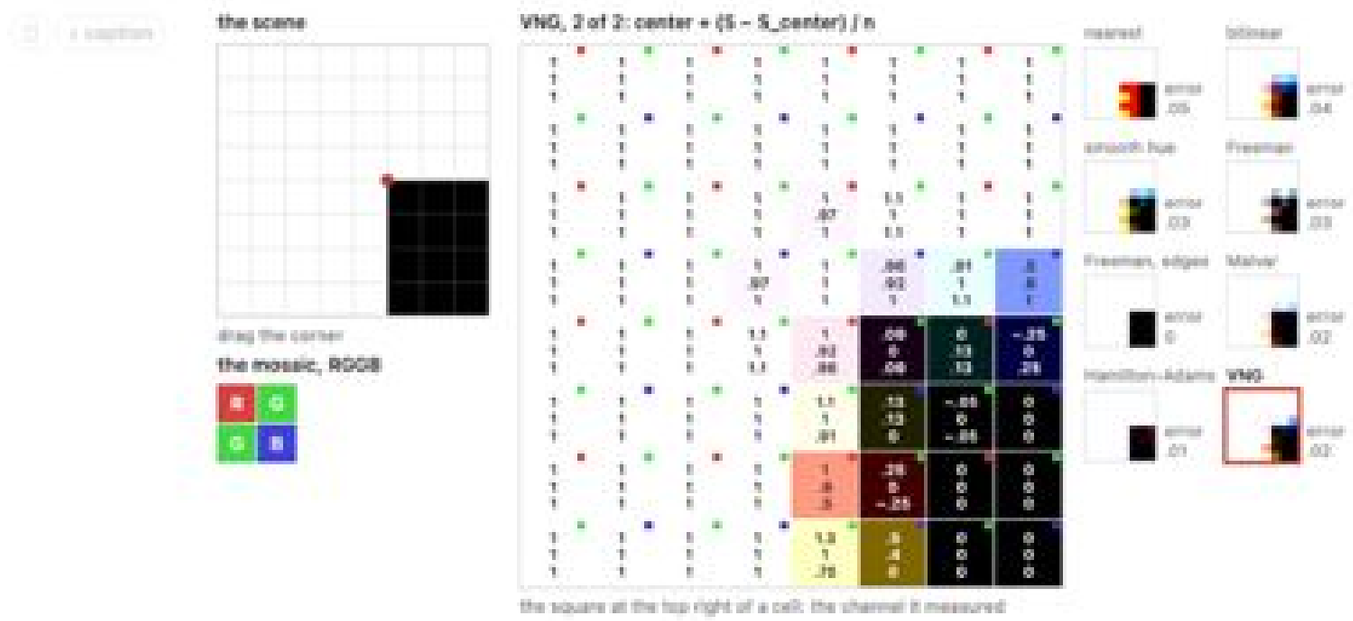
26 ARI: per-pixel choice | 27 frequency: spectrum | 28 frequency: C_0 | 29 frequency: C_1, two copies | 30 frequency: luma | 31 frequency: R, G, B ▶

input: corner | colors: black on white | grid: 8 × 8, with values

angle: 0° | then Freeman's 5 × 5 median | reset

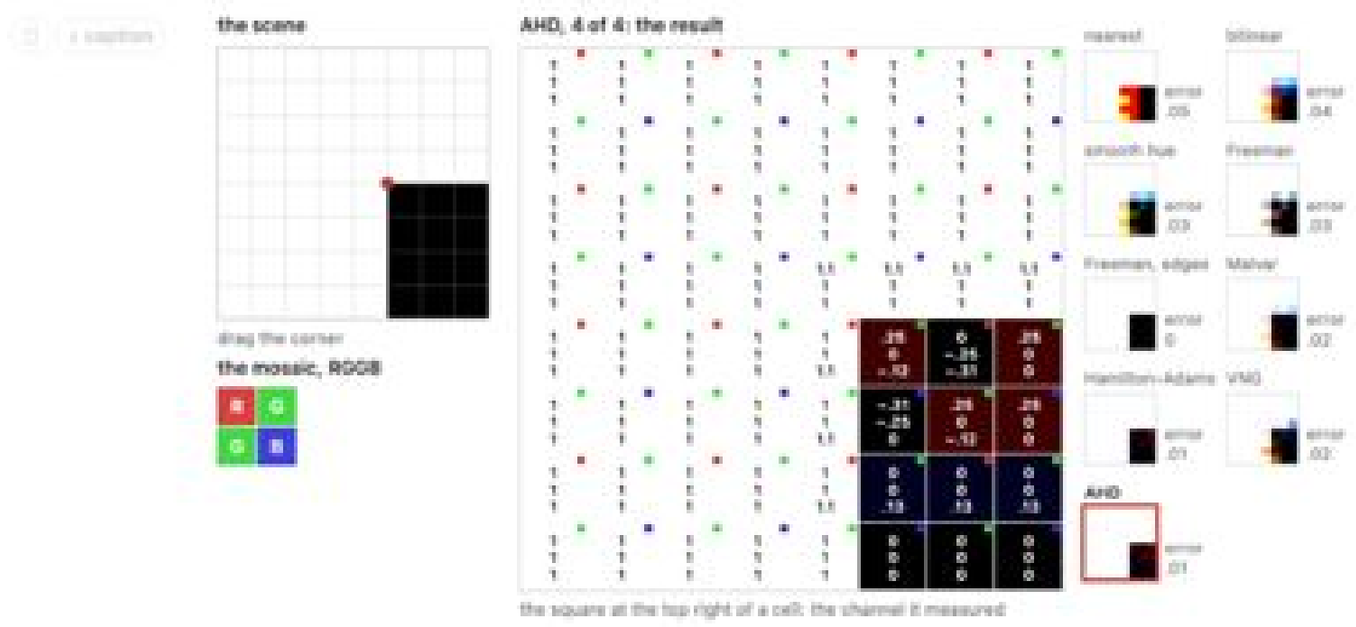
Demo courtesy of Frédo Durand, MIT.

Demo: The classical methods, step by step: VNG: average



Demo courtesy of Frédo Durand, MIT.

Demo: The classical methods, step by step: AHD: result



1 input: mosaic | 2 nearest | 3 bilinear | 4 smooth hue: R/G, B/G | 5 smooth hue: interpolate | 6 smooth hue: times G | 7 Freeman: R−G, B−G

8 Freeman: median | 9 Freeman: add G back | 10 Freeman on edges, 5×5: add G back | 11 Malvar: correction | 12 Malvar: add to bilinear

13 Hamilton–Adams: G | 14 Hamilton–Adams: R, B at G | 15 Hamilton–Adams: diagonals | 16 VNG: directions | 17 VNG: average | 18 AHD: horizontal

19 AHD: vertical | 20 AHD: choose | **21 AHD: result** | 22 RI: G by residuals | 23 RI: tentative $\hat{R} = aG + b$ | 24 RI: residual vs G − R | 25 RI: add back

26 ARI: per-pixel choice | 27 frequency: spectrum | 28 frequency: C_0 | 29 frequency: C_1, two copies | 30 frequency: luma | 31 frequency: R, G, B ▶

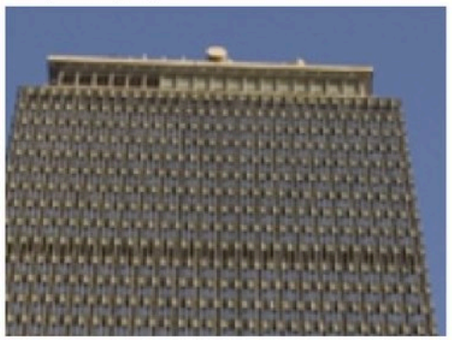
input: corner | colors: black on white | grid: 8 × 8, with values

angle: 0° | then Freeman's 5 × 5 median | reset

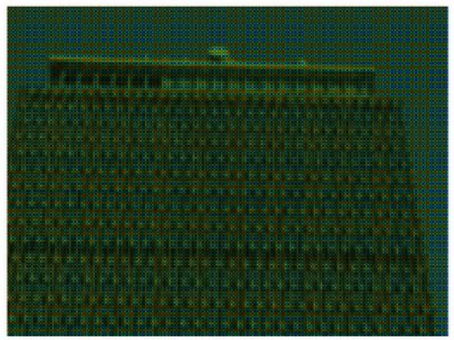
Demo courtesy of Frédo Durand, MIT.

Demo: How AHD chooses

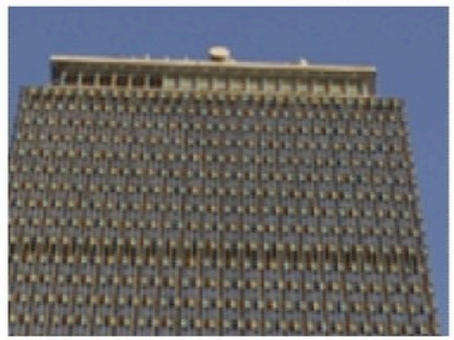
the photograph



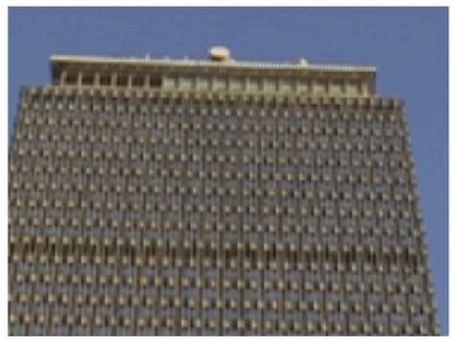
its RGGB mosaic



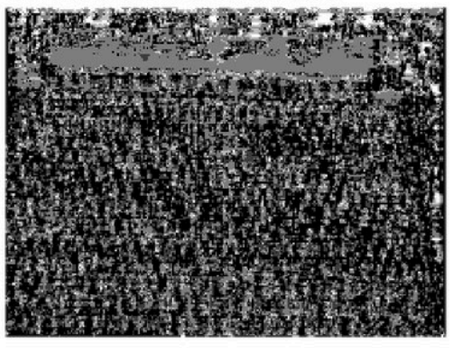
horizontal candidate (error 2.4)



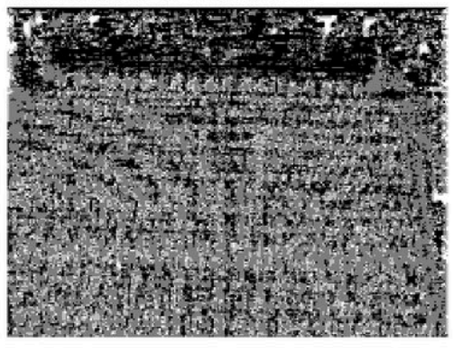
vertical candidate (error 1.9)



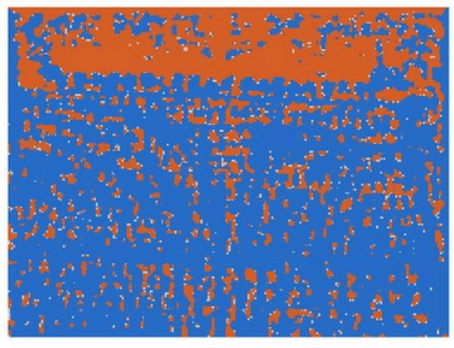
homogeneity of the horizontal



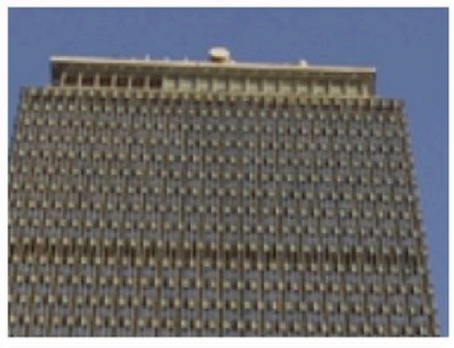
homogeneity of the vertical



choice: orange H, blue V



AHD result (error 1.7)



The choice picks the more accurate candidate at 68% of the pixels where the two differ. Photograph © Frédo Durand.

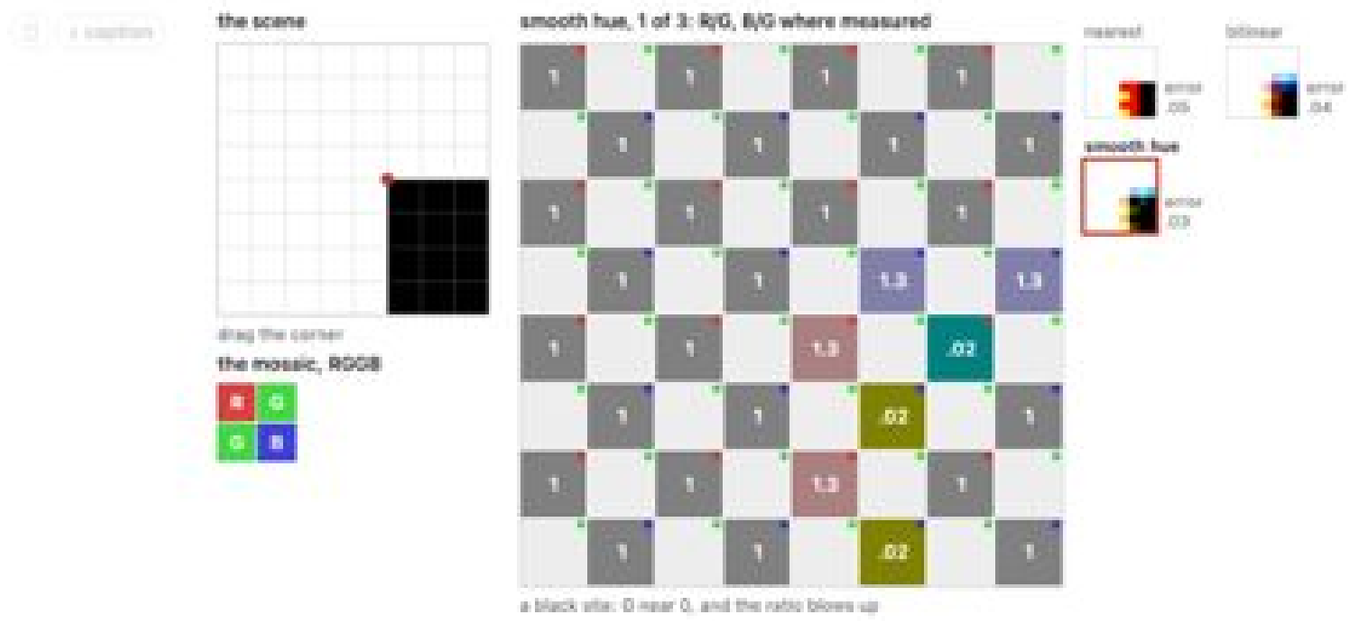
Reference

- What we do essentially follows
  https://patents.google.com/patent/US5382976A/en
  Inventor Robert H. Hibbard (Kodak)

Questions?



Demo: The classical methods, step by step: smooth hue: R/G, B/G



1 input: mosaic | 2 nearest | 3 bilinear | **4 smooth hue: R/G, B/G** | 5 smooth hue: interpolate | 6 smooth hue: times G | 7 Freeman: R−G, B−G

8 Freeman: median | 9 Freeman: add G back | 10 Freeman on edges, 5×5: add G back | 11 Malvar: correction | 12 Malvar: add to bilinear

13 Hamilton–Adams: G | 14 Hamilton–Adams: R, B at G | 15 Hamilton–Adams: diagonals | 16 VNG: directions | 17 VNG: average | 18 AHD: horizontal

19 AHD: vertical | 20 AHD: choose | 21 AHD: result | 22 RI: G by residuals | 23 RI: tentative $\hat{R} = aG + b$ | 24 RI: residual vs G − R | 25 RI: add back

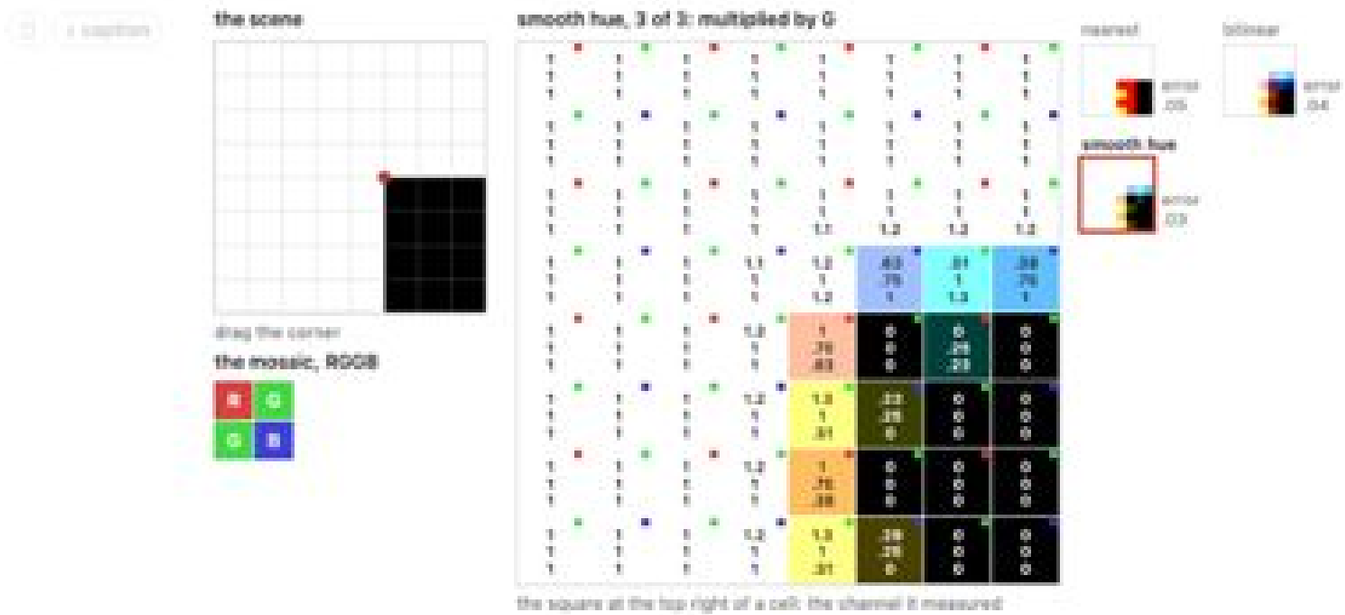
26 ARI: per-pixel choice | 27 frequency: spectrum | 28 frequency: C_0 | 29 frequency: C_1, two copies | 30 frequency: luma | 31 frequency: R, G, B

input: corner | colors: black on white | grid: 8 × 8, with values

angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

Demo: The classical methods, step by step: smooth hue: times G



1 input: mosaic | 2 nearest | 3 bilinear | 4 smooth hue: R/G, B/G | 5 smooth hue: interpolate | **6 smooth hue: times G** | 7 Freeman: R−G, B−G

8 Freeman: median | 9 Freeman: add G back | 10 Freeman on edges, 5×5: add G back | 11 Malvar: correction | 12 Malvar: add to bilinear

13 Hamilton–Adams: G | 14 Hamilton–Adams: R, B at G | 15 Hamilton–Adams: diagonals | 16 VNG: directions | 17 VNG: average | 18 AHD: horizontal

19 AHD: vertical | 20 AHD: choose | 21 AHD: result | 22 RI: G by residuals | 23 RI: tentative $\hat{R} = aG + b$ | 24 RI: residual vs G − R | 25 RI: add back

26 AHI: per-pixel choice | 27 frequency: spectrum | 28 frequency: C_g | 29 frequency: C_r, two copies | 30 frequency: luma | 31 frequency: R, G, B ▶

input: corner | colors: black on white | grid: 8 × 8, with values

angle: 0° | reset

Demo courtesy of Frédo Durand, MIT.

