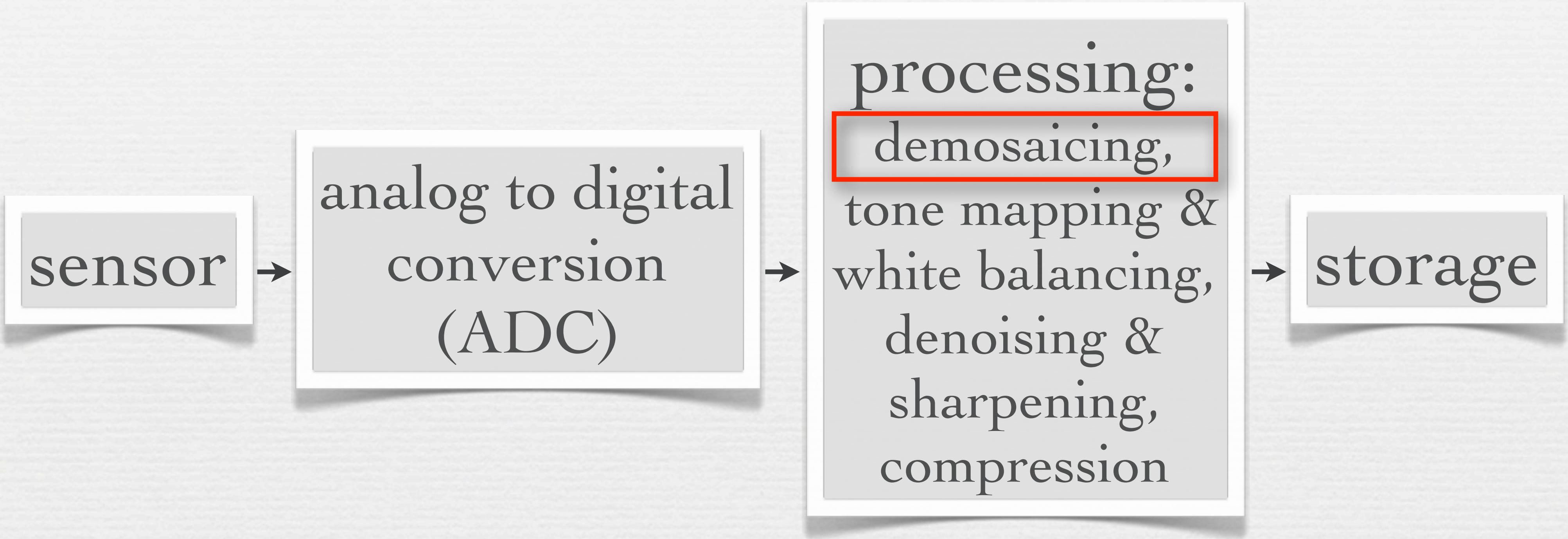
Post-processing pipeline

CS 178, Spring 2014

Begun 5/29/14, finished 6/3.

Marc Levoy
Computer Science Department
Stanford University

Camera pixel pipeline

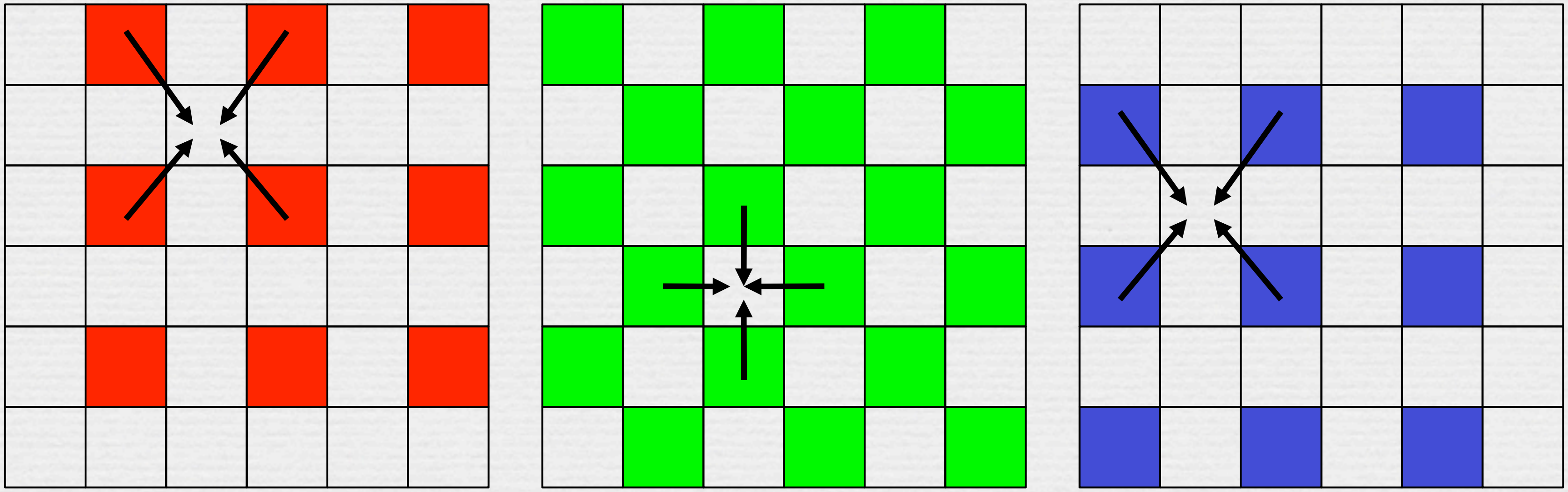


sensor → analog to digital conversion (ADC) → processing: demosaicing, tone mapping & white balancing, denoising & sharpening, compression → storage

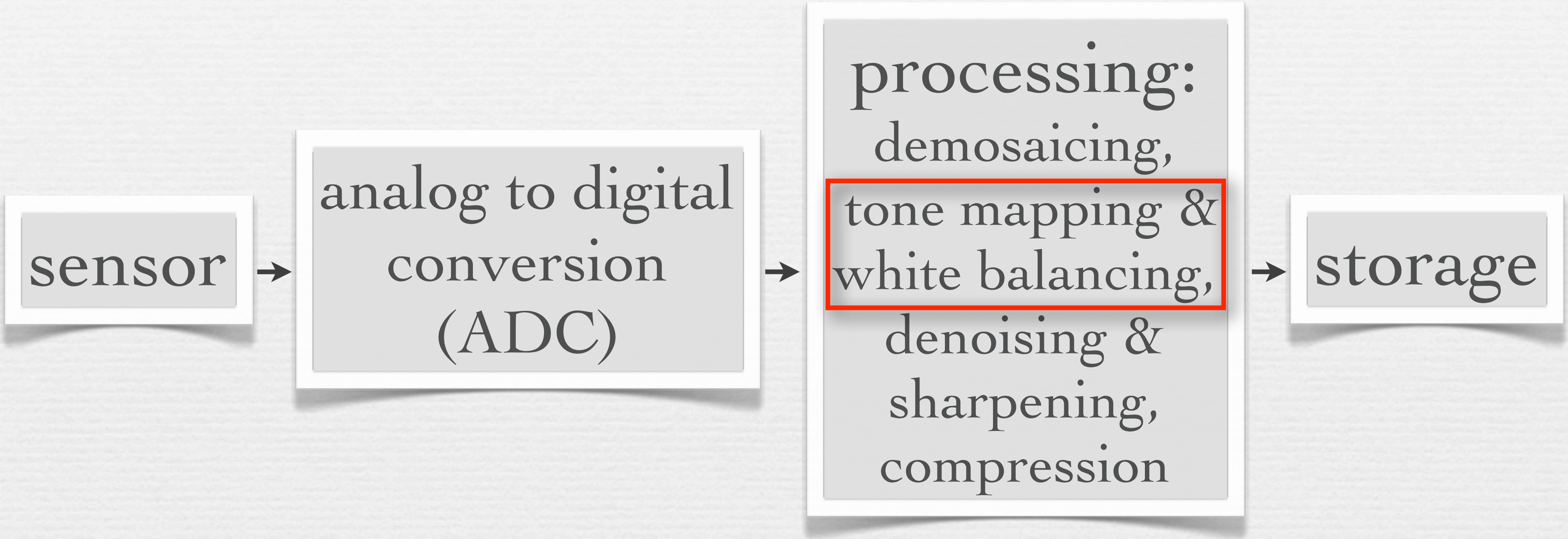
- ✦ every camera uses different algorithms
- ✦ the processing order may vary
- ✦ most of it is proprietary

Demosaicing (review)

- linear interpolation
  - average of the 4 nearest neighbors of the same color
- cameras typically use more complicated scheme
  - try to avoid interpolating across feature boundaries
  - demosaicing is often combined with denoising, sharpening...



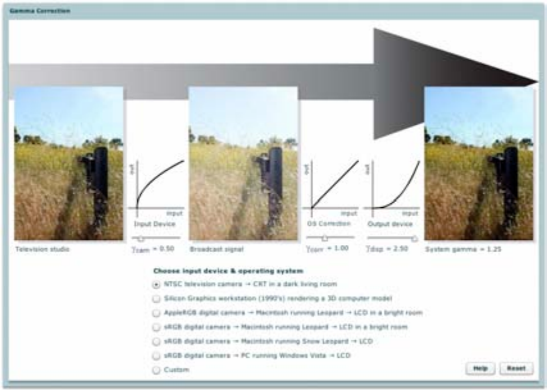
Camera pixel pipeline



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

Gamma and gamma correction

- the goal of digital imaging is to accurately reproduce <u>relative</u> scene luminances on a display screen
  - absolute luminance is impossible to reproduce
  - humans are sensitive to relative luminance anyway
- in some workflows, pixel value is made proportional to scene luminance, in other systems to perceived brightness
  - in CRTs luminance was proportional to voltage$^\gamma$ with $\gamma \approx 2.5$, so TV cameras were designed to output voltage $\propto$ scene luminance$^{1/\gamma}$
  - pixel value $\propto$ luminance$^{1/2.5}$ is roughly perceptually uniform, so it's a good space for quantization, for example in JPEG files



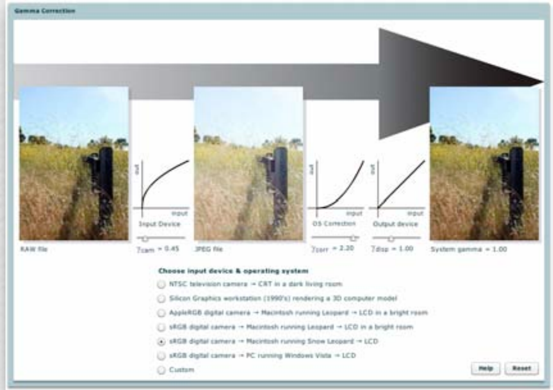
Television studio | γ_{cam} = 0.50 | Broadcast signal | γ_{corr} = 1.00 | γ_{disp} = 2.50 | System gamma = 1.25

Choose input device & operating system

● NTSC television camera → CRT in a dark living room
○ Silicon Graphics workstation (1990's) rendering a 3D computer model
○ AppleRGB digital camera → Macintosh running Leopard → LCD in a bright room
○ sRGB digital camera → Macintosh running Leopard → LCD in a bright room
○ sRGB digital camera → Macintosh running Snow Leopard → LCD
○ sRGB digital camera → PC running Windows Vista → LCD
○ Custom

<u>(FLASH DEMO)</u>

<u>http://graphics.stanford.edu/courses/cs178/applets/gamma.html</u>



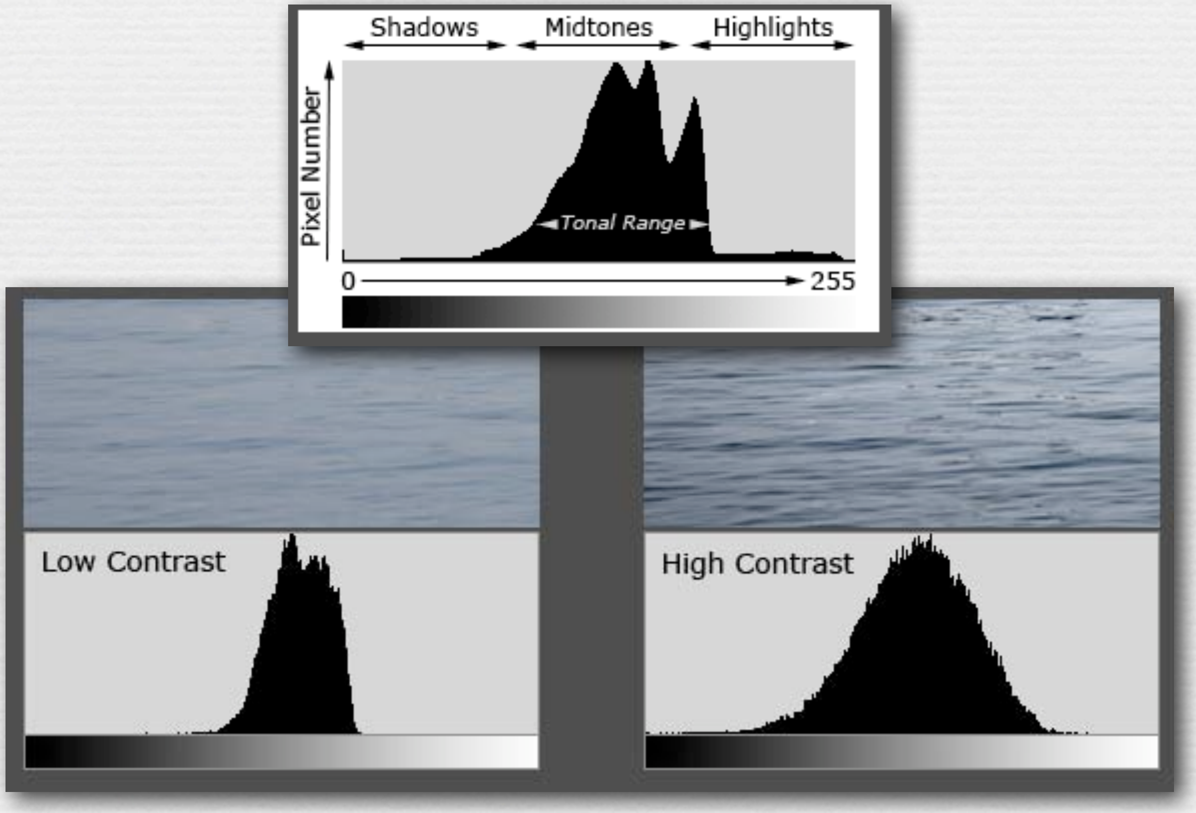
RAW file | γ_{cam} = 0.45 | JPEG file | γ_{corr} = 2.20 | γ_{disp} = 1.00 | System gamma = 1.00

Choose input device & operating system

○ NTSC television camera → CRT in a dark living room
○ Silicon Graphics workstation (1990's) rendering a 3D computer model
○ AppleRGB digital camera → Macintosh running Leopard → LCD in a bright room
○ sRGB digital camera → Macintosh running Leopard → LCD in a bright room
● sRGB digital camera → Macintosh running Snow Leopard → LCD
○ sRGB digital camera → PC running Windows Vista → LCD
○ Custom

Contrast correction (a.k.a. tone mapping)

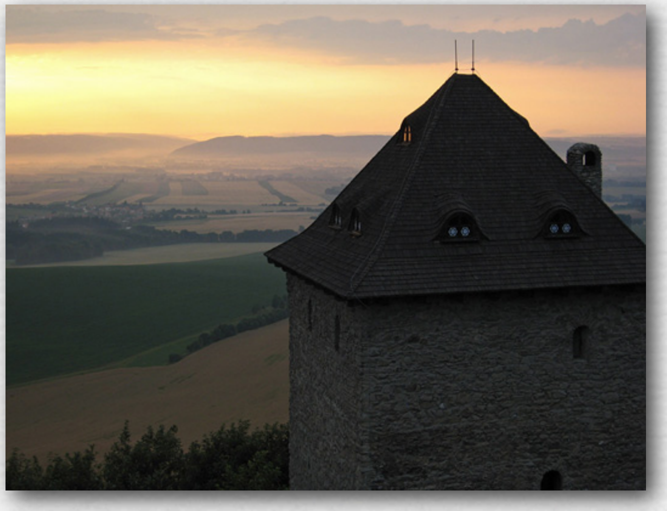
- manual editing
  - capture image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.
  - to expand contrast, apply an S-curve to pixel values



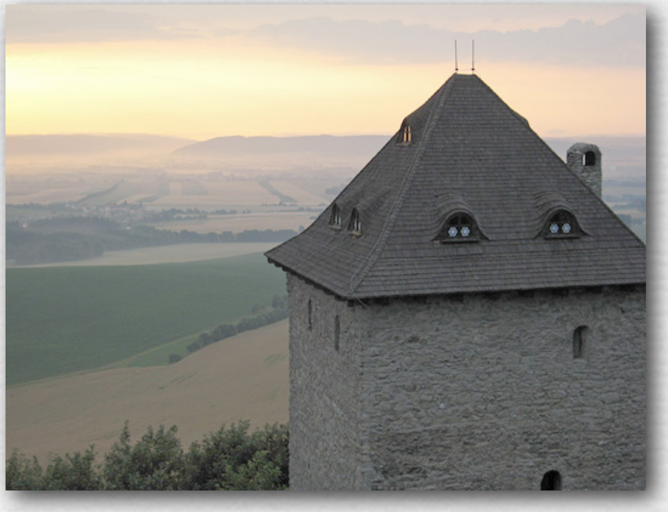
(cambridgeincoulour.com)

Contrast correction (a.k.a. tone mapping)

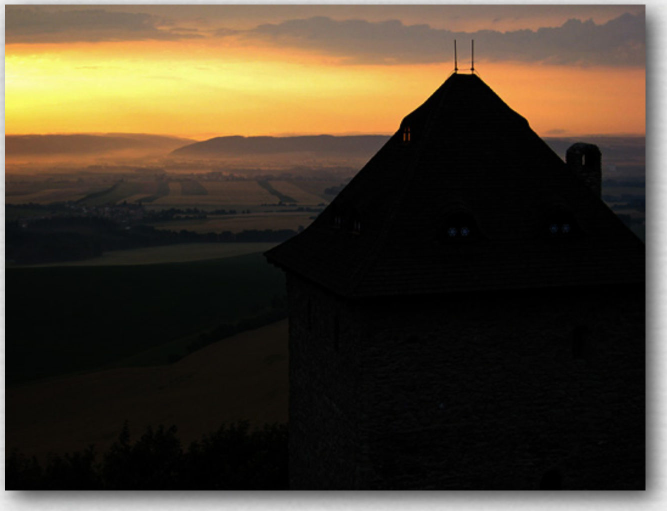
- manual editing
  - capture image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.
  - to expand contrast, apply an S-curve to pixel values
- gamma transform (in addition to RAW→JPEG gamma)
  - output = input$^{\gamma}$ (for $0 \leq I_i \leq 1$)
  - simple but crude



original



$\gamma = 0.5$

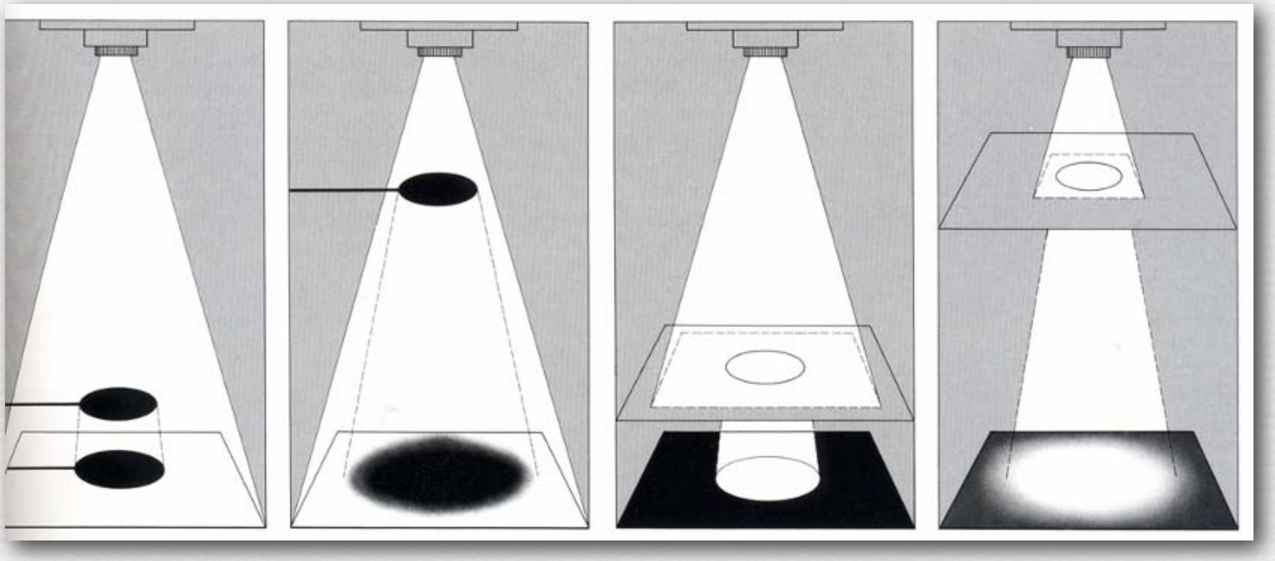


$\gamma = 2.0$

Contrast correction (a.k.a. tone mapping)

- manual editing
  - capture image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.
  - to expand contrast, apply an S-curve to pixel values
- gamma transform (in addition to RAW→JPEG gamma)
  - output = input$^\gamma$ (for $0 \le I_i \le 1$)
  - simple but crude
- global versus local transformations

Traditional dodging and burning

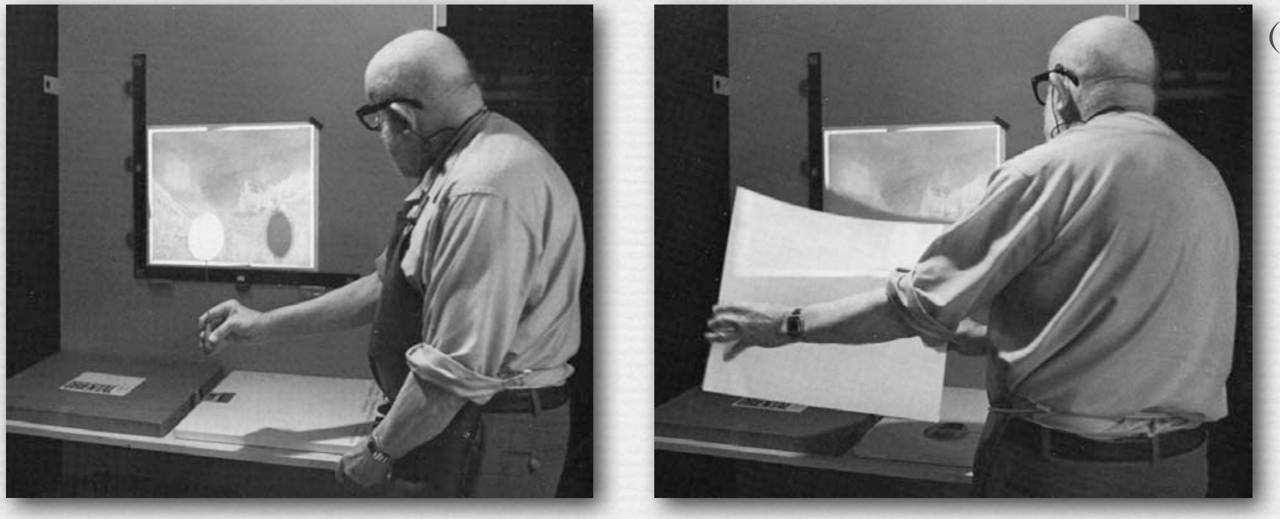


(Rudman)

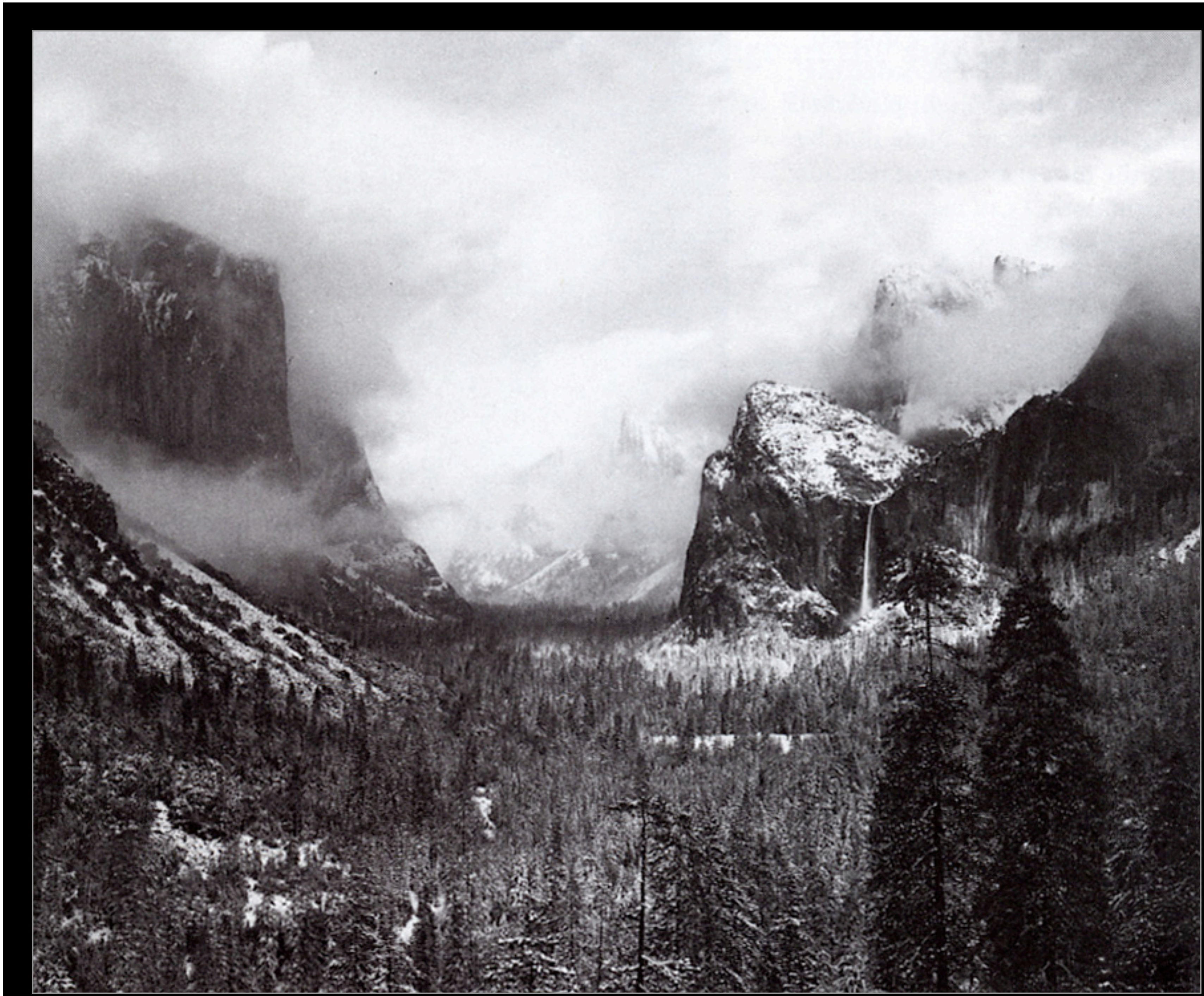
dodging
(leaves print lighter)

burning
(makes print darker)

Ansel Adams in
his darkroom

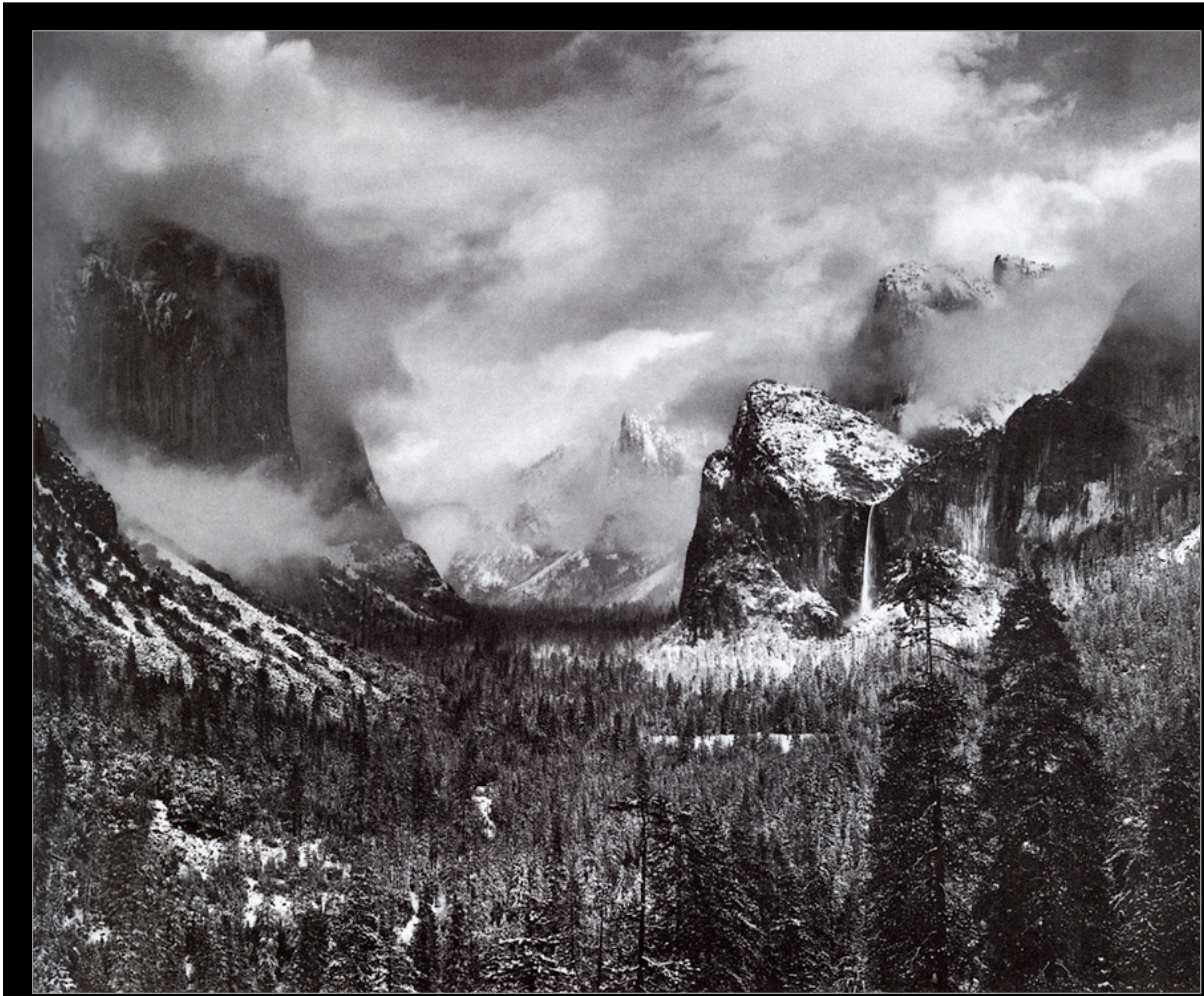


(Adams)



straight
print

Ansel Adams, Clearing Winter Storm, 1942



toned
print

Ansel Adams, Clearing Winter Storm, 1942

Recap

- in CRTs luminance = voltage$^{\gamma}$ where $\gamma \approx 2.5$, so television cameras output luminance$^{1/\gamma}$ to compensate
  - NTSC cameras use luminance$^{0.5}$, yielding a *system gamma*, to compensate for human *dark adaptation* during viewing
- digital cameras also gamma transform sensed pixels before storing them in JPEG files
  - while this matches television cameras, another good reason is perceptual uniformity, thereby reducing quantization artifacts
  - for sRGB cameras, $\gamma = 1/2.2$
- *tone mapping* methods may include
  - contrast expansion
  - additional gamma mapping
  - local methods, like dodging & burning

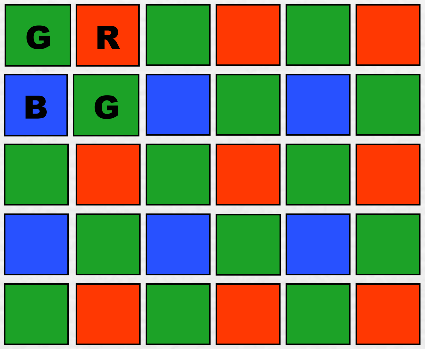
Questions?