Challenges



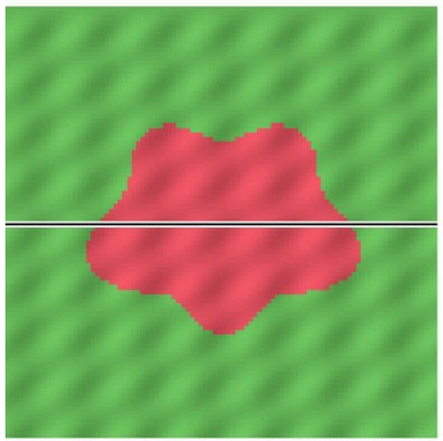
Edge cases

- http://www.luminous-landscape.com/contents/DNG-Recover-Edges.shtml
- http://www.luminous-landscape.com/forum/index.php?topic=51328.0

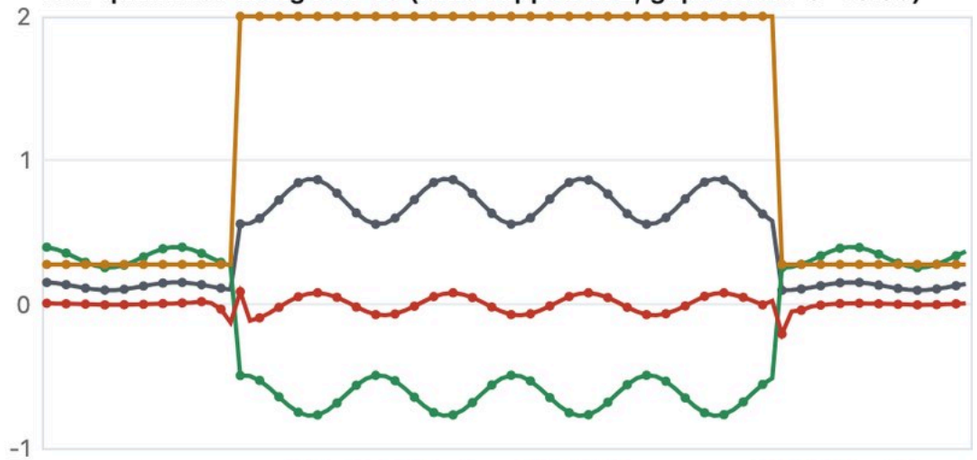
More approaches

Demo: Residual interpolation along a scanline

the scene

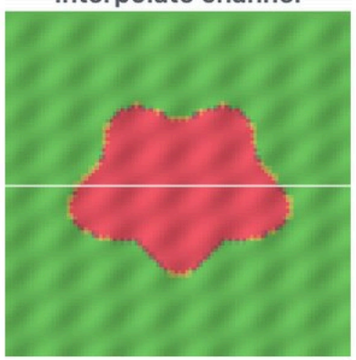
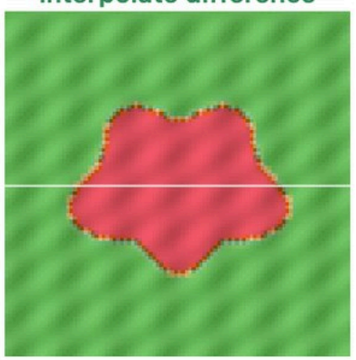
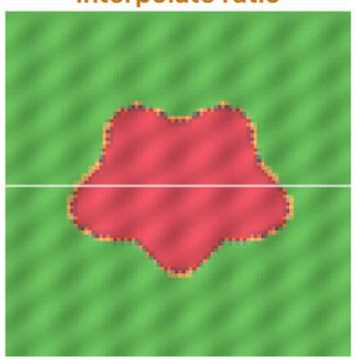
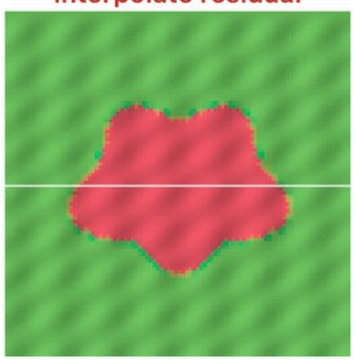


four quantities along row 48 (ratio clipped at 2; gaps where G < 0.05)



dots: measured red sites; x across the image →

reconstruction

interpolate channel	interpolate difference	interpolate ratio	interpolate residual
			
28.70 dB	25.63 dB	23.07 dB	**31.56 dB best here**

PSNR of red and blue at unmeasured sites; green is the same bilinear-style interpolation in all four.

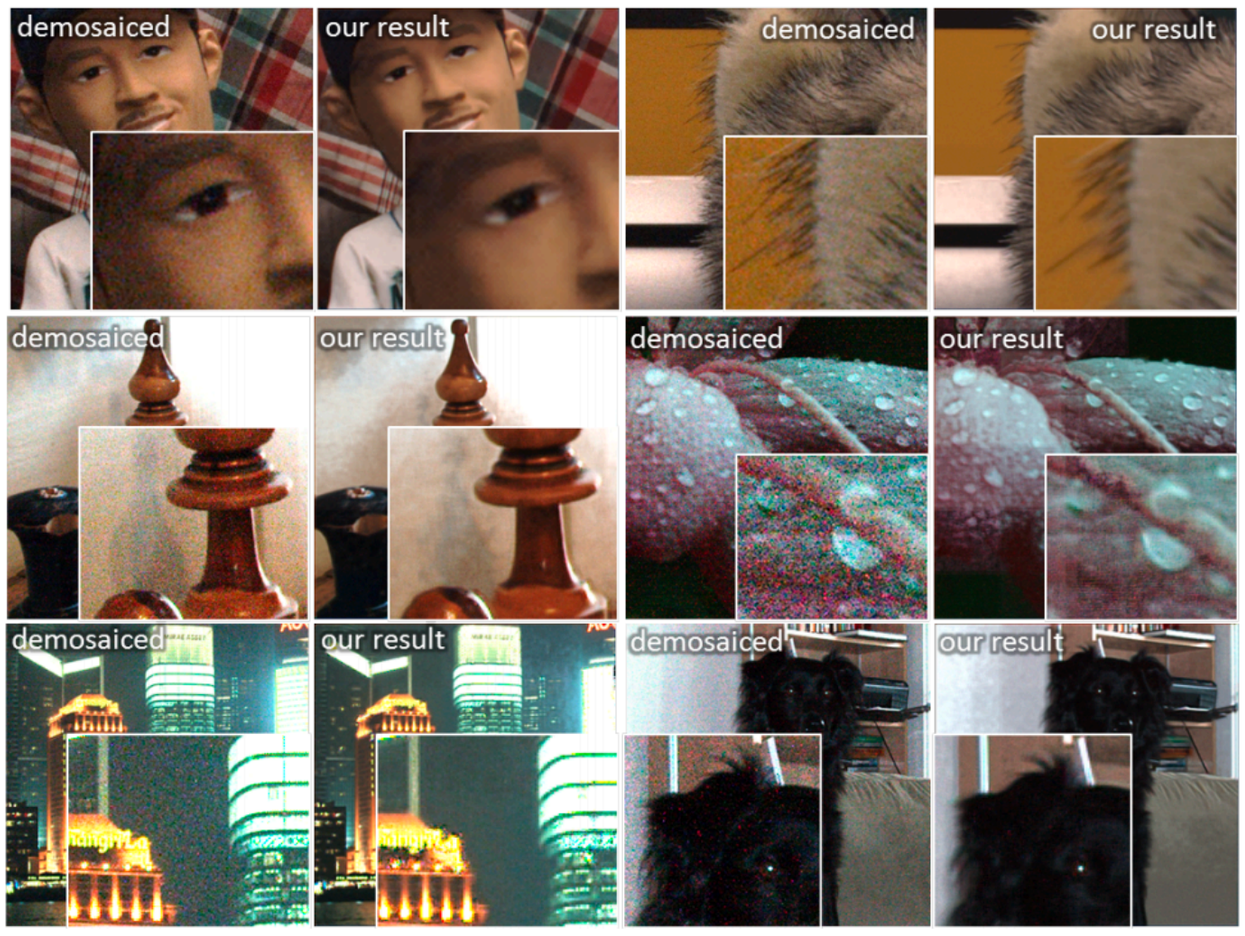
Residual interpolation and ARI

1. **Green first.** Estimate green along rows and along columns separately, each time guiding by the measured red or blue along that line and interpolating the residual along it, then combine the two directional estimates with weights that favor the direction whose color differences vary less locally.
2. **Tentative red.** With the full green as the guide, fit a local linear model $\tilde{R} = aG + b$ in each window, using only the sites where red was measured, and evaluate it everywhere.
3. **Residual.** At the measured red sites, form $R - \tilde{R}$.
4. **Interpolate and add back.** Interpolate the residual bilinearly to every site and add it to $\tilde{R}$. Blue is the same with blue's samples.

Joint and learned methods

Denoising & Demosaicking

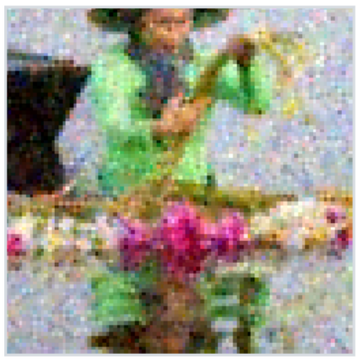
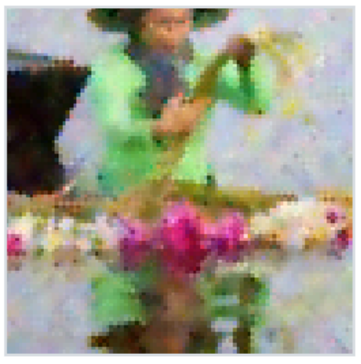
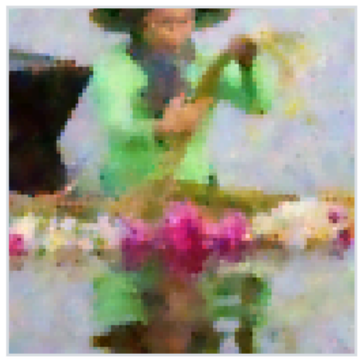
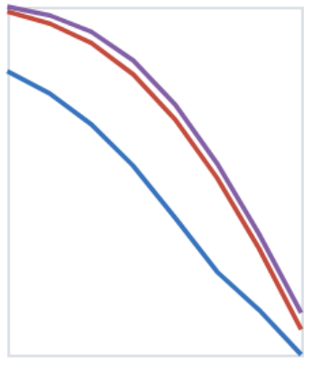
- **http://research.microsoft.com/en-us/UM/people/yasumat/papers/lowlight_CVPR11.pdf**



Demo: Joint against sequential denoising and demosaicking

compare three simple denoising and demosaicking pipelines

Toy noise model: ISO 3200 · σ 0.0679 on the mosaic

no denoising	denoise → demosaic	demosaic → RGB	demosaic → Y/C	
				
22.79 dB	24.32 dB same-colour filter	24.57 dB RGB filter	22.73 dB luma/chroma filters	ISO 100 → 12800 luma/chroma denoise first demosaic first

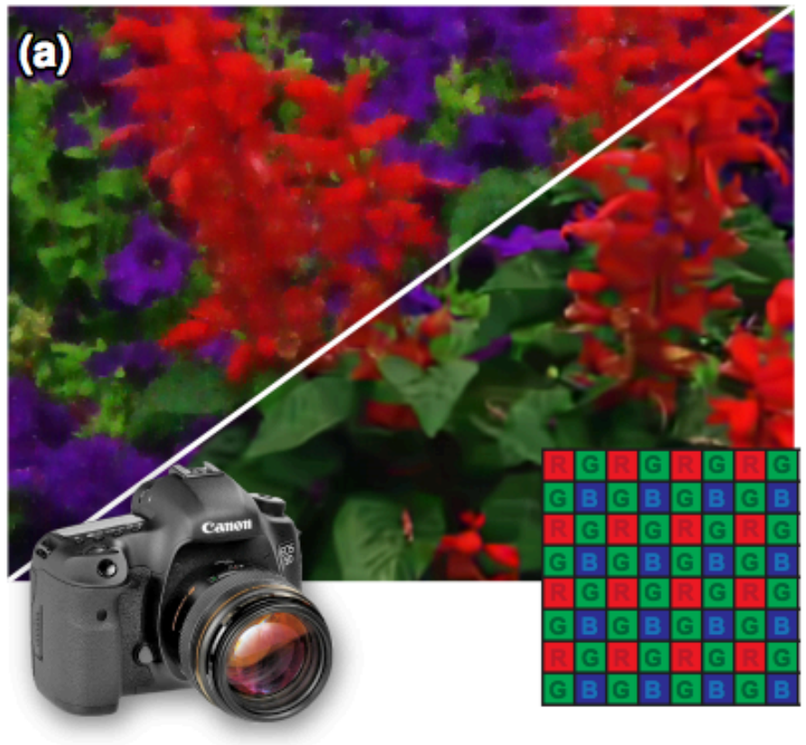
The three orderings differ by 1.09 dB at ISO 100 and 0.71 dB at ISO 12800.

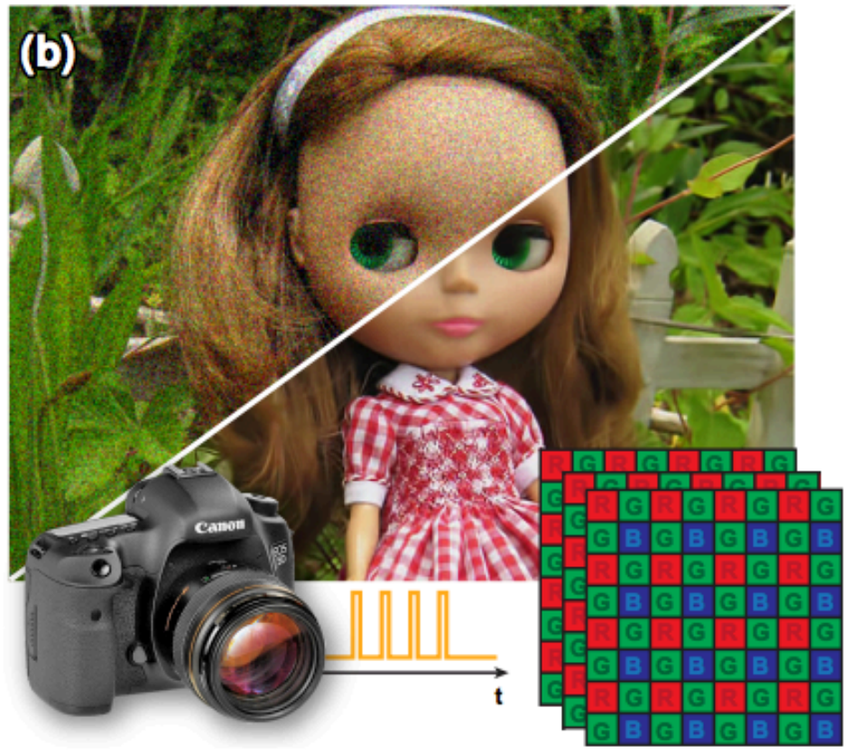
Gaussian mosaic noise σ = 0.012 √(ISO/100), clamped; colour-difference demosaicking; all sequential.

These measurements do not test a joint estimator or establish how learned methods rank.