High dynamic range (HDR) imaging

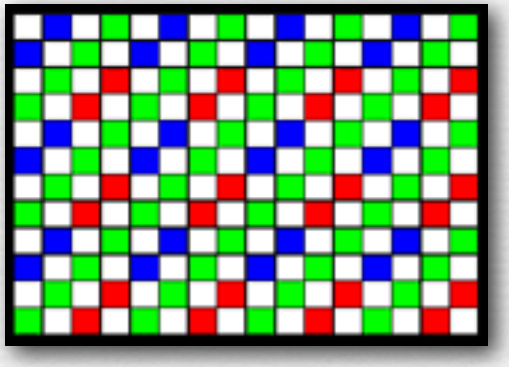
- step 1: capturing HDR images
- step 2a: direct display of HDR images, or
- step 2b: tone mapping to create an LDR image for display

Capturing HDR images

- alternative color filter arrays



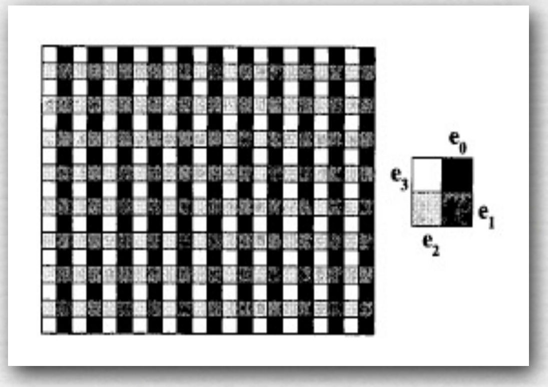
Bayer pattern



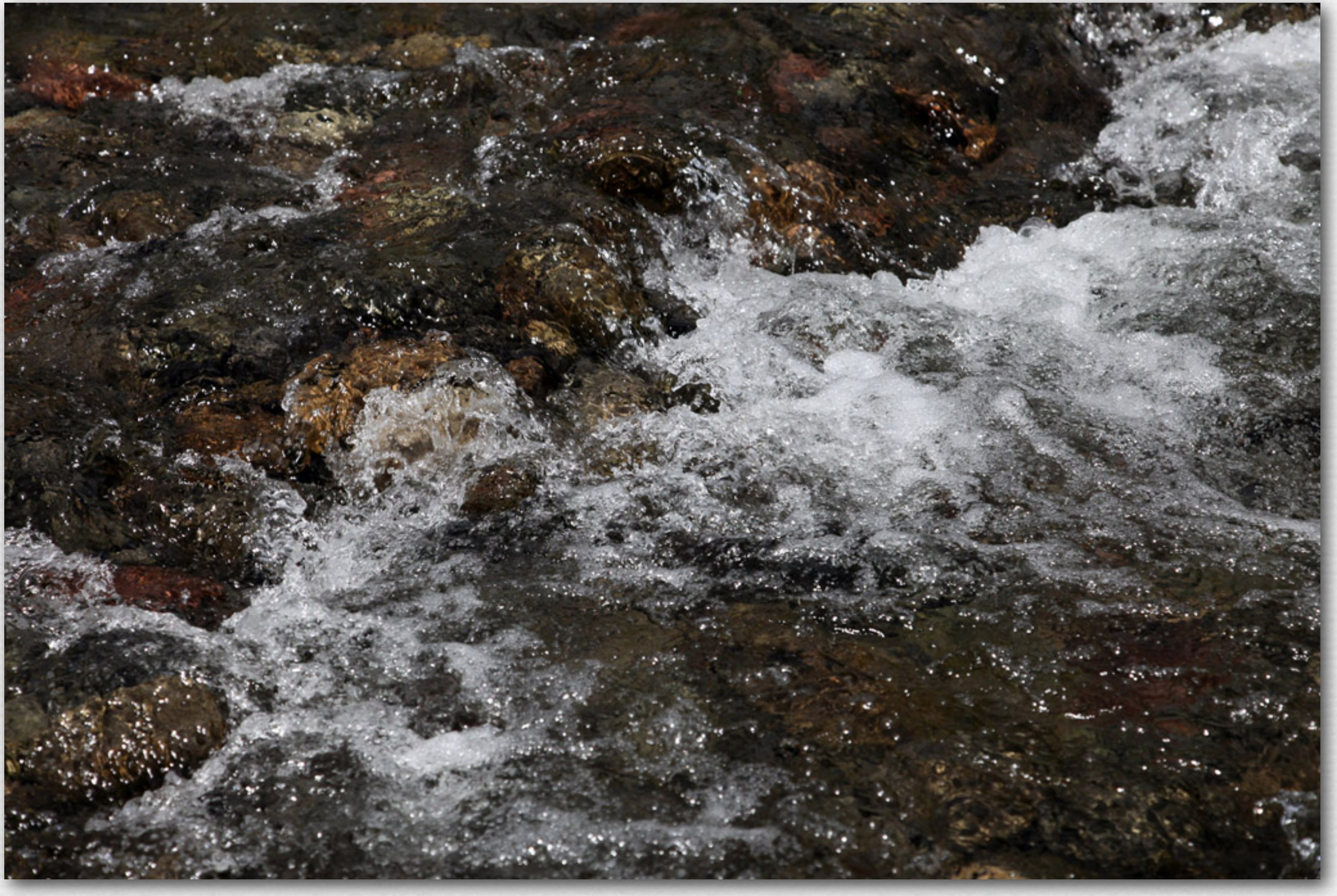
RGBC (a.k.a. RGBW)

- per-pixel neutral density filters [Nayar CPVR 2000]
  - trades spatial resolution for dynamic range
  - throws away photons