Demosaick+denoise+…

- https://research.nvidia.com/publication/flexisp-flexible-camera-image-processing-framework





Video: FlexISP: A Flexible Camera Image Processing Framework

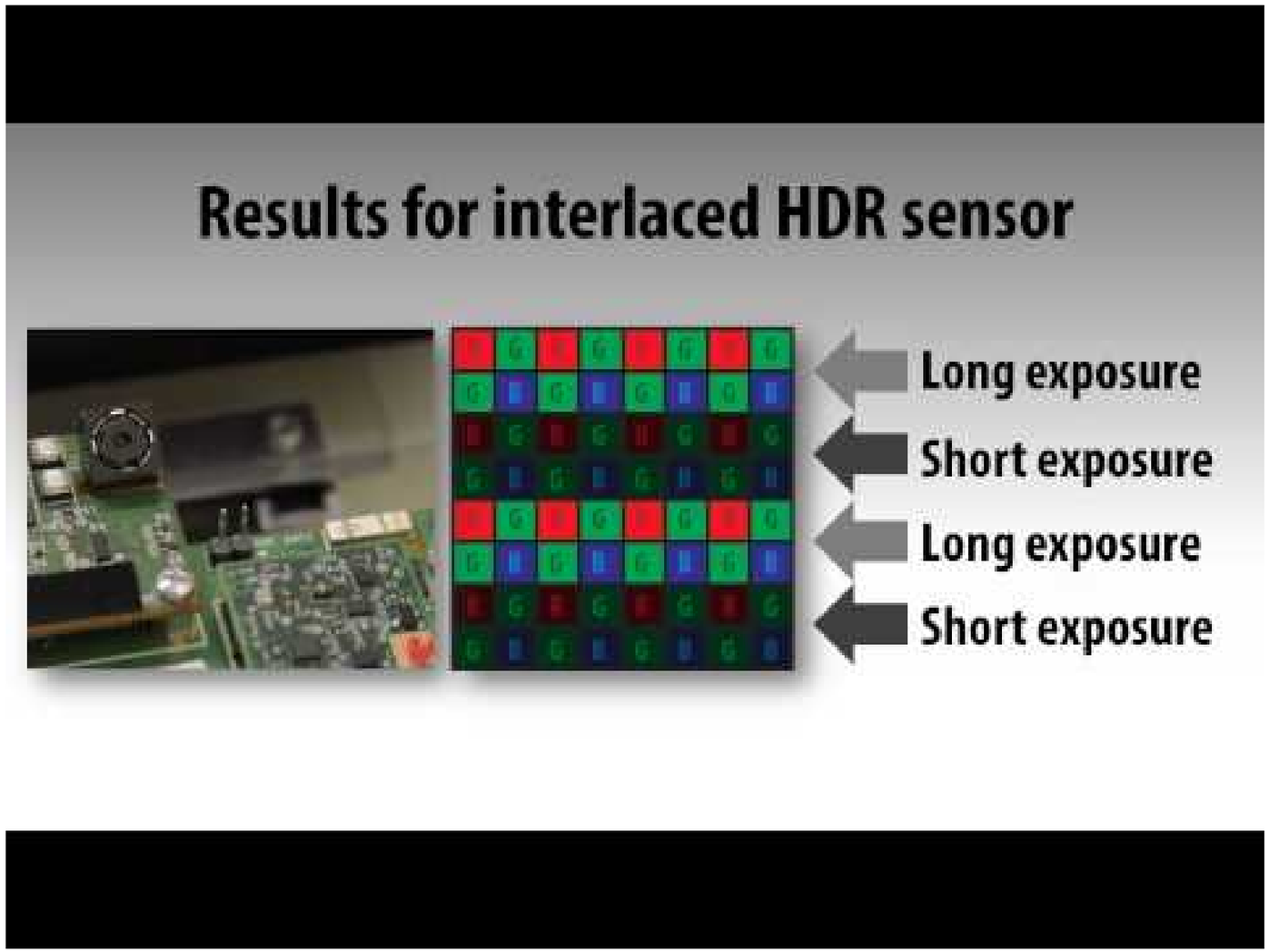
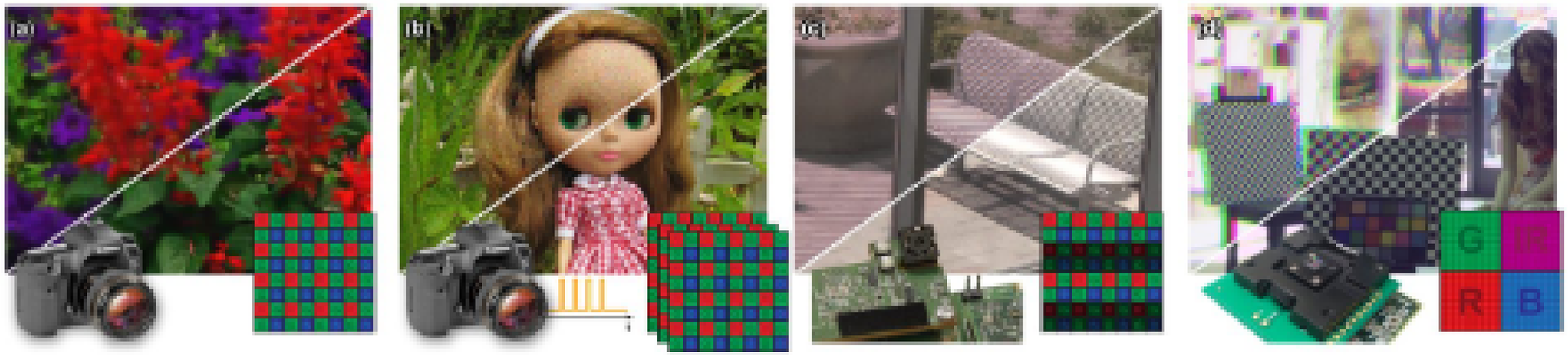


Figure 24.3.8: One optimization, many imaging problems



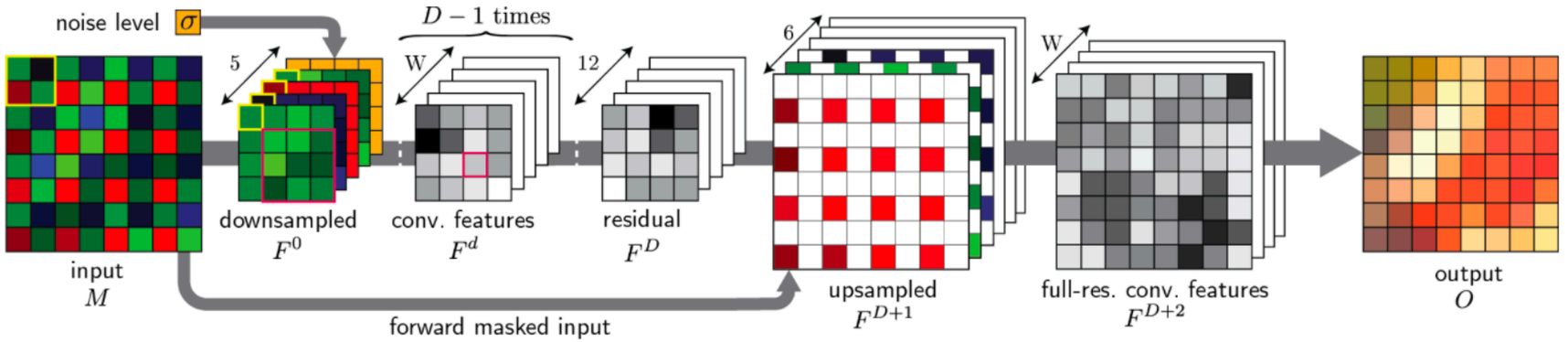
Reproduced from Heide et al. (2014), with permission of F. Heide.

FlexISP poses the whole image-processing pipeline as a single objective (a data term tying the latent image to the raw measurements through the sensor's forward model…

Deep Joint Demosaicking and Denoising

Siggraph Asia 2016

Michaël Gharbi MIT CSAIL
Gaurav Chaurasia MIT CSAIL
Sylvain Paris Adobe
Frédo Durand MIT CSAIL



- **https://groups.csail.mit.edu/graphics/demosaicnet/**
- Now in Adobe Camera Raw/Photoshop/Lightroom

Video: **Deep Joint Demosaicking and Denoising, Michaël Gharbi, Gaurav Chaurasia…**

[redacted]



[redacted]

Figure 15.4.2: **The learned model's result, set against the methods of this chapter**



From Gharbi, Chaurasia, Paris & Durand (2016).

From a single noisy Bayer mosaic (the regime where green-based and edge-directed demosaicking start to crumble because the noise corrupts the very gradients and color…

Adobe Camera Raw: Enhance Details



Adobe Camera Raw "Enhance Details" Feature

https://diglloyd.com/blog/2019/20190319_1150-Adobe-EnhanceDetails-commer

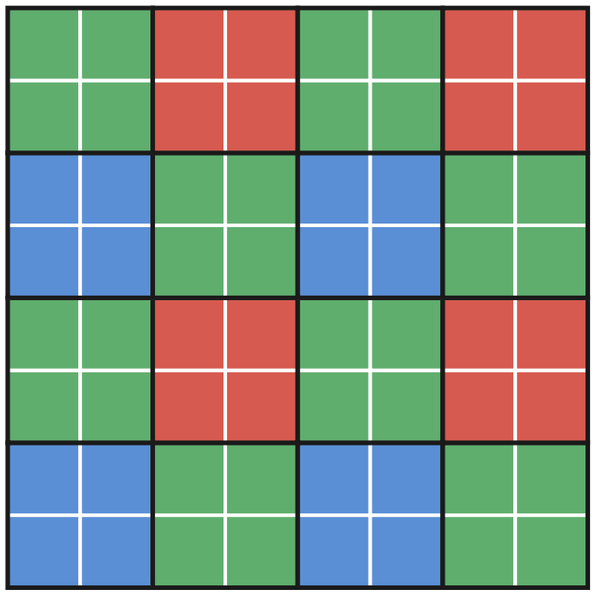
Trends in mosaics and demosaicking

Quad sensors

2x2 blocks of the same color

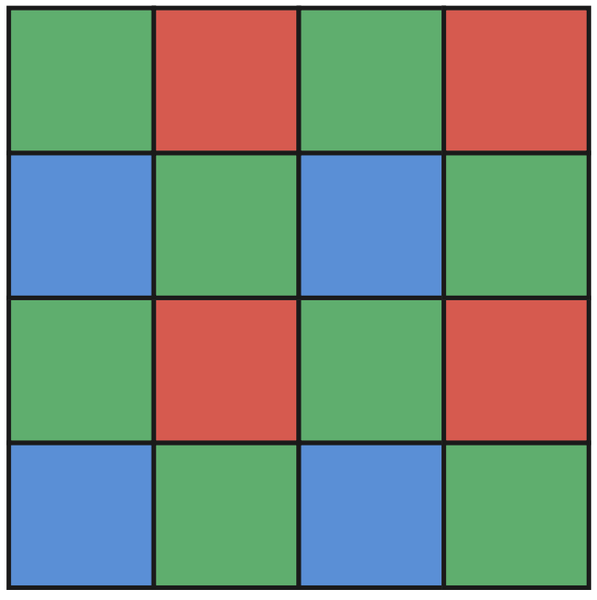
full res mode and half-resolution pooled mode

full resolution



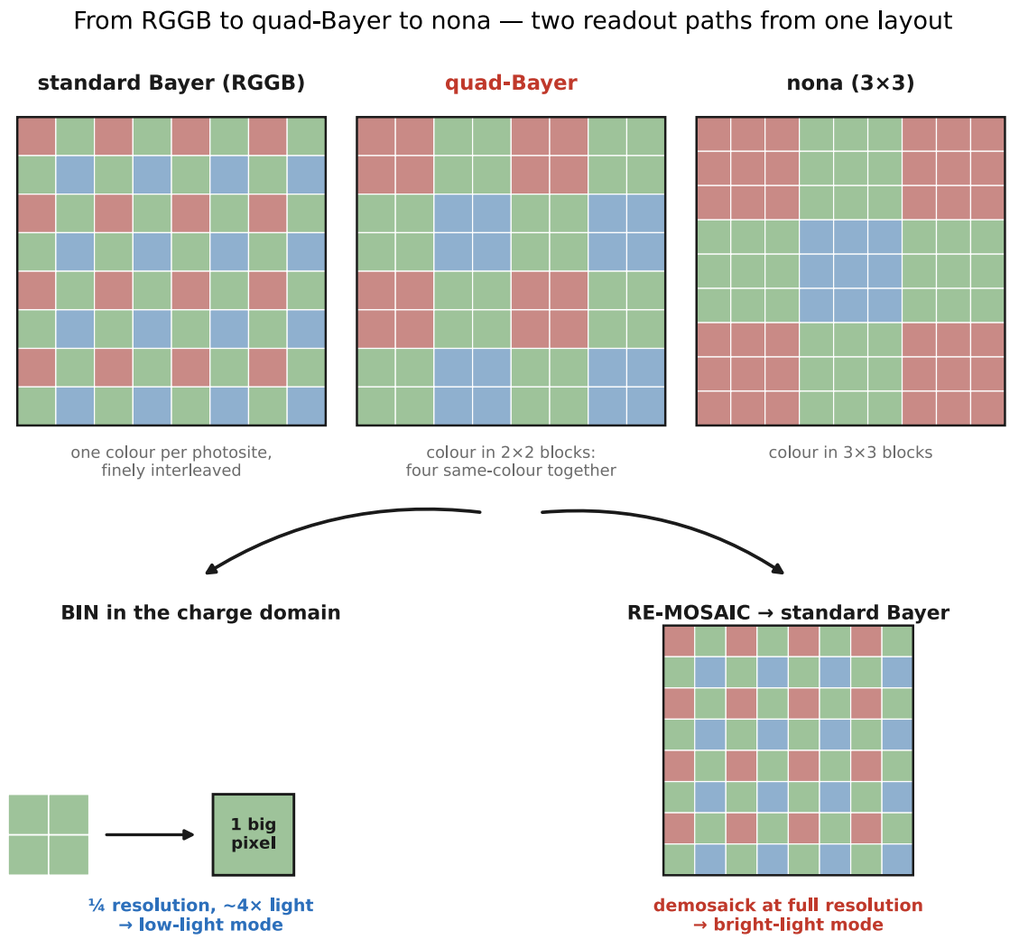
every photosite read: 8 × 8 pixels

binned (low light)



each 2 × 2 block summed: 4 × 4 pixels

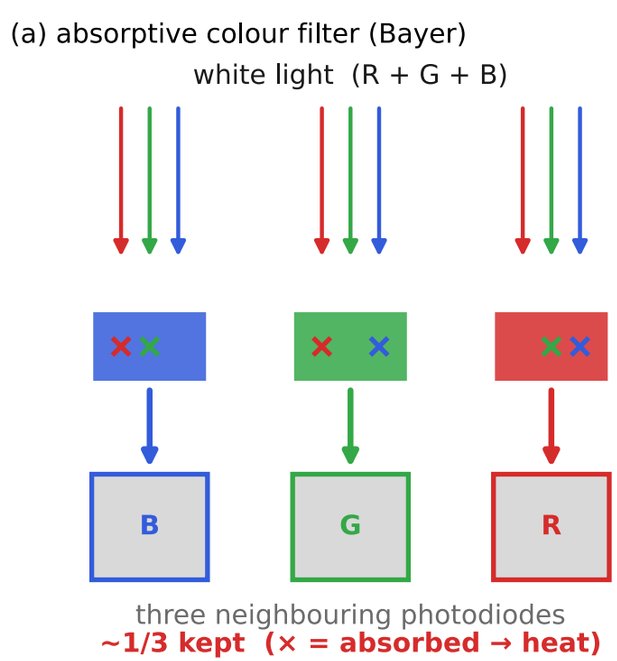
Figure 34.2.1: From RGGB to quad Bayer to nona

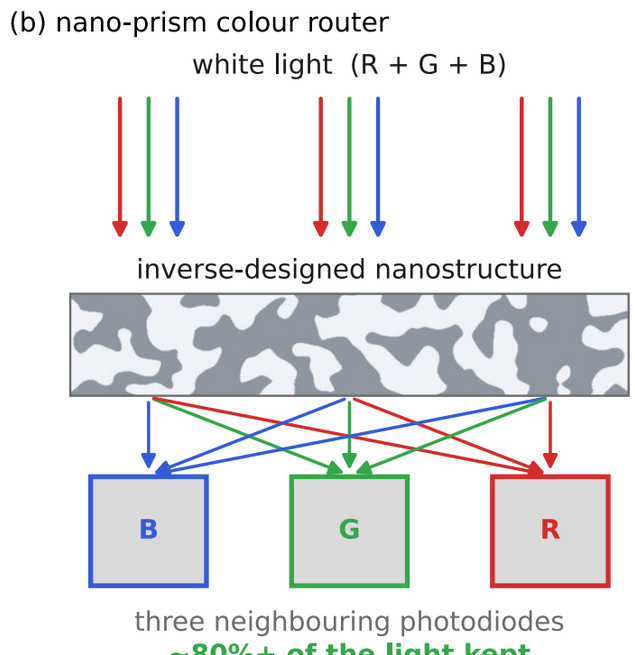


Left: the standard Bayer tile, one color per photosite, finely interleaved.

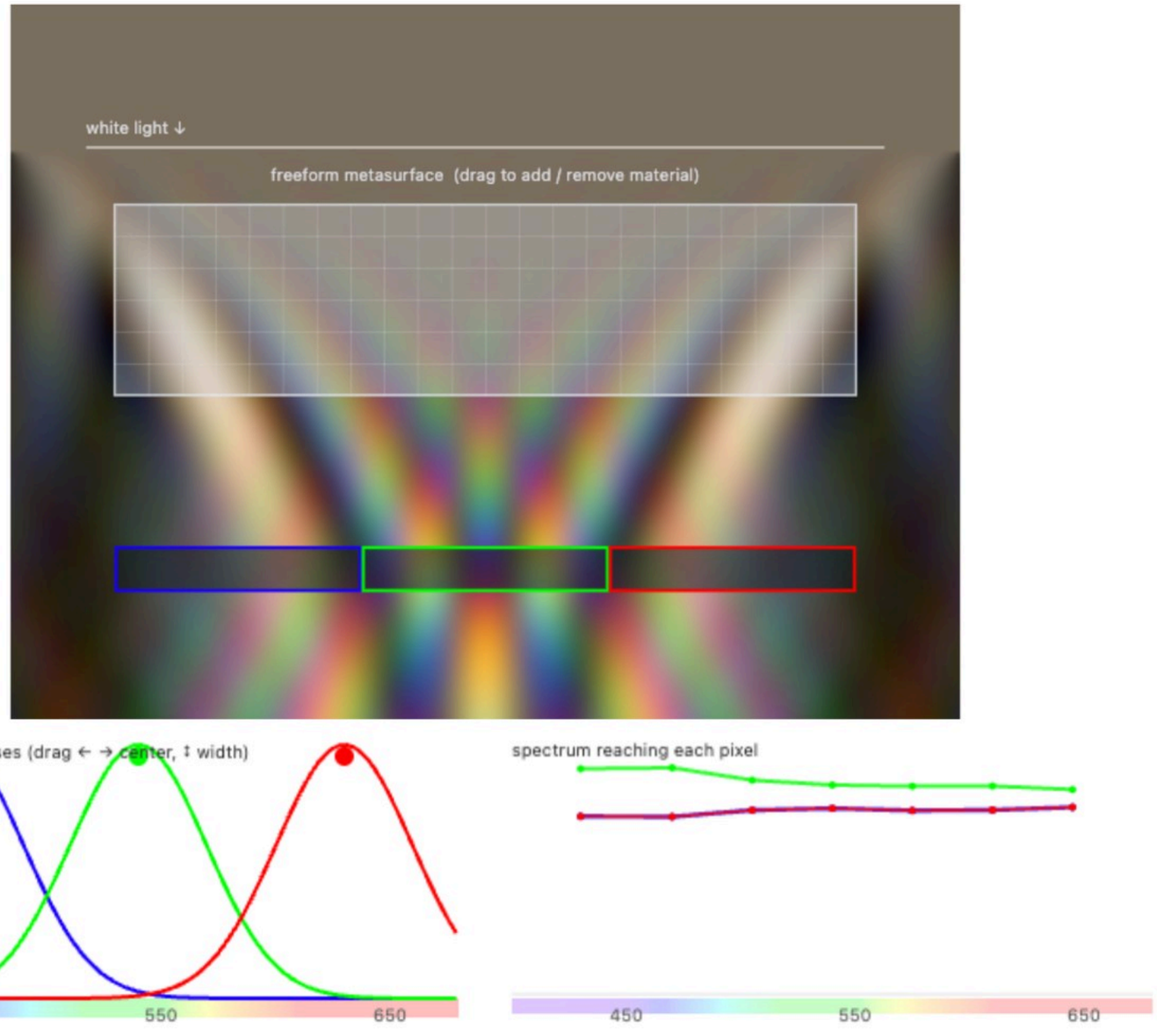
Nano prisms

Meta-optics routes light according to wavelength





Illustrative nano-prism: uniform density, white light



Seven wavelengths, 430–645 nm; one-way scalar beam propagation. Index 1–2.1. Colors and intensity are normalized for visibility; no reflections or polarization.