Sony

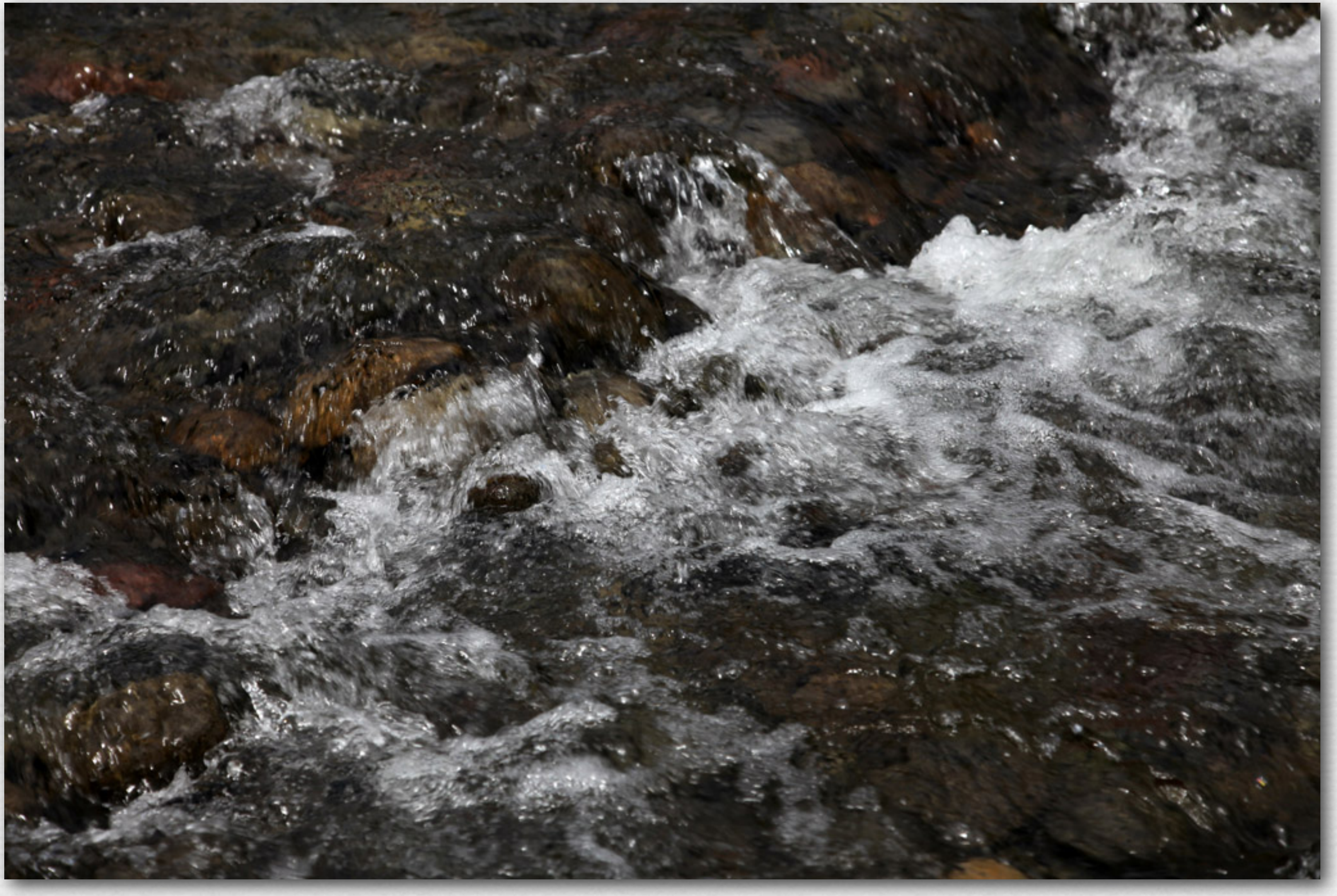


Games with ND filters



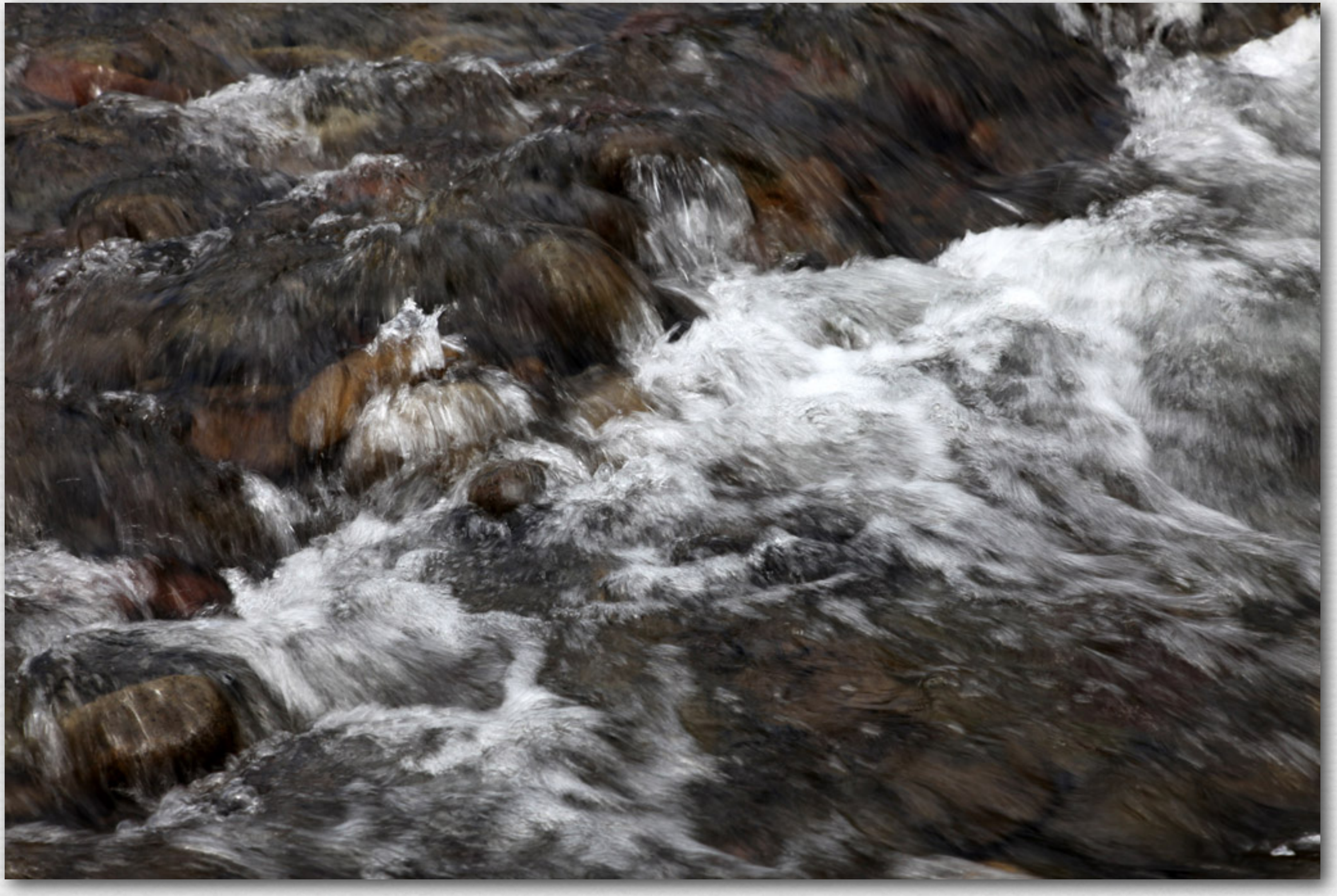
1/500s, f/5.6, ISO 800

Games with ND filters



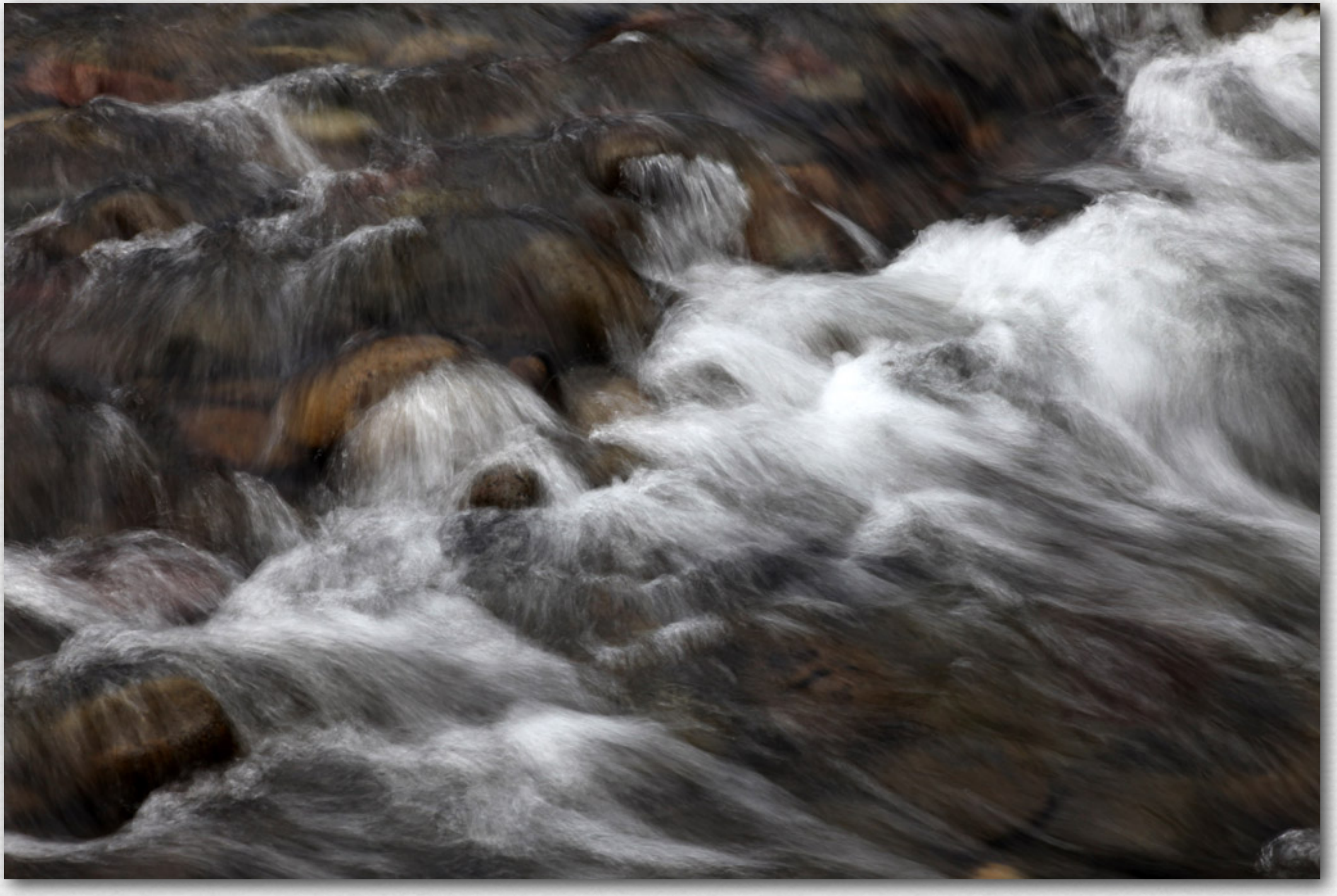
1/125s

Games with ND filters



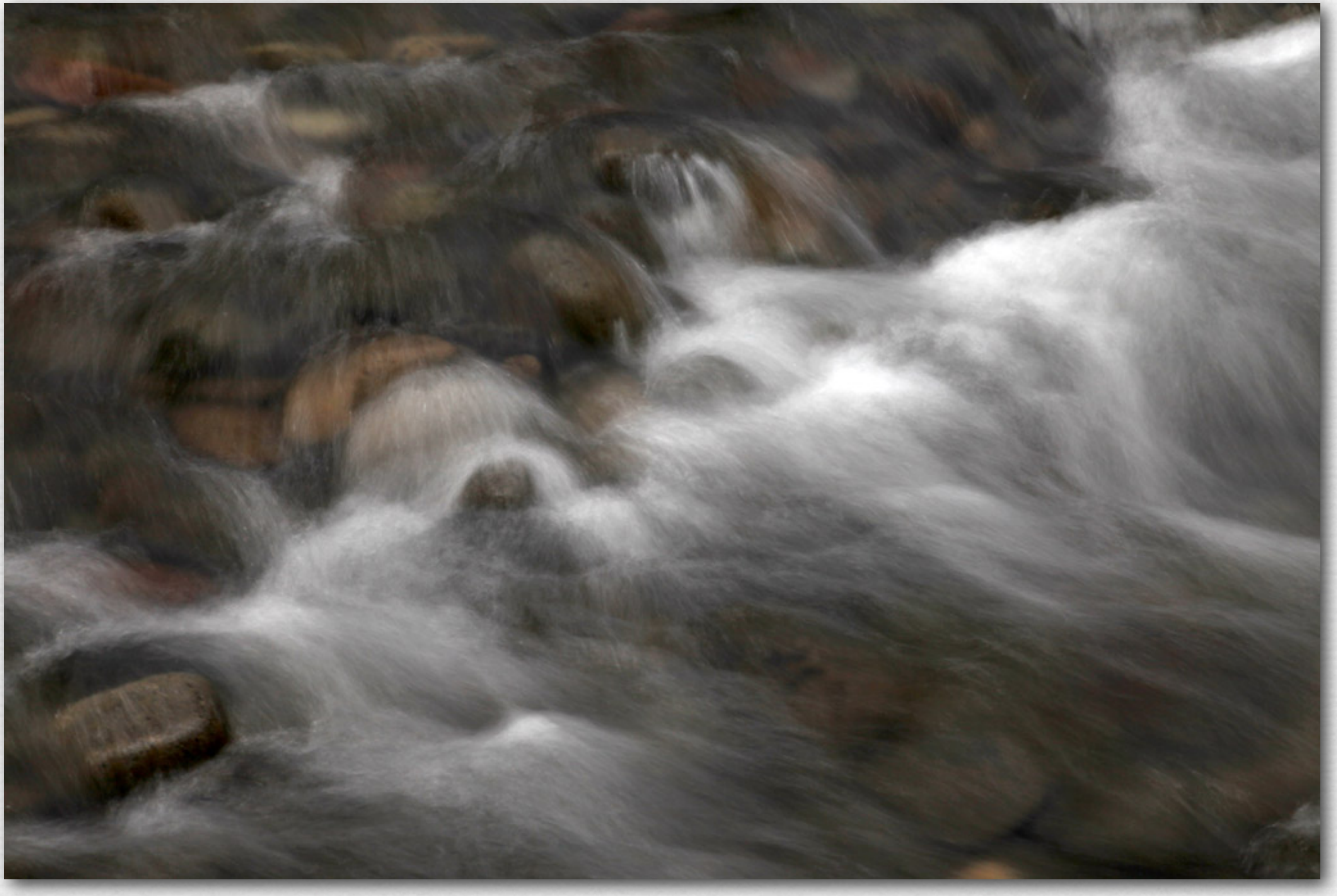
1/30s

Games with ND filters



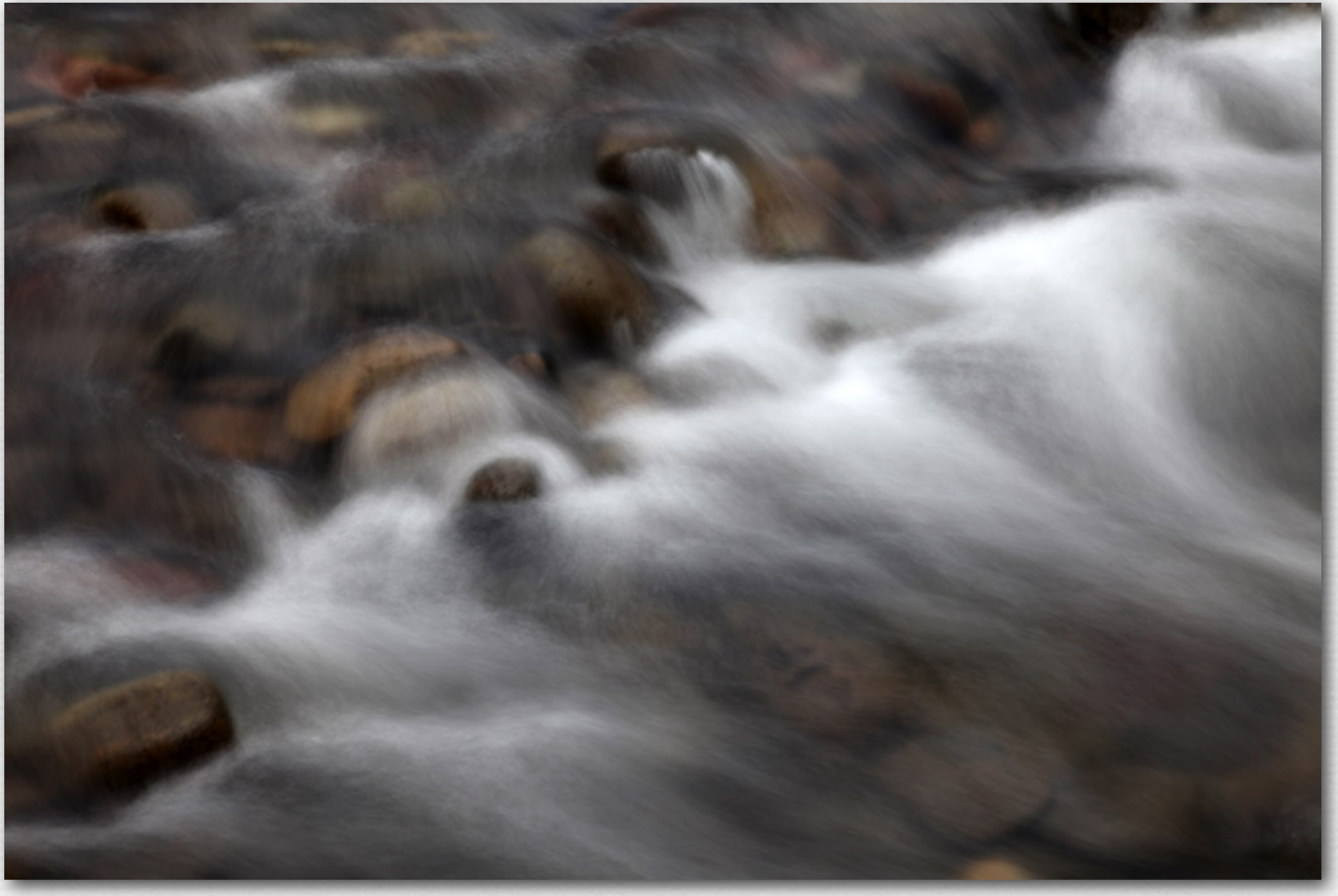
1/8s

Games with ND filters



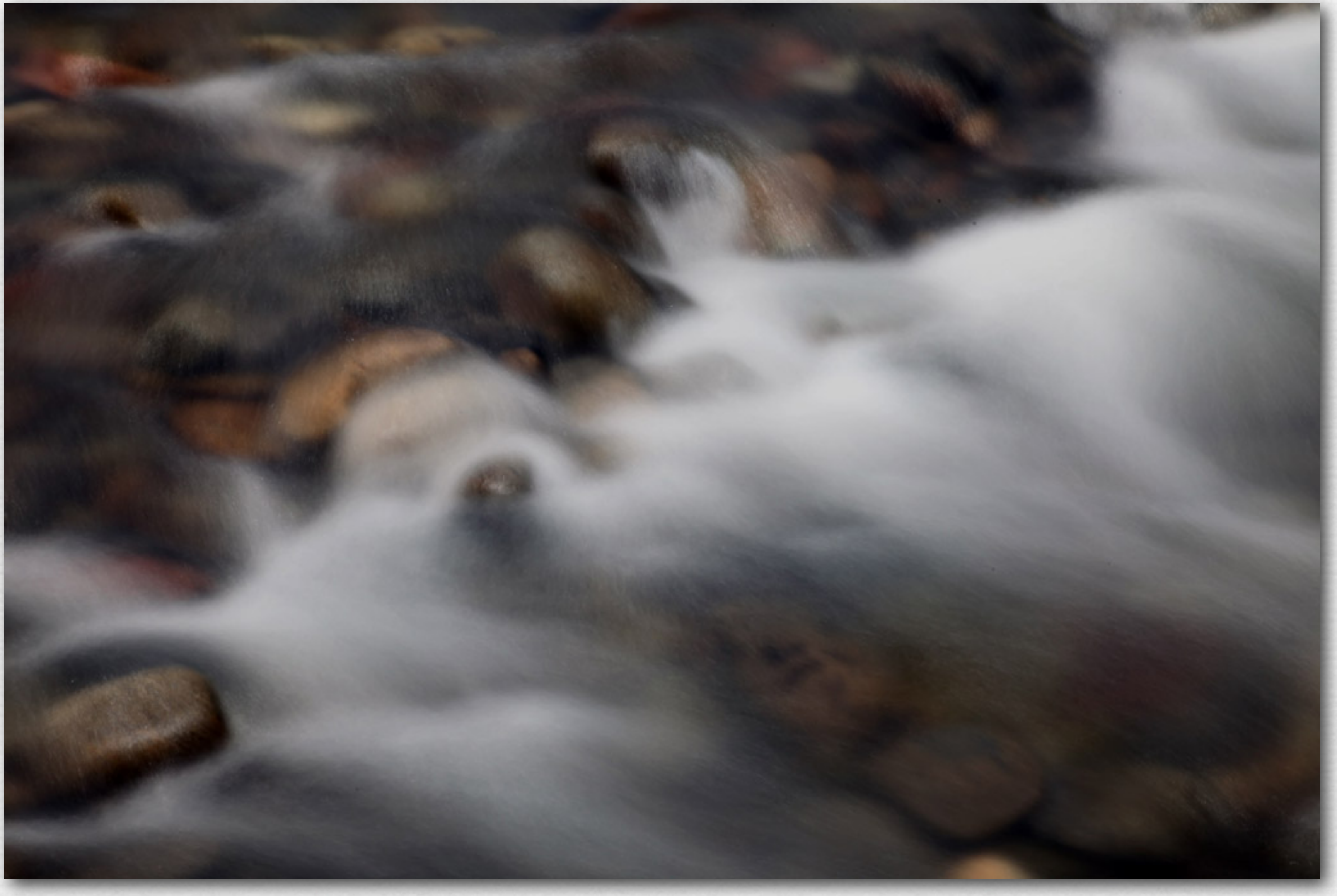
1/2s

Games with ND filters



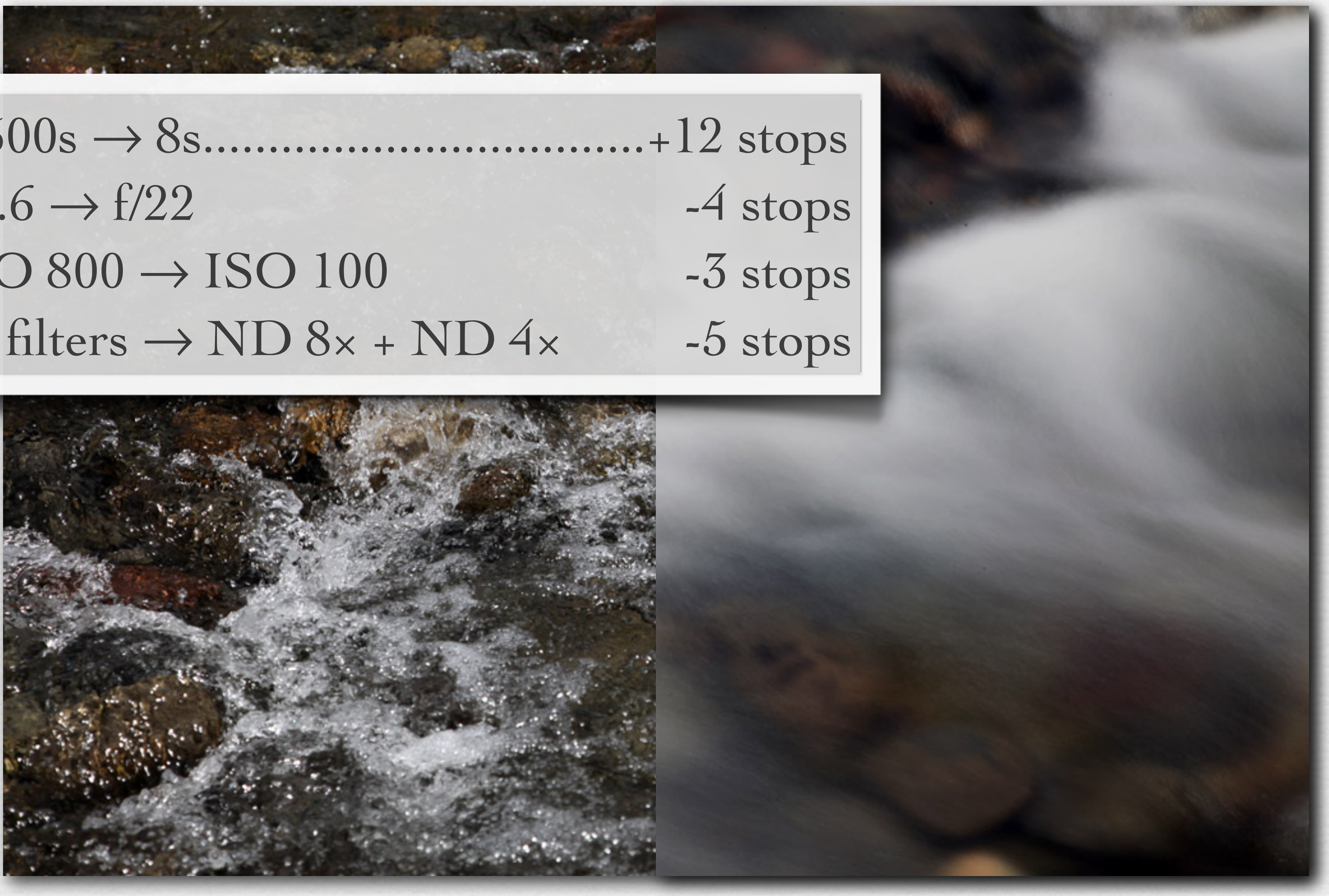
2s

Games with ND filters



8s

Games with ND filters

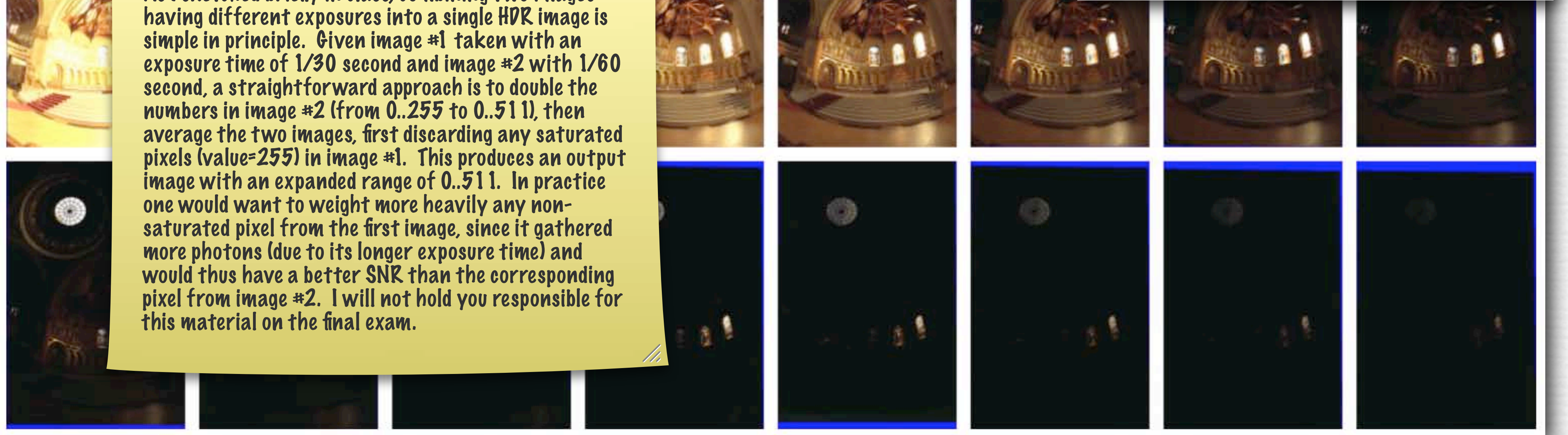


1/500s → 8s................................	+12 stops
f/5.6 → f/22	-4 stops
ISO 800 → ISO 100	-3 stops
no filters → ND 8× + ND 4×	-5 stops

Capturing HDR images

- multiple bracketed exposures [Debevec SIGGRAPH 1997]
- changing the exposure time is usually better than changing the aperture

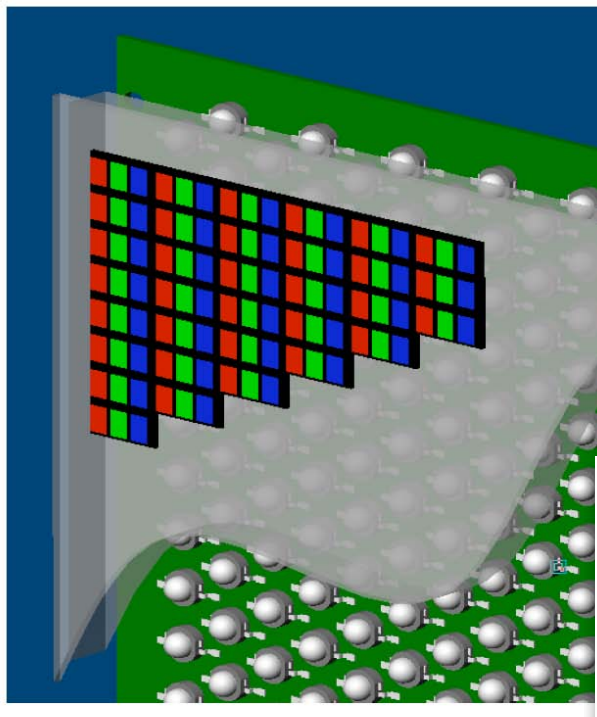
Q. How about changing the ISO?

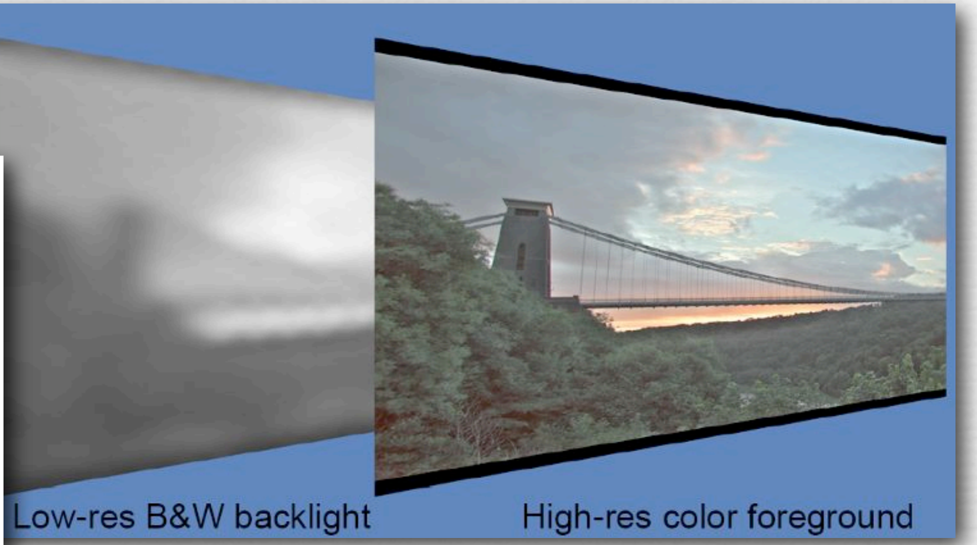


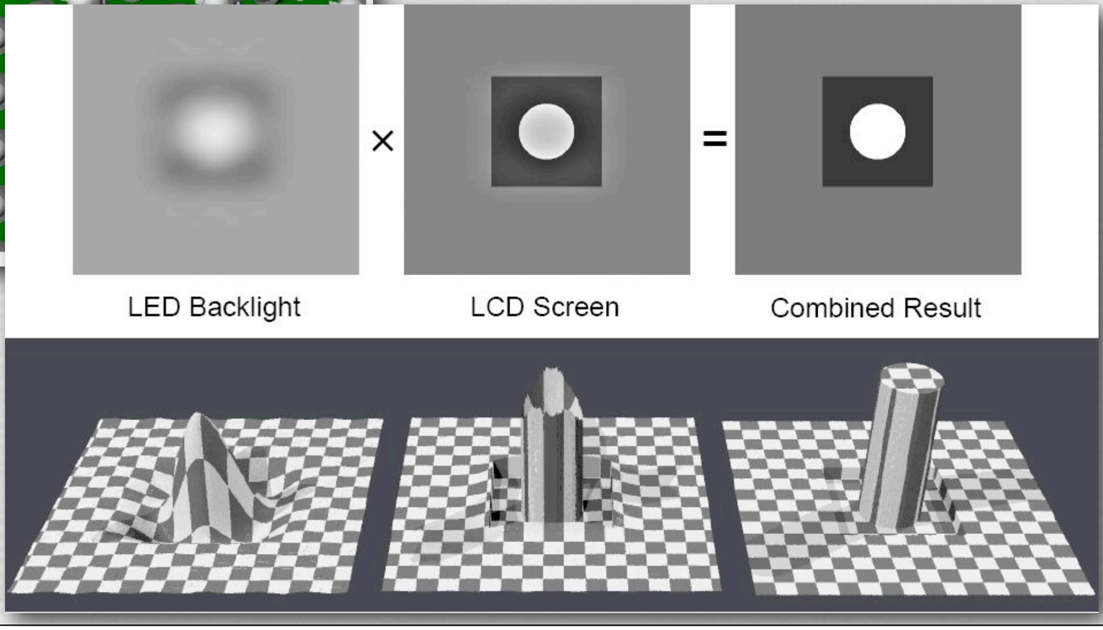
As I sketched briefly in class, combining two images having different exposures into a single HDR image is simple in principle. Given image #1 taken with an exposure time of 1/30 second and image #2 with 1/60 second, a straightforward approach is to double the numbers in image #2 (from 0..255 to 0..511), then average the two images, first discarding any saturated pixels (value=255) in image #1. This produces an output image with an expanded range of 0..511. In practice one would want to weight more heavily any non-saturated pixel from the first image, since it gathered more photons (due to its longer exposure time) and would thus have a better SNR than the corresponding pixel from image #2. I will not hold you responsible for this material on the final exam.

Direct display of HDR images

- Sunnybrook HDR display







Brightside HDR display

conjectured iWatch
(isource.com)

Panasonic curved OLED screens

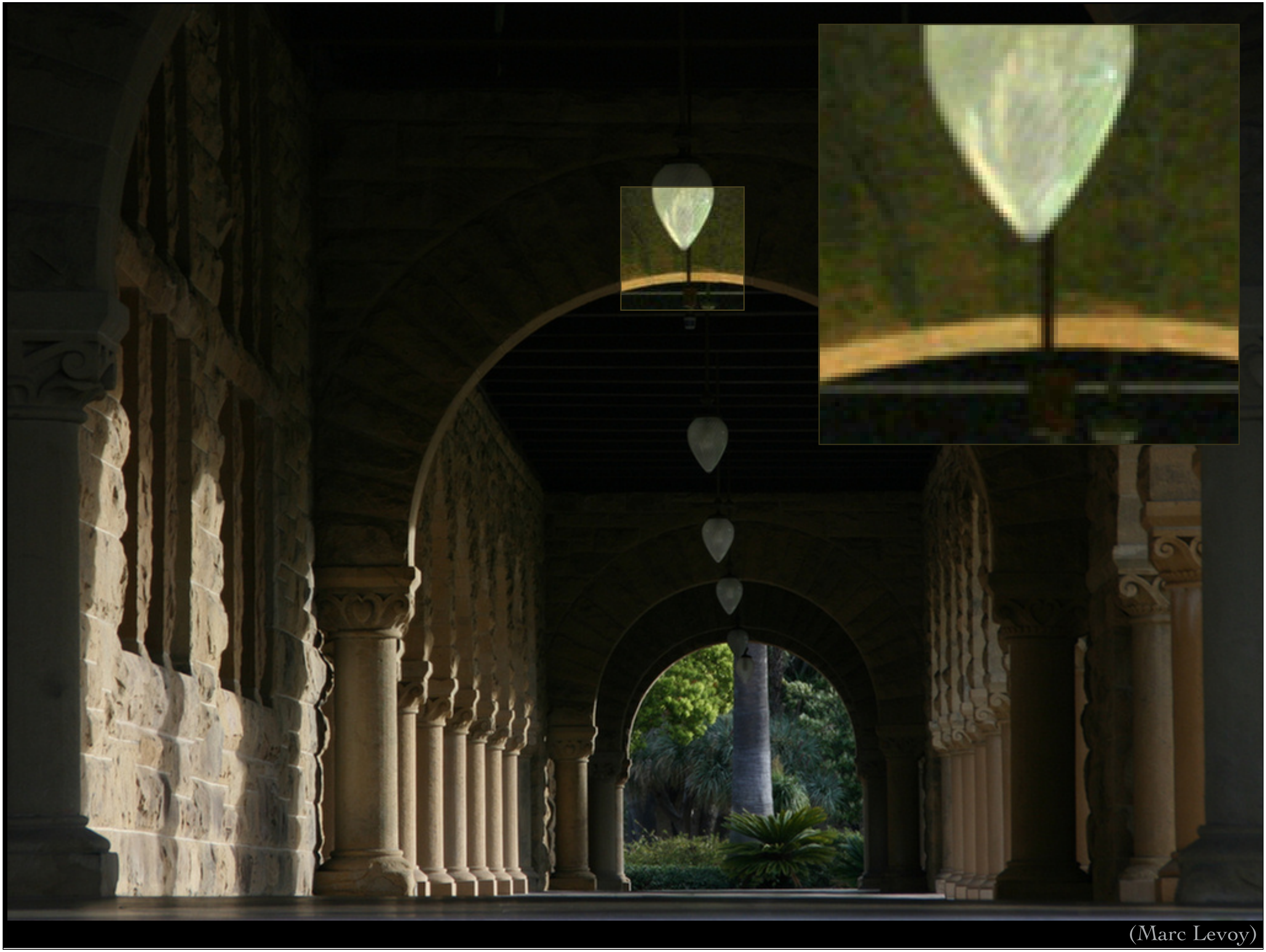
High dynamic range (HDR) imaging

- step 1: capturing HDR images
- step 2a: direct display of HDR images, or
- step 2b: tone mapping to create an LDR image for display

you're not responsible for HDR tone mapping on your final

- goals of HDR → LDR tone mapping
  - squeeze >12 bits of HDR image into 8 bits for JPEG
  - hint: just scaling the pixel values looks poor, as we'll see...

(Marc Levoy)



(Marc Levoy)

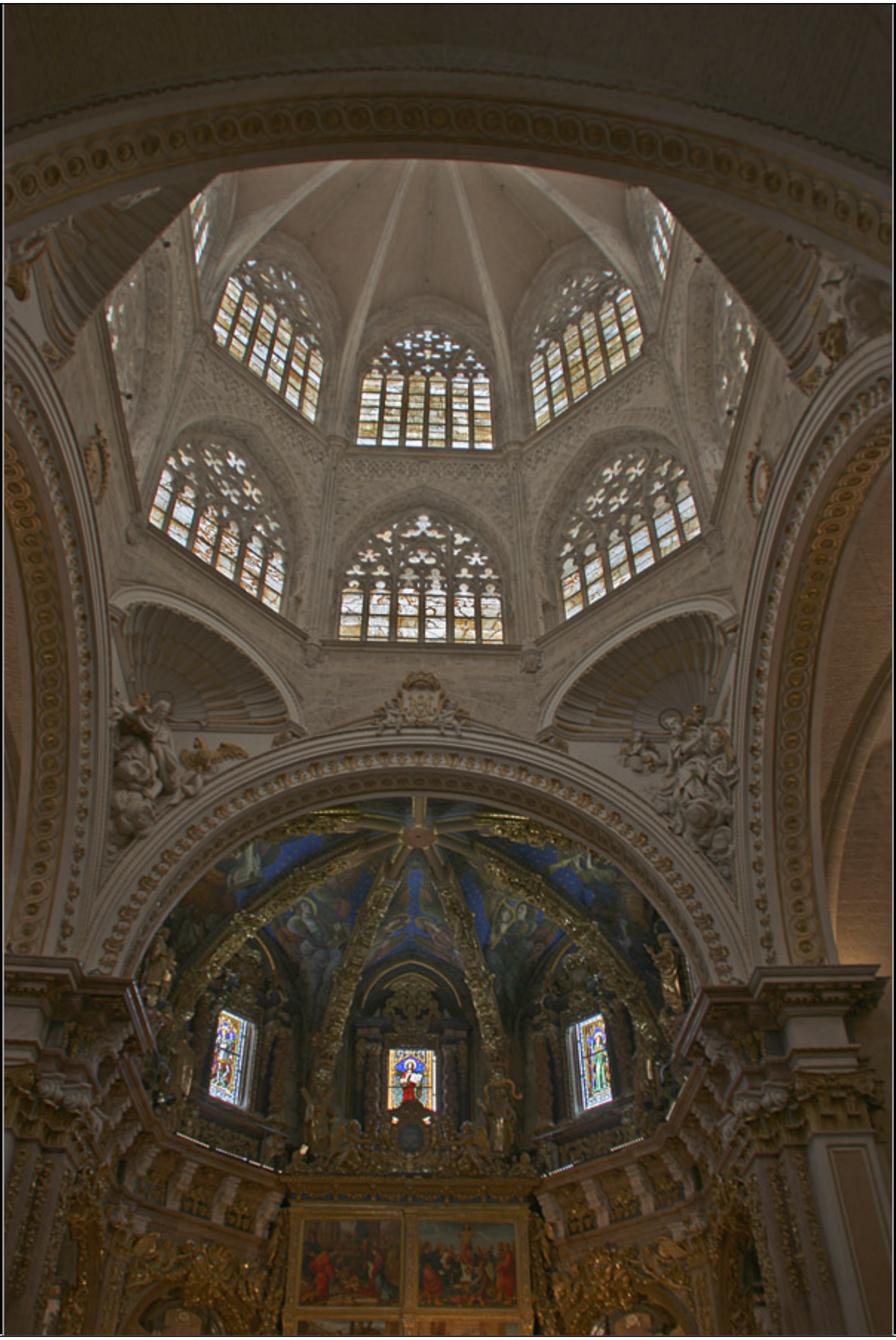
(Marc Levoy)

Cathedral,
Valencia

(Marc Levoy)

tone mapping in Photoshop CS4 by "exposure and gamma" method

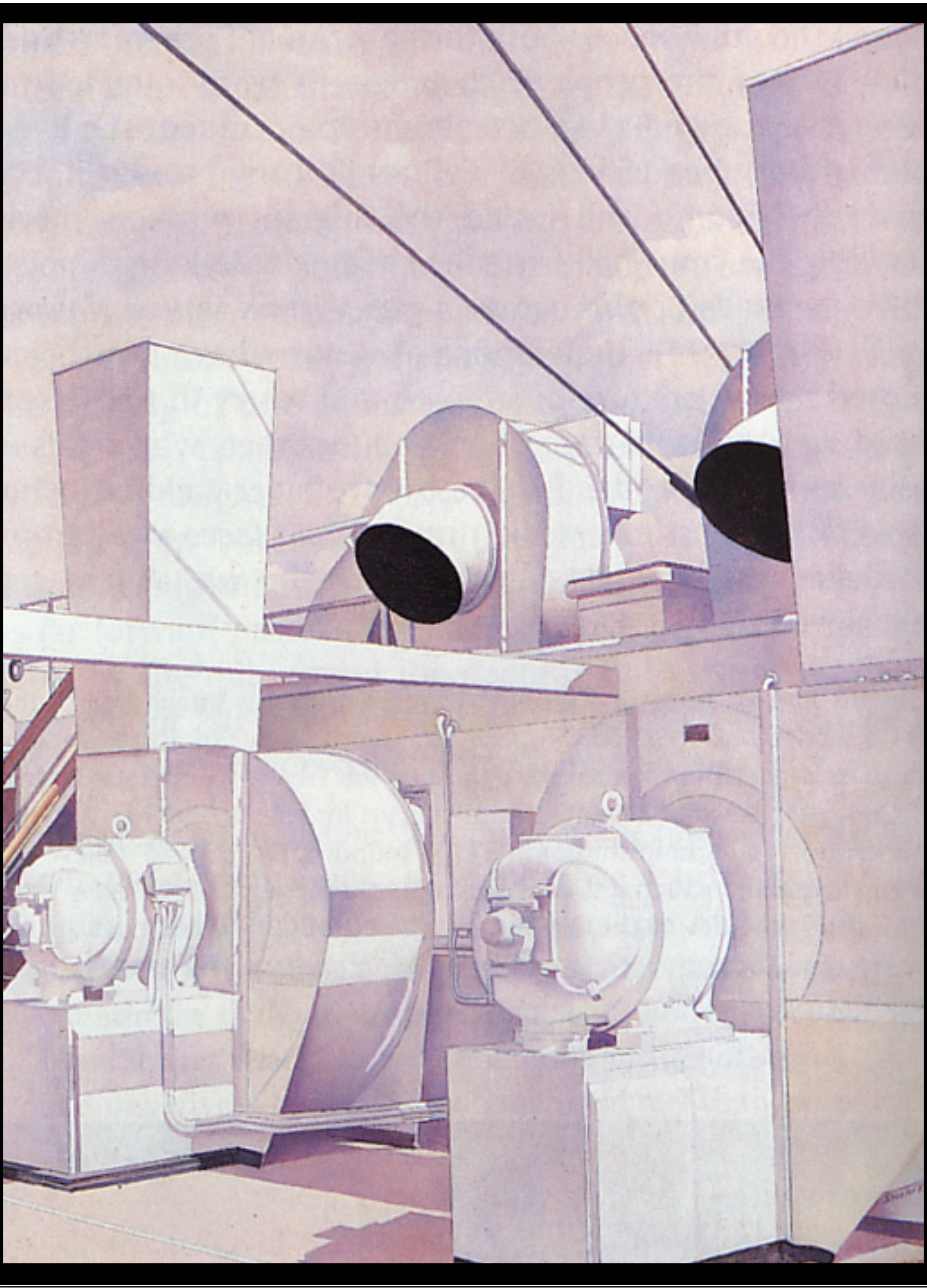
(if gamma=1, this is just scaling the pixel values)



Cathedral, Valencia

(Marc Levoy)

How do artists solve the tone mapping problem?



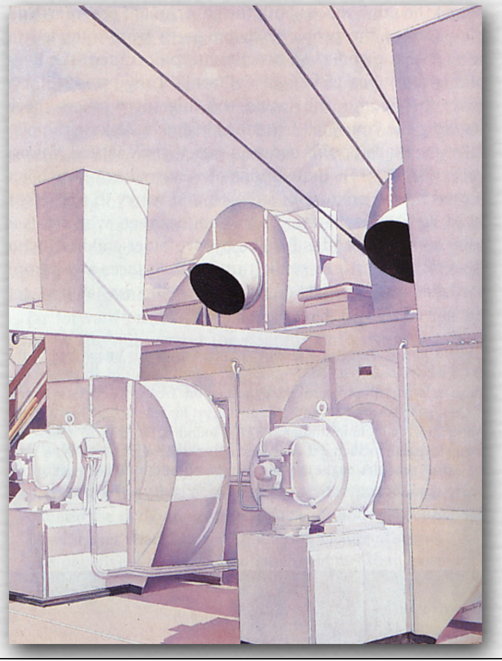
Charles Sheeler,
The Upper Deck
(1929)

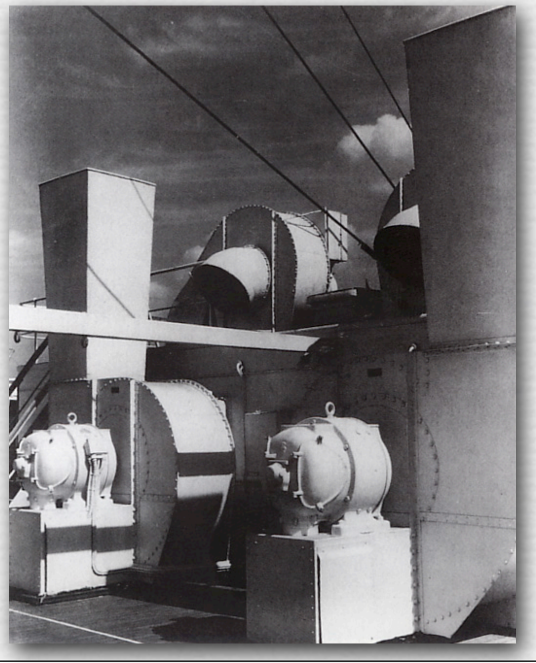
Joseph Wright,The Orrery (1765)

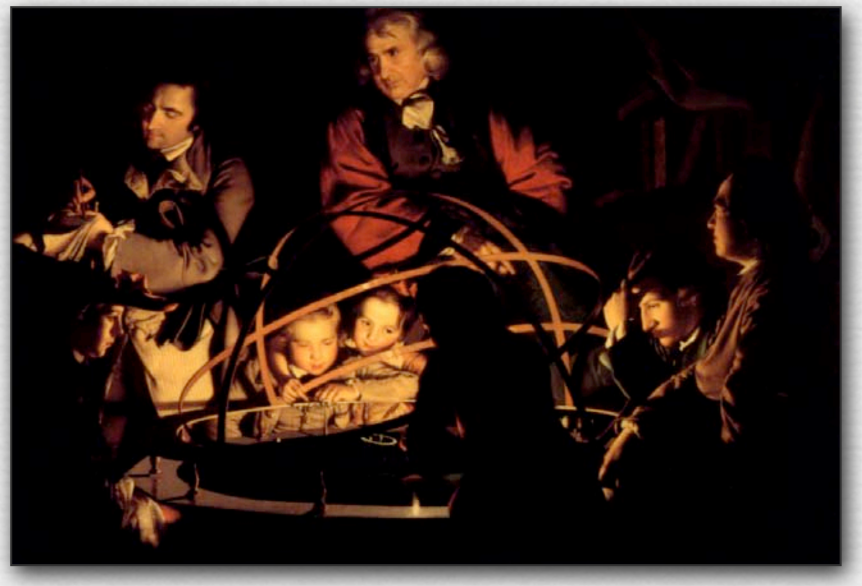
How do artists solve the tone mapping problem?