Searching and optimizing demosaicking

http://gradient.halide.ai/

https://dspace.mit.edu/entities/publication/d08a6c07-8213-4076-8f31-19344f1c96d8



Fig. 5. We use our automatic gradients to relax the AHD demosaicking algorithm (a) by adding more filters to interpolate the green channel (8 instead of 2 here, with 5x5 footprint instead of 5x1). With this simple tweak, and by optimizing the filters using our automatically generated derivatives, we can obtain sharper images in difficult cases (b), first row. The small-footprint of this simple demosaicking method nevertheless inherits some of the limitations of AHD. In particular, it leads to artifacts in complex, moiré-prone patterns (second row). Images taken from deep demosaicking dataset [Gharbi et al. 2016].

Differentiable Programming for
Image Processing and Deep Learning in Halide

TZU-MAO LI, MIT CSAIL
MICHAËL GHARBI, MIT CSAIL
ANDREW ADAMS, Facebook AI Research
FRÉDO DURAND, MIT CSAIL
JONATHAN RAGAN-KELLEY, UC Berkeley & Google

Searching for Fast Demosaicking Algorithms

KARIMA MA, Massachusetts Institute of Technology
MICHAEL GHARBI, ANDREW ADAMS, and SHOAIB KAMIL, Adobe Systems Inc
TZU-MAO LI, University of California San Diego
CONNELLY BARNES, Adobe Systems Inc
JONATHAN RAGAN-KELLEY, Massachusetts Institute of Technology

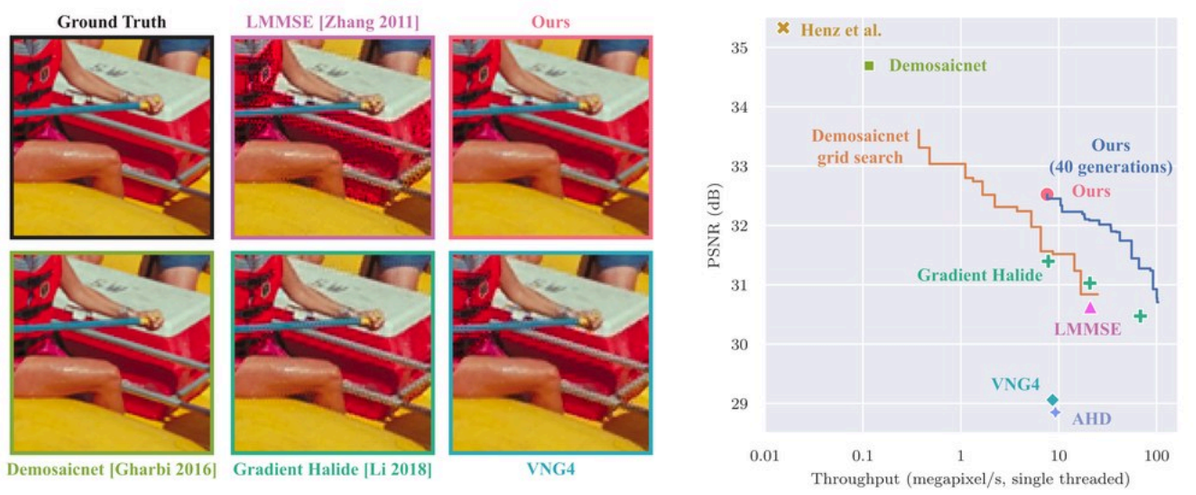


Fig. 1. We use a combination of learning and program search to automatically synthesize efficient, high-quality demosaicking algorithms. They significantly advance the Pareto frontier of cost vs. quality over prior state-of-the-art methods from 10 s to 1000 s of operations per pixel (plot, right). They are at least 1dB higher quality at the same cost, or 5–10× faster at the same quality, relative to prior published algorithms. Visual quality is noticeably improved on challenging image content (note the Bayer grid speckling and zippering artifacts in the LMMSE, GradientHalide, and VNG outputs). The only prior methods which offer higher quality than ours are large convolutional models 2–3 orders of magnitude more computationally expensive (Demosaicnet, Henz at al.). In addition to traditional Bayer demosaicking shown here, we present Pareto-dominant algorithms for demosaicking from X-Trans sensors, and for joint demosaicking superresolution and superresolution alone.

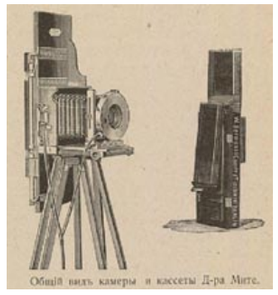
Other Color Acquisition

Color sampling

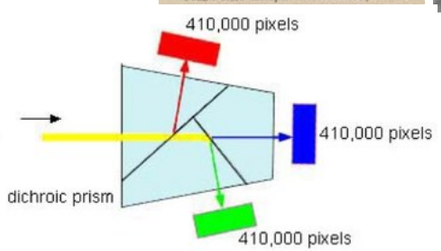
- Problem: a photosite can record only one number
- We need 3 numbers for color
- What can we do?

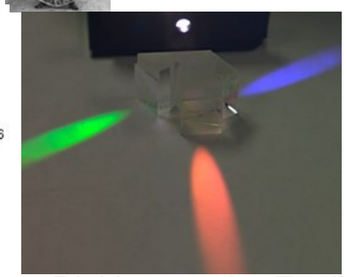
What are some approaches to sensing color images?

- Scan 3 times (temporal multiplexing)
- Use 3 detectors (3-ccd camera)
- Use offset color samples (spatial multiplexing)
- Multiplex in depth (Tripack film, Foveon)

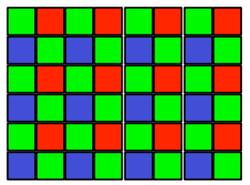








Dicklyon at the English Wikipedia project.





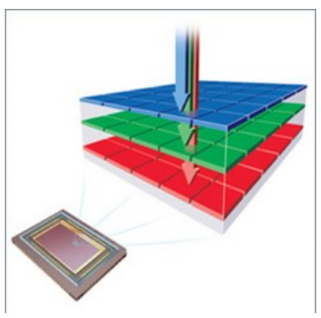
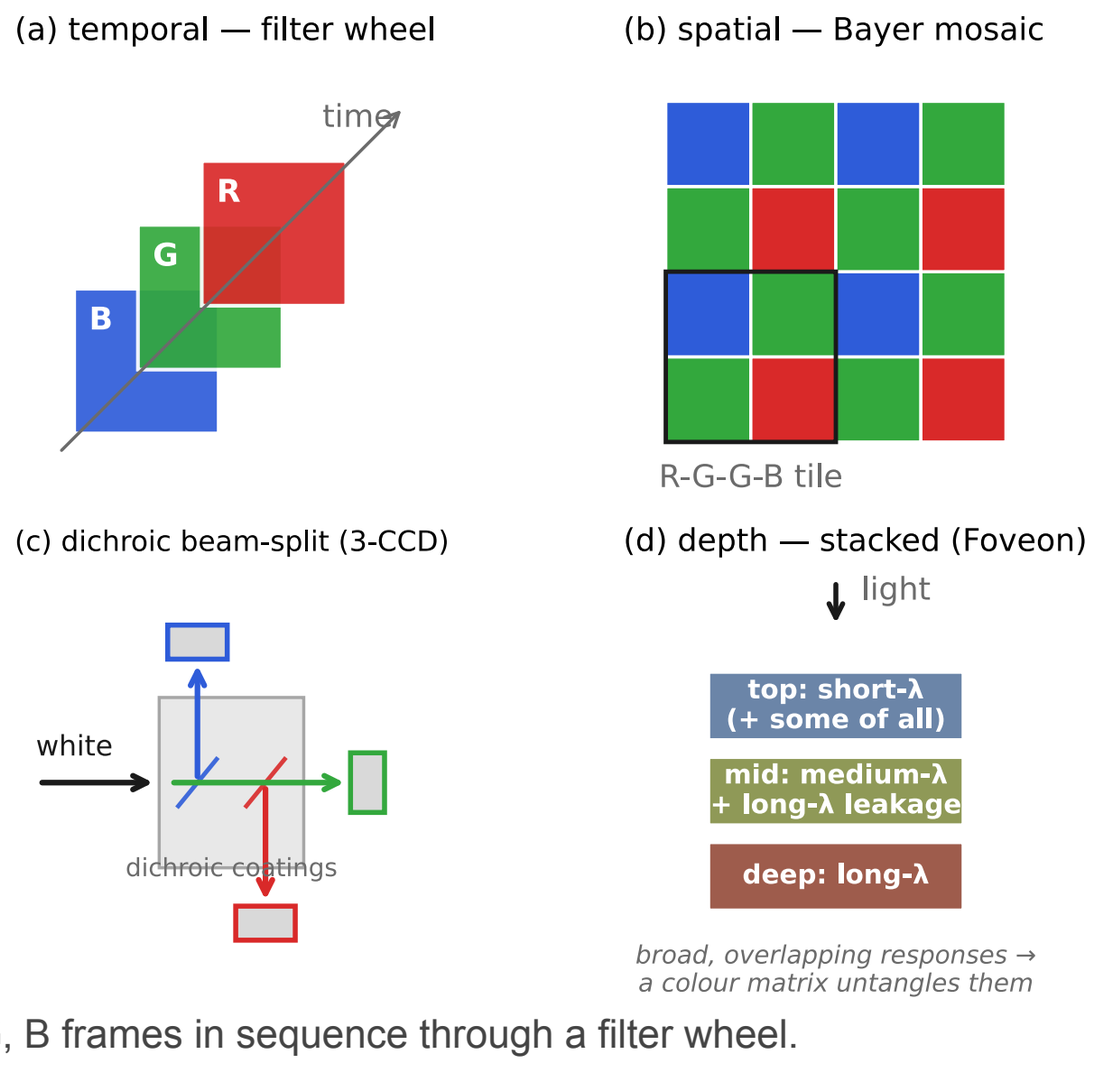


Figure 4.6.1: Four color-sensing multiplexing strategies



Temporal: shoot R, G, B frames in sequence through a filter wheel.

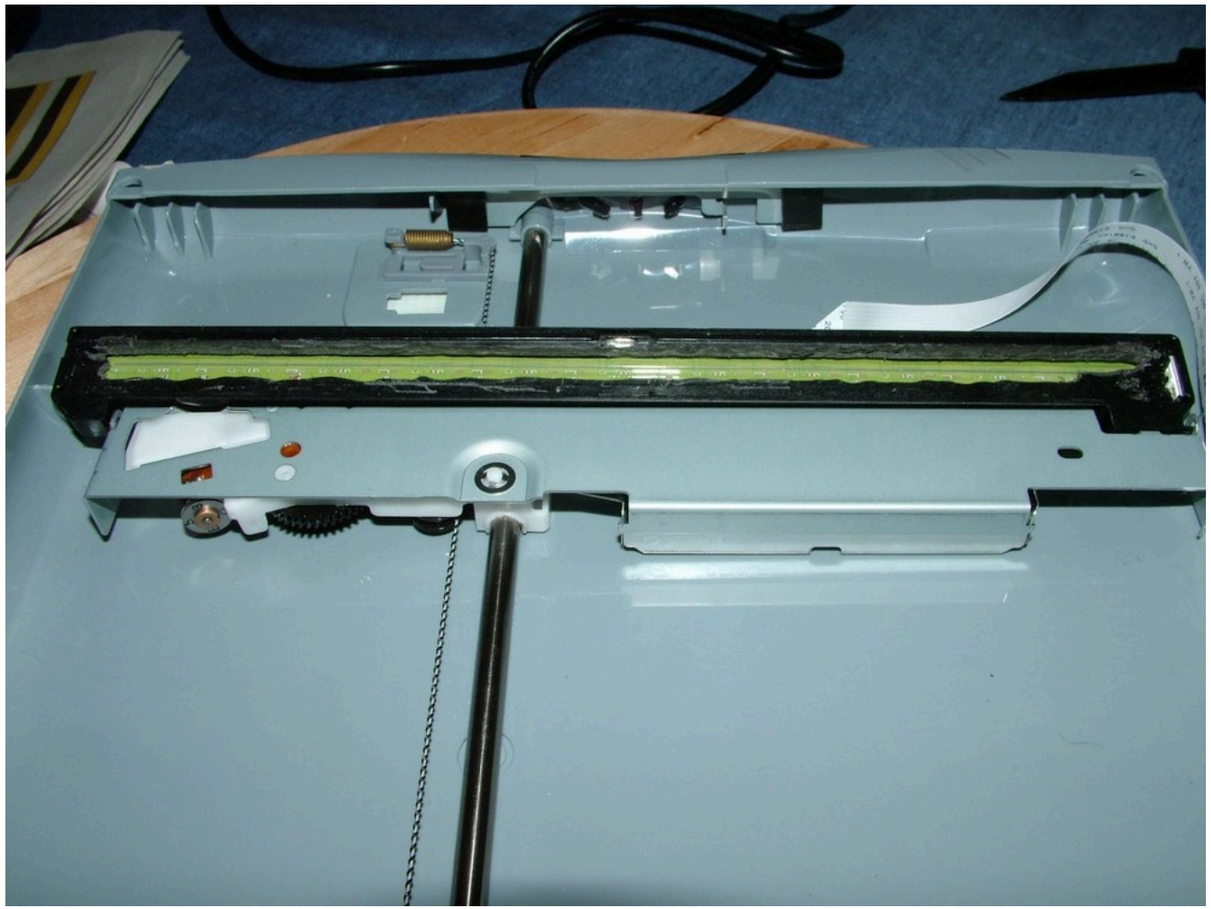
Temporal multiplexing

- 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



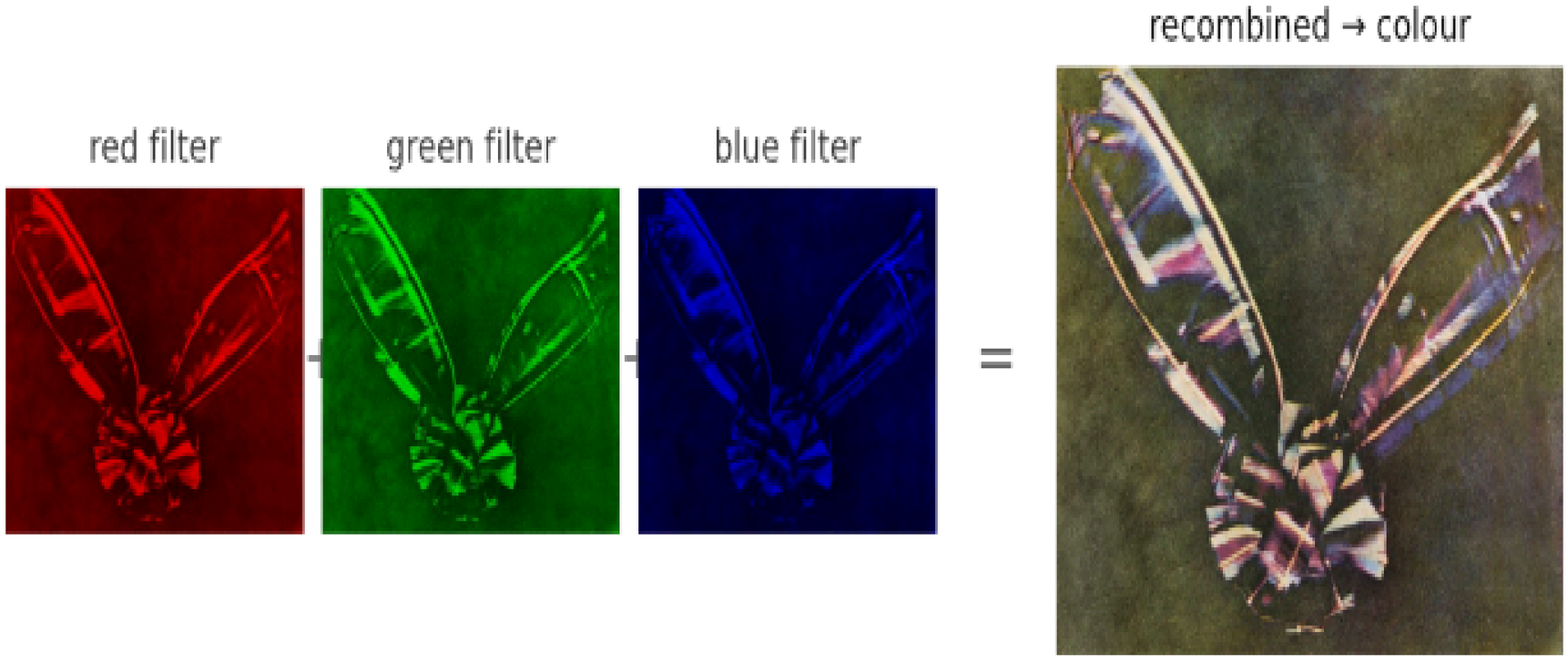
Figure 4.6.2: A scanner's linear sensor bar