- for bright scenes
  - human vision is dazzled, compressing brightnesses
- for dark scenes
  - shadows are below threshold, so completely black

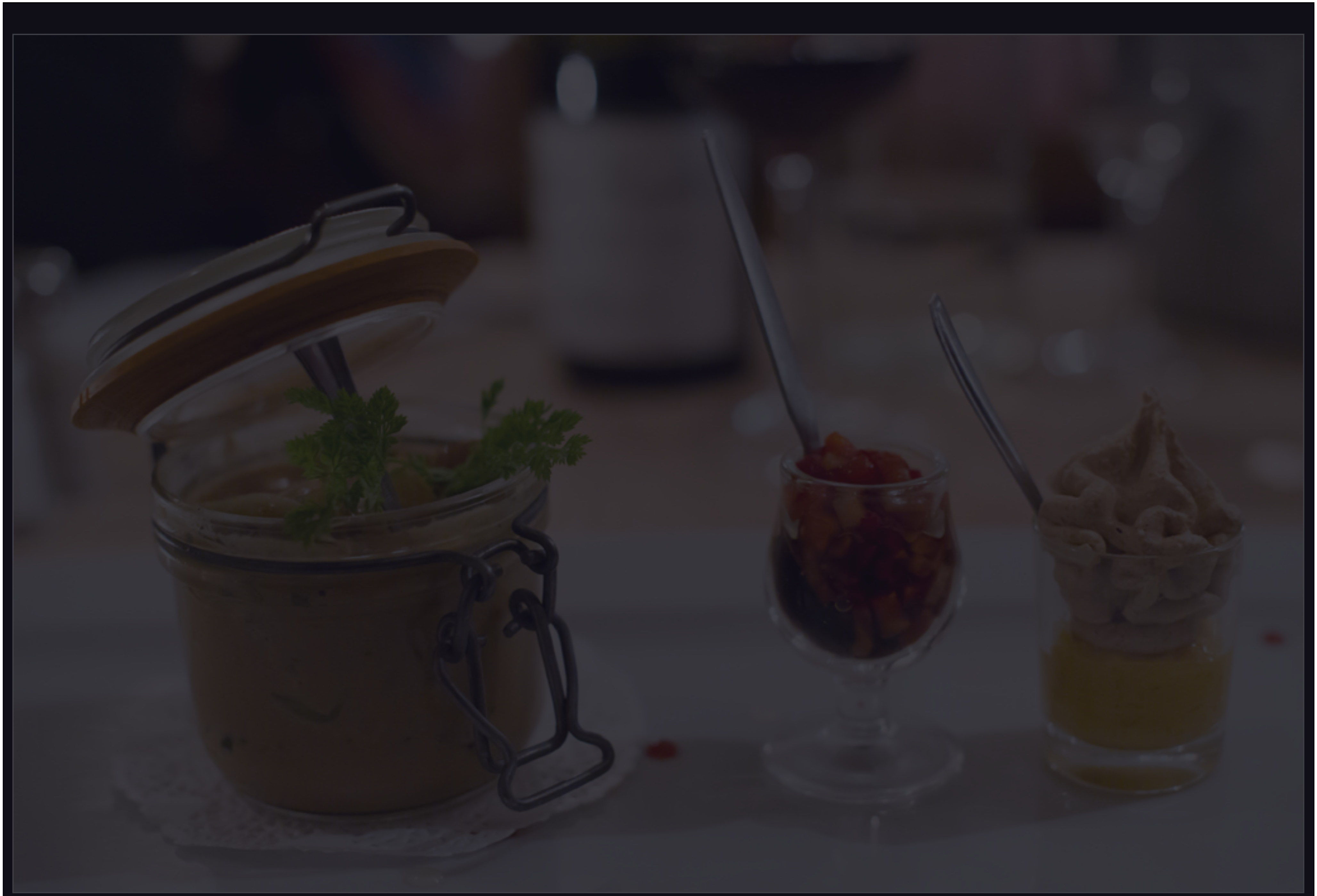
Hermann von Helmholtz
(1821-1894)
"The relation of optics to painting"



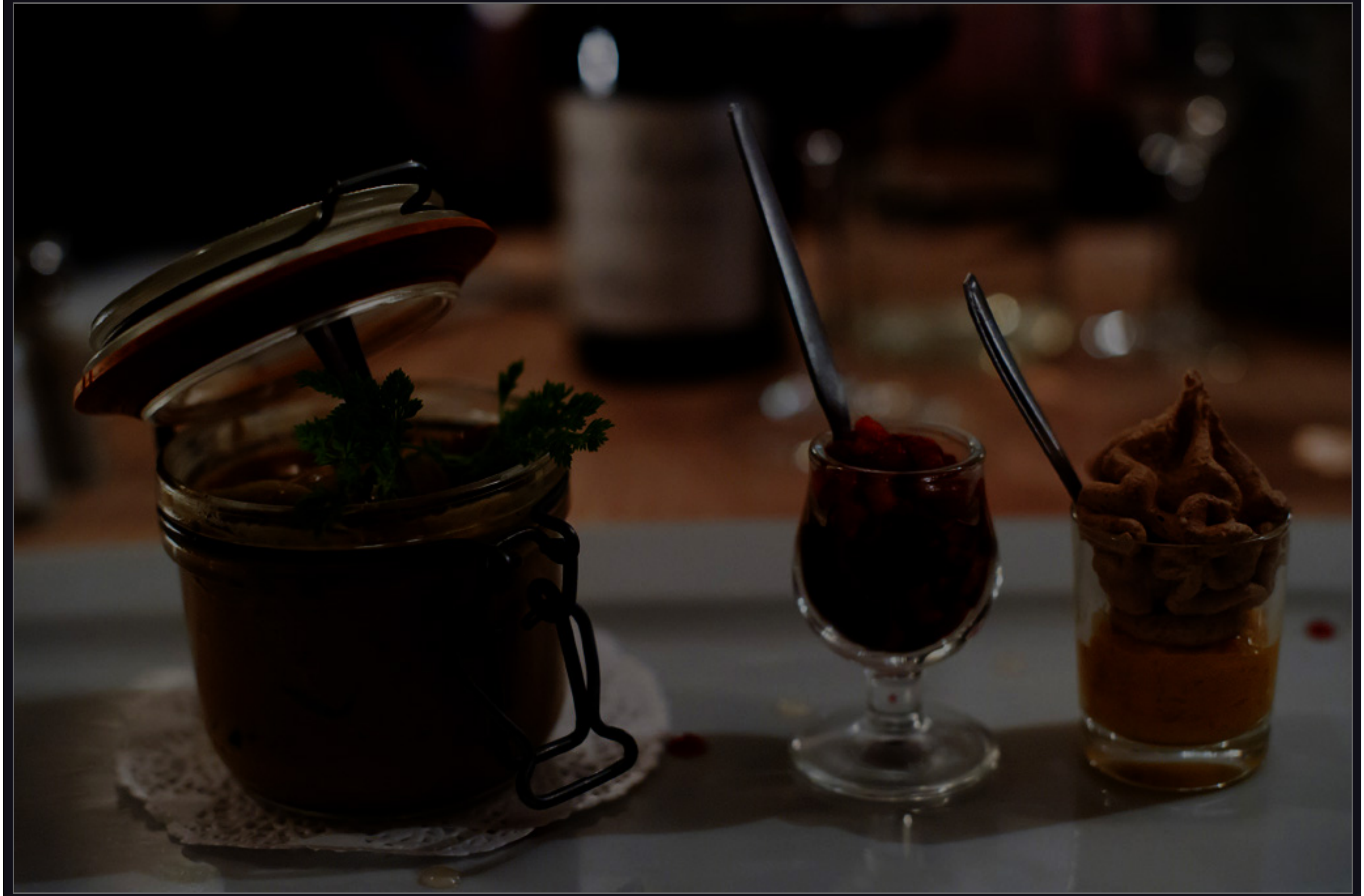




Nikon D3S, ISO 25,600, denoised in Lightroom 3, photograph by Fredo Durand



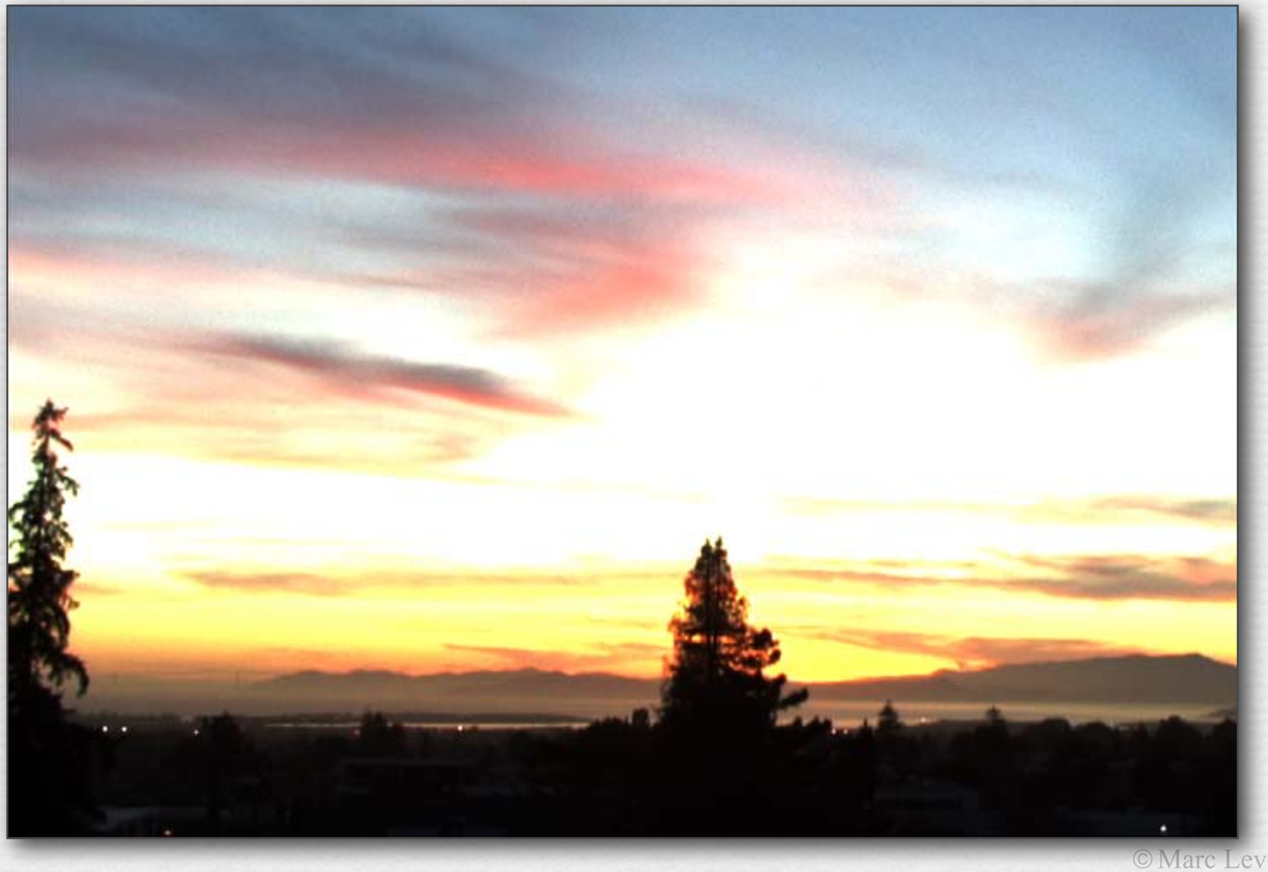
Linear scaling of pixel values (followed by RAW ⟶ JPEG gamma transform)



Thresholding of shadows (and maybe an additional S-curve)

Tone mapping for very HDR scenes
(slides from Fredo Durand)

- scene has >100,000:1 dynamic range, JPEG has 255:1
- how can we compress the scene's dynamic range?



- in the mapping shown here, the sun is blown out
- if scaled linearly from 100,000:1 scene to 255:1, almost all pixels would be dark

Global tone mapping operators

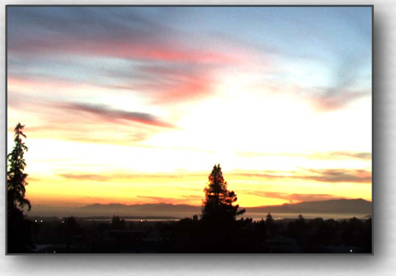
- gamma compression, applied independently on R,G,B

  output = input$^{\gamma}$ (γ = 0.5 here)

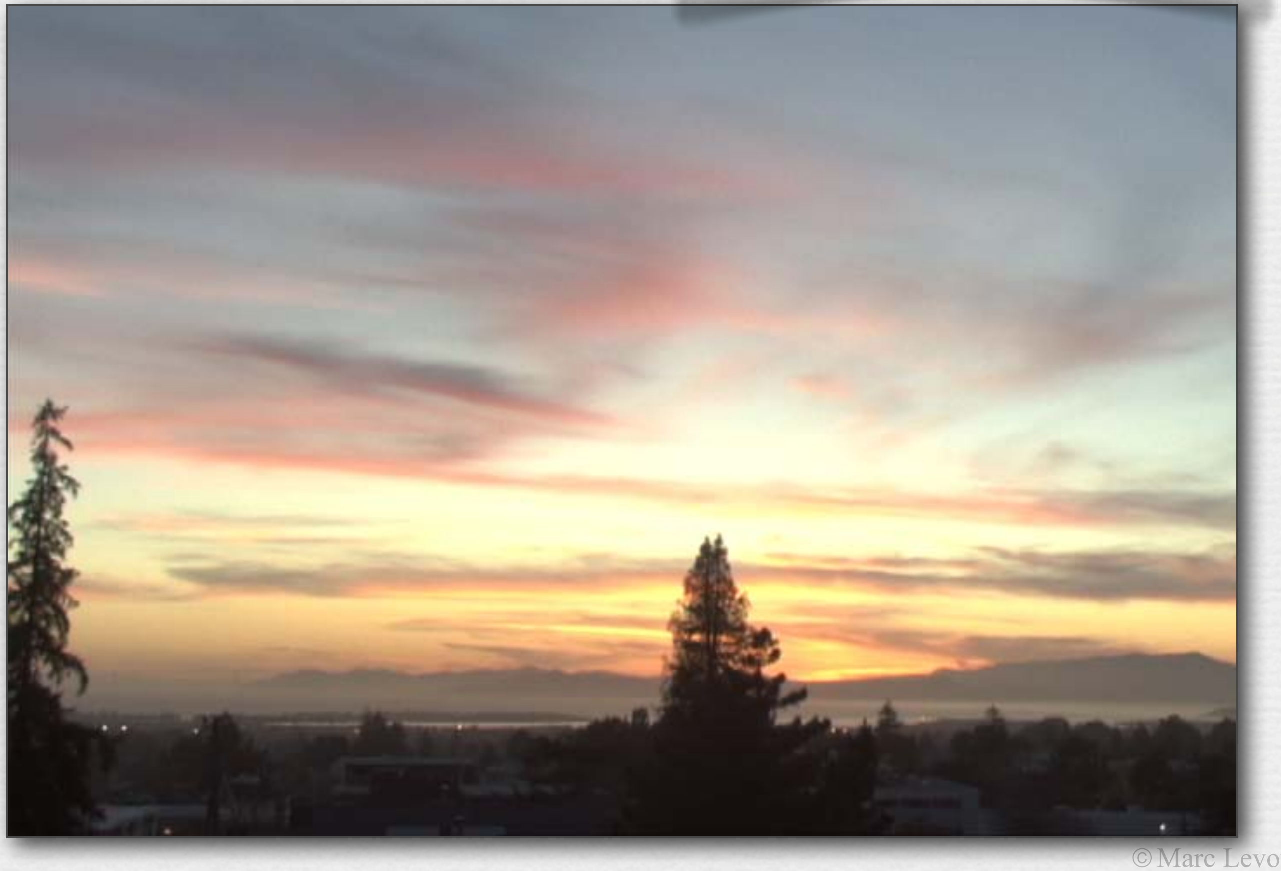
- colors become washed out

in addition to the gamma transform during RAW → JPEG conversion

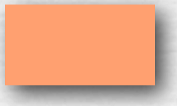
input



output

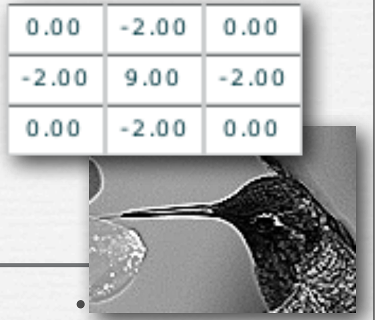


$(1.0, 0.4, 0.2)^{0.5} = (1.0, 0.63, 0.44)$



(try it yourself in Photoshop)

(from our convolution applet)

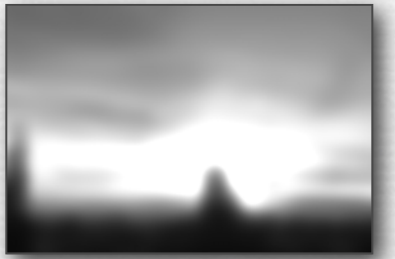


0.00	-2.00	0.00
-2.00	9.00	-2.00
0.00	-2.00	0.00

Local tone mapping operators

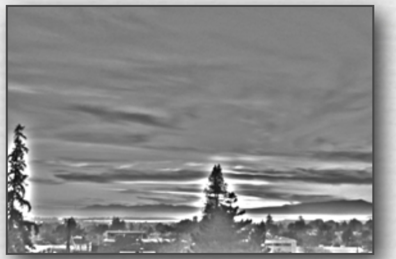
- ✦ reduce contrast of low frequencies, while preserving high frequencies [Oppenheim 1968, Chiu et al. 1993]
- ✦ produces halos!

low frequency



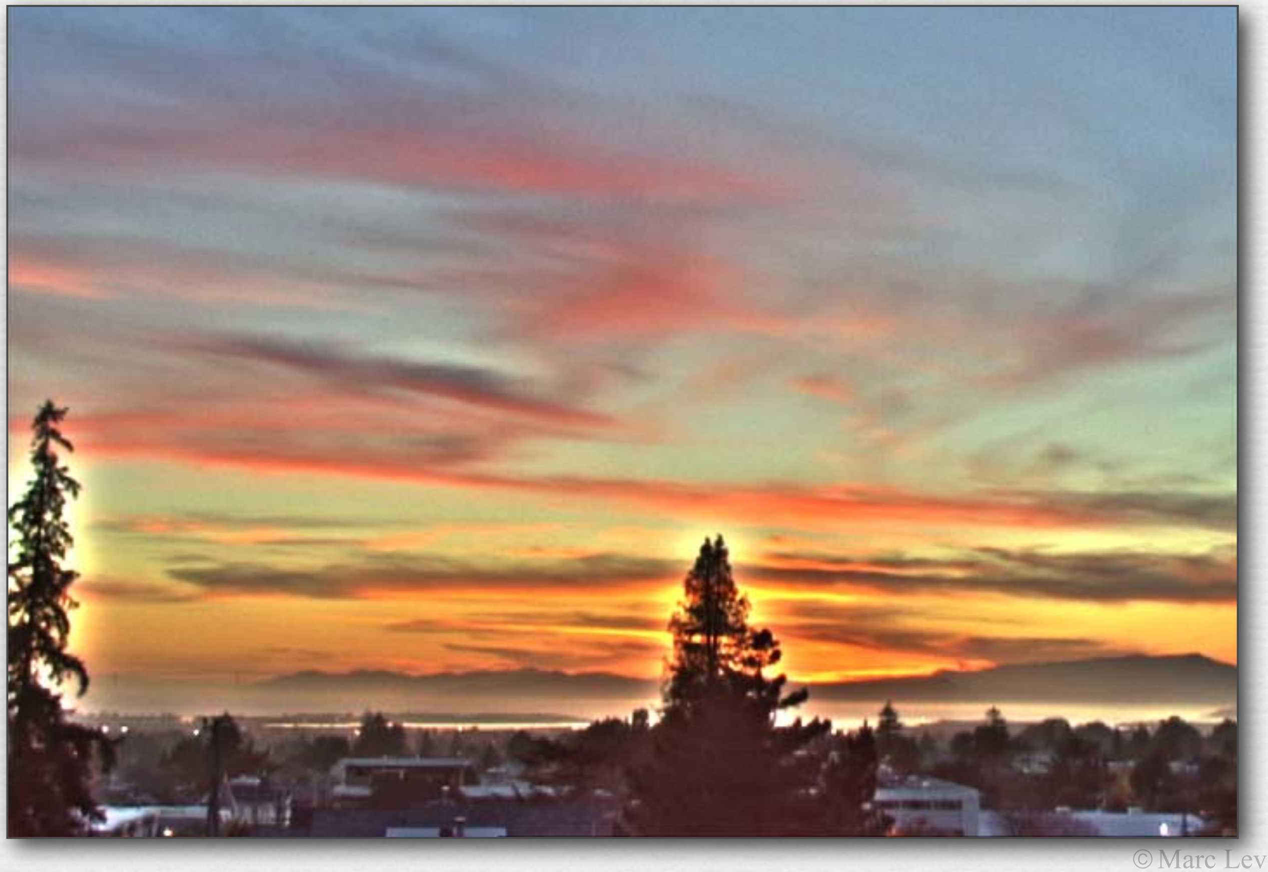
(e.g. Gaussian blur)

high frequency



(e.g. original minus Gaussian)

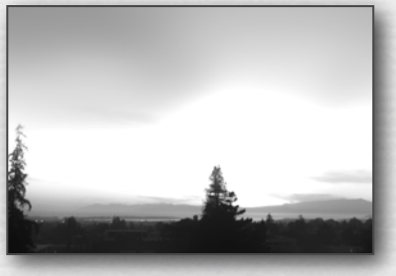
chrominance



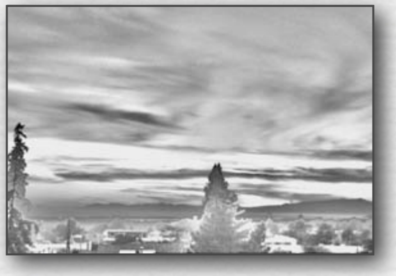
Local tone mapping operators

- bilateral filtering to compute large scale image without blurring across edges, remainder is detail image (no halos!); reduce contrast of large scale, while preserving details [Durand and Dorsey SIGGRAPH 2002]

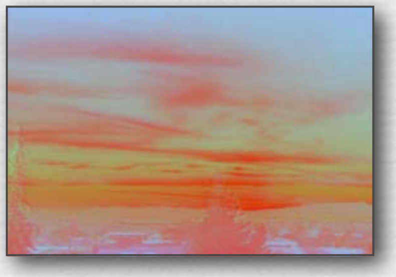
large scale

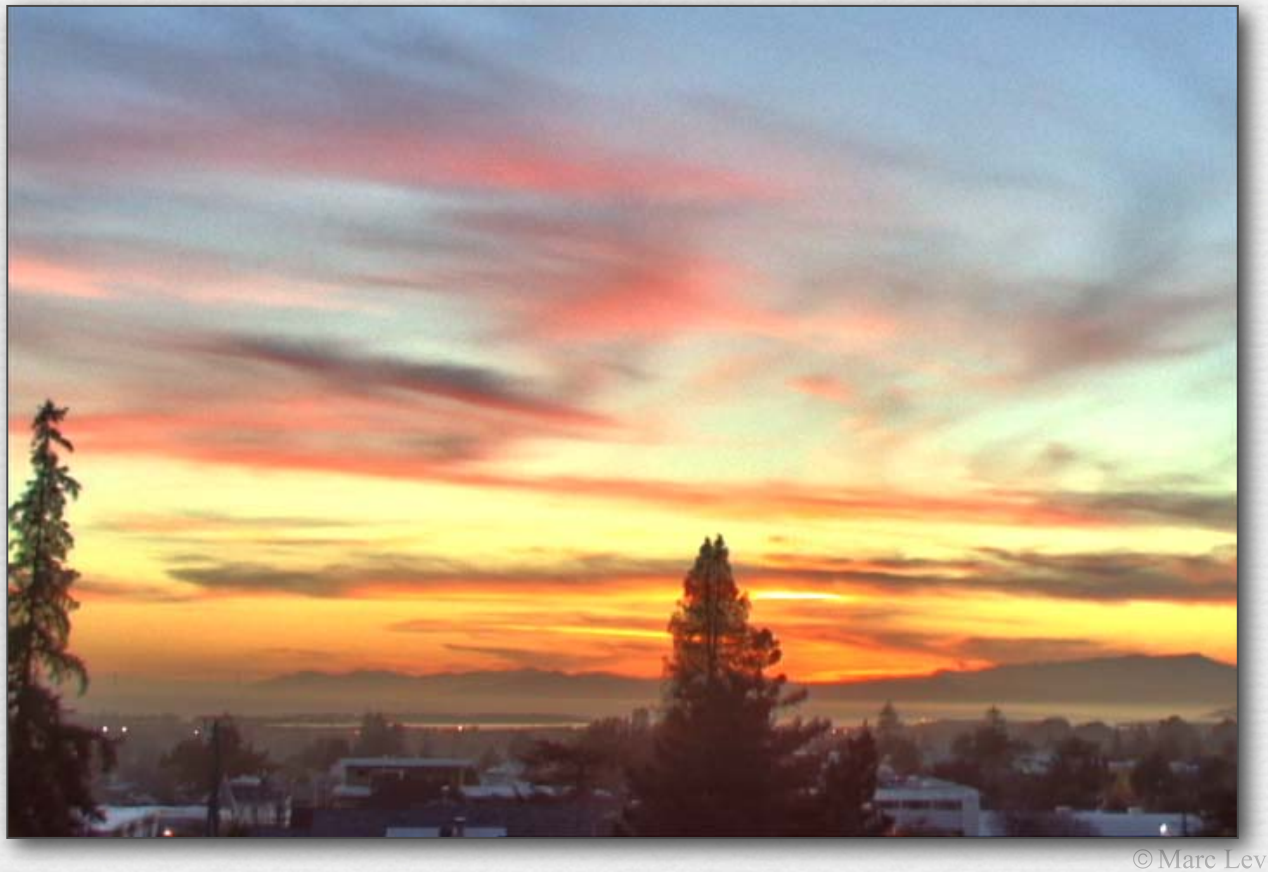


detail

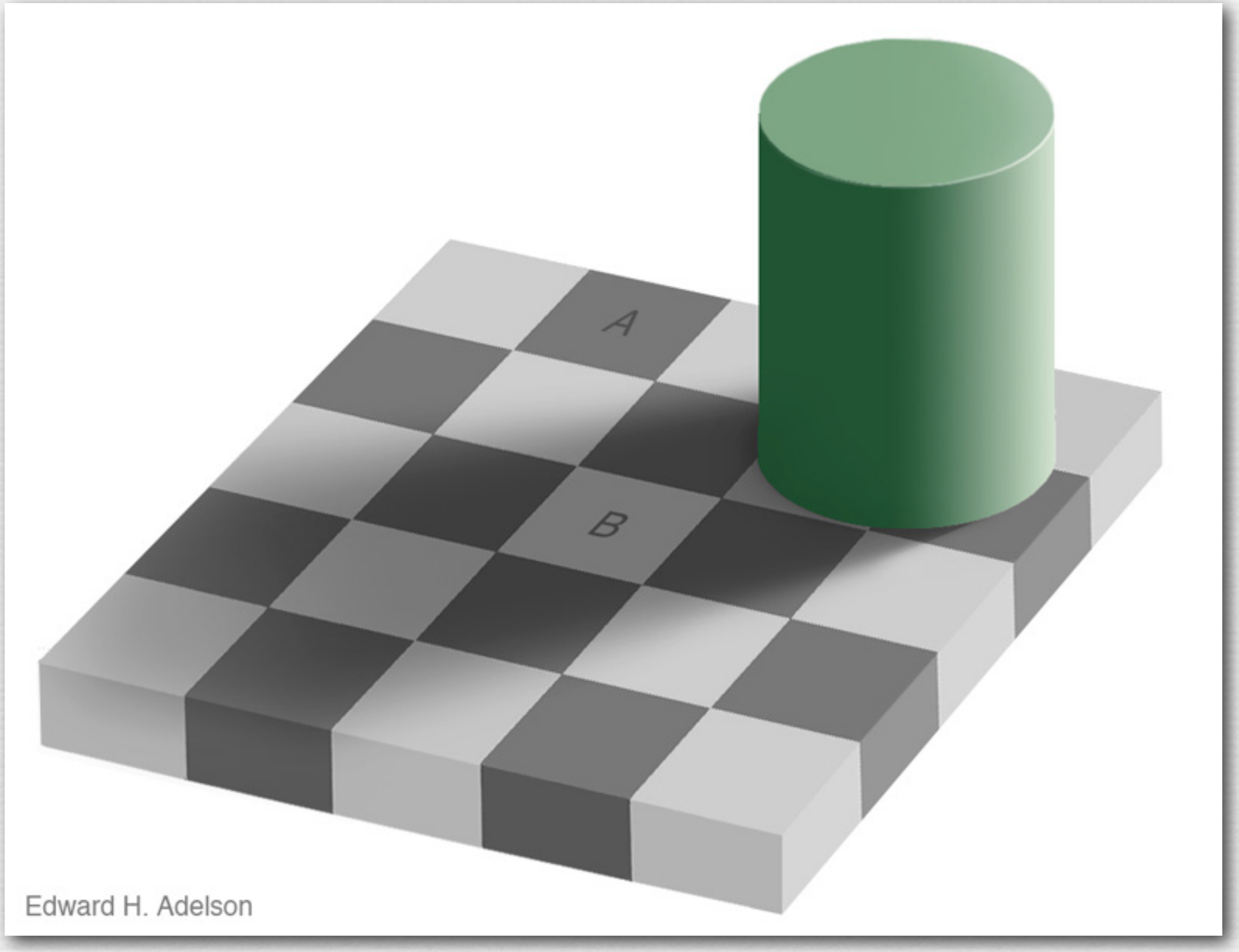


chrominance

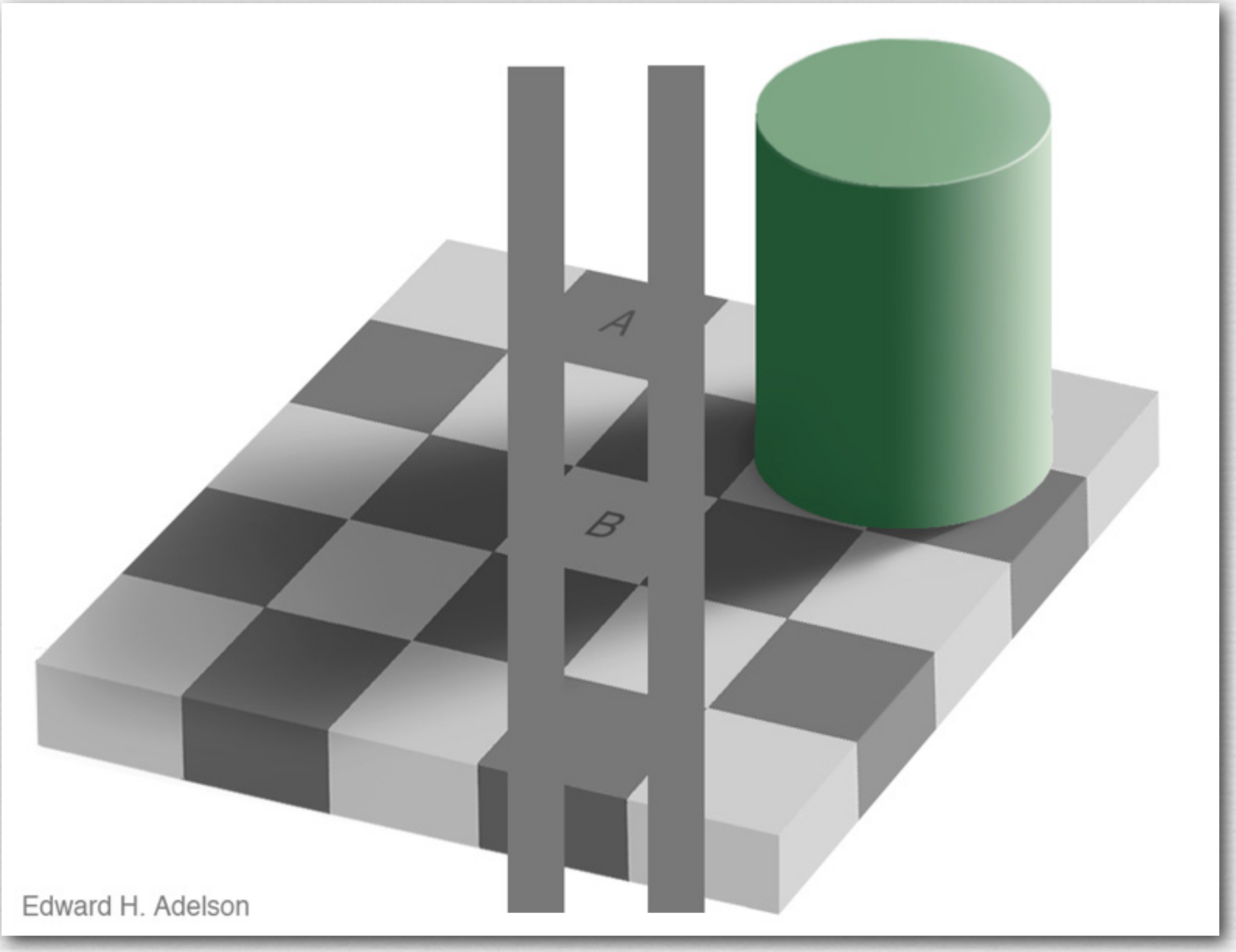




The importance of local contrast

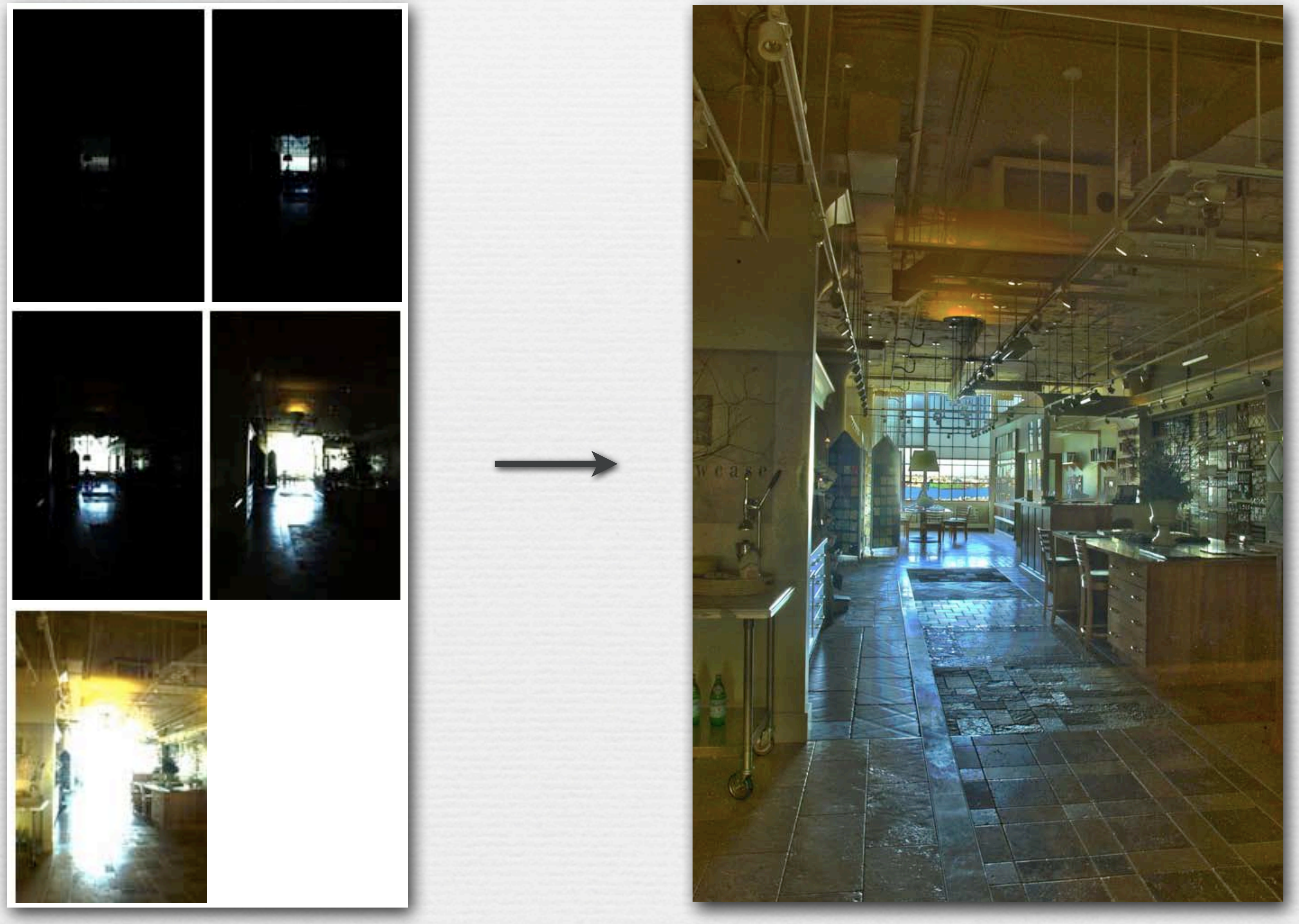


The importance of local contrast

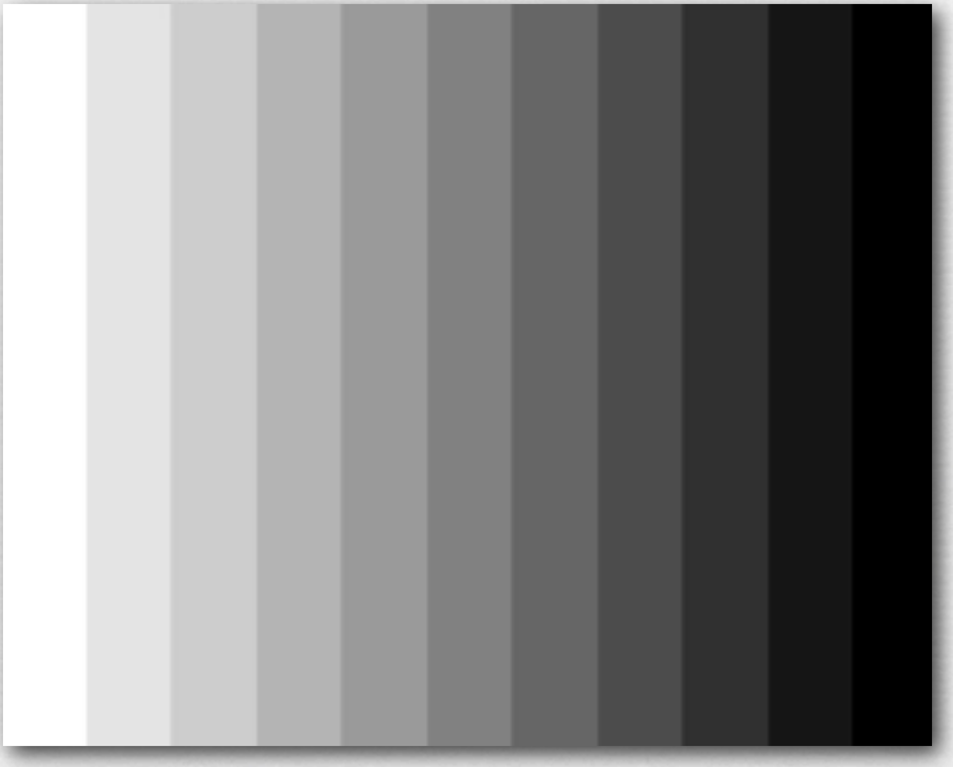


Tone mapping using bilateral filters

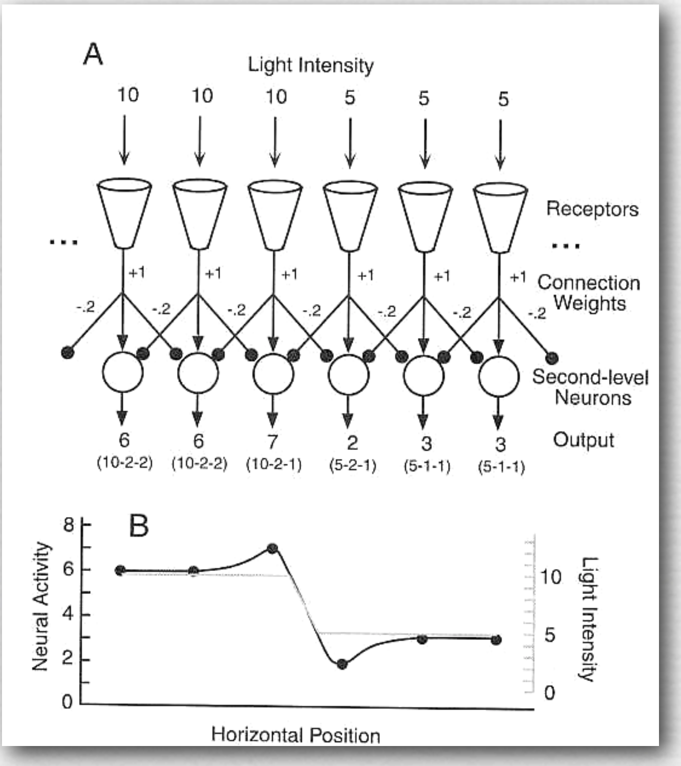
[Durand and Dorsey SIGGRAPH 2002]



Mach bands and lateral inhibition



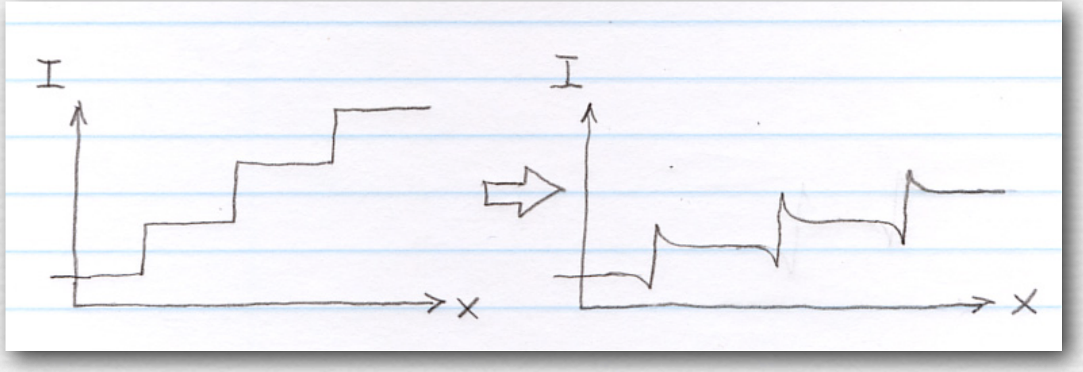
the Mach band illusion: each wedge should appear brighter on its right side



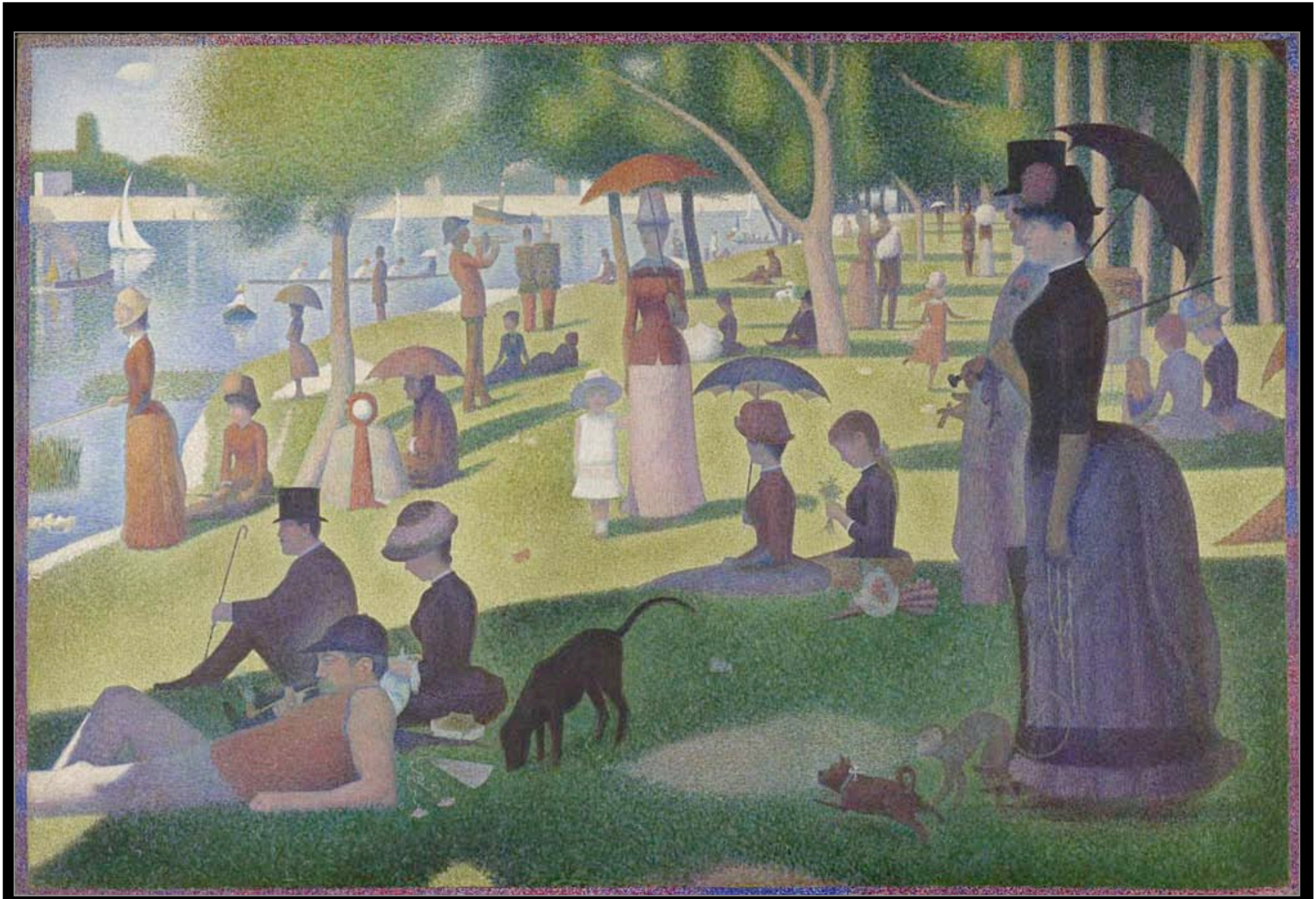
(Goldstein or Wolfe)

- lateral inhibition among receptive fields in the retina is equivalent to image convolution with a sharpening kernel

Why might tone mapping look cartoony? (contents of whiteboard)



- a step wedge (at left) is converted by a tone mapping operator that enhances local contrast to the plot at right
  - the human eye does this internally due to lateral inhibition, but that doesn't necessarily mean we want to present an image like this to the human eye!

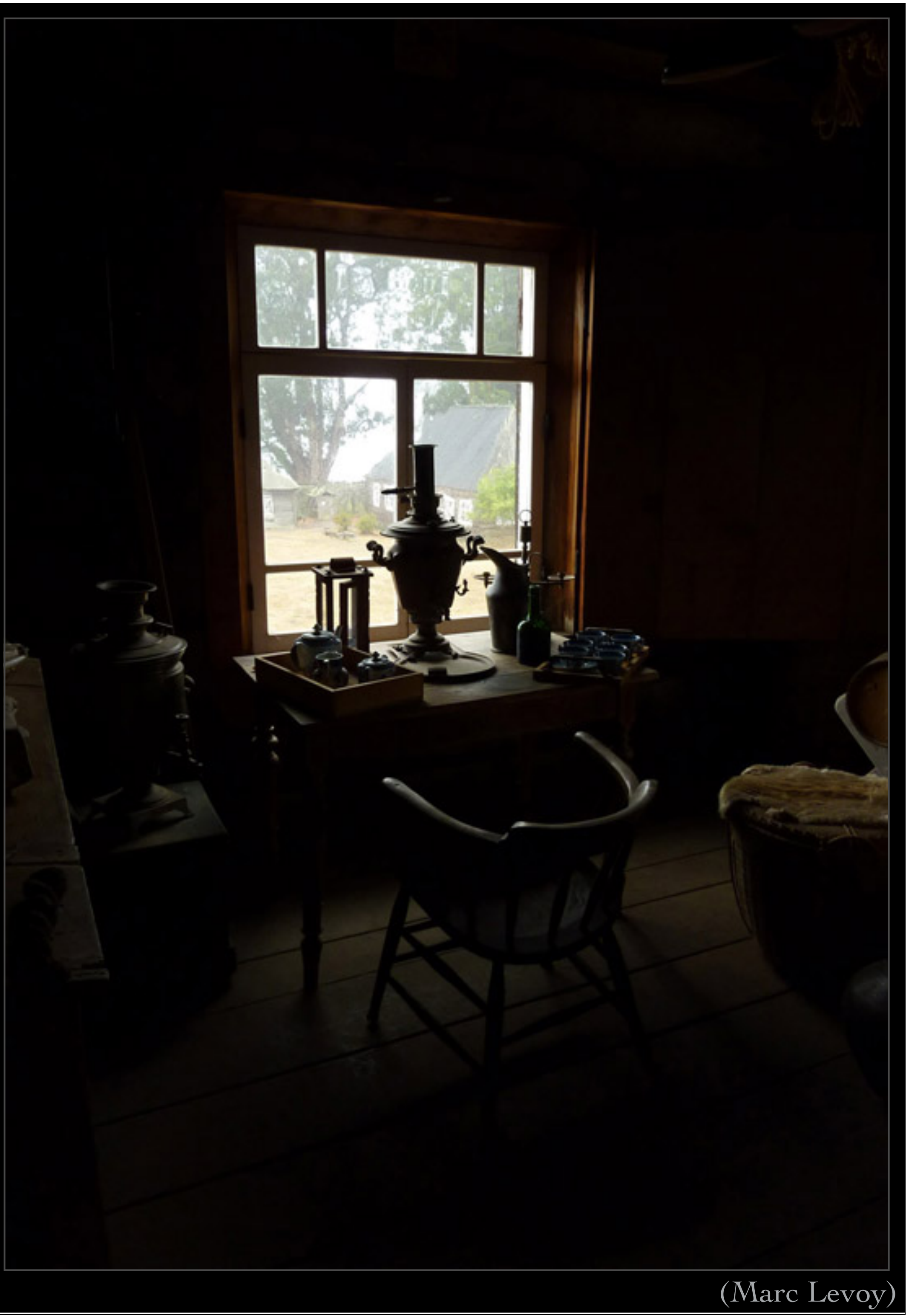


La Grande Jatte, Georges Seurat, 1884

(Marc Levoy)

Commissary,
Fort Ross, CA,
2010

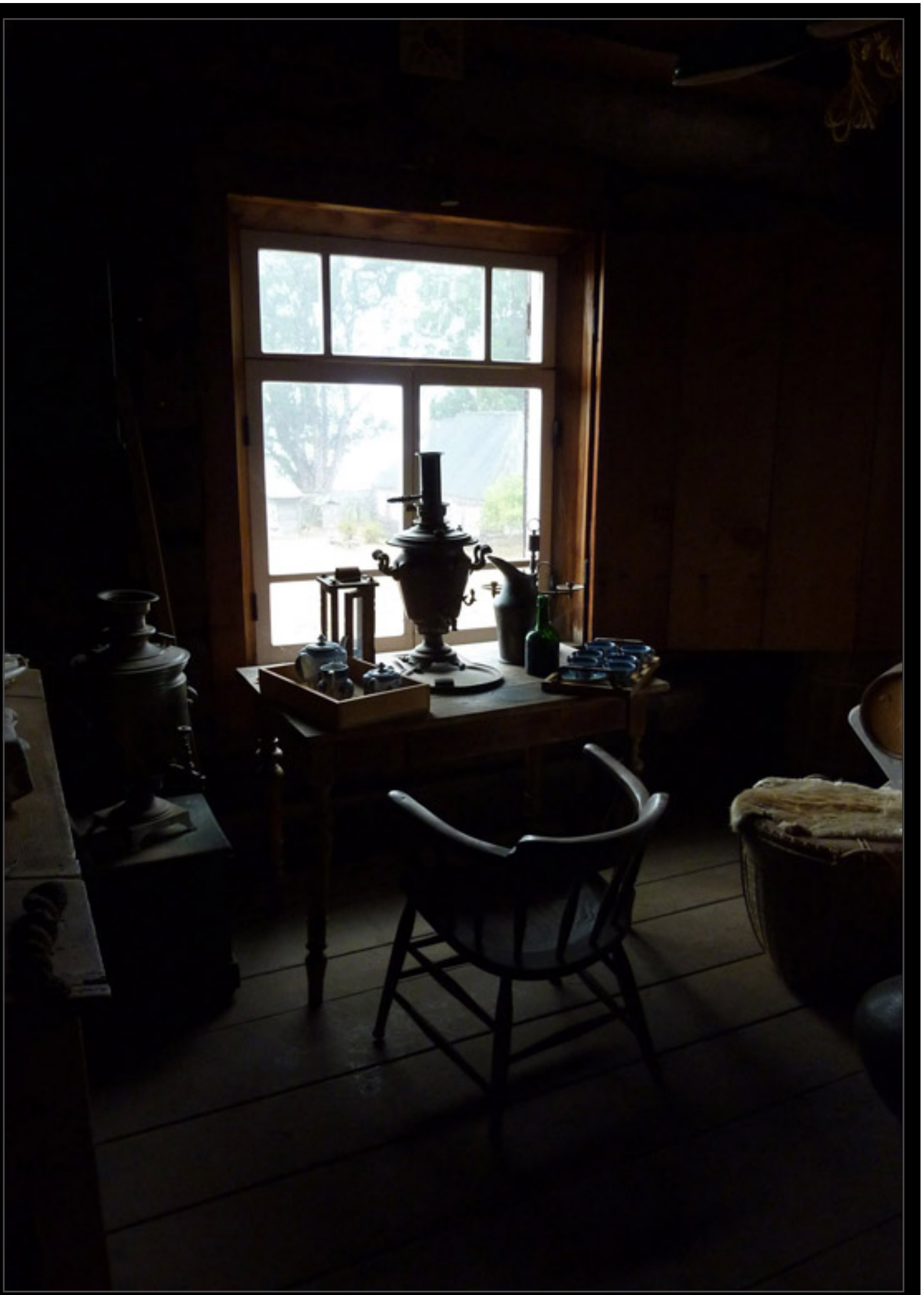
(Panasonic ZS3,
1/30s, ISO 125)



(Marc Levoy)

Commissary,
Fort Ross, CA,
2010

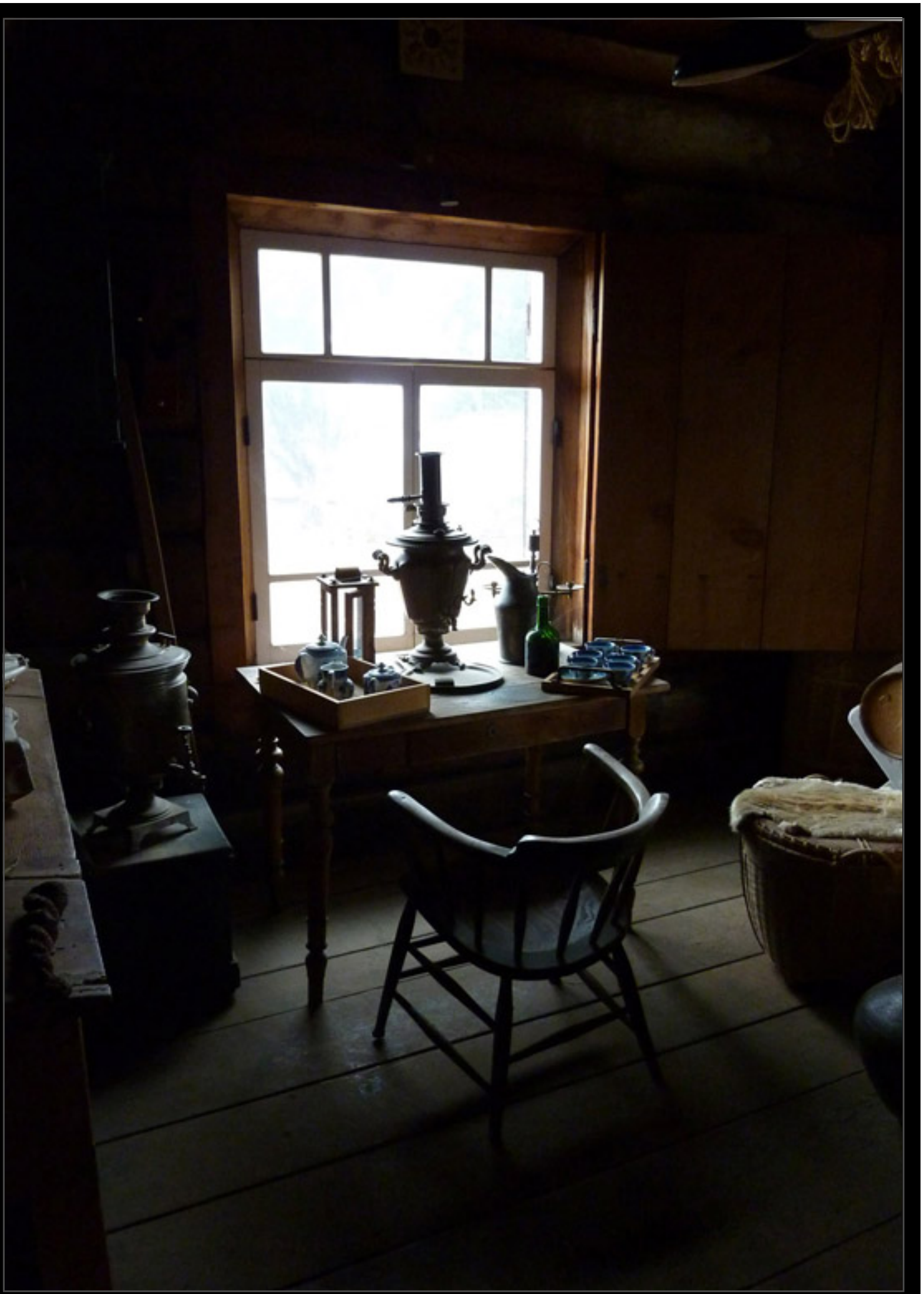
(Panasonic ZS3,
1/30s, ISO 250)



(Marc Levoy)

Commissary,
Fort Ross, CA,
2010

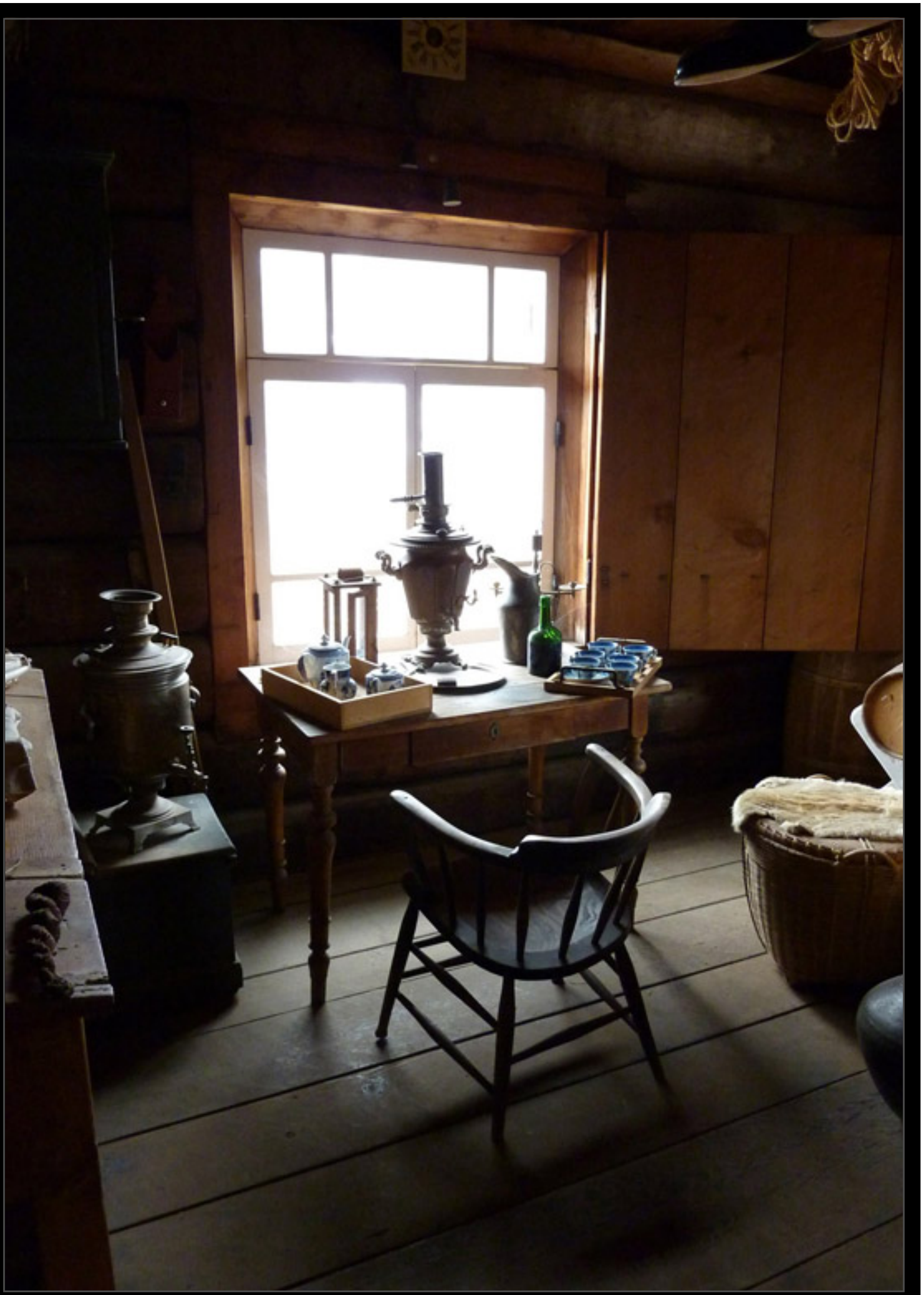
(Panasonic ZS3,
1/25s, ISO 400)