A scanner's linear sensor bar

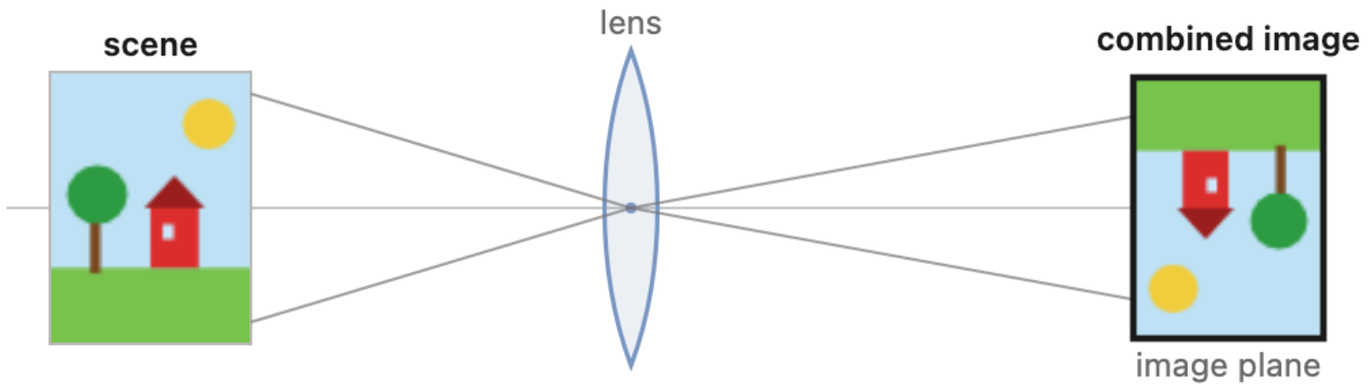


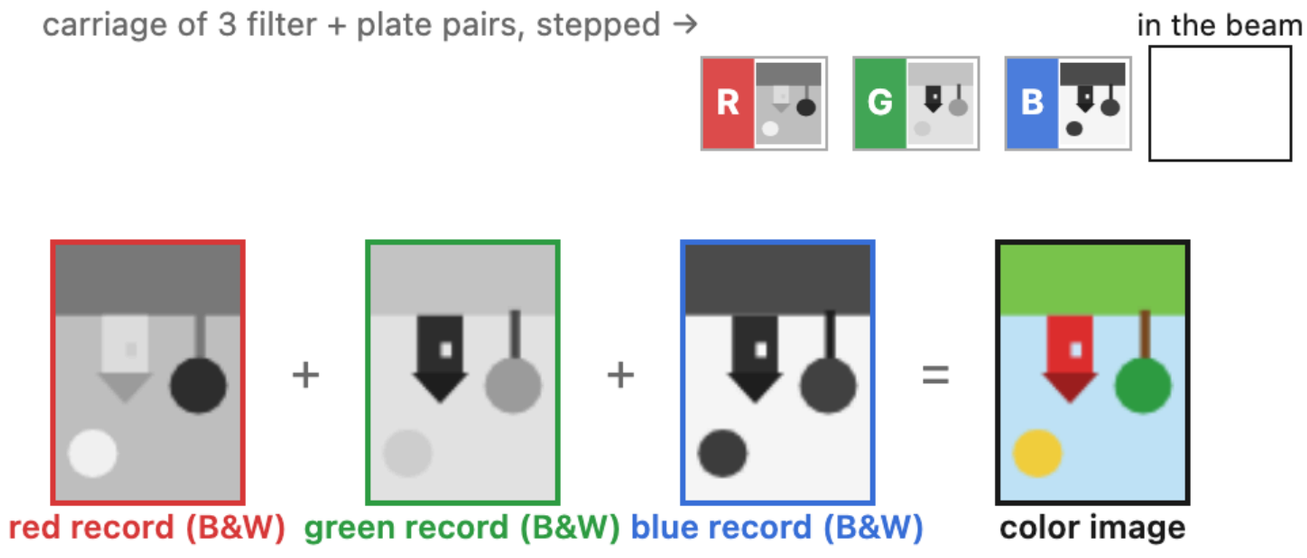
The contact-image-sensor (CIS) bar of a flatbed scanner reads one line of the page at a time; the carriage sweeps it down the page, spending time to build the second…

Maxwell's tartan ribbon, 1861 — the first colour photograph (colour multiplexed in time)



Demo: Temporal multiplex plates





Prokudin-Gorskii

- Early 1900s
- http://en.wikipedia.org/wiki/Prokudin-Gorskii
- Pset 3 for 6.8370

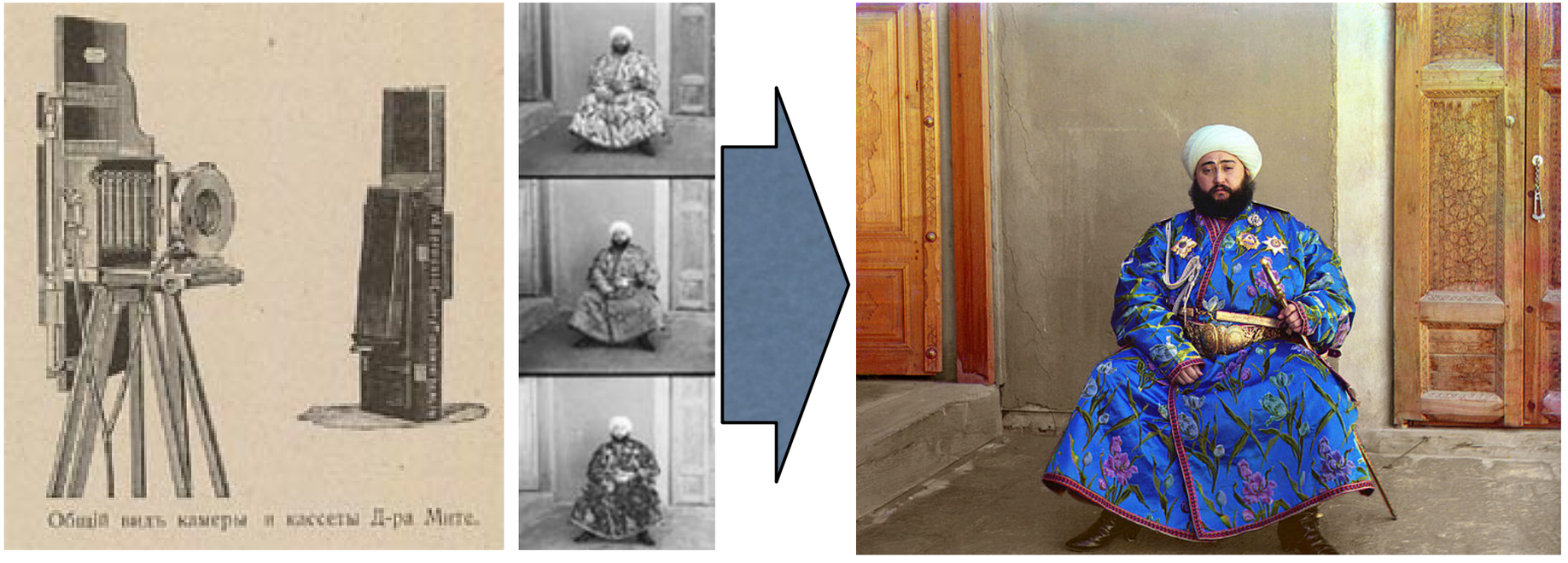
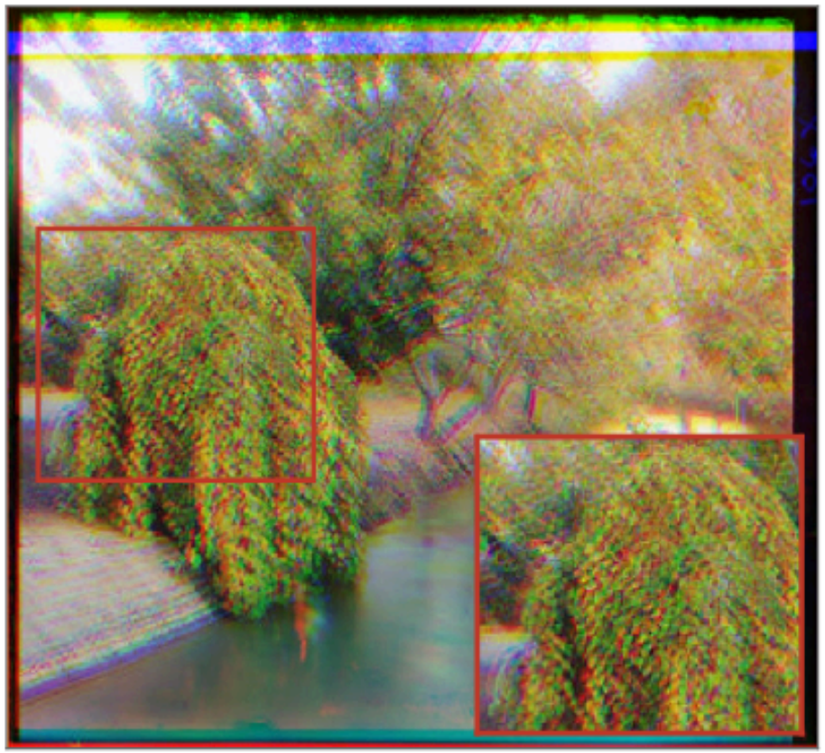


Figure 4.6.5: Recovering color from the three plates: alignment is the catch

Temporal colour multiplexing (Prokudin-Gorskii, c.1910): three exposures, one per filter — then register them

(a) three plates stacked, NOT aligned



channels offset → red / blue fringes, smeared edges

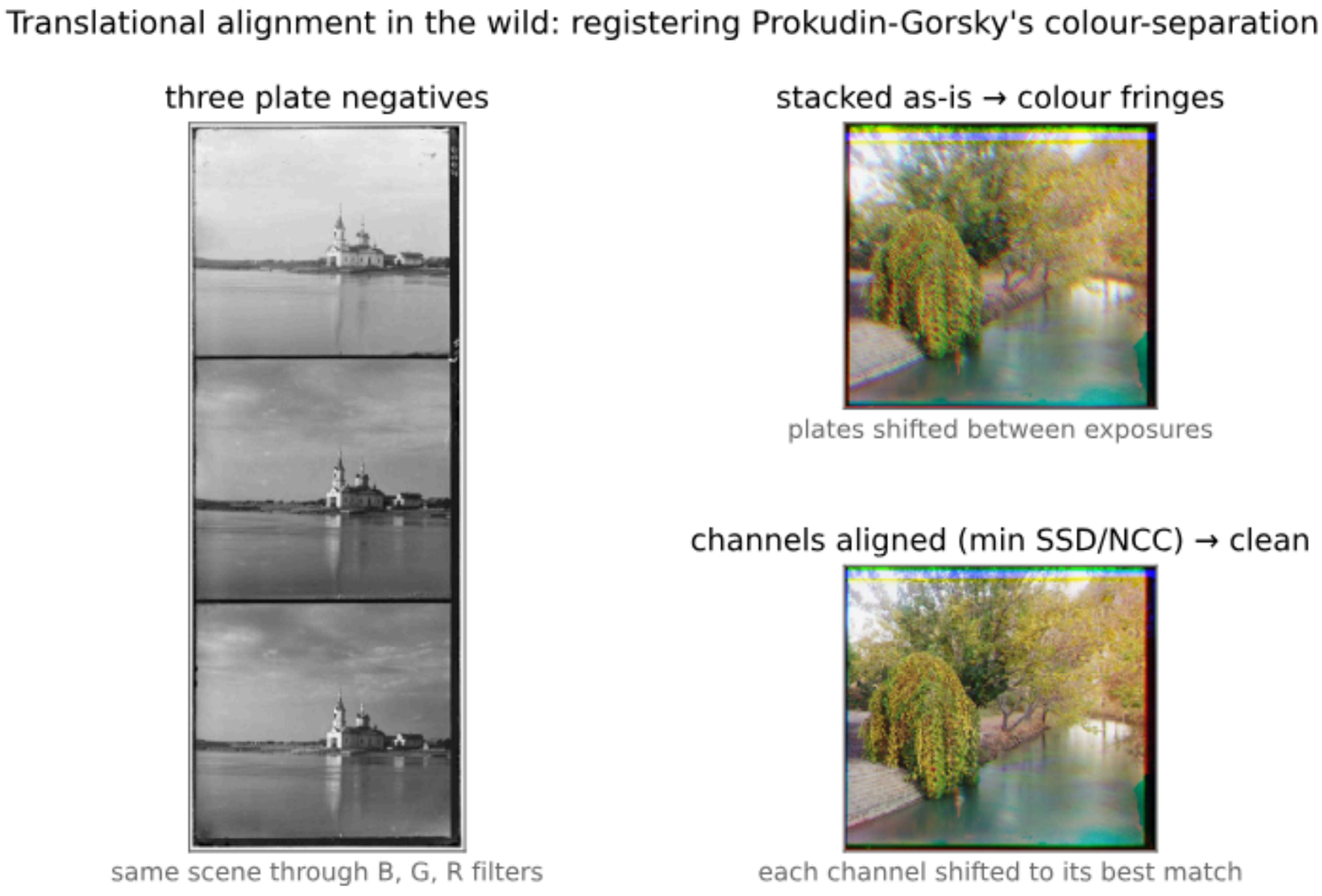
(b) after registering the three plates



one true R, G, B per pixel → sharp full-colour photo

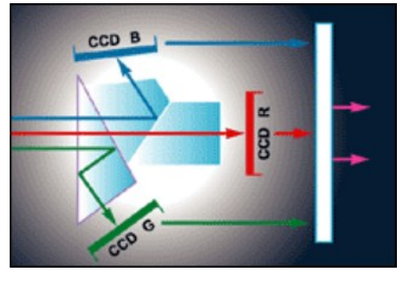
Because the red, green, and blue exposures were shot sequentially, the plates are slightly shifted relative to one another; stacked as-is (left) the color image is…

Figure 12.1.4: The translational aligner, a century early

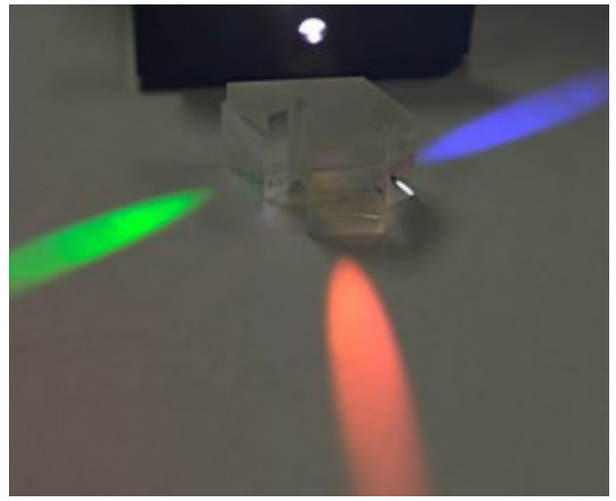


Prokudin-Gorsky's three color-separation plates (blue, green, red), shot in sequence with a shifting camera.

3-chip

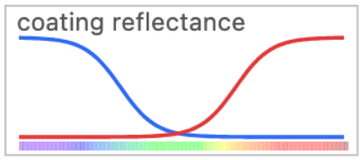


- High-end 3-ccd video cameras
- Use separation prisms
  - prisms that split wavelengths
- Pros
  - 3 real values per pixel
  - Little photon loss
- Cons
  - costly (needs 3 sensors)
  - space

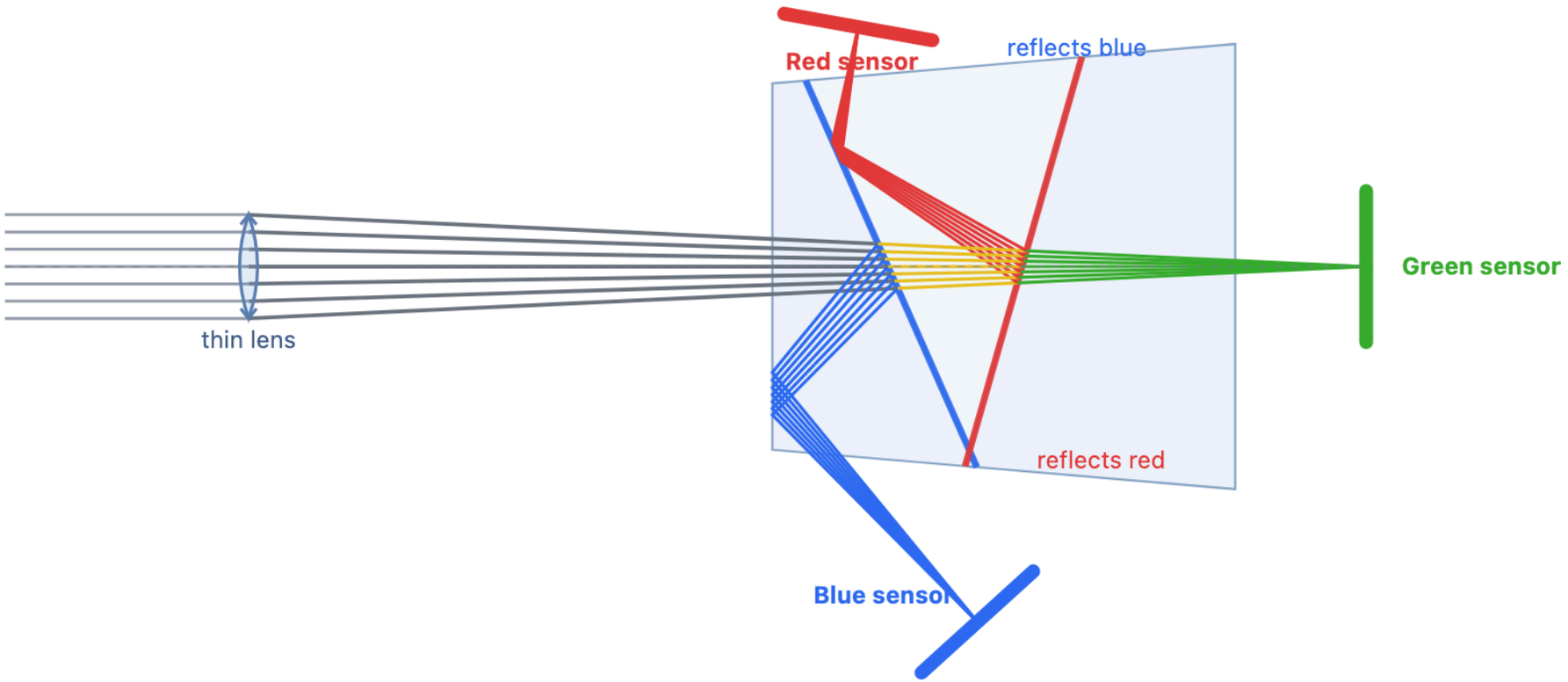


Dicklyon at the English Wikipedia project.