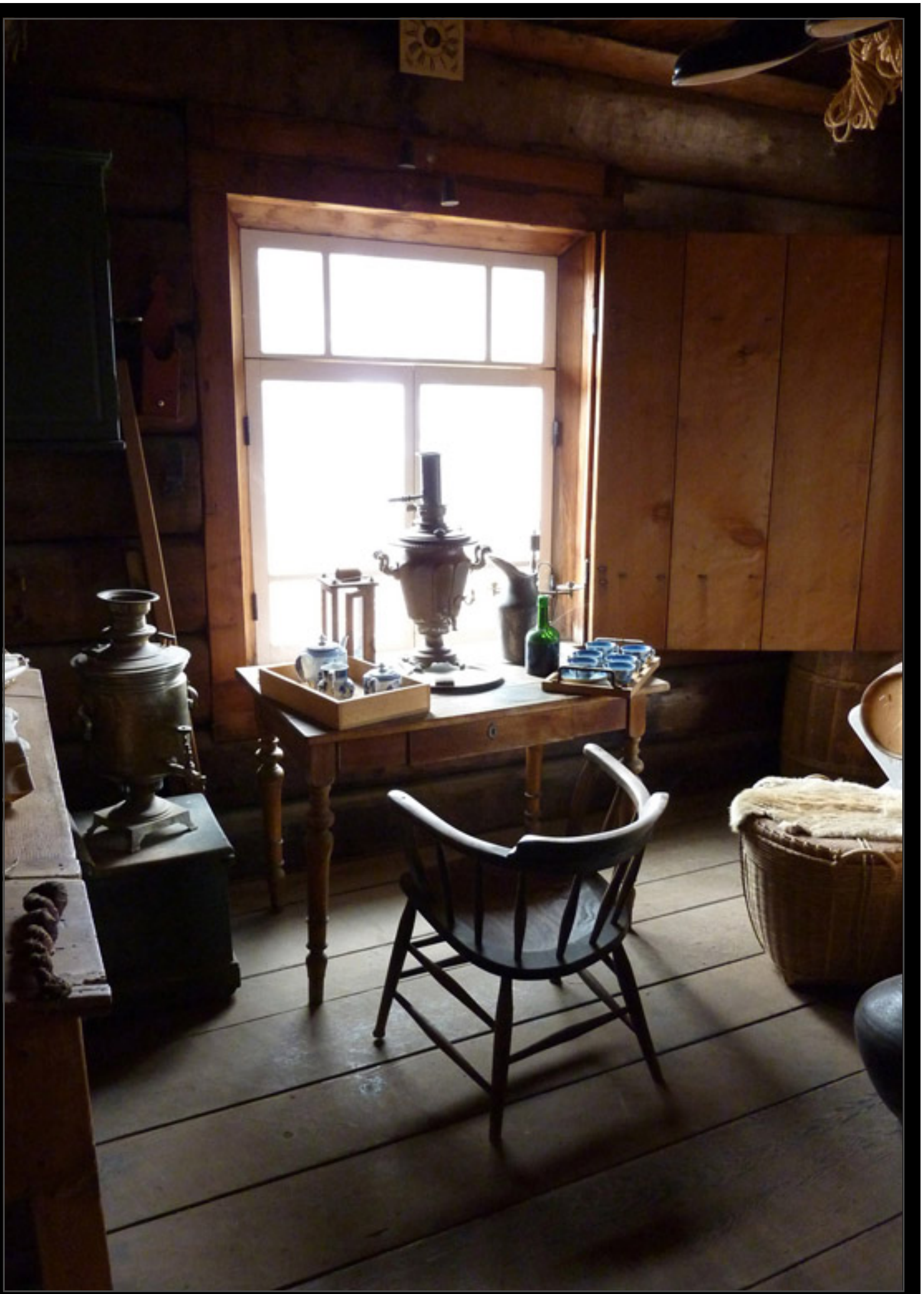
(Marc Levoy)

Commissary,
Fort Ross, CA,
2010

(Panasonic ZS3,
1/13s, ISO 400)



(Marc Levoy)



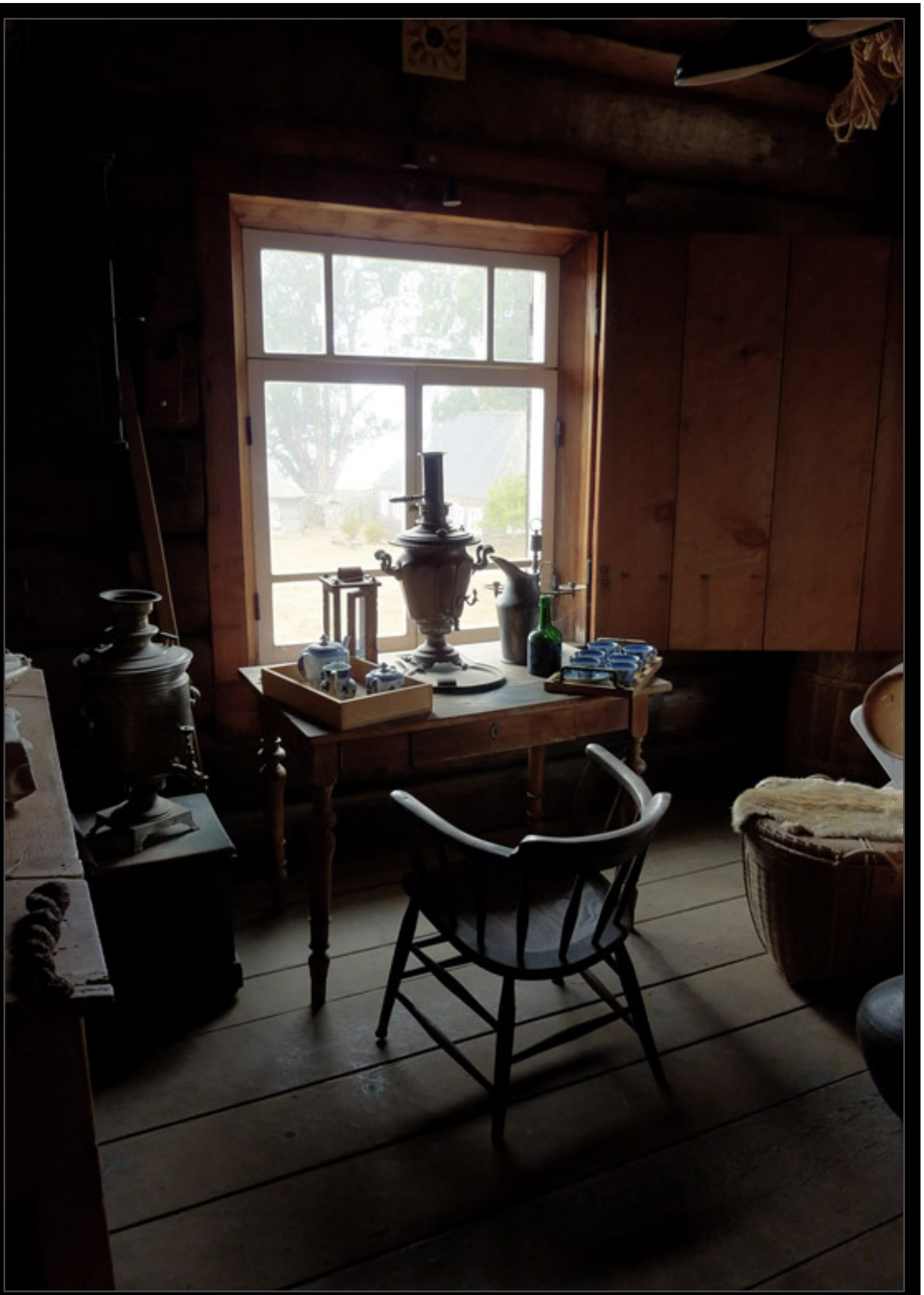
Commissary,
Fort Ross, CA,
2010

(Panasonic ZS3,
1/8s, ISO 400)

(Marc Levoy)

Commissary,
Fort Ross, CA,
2010

(tone mapped HDR using Photomatix v3.3.2's "tone compressor" algorithm)

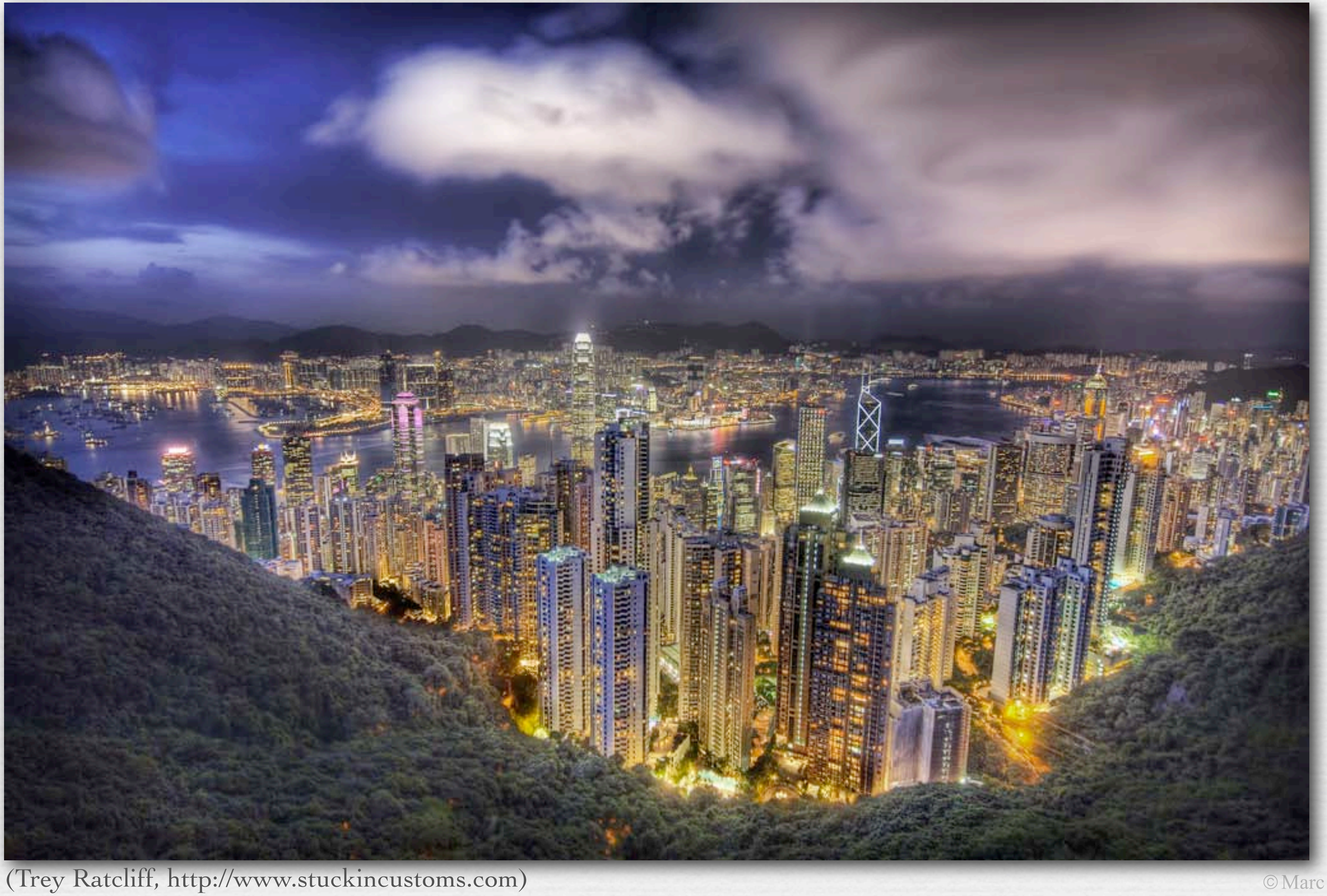


(Marc Levoy)

LEATHER

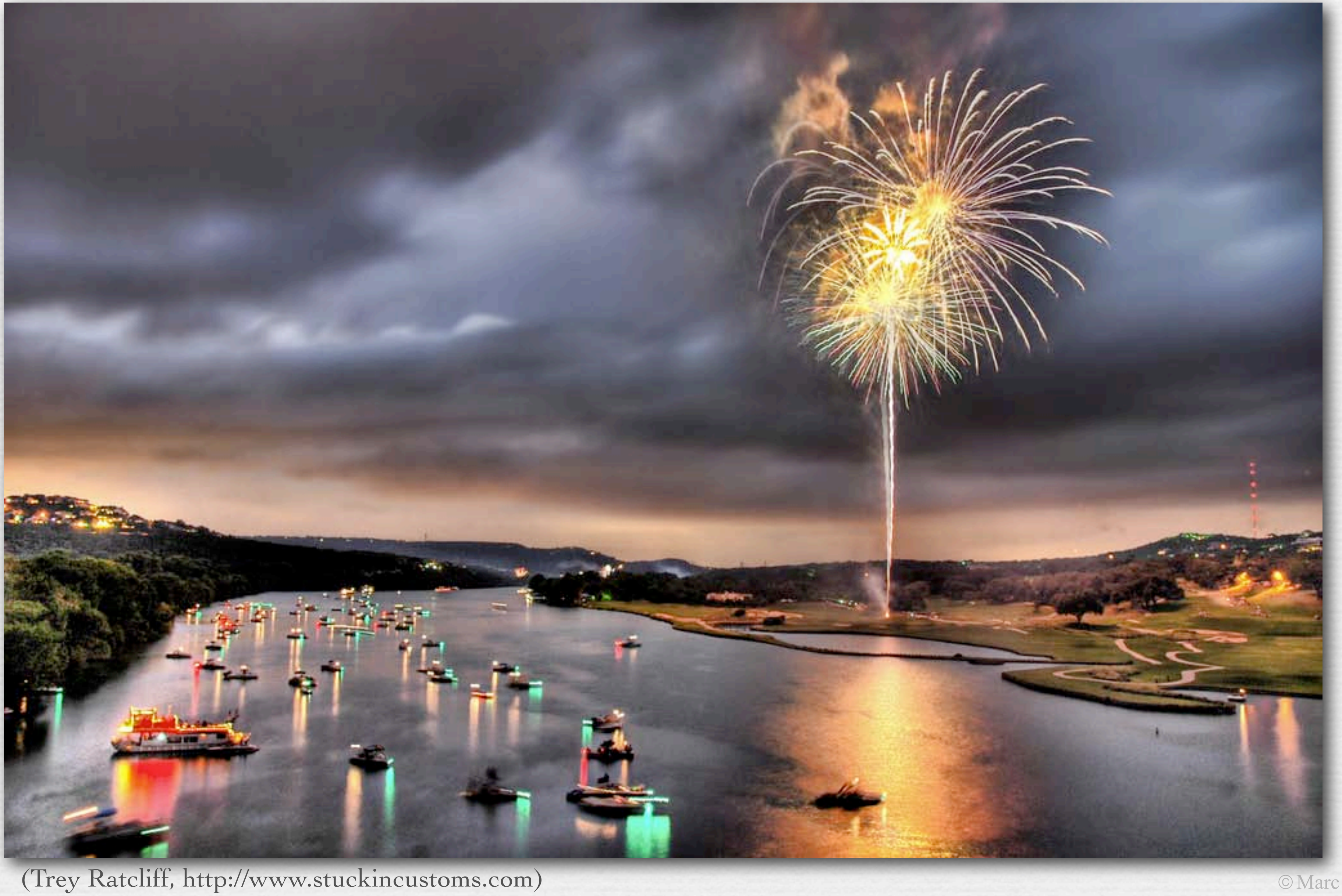
LEATHER

The HDR “look”



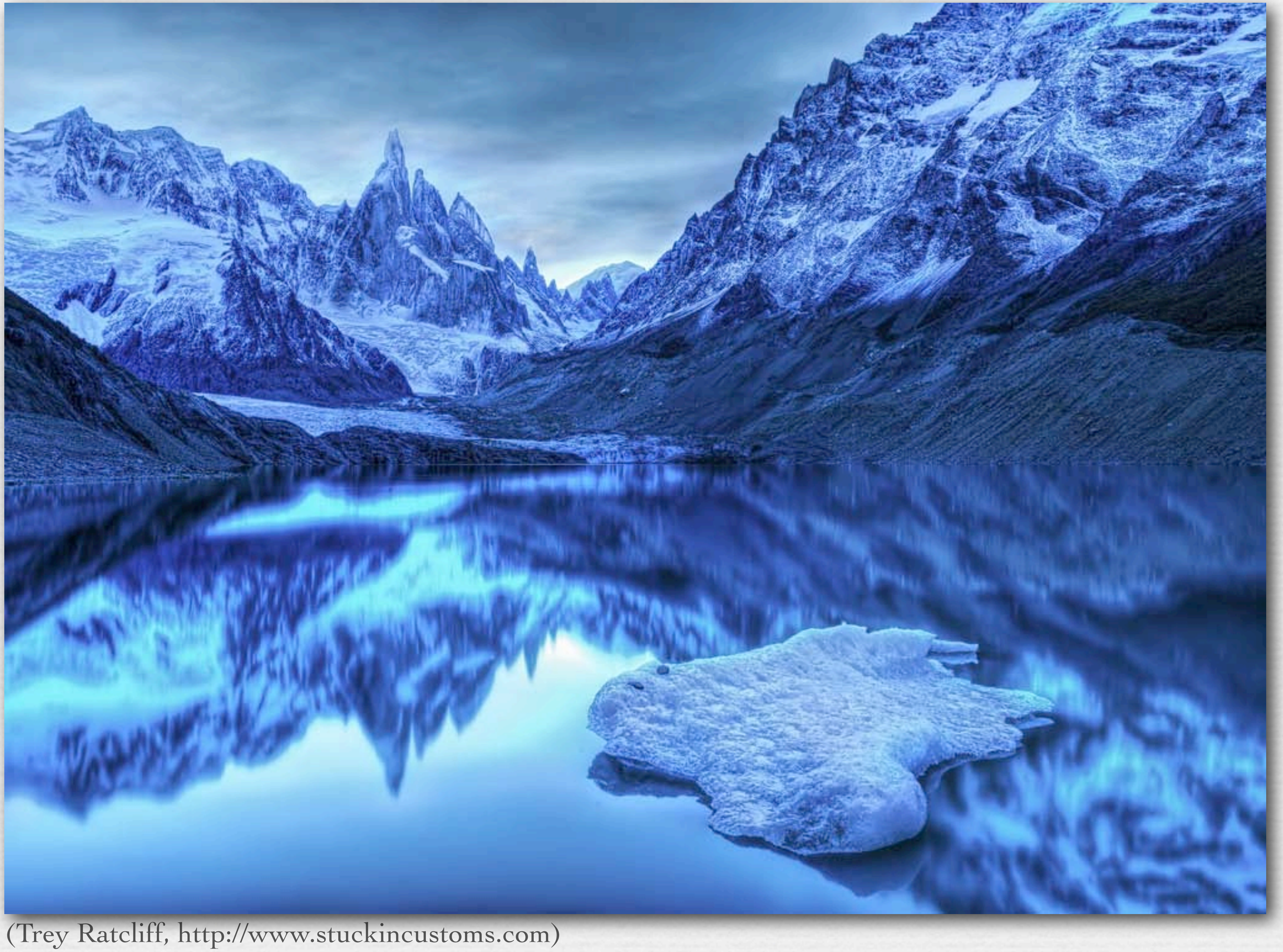
(Trey Ratcliff, http://www.stuckincustoms.com)

The HDR "look"



(Trey Ratcliff, http://www.stuckincustoms.com)

The HDR "look"



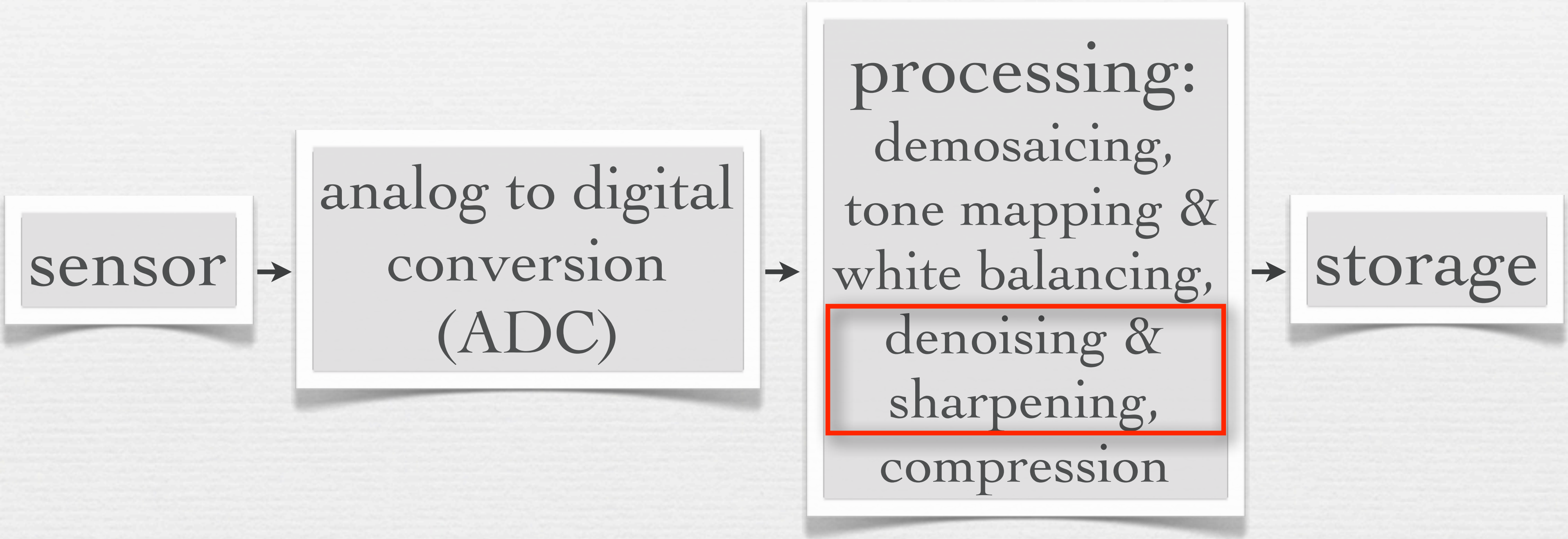
(Trey Ratcliff, http://www.stuckincustoms.com)

Recap

- high dynamic range (HDR) imaging is useful, and a new aesthetic
  - but is not necessary in all photographic situations
- low dynamic range (LDR) tone mapping methods can also be applied to HDR scenes
  - but reducing very HDR scenes to 8 bits for JPEG using only *global methods* is hard
- *local methods* reduce large-scale luminance changes (across the image) while preserving *local contrast* (across edges)
  - use bilateral filtering to avoid halos

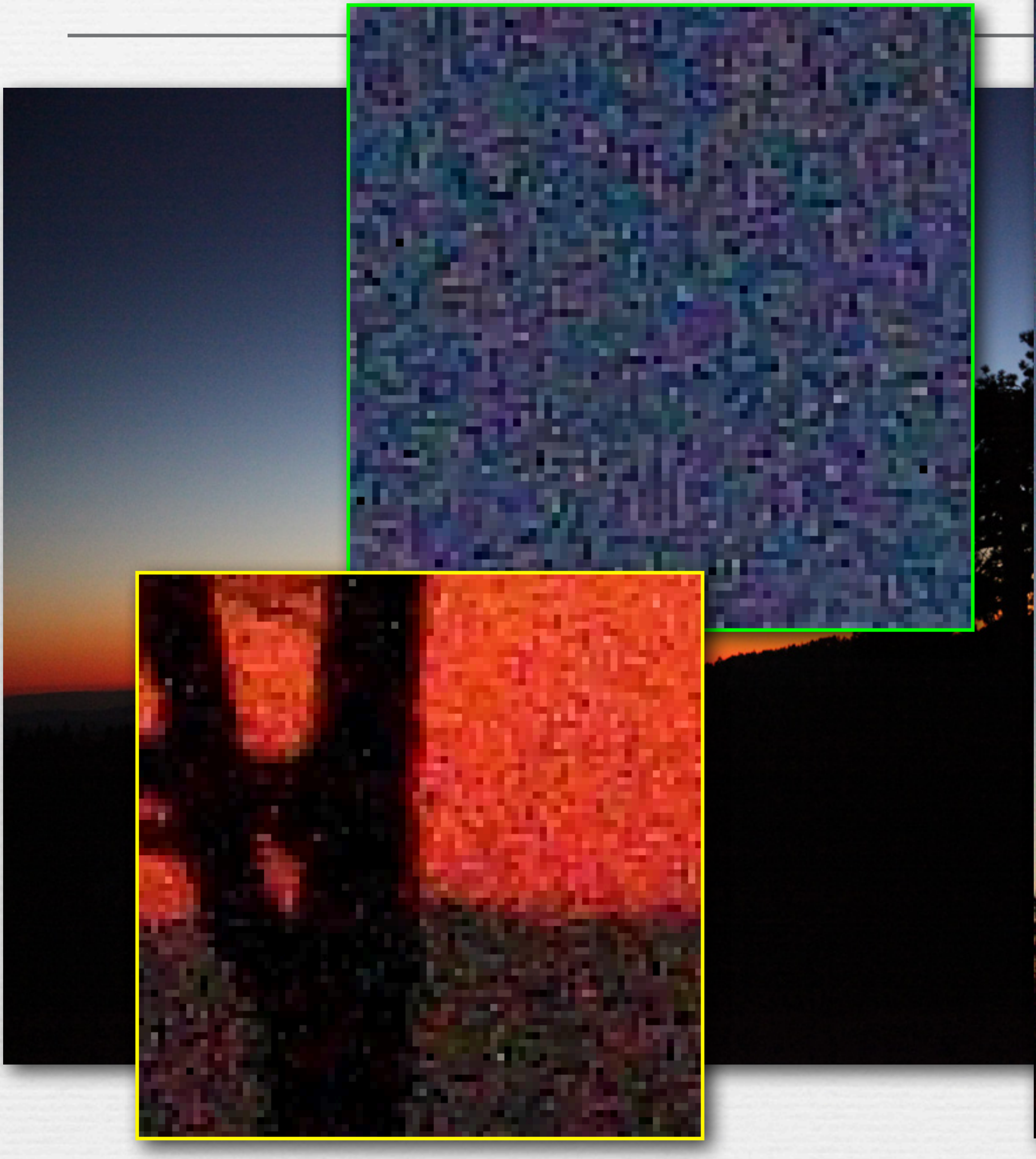
Questions?