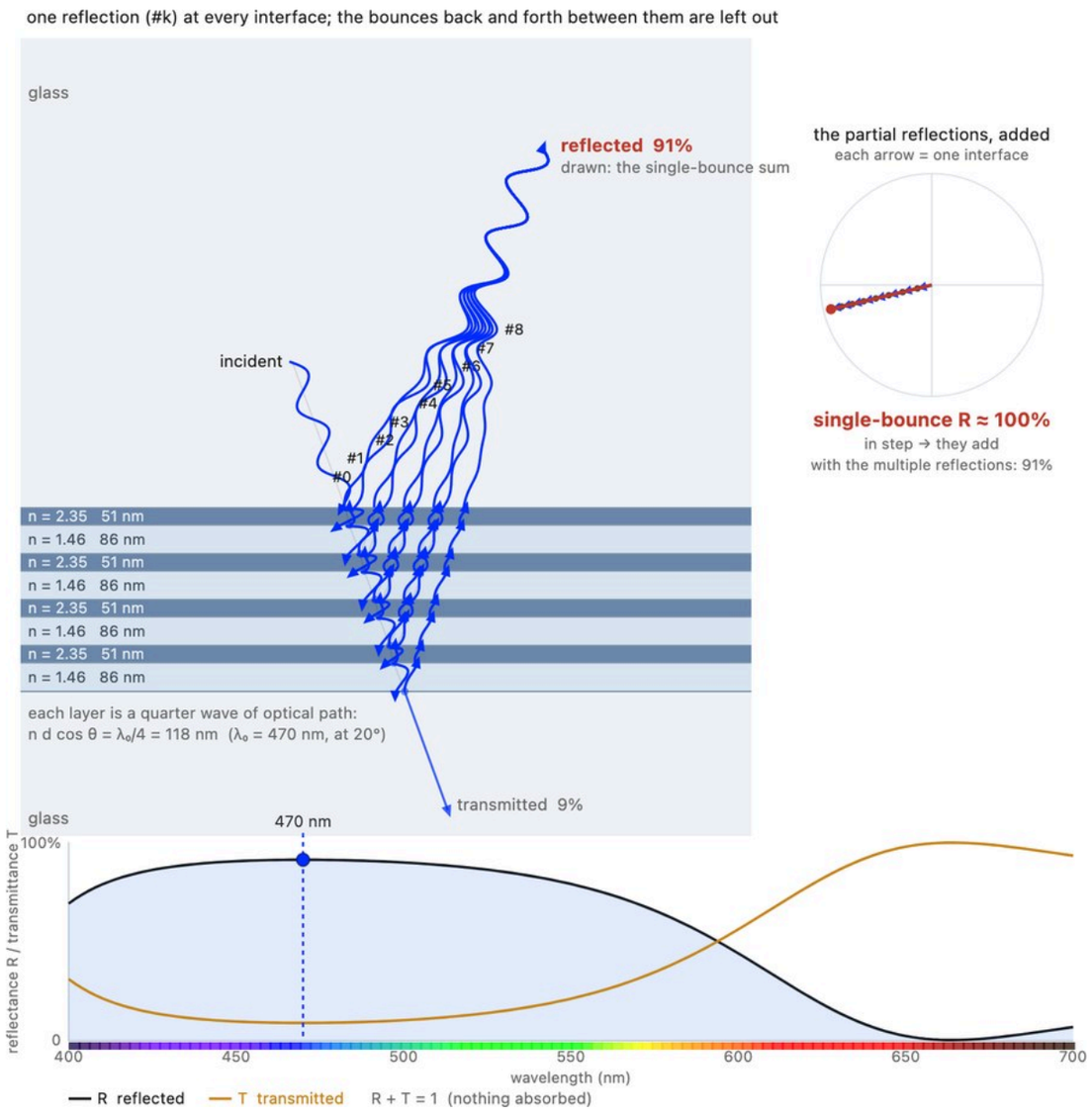
Demo: 3-CCD prism



scene point at ∞ (direction 0.00°)

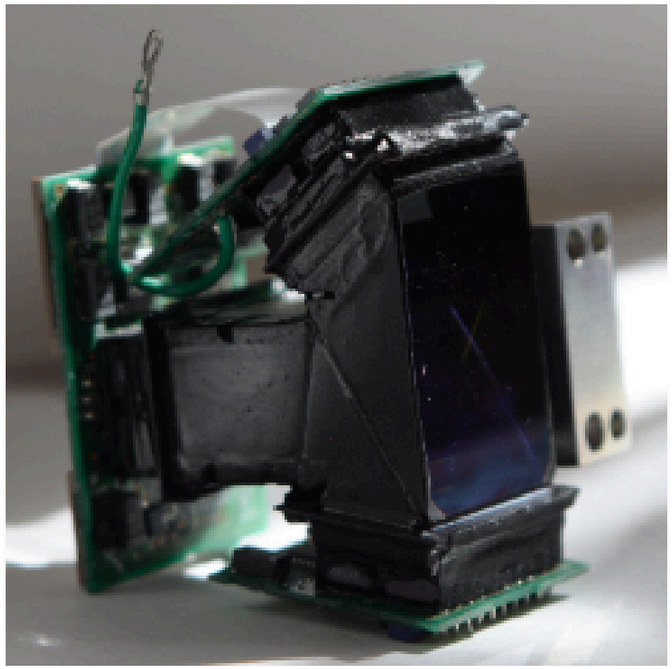


Demo: Inside a dichroic coating

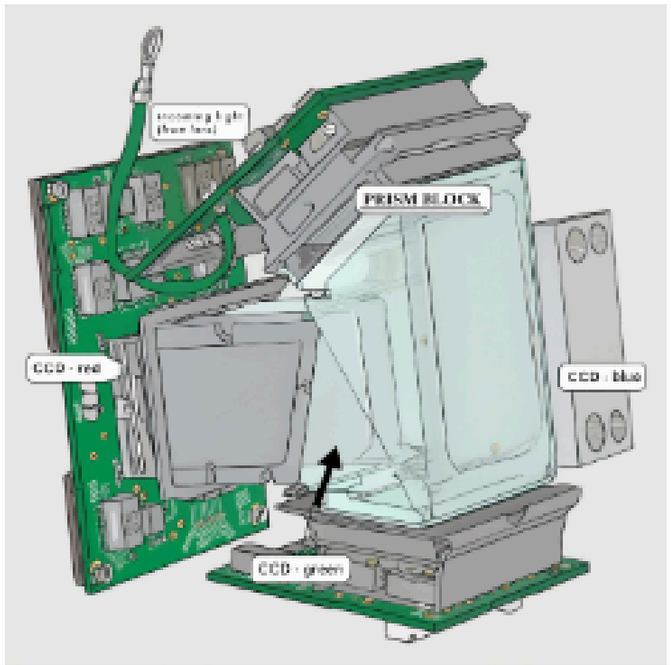


A trichroic (3-CCD) color-separation prism, and a bare prism splitting white light

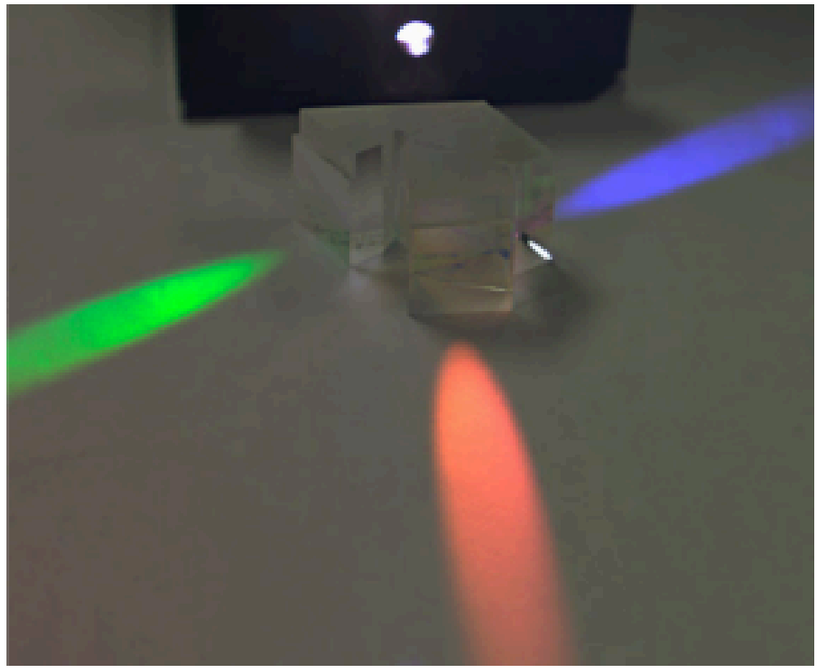
(a) the real 3-CCD prism block

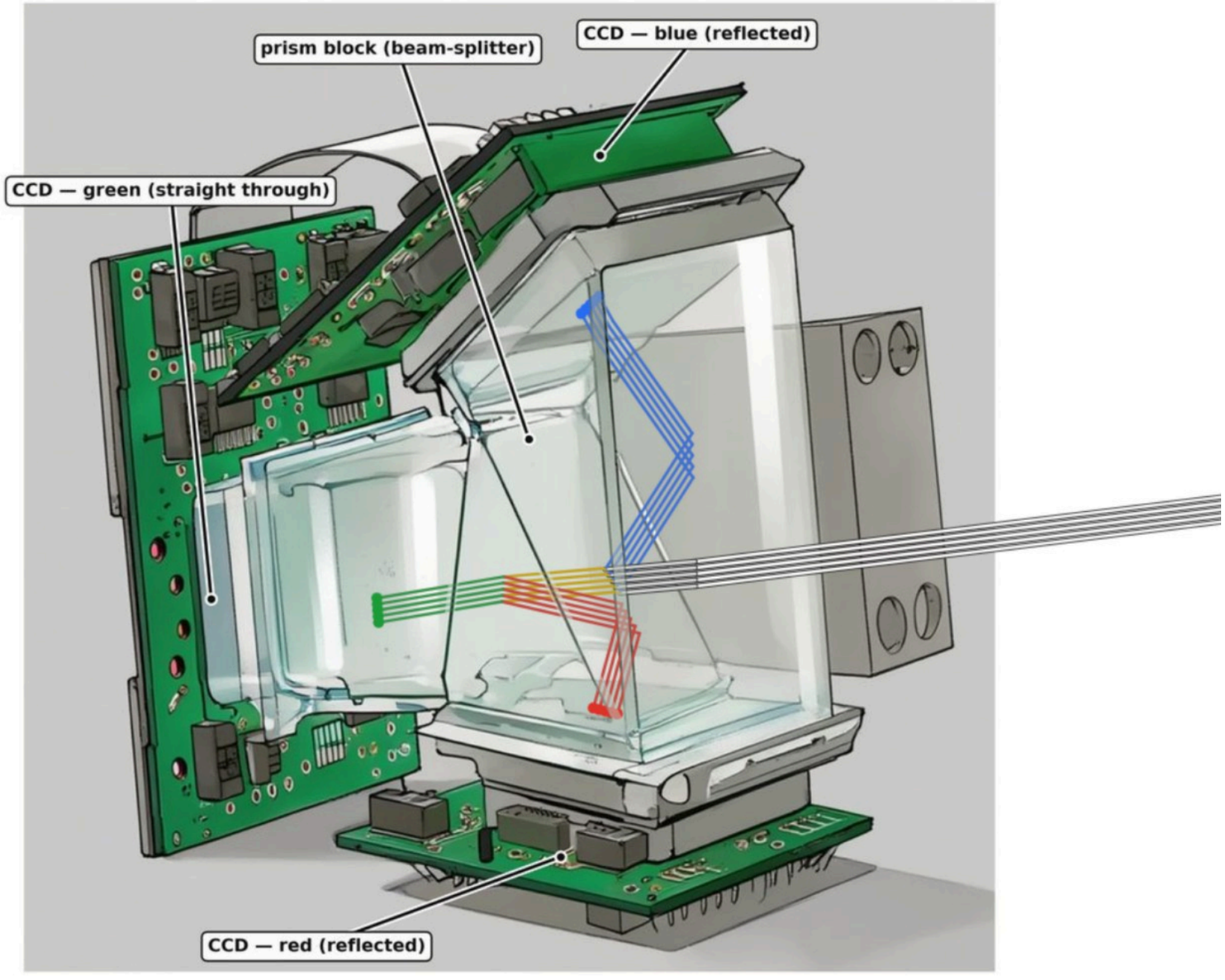


(b) the same block, clarified and labeled



(c) bare prism: white light split into R, G, B




prism block (beam-splitter)
CCD — blue (reflected)
CCD — green (straight through)
CCD — red (reflected)

Brewster's version

- Swiss cheese mirror

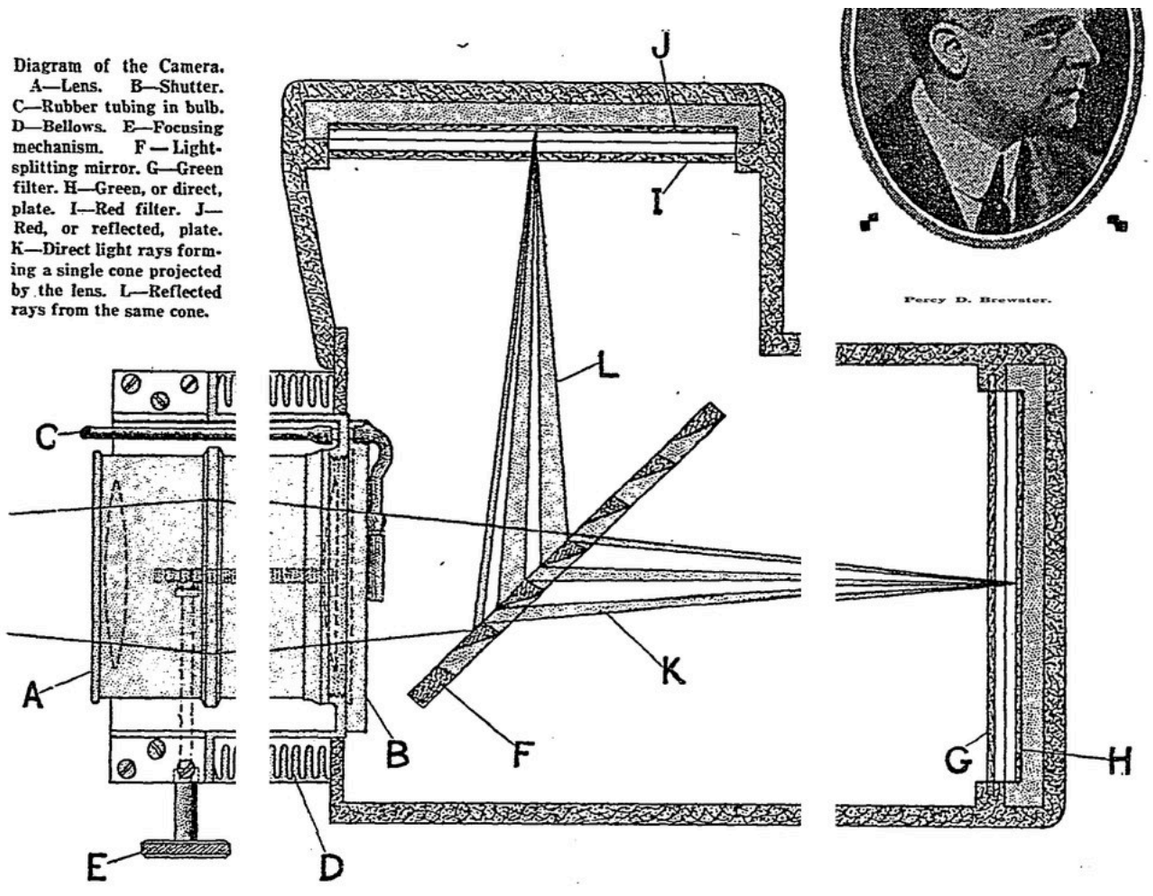
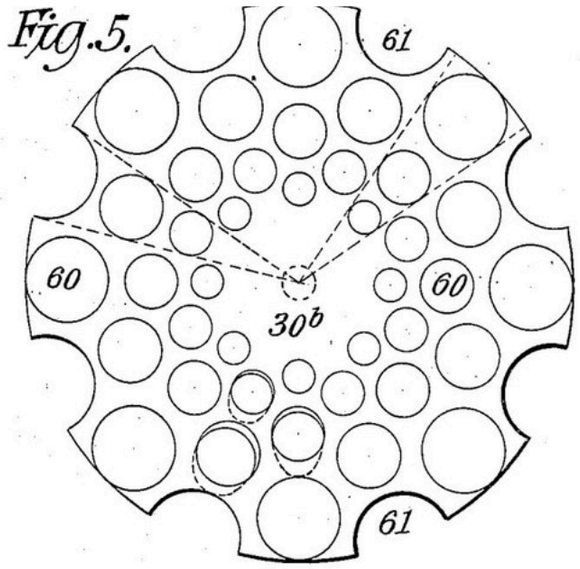


Diagram of the Camera. A—Lens. B—Shutter. C—Rubber tubing in bulb. D—Bellows. E—Focusing mechanism. F—Light-splitting mirror. G—Green filter. H—Green, or direct, plate. I—Red filter. J—Red, or reflected, plate. K—Direct light rays forming a single cone projected by the lens. L—Reflected rays from the same cone.

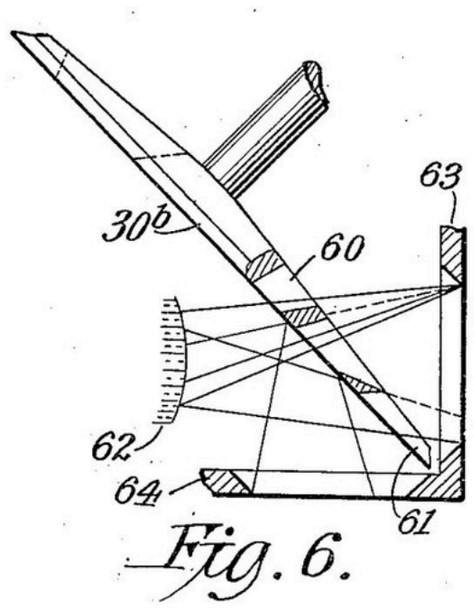
Percy D. Brewster.

Brewster's “cheese-grater” beam splitter: color by a mirror full of holes (1921)

The perforated disc, face-on

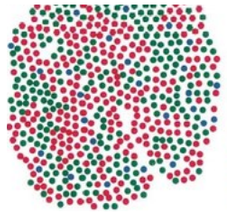


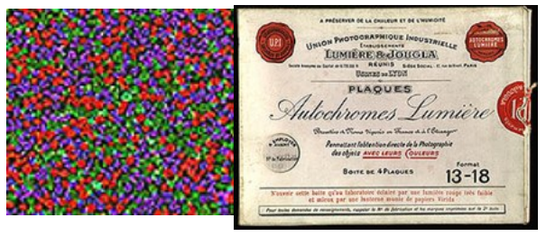
At 45°, splitting the beam to two gates

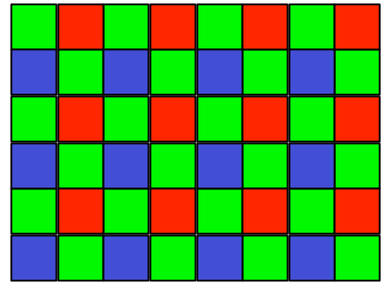


Light hitting the solid metal (61) reflects to one gate; light through the holes (60) passes to the other, no coating needed.

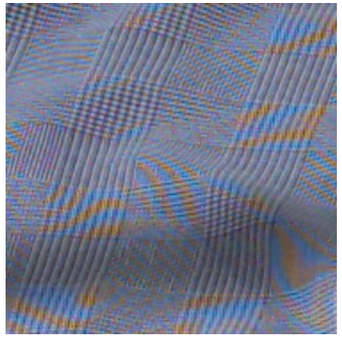
Spatial multiplexing







- Human eye (cone mosaic)
- Older color film (Autochrome)
- Bayer mosaic or CFA (color filter array)
- Most still cameras, most cheap camcorder, some high-end video cameras (e.g. RED)
- Pros
  - single sensor
  - well mastered technology, high resolution
- Cons
  - needs interpolation, color jaggies
  - requires antialiasing filter (reduces sharpness)
  - loss of light

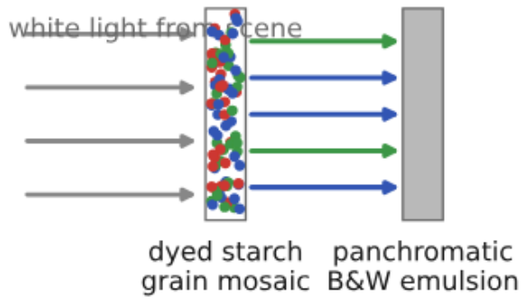


Demo: Autochrome, a color filter array in film

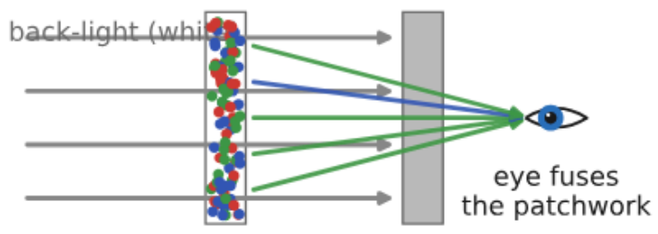
The Lumière Autochrome (1907): a random-tiled colour filter array in film

A additive-screen process: a colour filter array in film

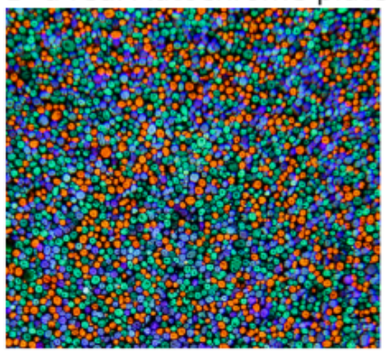
① **Capture**



② **Viewing (through the same screen)**



B a real Autochrome plate (photomicrograph)



C an original box of Lumière plates

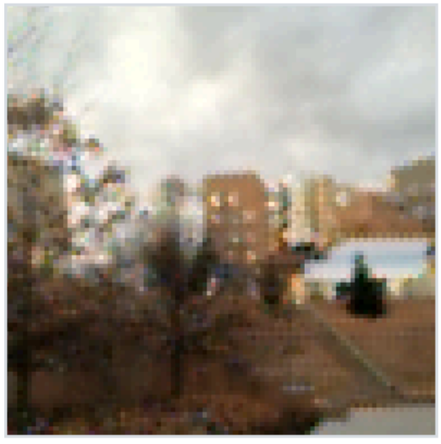
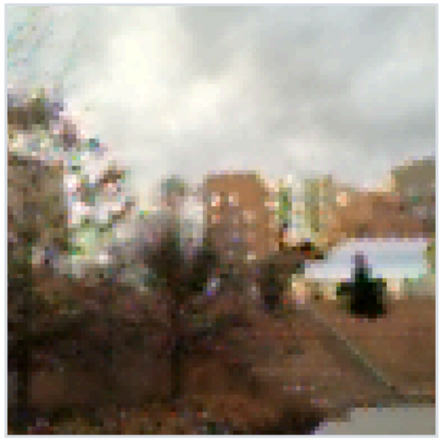
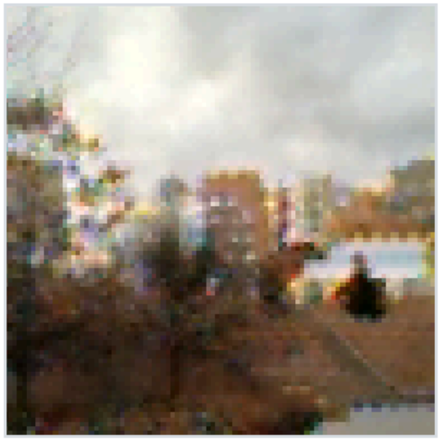
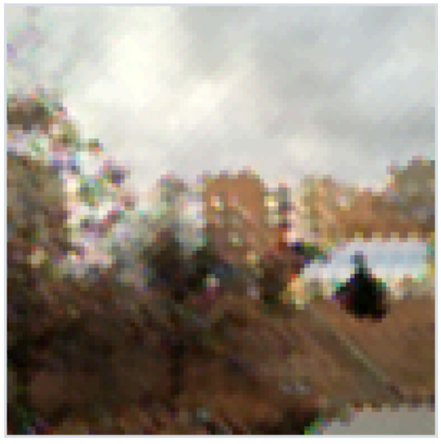


Scattered dyed potato-starch grains over a panchromatic emulsion; each grain is one photosite's colour filter and the eye does the demosaicking, the direct ancestor of the Bayer mosaic. Micrograph: Janke, CC0, via Wikimedia Commons. Box: Anne Jea, CC BY-SA 4.0, via Wikimedia Commons.

Demo: Bayer, X-Trans, Quad-Bayer and RGBW compared

four colour filter arrays, one scene

no optical low-pass filter

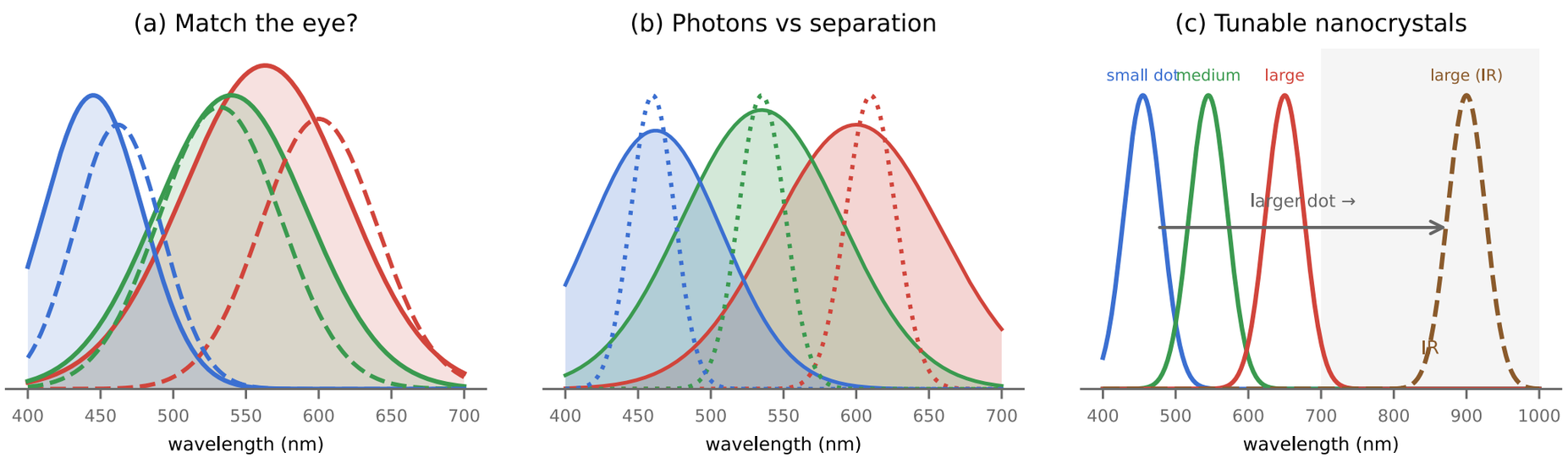
Bayer	X-Trans	Quad-Bayer	RGBW
			
31.34 dB	31.05 dB	28.33 dB	27.25 dB

Bayer has the highest PSNR with this interpolator, by 0.28 dB.

Generic interpolation, not an optimized camera comparison. RGBW reconstruction ignores the white samples.

Encoded RGB toy mosaics; no sensor-noise or low-light sensitivity model. Pattern-specific algorithms can change the ranking.

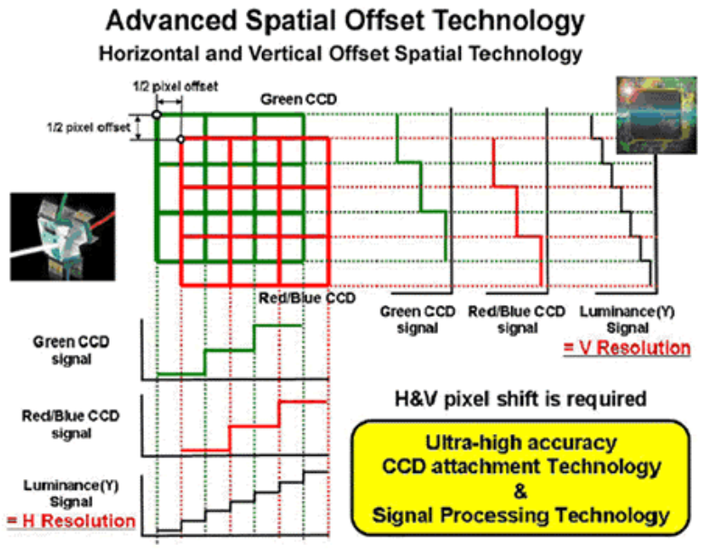
Figure 4.6.24: The spectral-sensitivity trilemma



(a) Match the eye: the human cone fundamentals (smooth, heavily overlapping) beside a real sensor's RGB, the mismatch (the sensor is not a linear transform of the cones…

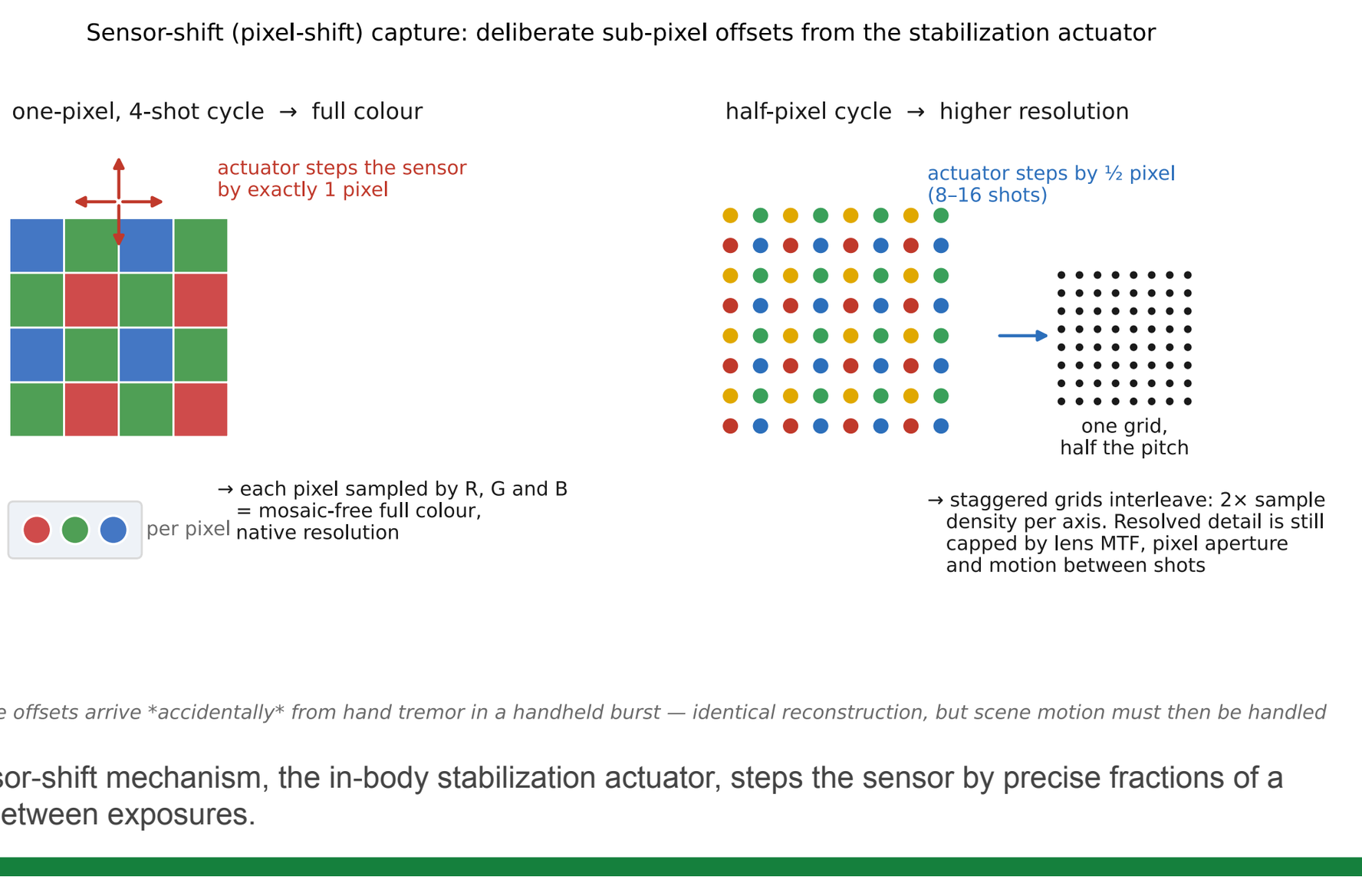
Combination: pixel shift

- 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

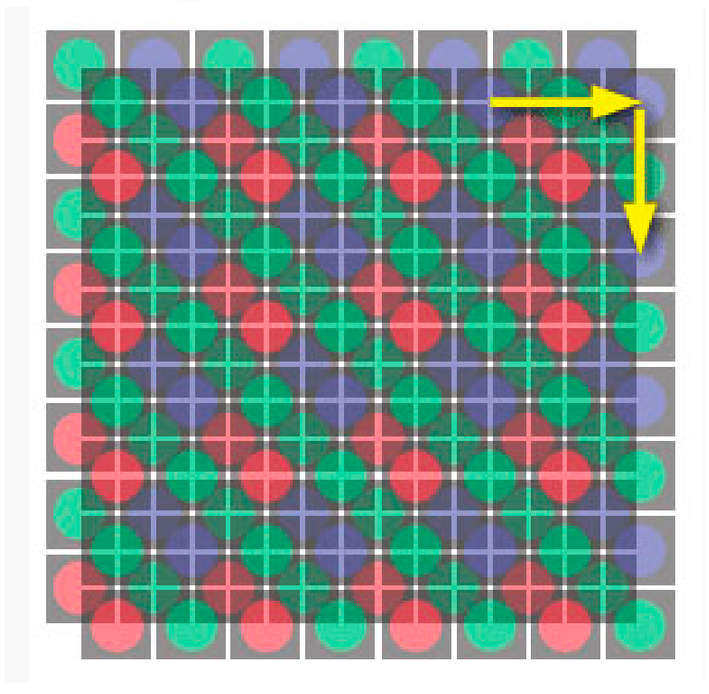
Figure 15.10.1: Manufacturing the offsets



*the same offsets arrive *accidentally* from hand tremor in a handheld burst — identical reconstruction, but scene motion must then be handled*

A sensor-shift mechanism, the in-body stabilization actuator, steps the sensor by precise fractions of a pixel between exposures.

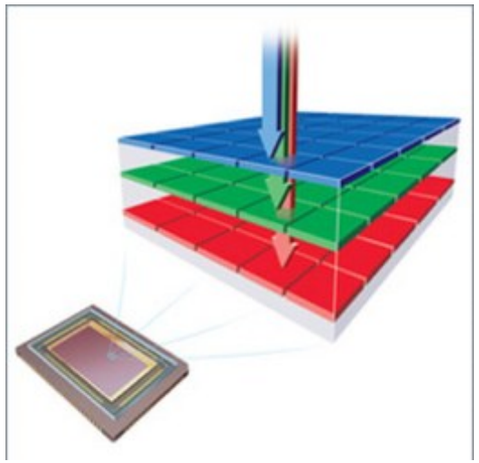
- **http://www.petapixel.com/2011/05/26/hasselblad-h4d-200ms-shoots-200mp-photos-with-a-50mp-sensor/**

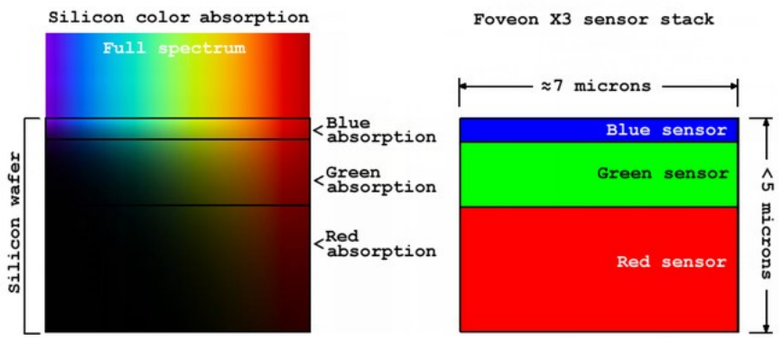




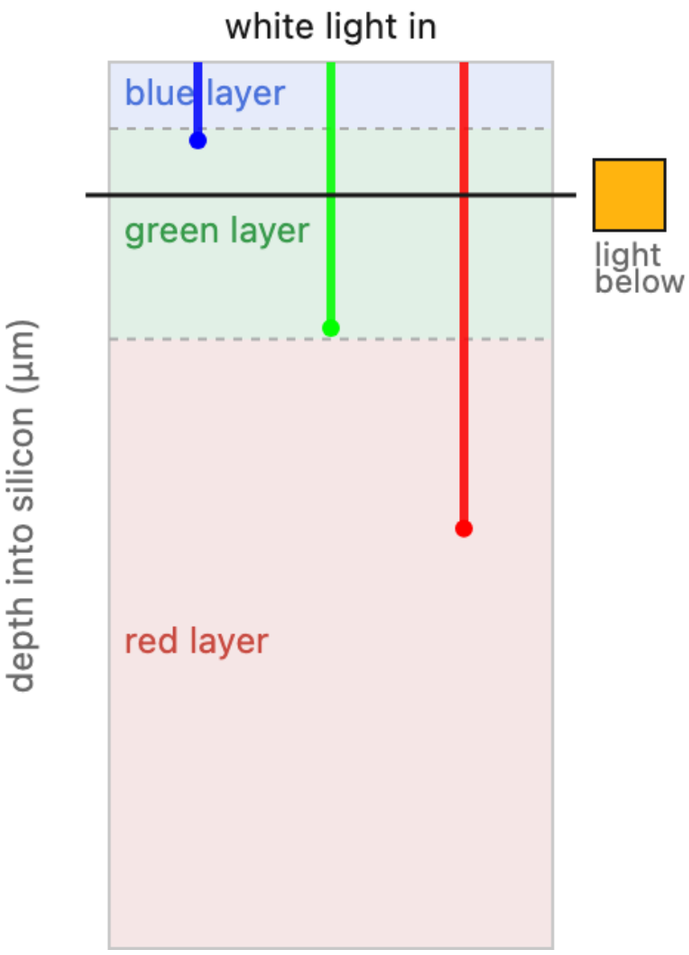
Depth multiplexing (Foveon X3 sensor)

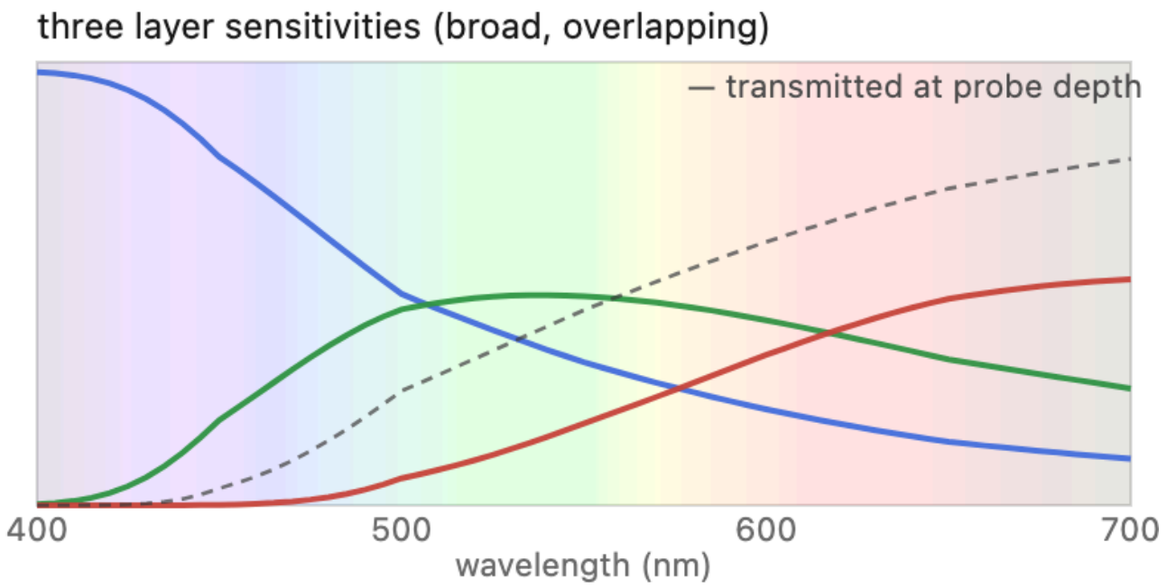
- Leverage difference in absorption per wavelength
- Pros
  - 3 real numbers per pixel
  - Less light loss
- Cons
  - Requires more color processing (3 numbers must be multiplied by matrix to get RGB)





Demo: Foveon depth stack





un-mixing matrix $[R\ G\ B]^T = M \cdot [\text{layer signals}]^T$:

```
R = [  0.83  −2.50   4.93 ]
G = [ −1.69   5.01  −4.54 ]
B = [  1.87  −1.54   0.66 ]
```

Figure 4.6.16: A camera built around a stacked-photodiode (Foveon) sensor: the Sigma…

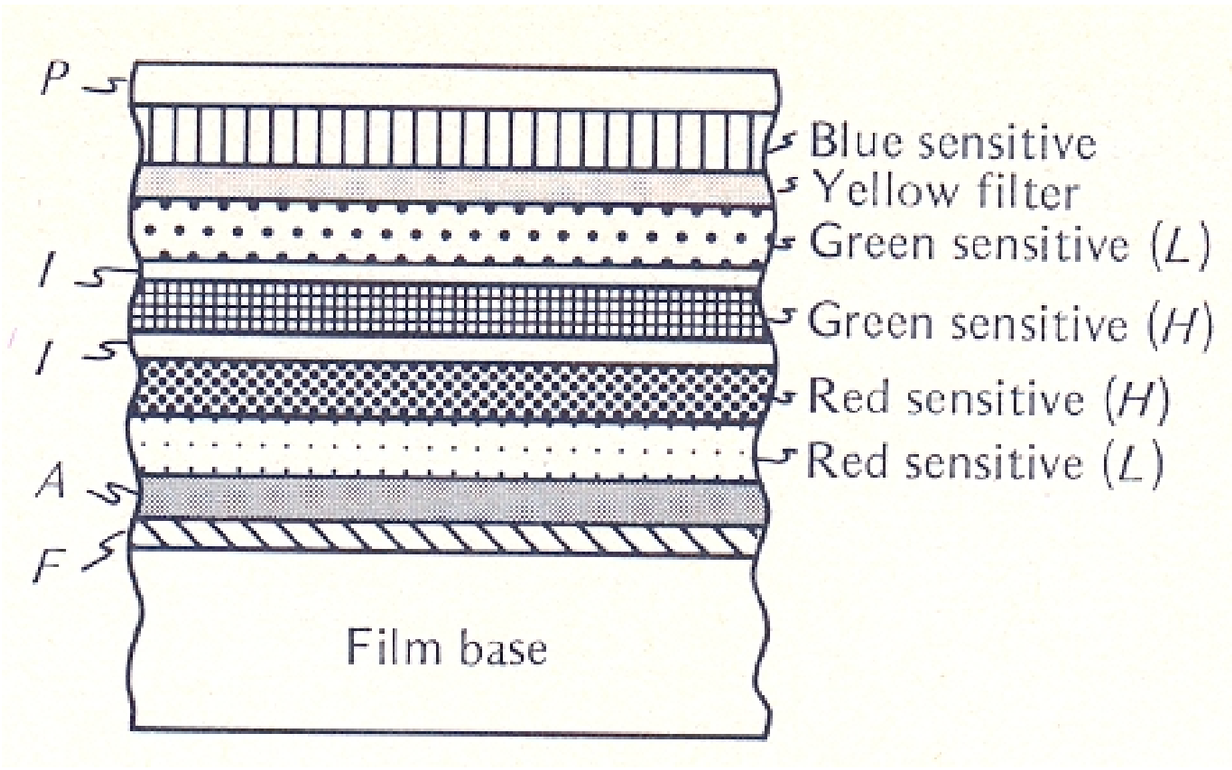
A stacked-photodiode (Foveon) sensor camera



A camera built around a stacked-photodiode (Foveon) sensor: the Sigma dp2.

Depth multiplexing

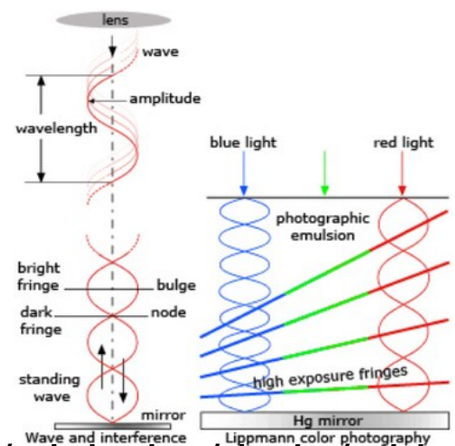
- Good old color film (tripack)



Interferences (Lippmann process)

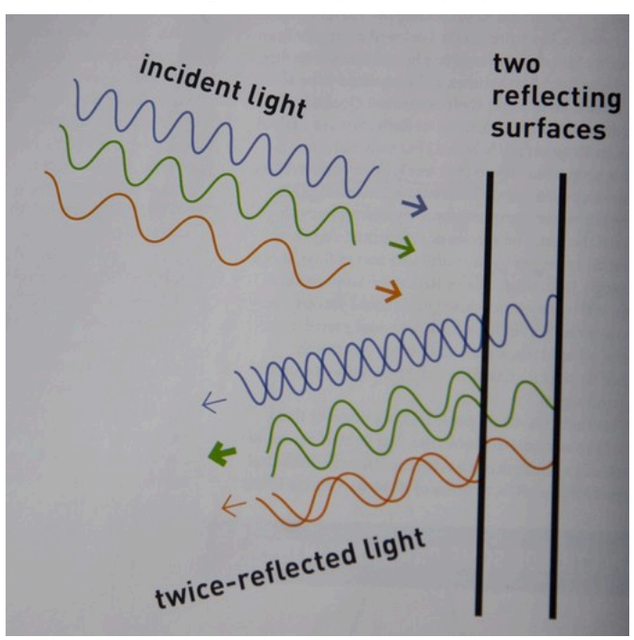


- Metal mirror to create interferences
  - ancestor of holography
  - similar to colors in thin oil film



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

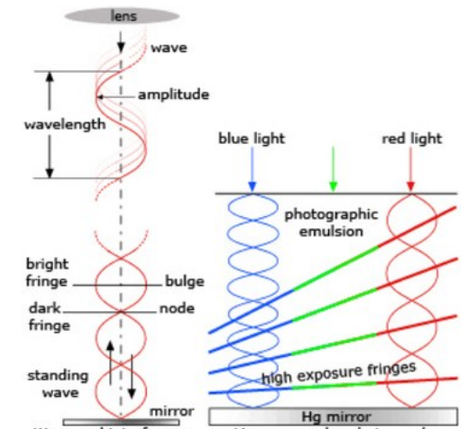




Interferences (Lippmann process)



- 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



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



'Saint-Maxime", 1891-1899
Photographer: Gabriel Lippmann (1845-1929)

Demo: Lippmann standing wave

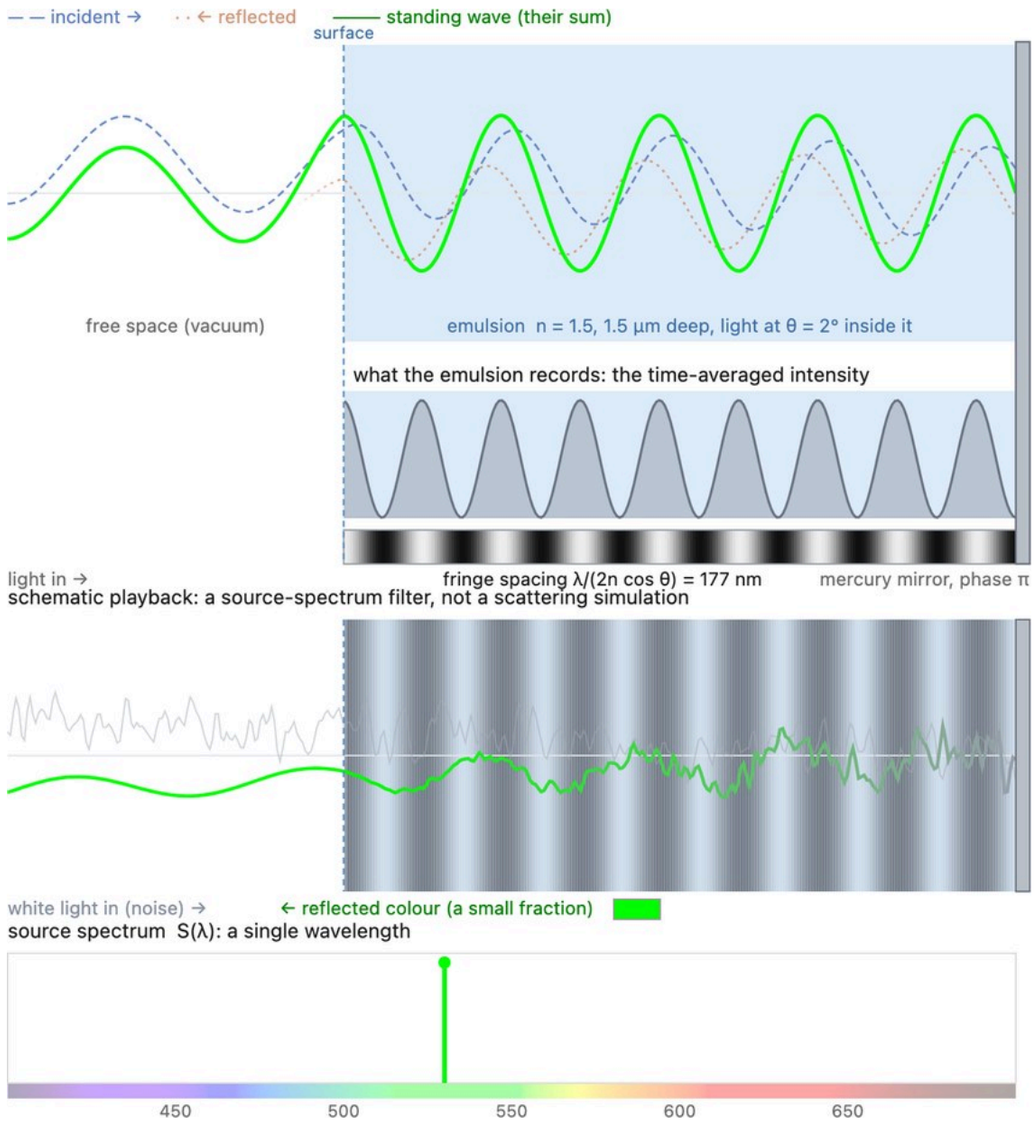
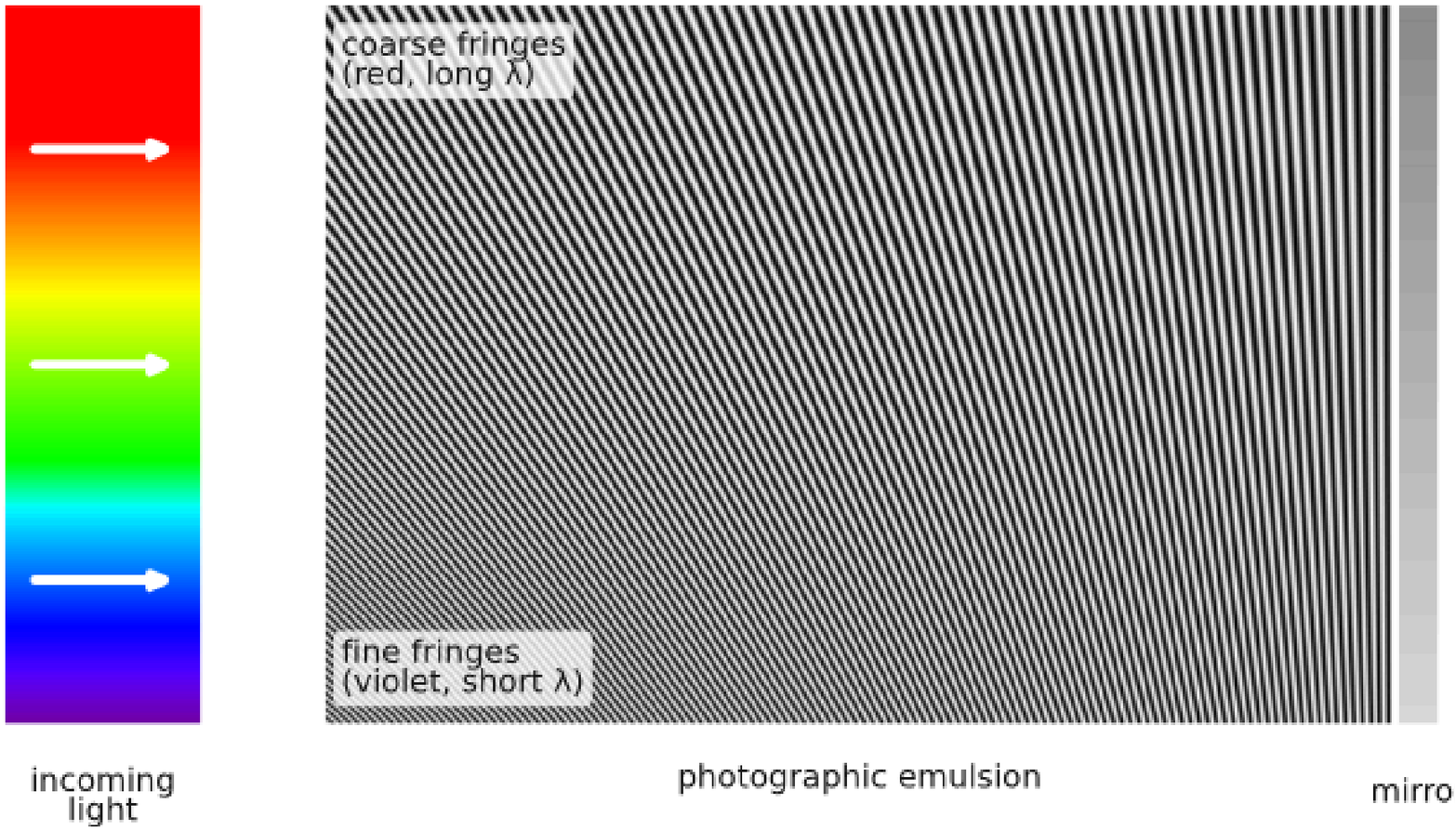


Figure 4.6.20: Lippmann interference color photography



Recording (top-left): light enters a transparent emulsion backed by a liquid-mercury mirror; the incident and reflected waves form a standing wave whose antinodes expose…

A whole spectrum recorded by a Lippmann plate: each colour its own fringe spacing

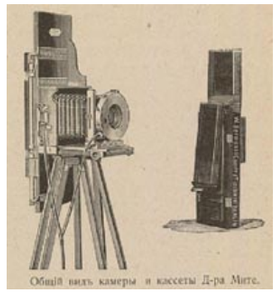


Video: Steve Mold demonstrates interference color photography

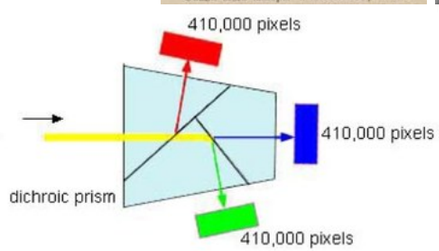


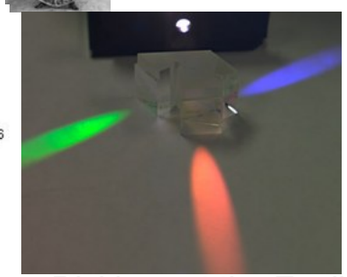
Recap & Questions?

- Scan 3 times (temporal multiplexing)
- Use 3 detectors (3-ccd camera)
- Use offset color samples (spatial multiplexing)
- Multiplex in depth (Tripack film, Foveon)

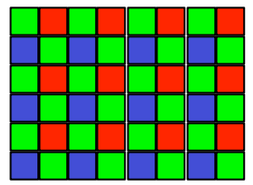




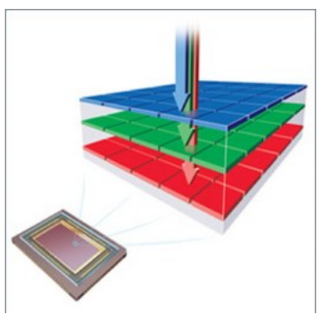




Dicklyon at the English Wikipedia project.







Links

- What we do essentially follows **https://patents.google.com/patent/US5382976A/en** , Inventor Robert H. Hibbard (Kodak)

- **http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.634&rep=rep1&type=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**

Speaker notes

More

- **http://www.ists.dartmouth.edu/library/edf0205.pdf**
- **http://ieeexplore.ieee.org/iel5/79/30519/01407714.pdf?isnumber=30519&prod=JNL&arnumber=1407714&arSt=+44&ared=+54&arAuthor=Gunturk%2C+B.K.%3B+Glotzbach%2C+J.%3B+Altunbasak%2C+Y.%3B+Schafer%2C+R.W.%3B+Mersereau%2C+R.M**.