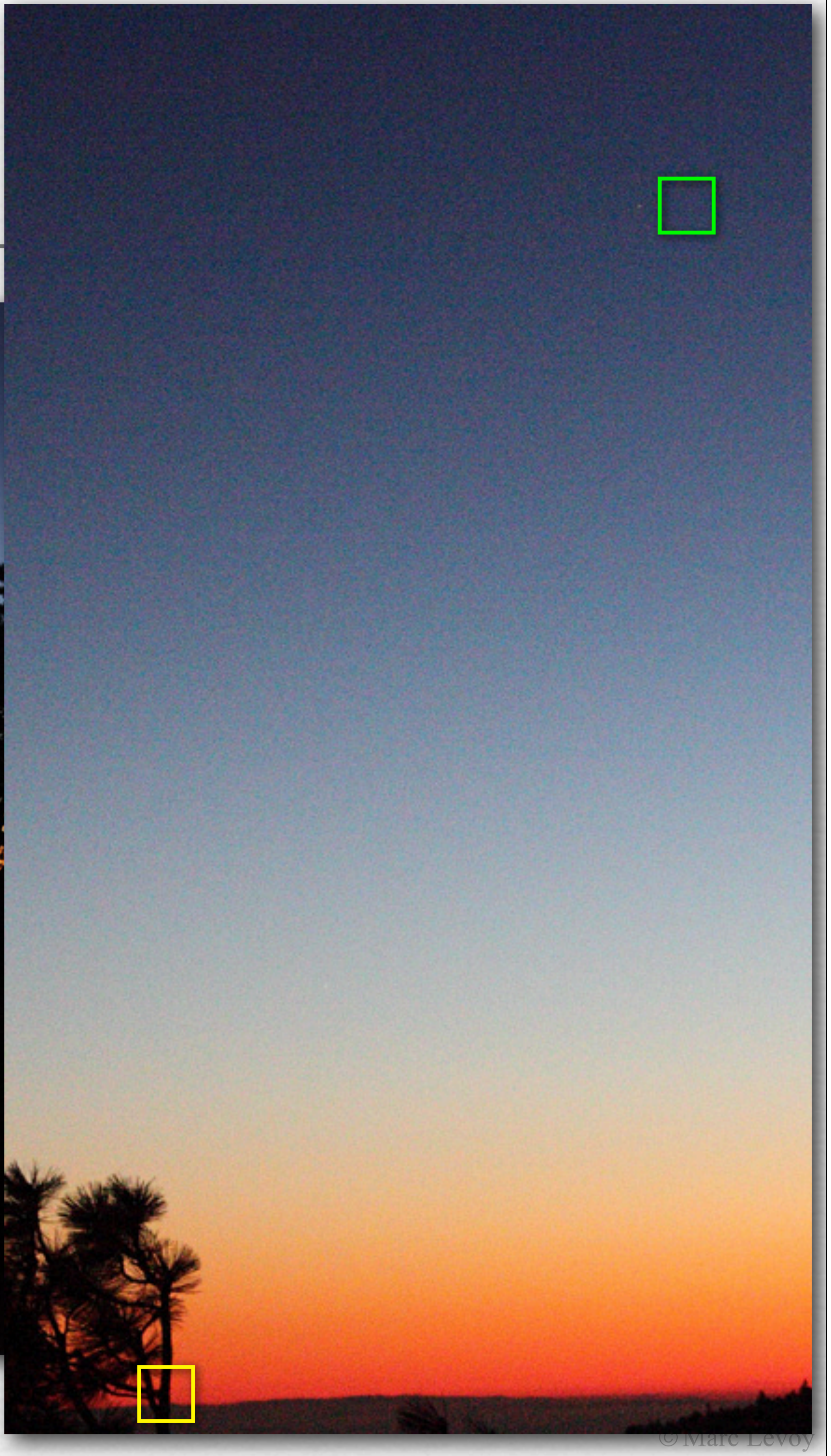
Camera pixel pipeline



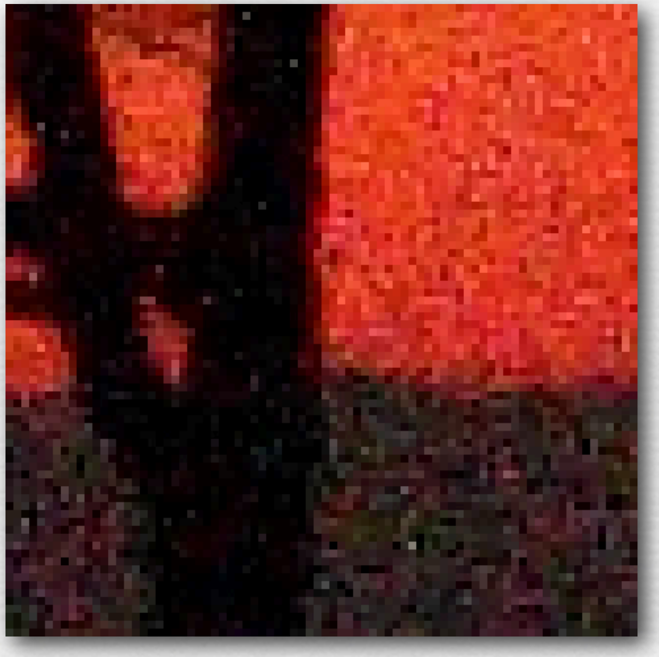
sensor → analog to digital conversion (ADC) → processing: demosaicing, tone mapping & white balancing, denoising & sharpening, compression → storage

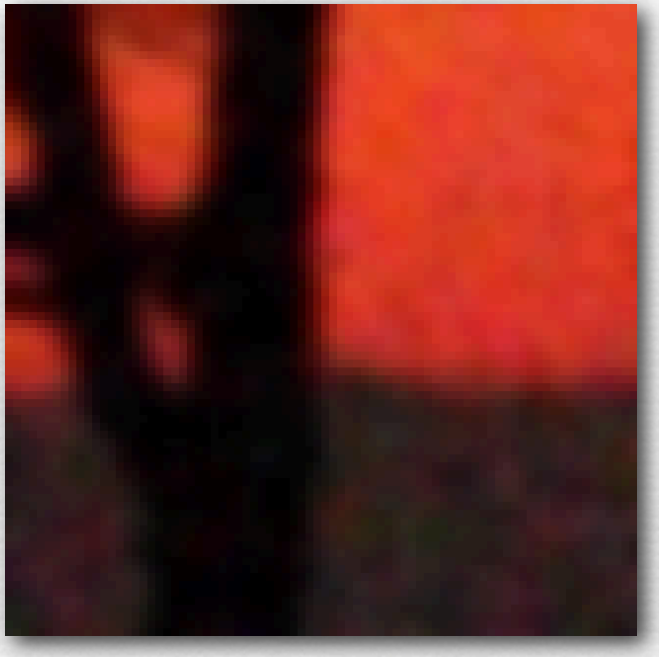
Canon 5D II at dusk

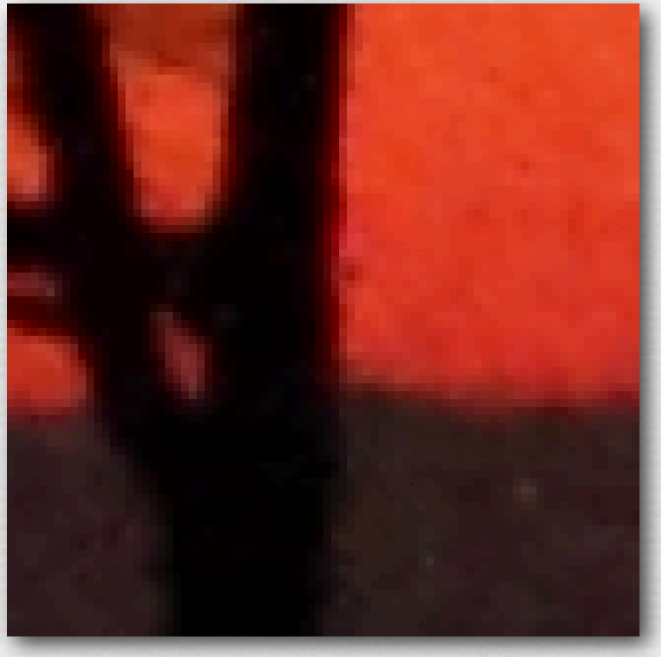




Denoising


RAW (ISO 6400)

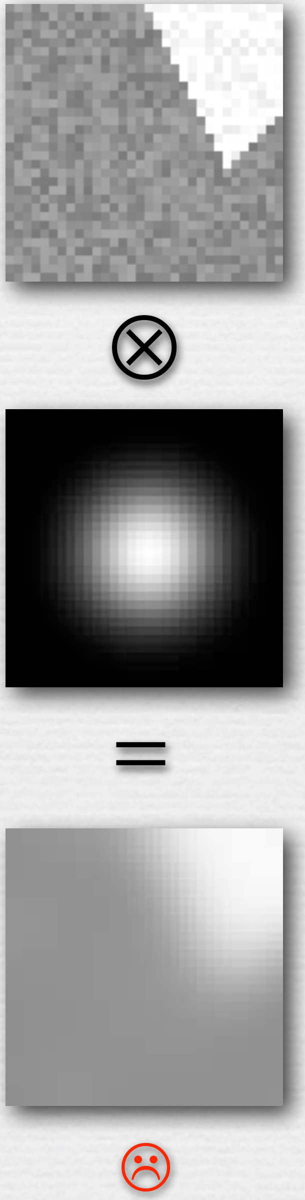

Gaussian blur, radius = 1.3


Canon denoising

- goal is to remove sensor noise
  - blurring works, but also destroys edges
  - I don’t know what Canon does,
    but here’s something that works...

Bilateral filtering [Tomasi ICCV 1998]

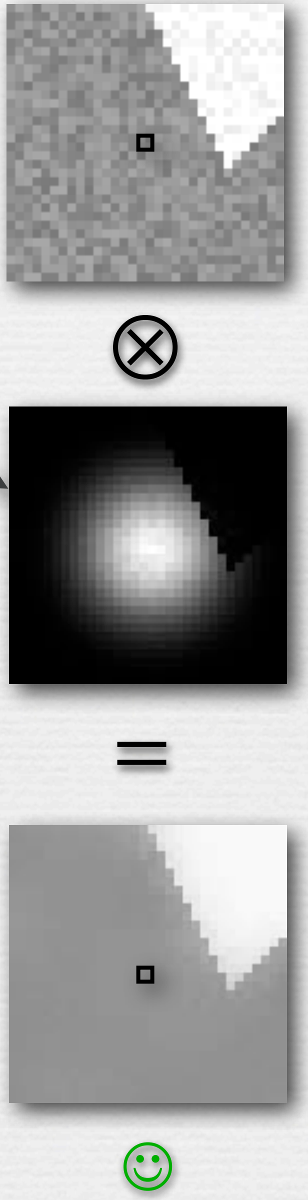
- images are often piecewise constant with noise added
  - in this case, nearby pixels are often a different noisy measurement of the same data


(simulated)

- simple blurring doesn’t work
  - because it also blurs the edges
- we should blur only within each constant-colored scene region
  - not across the edges between regions

Bilateral filtering

- if the pixels are similar in intensity, they are probably from the same region of the scene



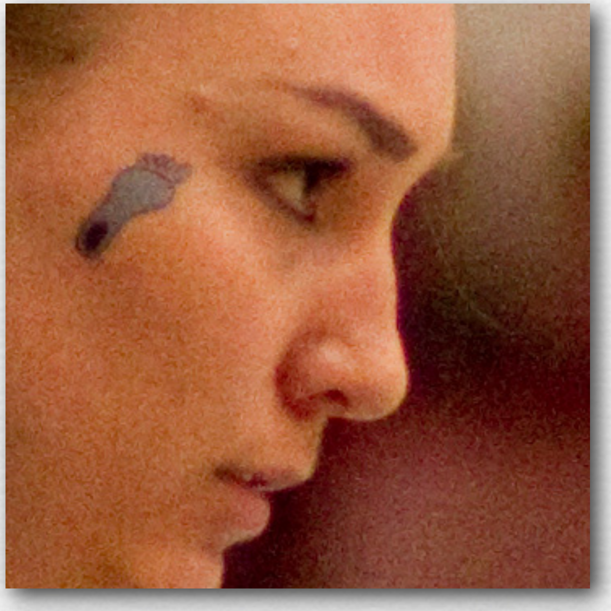
effective filter weights are thus different for each pixel of input

- so perform a convolution where the weight applied to nearby pixels in the summation falls off
  - with increasing *(x,y)* distance from the pixel being blurred, and
  - with increasing intensity difference from the pixel being blurred
- i.e. blur in *domain* and *range* dimensions!

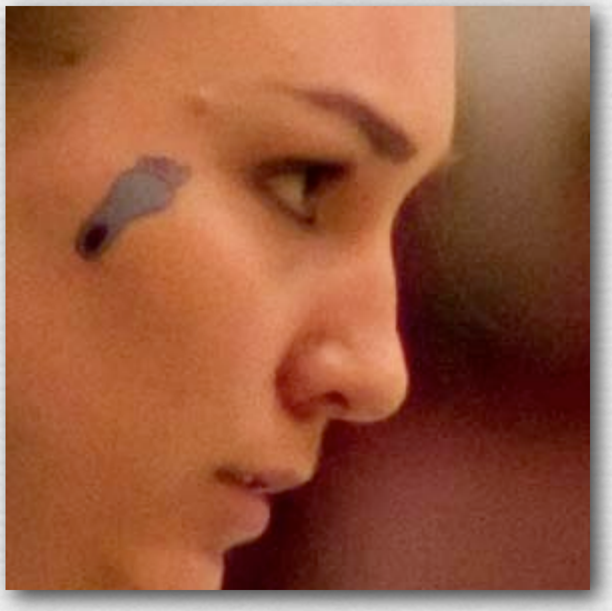
Example of bilateral filtering

Women's gymnastics

(Canon 7D, 1/1000 sec, ISO 3200, f/1.8, 85mm)

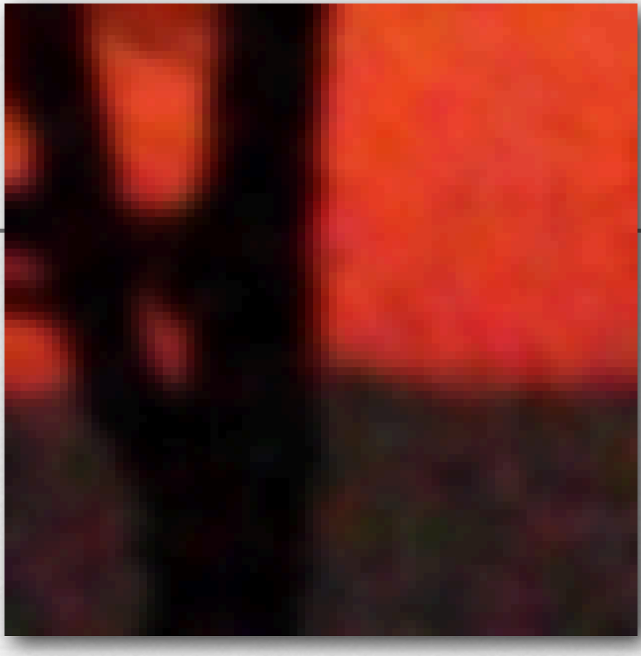


original

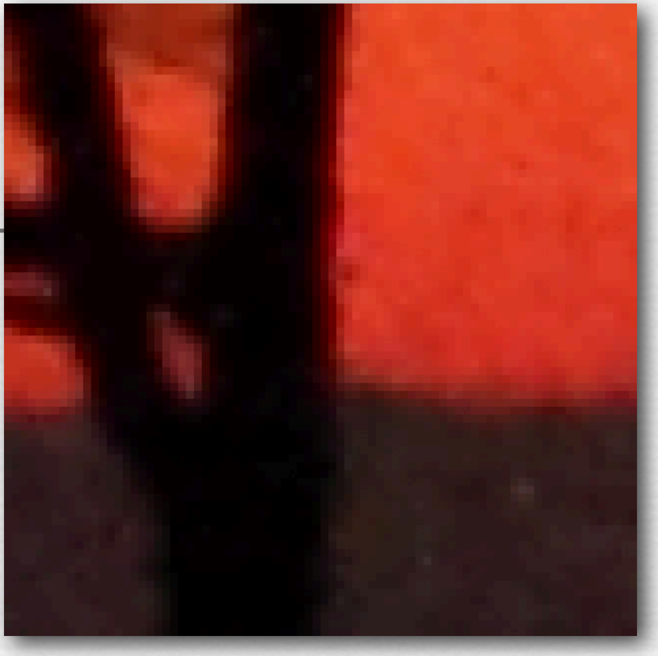


denoised in Noise Ninja

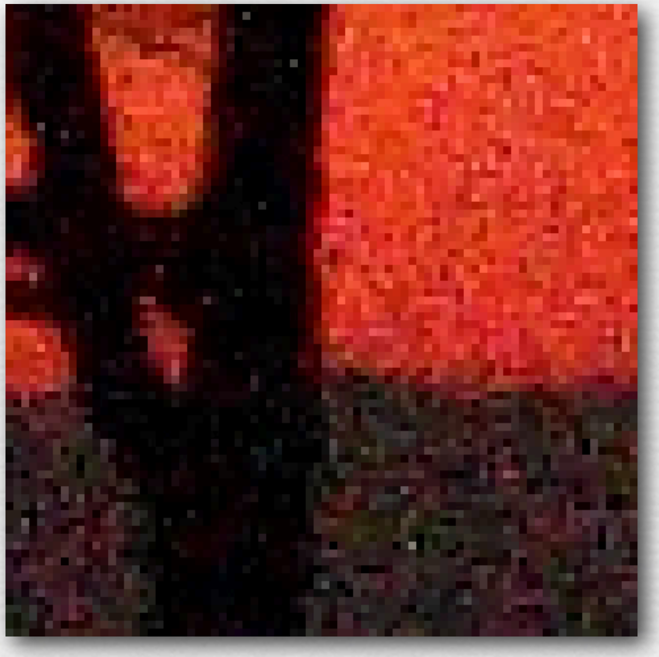
Denoising



Gaussian blur, radius = 1.3

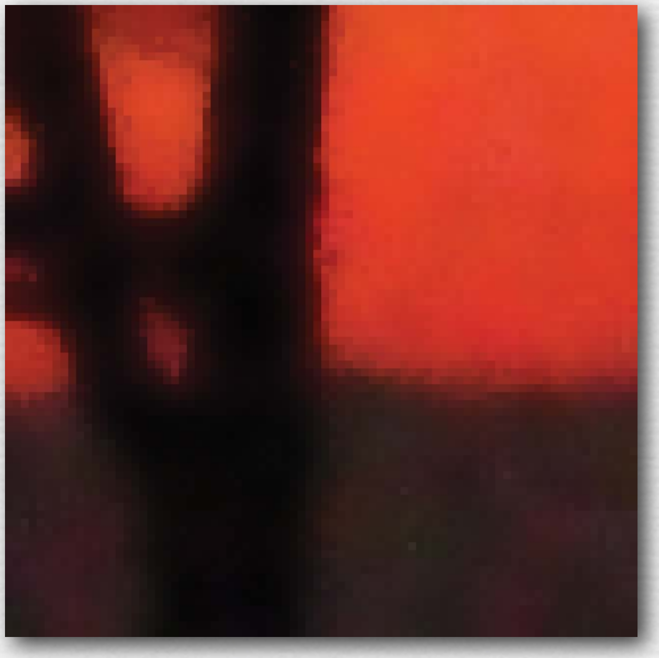


Canon denoising



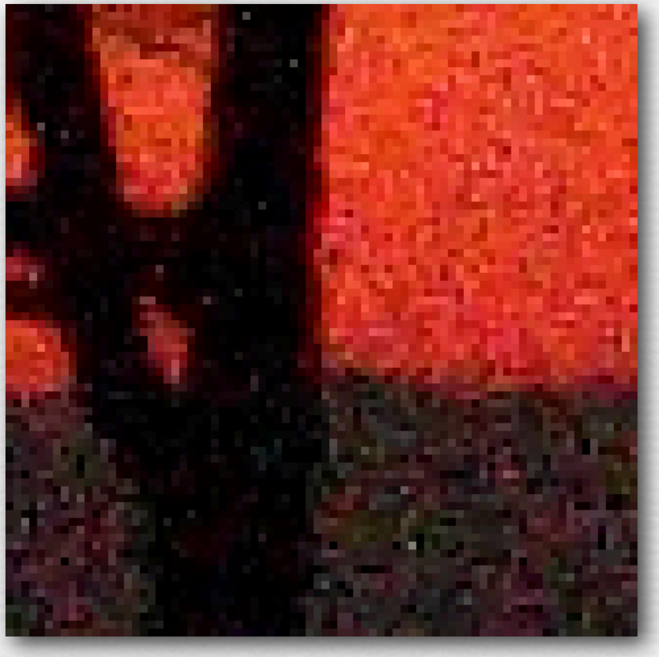
RAW (ISO 6400)

- bilateral filtering removes sensor noise without blurring edges
- can easily be extended to RGB

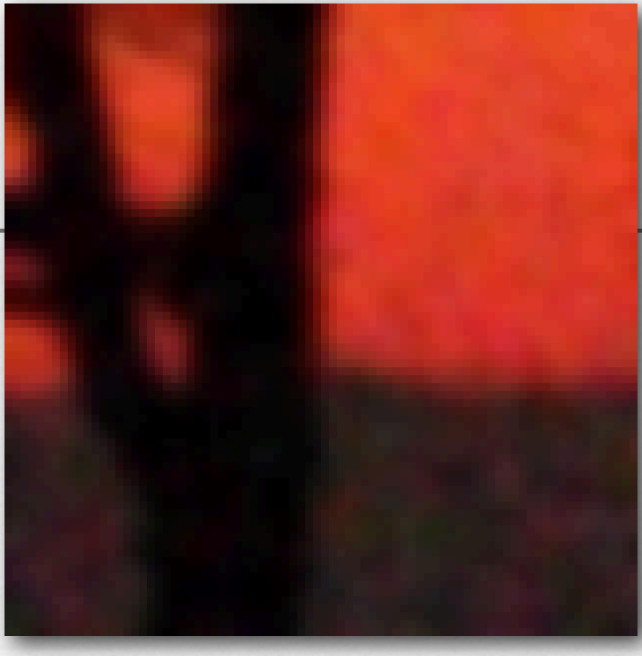


bilateral filtering

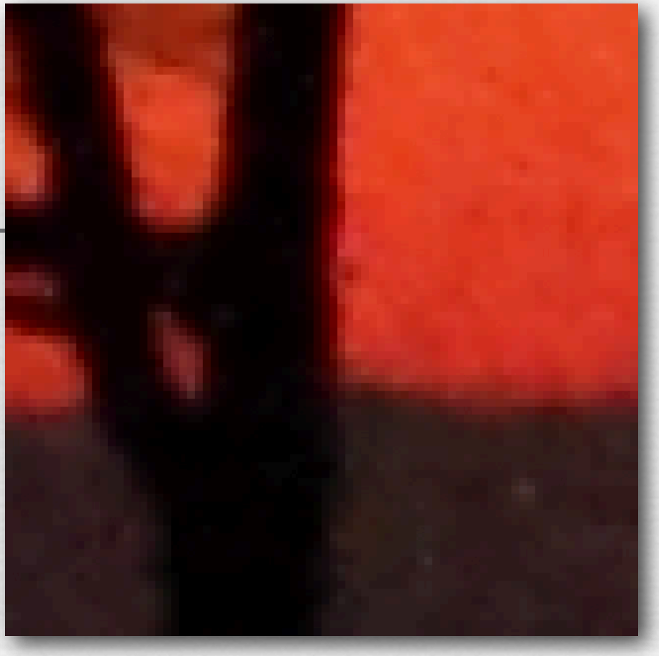
Denoising



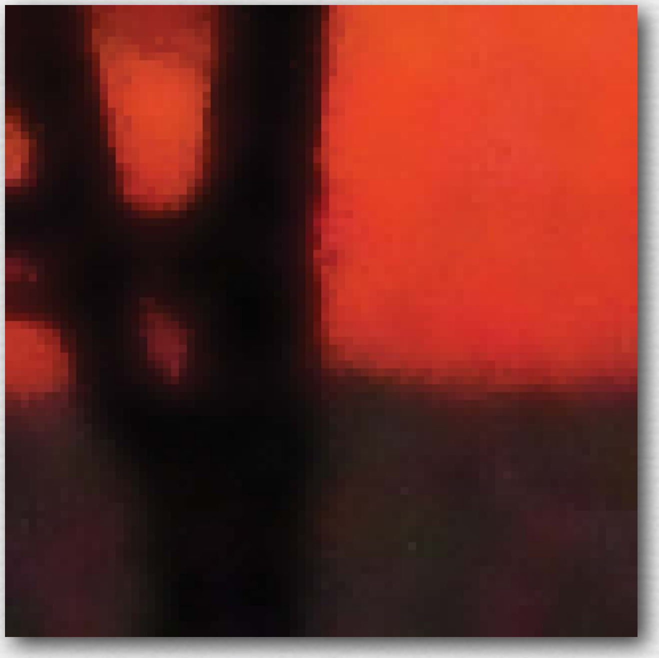
RAW (ISO 6400)



Gaussian blur, radius = 1.3



Canon denoising



bilateral filtering

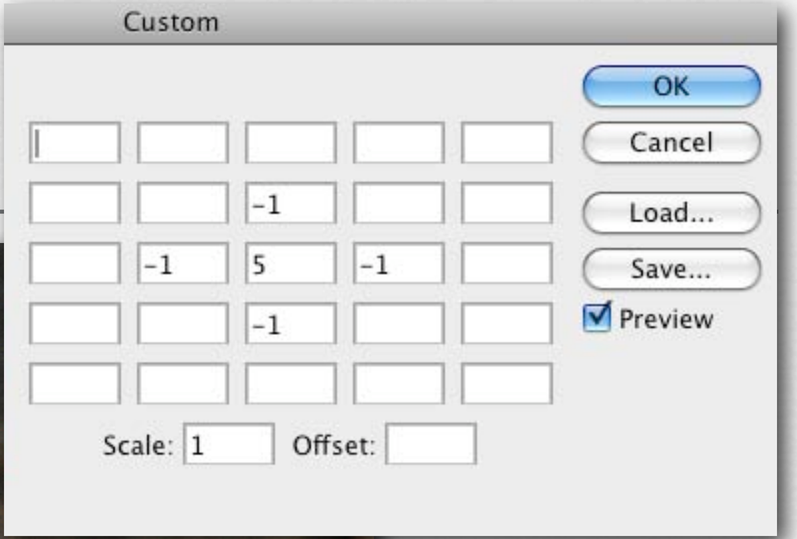
- can be applied more (or less) strongly to chrominance than luminance
- can be combined with demosaicing
- active area of research...

Sharpening

original

(Marc Levoy)

Sharpening



Custom

		-1		
	-1	5	-1	
		-1		

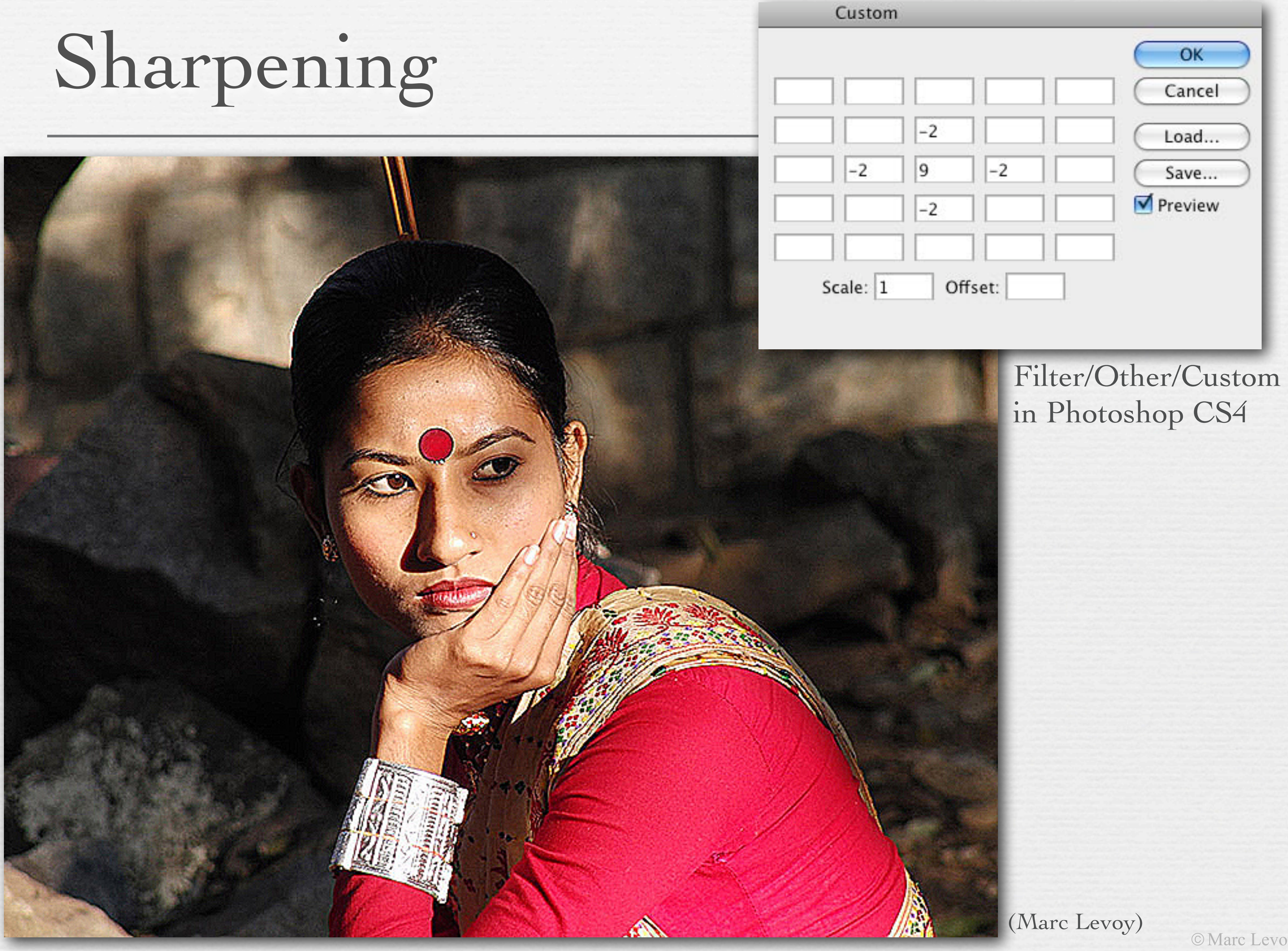
Scale: 1 Offset:

☑ Preview

Filter/Other/Custom in Photoshop CS4

(Marc Levoy)

Sharpening



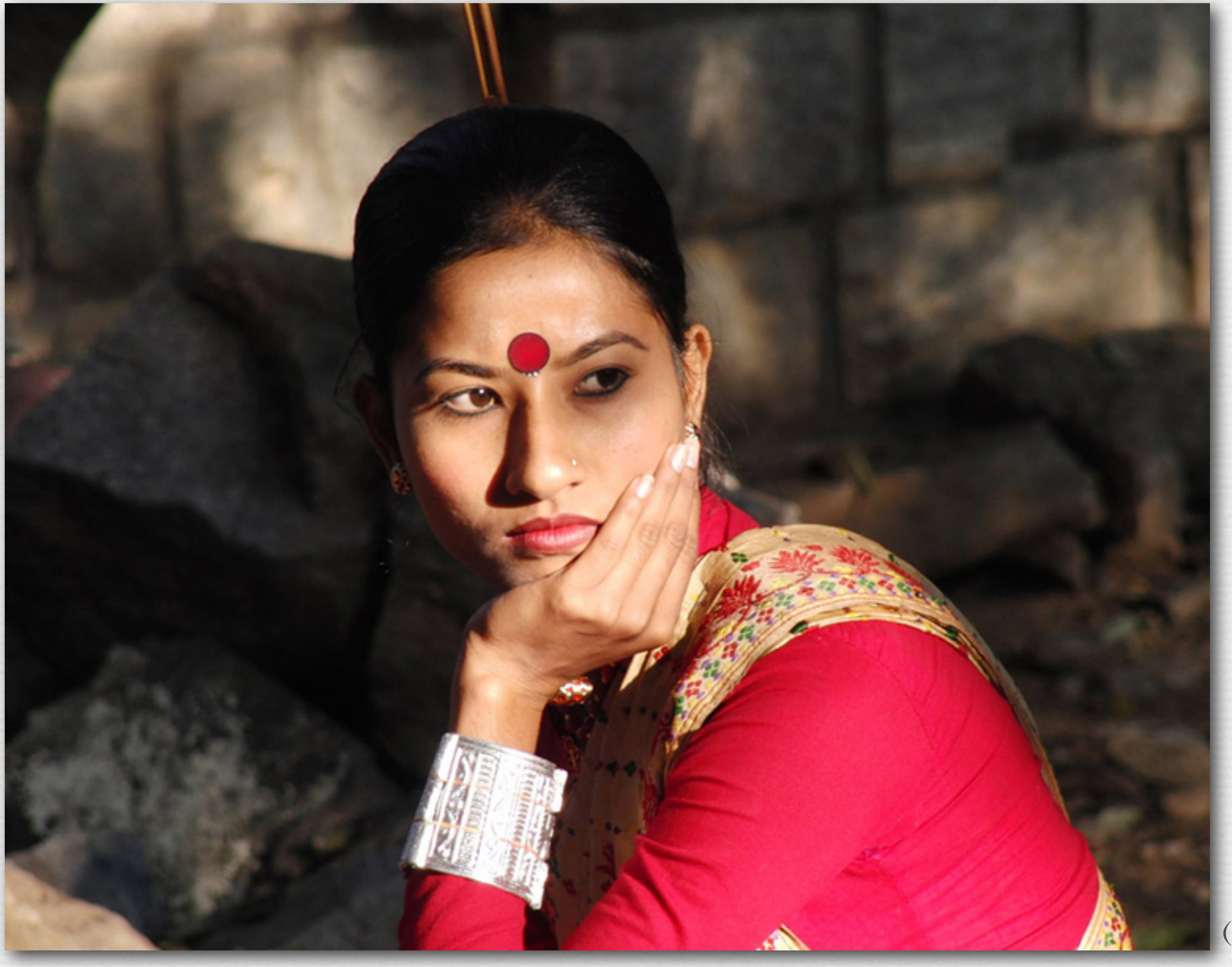
		-2		
	-2	9	-2	
		-2		

Scale: 1 Offset:

Filter/Other/Custom in Photoshop CS4

(Marc Levoy)

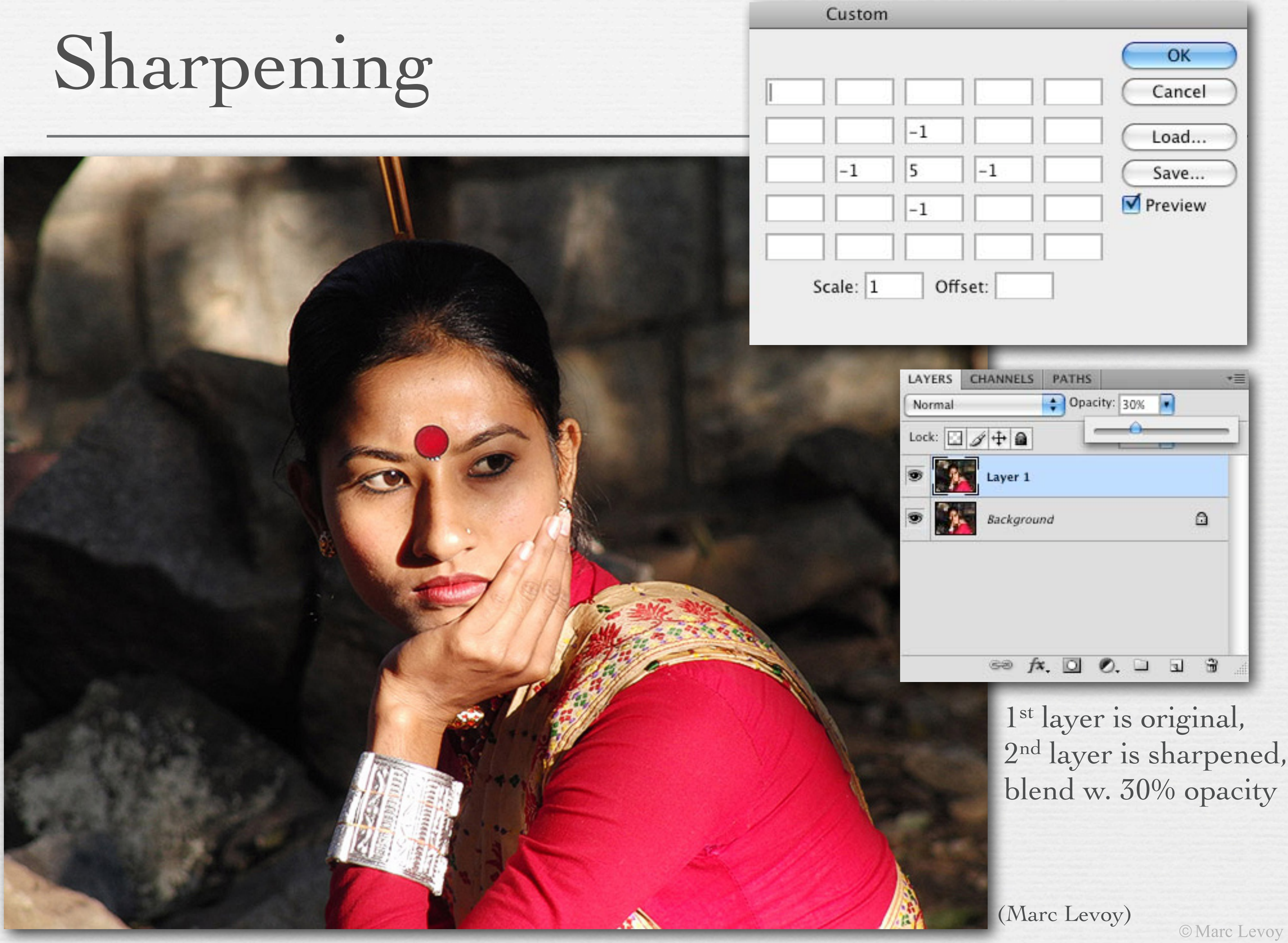
Sharpening



original

(Marc Levoy)

Sharpening



Custom

		-1		
	-1	5	-1	
		-1		

Scale: 1 Offset:

☑ Preview

Normal Opacity: 30%

Lock:

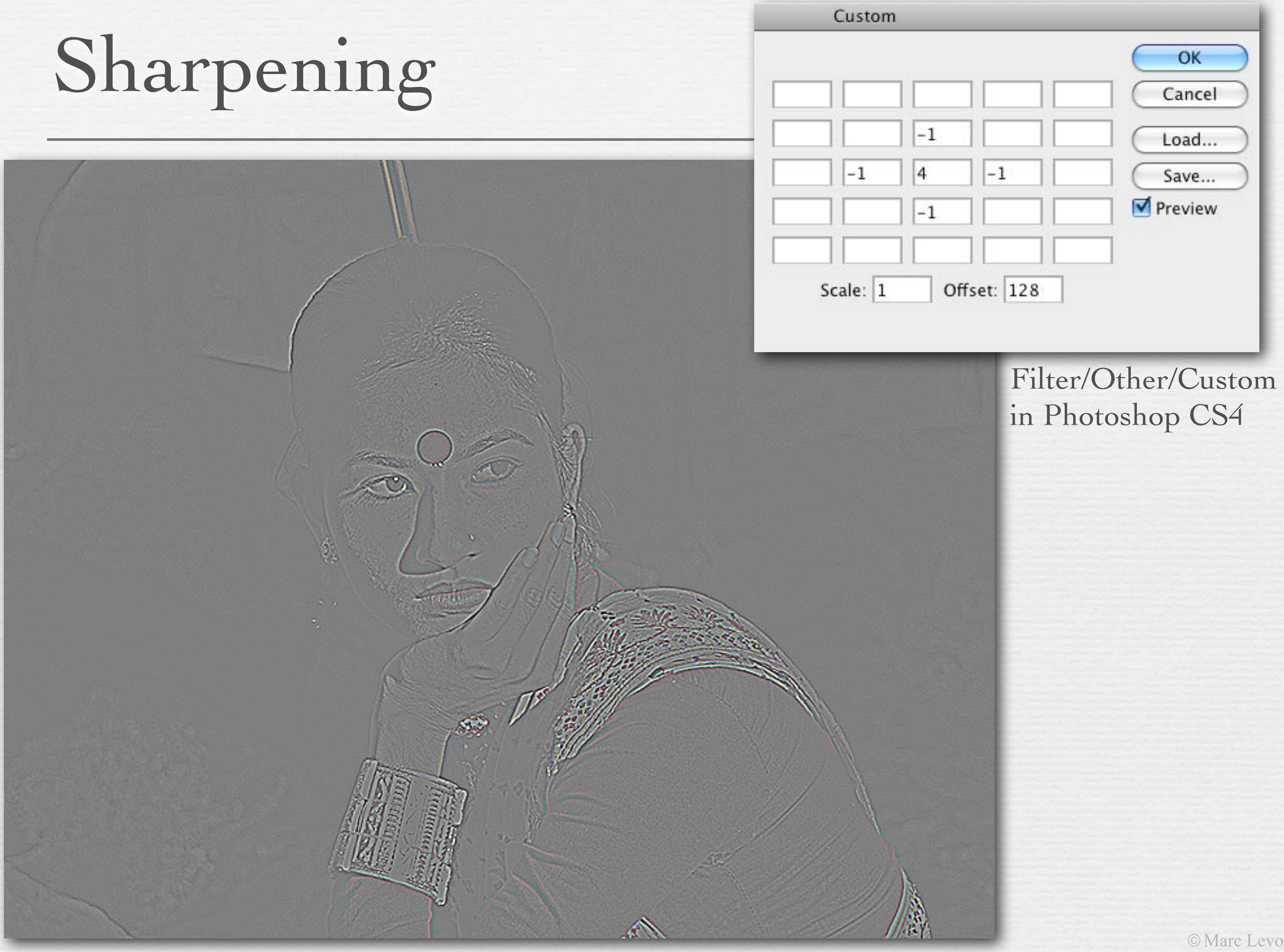
Layer 1

Background

1st layer is original,
2nd layer is sharpened,
blend w. 30% opacity

(Marc Levoy)

Sharpening

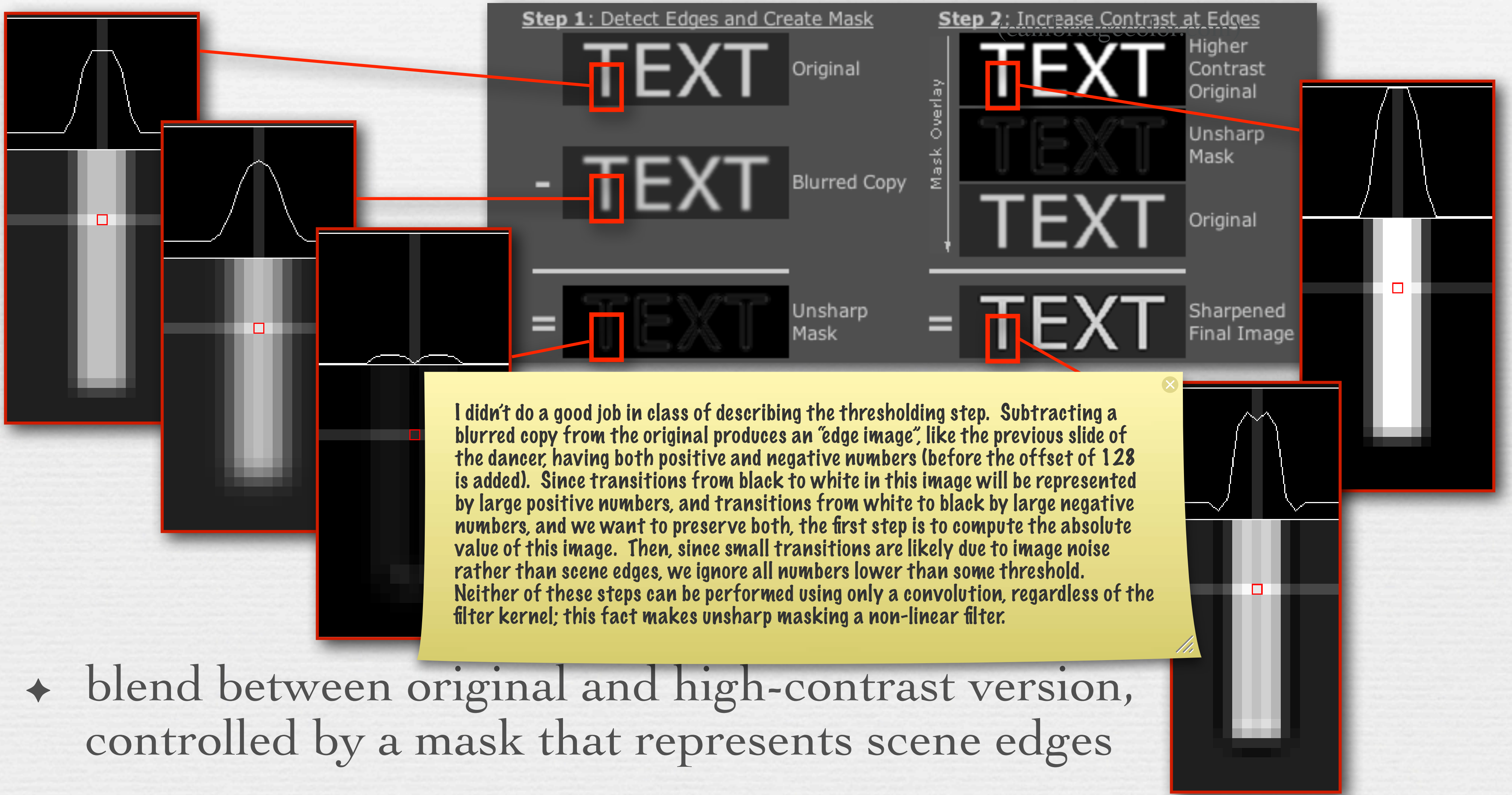


		-1		
	-1	4	-1	
		-1		

Scale: 1 Offset: 128

Filter/Other/Custom
in Photoshop CS4

Unsharp masking

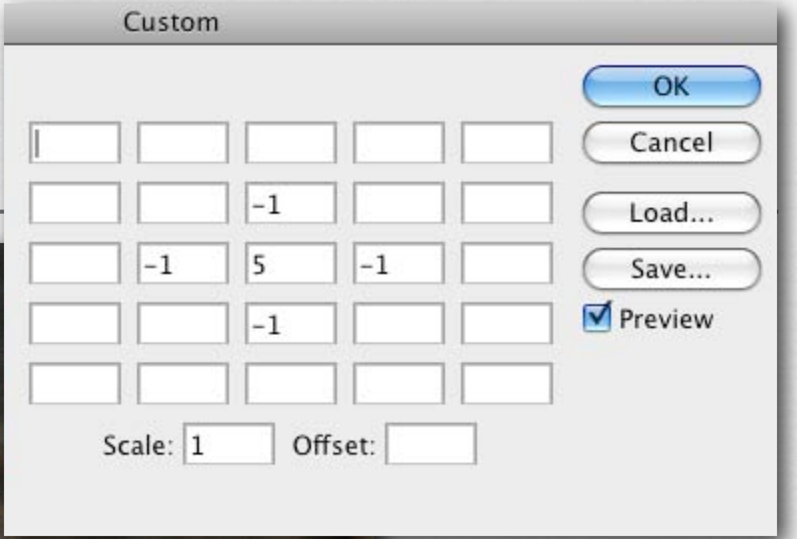


(cambridgecolor.com)

I didn't do a good job in class of describing the thresholding step. Subtracting a blurred copy from the original produces an "edge image", like the previous slide of the dancer, having both positive and negative numbers (before the offset of 128 is added). Since transitions from black to white in this image will be represented by large positive numbers, and transitions from white to black by large negative numbers, and we want to preserve both, the first step is to compute the absolute value of this image. Then, since small transitions are likely due to image noise rather than scene edges, we ignore all numbers lower than some threshold. Neither of these steps can be performed using only a convolution, regardless of the filter kernel; this fact makes unsharp masking a non-linear filter.

- blend between original and high-contrast version, controlled by a mask that represents scene edges
- dropping (thresholding) the darkest mask pixels avoids sharpening noise, and makes the filter non-linear

Sharpening



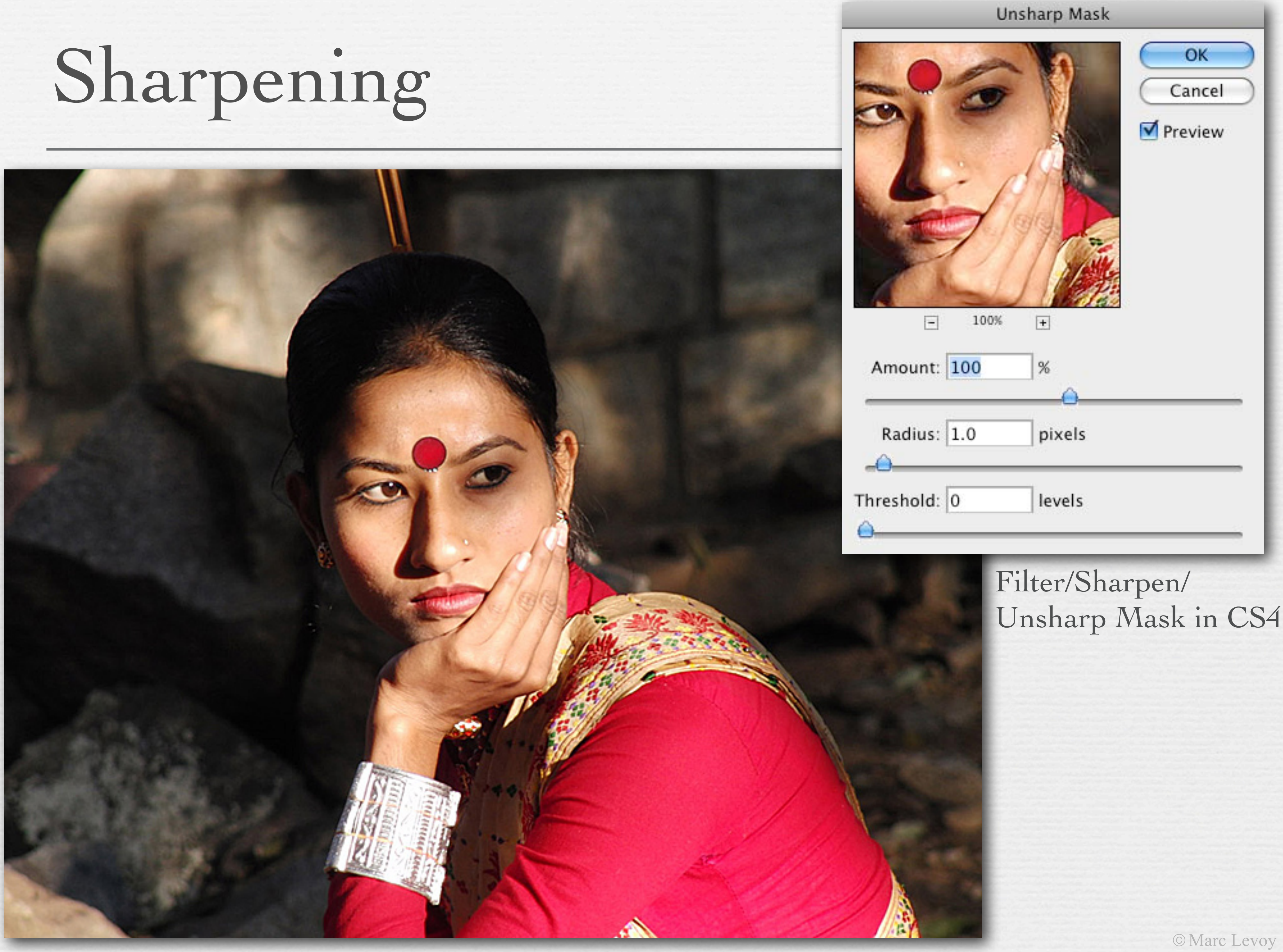
Custom

		-1		
	-1	5	-1	
		-1		

Scale: 1 Offset:

Filter/Other/Custom
in Photoshop CS4

Sharpening



Unsharp Mask

☑ Preview

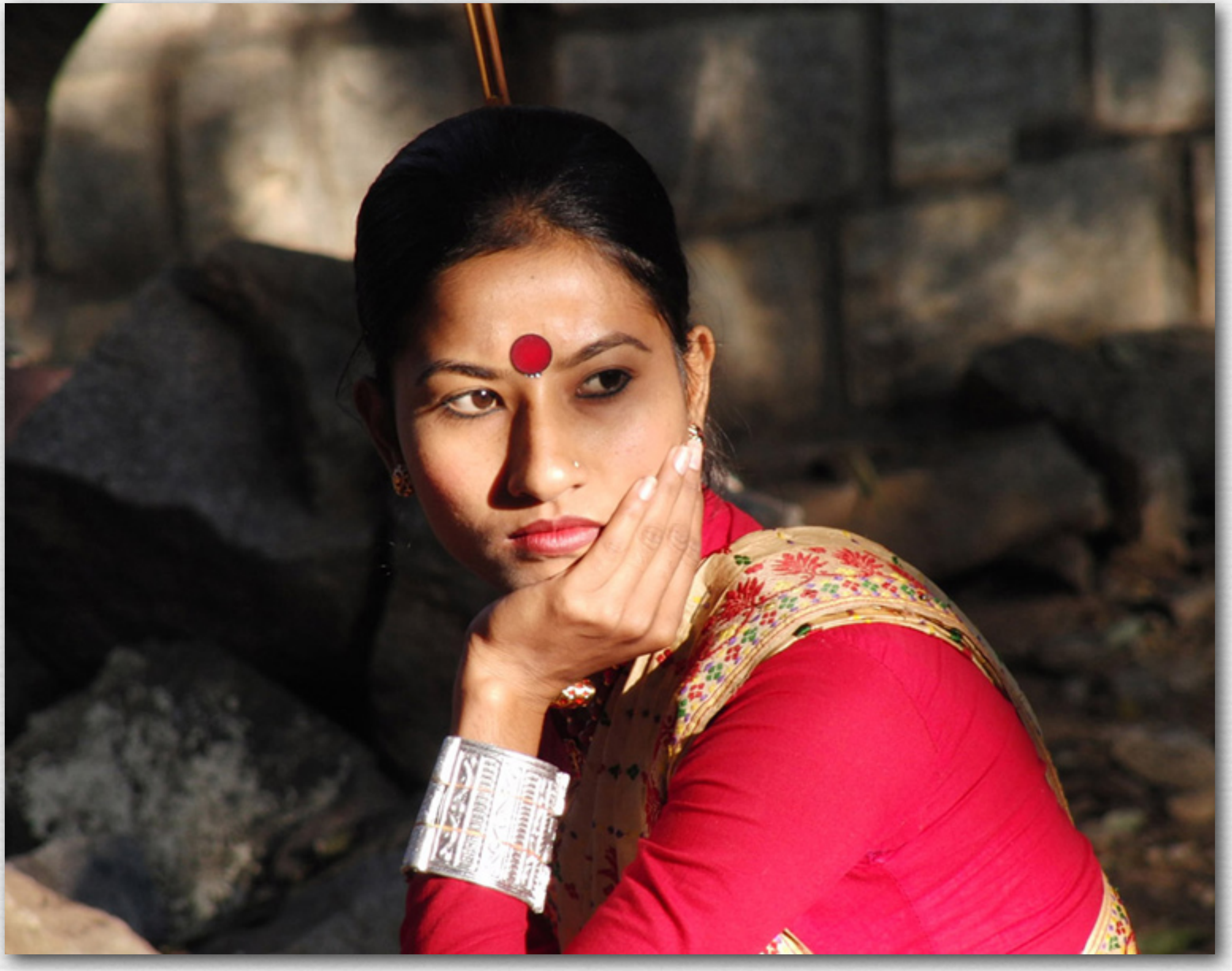
Amount: 100 %

Radius: 1.0 pixels

Threshold: 0 levels

Filter/Sharpen/
Unsharp Mask in CS4

Sharpening



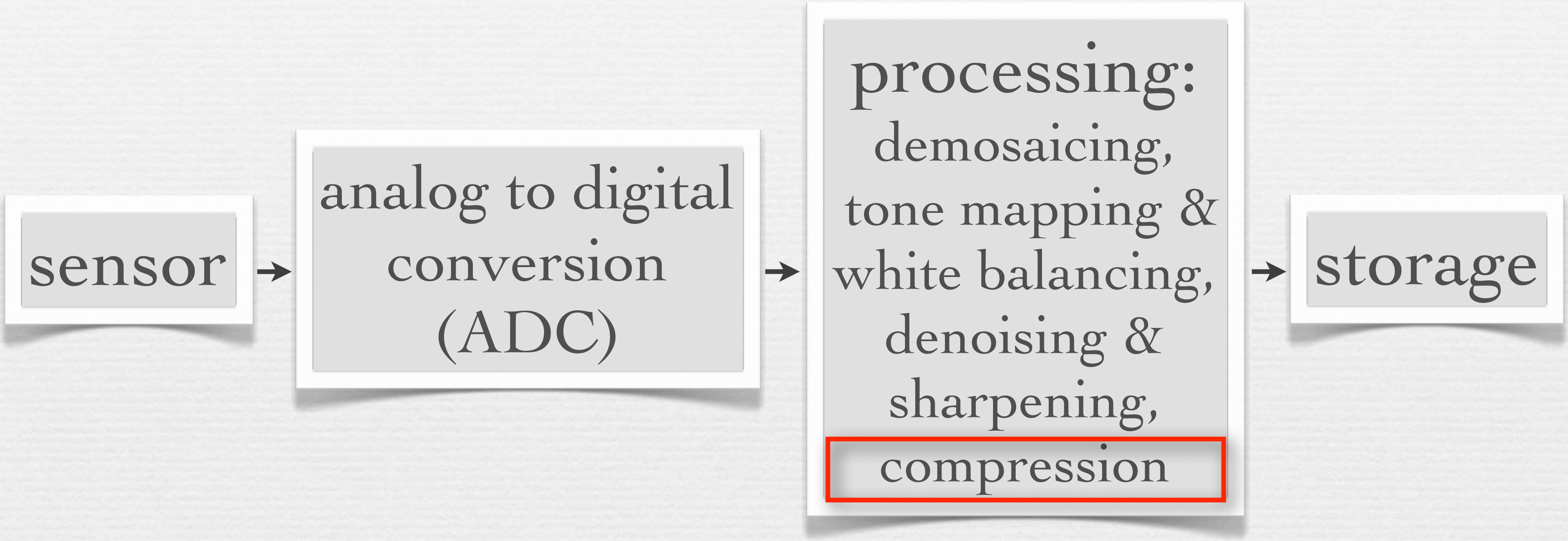
original

Recap

- *bilateral filtering* reduces noise while preserving edges
  - replaces each pixel with a weighted sum of its neighbors, where the weight drops with increasing distance from the pixel in X and Y and with increasing intensity difference
- *unsharp masking* sharpens edges but doesn't sharpen noise
  - replaces each pixel with a weighted sum of the original and a contrast-enhanced version, using the latter along edges, where the edge mask is `threshold(original-blur(original))`
- both are non-linear filters
  - i.e. they are not convolutions by a spatially invariant filter kernel

Questions?

Camera pixel pipeline



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

JPEG files

- Joint Photographic Experts Group
  - organized 1986, standard adopted 1994
- defines how an image is to be compressed (*codec*) into a stream of bytes, and the file format for storing that stream
  - file format is JFIF, but people use .JPG or .JPEG extensions
- good for compressing images of natural scenes
  - not so good for compressing drawings or graphics
- *lossy*, so loses quality <u>each</u> time you open → edit → save
  - especially if you crop or shift pixels (hence block boundaries)
  - for *lossless* compression, use PNG or TIFF

EXIF data

- ✦ Exchangeable Image File Format
  - created by Japan Electronic Industries Development Assoc.
- ✦ used by nearly every digital camera manufactured today
  - actually a file format
  - JPEG or TIFF file + metadata about the camera and shot
  - .JPG or .JPEG extension is used, not .EXIF

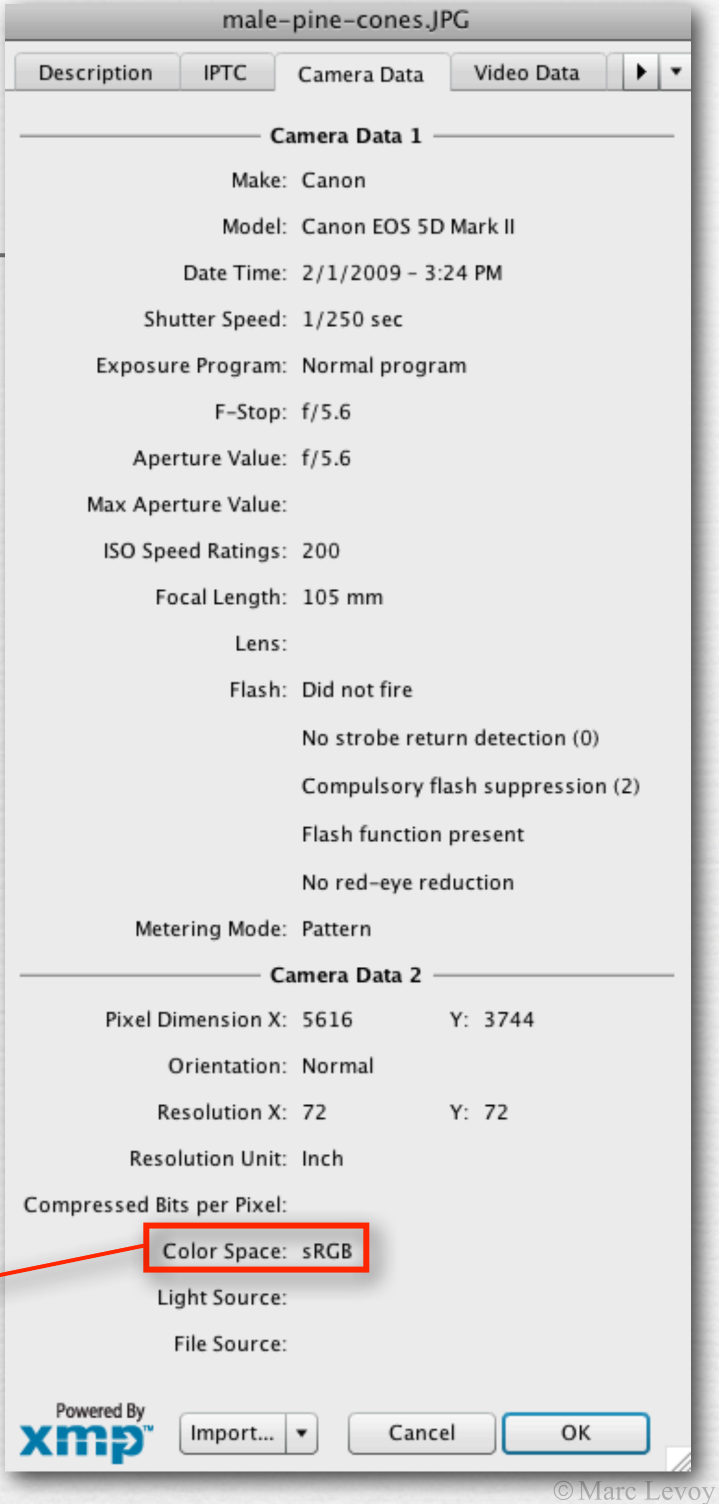
EXIF data

File/File Info in Photoshop CS4

(Marc Levoy)

shot with Canon 5D Mark II

Color Space: sRGB



Camera Data 1

Make: Canon

Model: Canon EOS 5D Mark II

Date Time: 2/1/2009 - 3:24 PM

Shutter Speed: 1/250 sec

Exposure Program: Normal program

F-Stop: f/5.6

Aperture Value: f/5.6

Max Aperture Value:

ISO Speed Ratings: 200

Focal Length: 105 mm

Lens:

Flash: Did not fire
No strobe return detection (0)
Compulsory flash suppression (2)
Flash function present
No red-eye reduction

Metering Mode: Pattern

Camera Data 2

Pixel Dimension X: 5616 Y: 3744

Orientation: Normal

Resolution X: 72 Y: 72

Resolution Unit: Inch

Compressed Bits per Pixel:

Color Space: sRGB

Light Source:

File Source:

EXIF data

exiftool

```
Focal Length                    : 105.0 mm
```

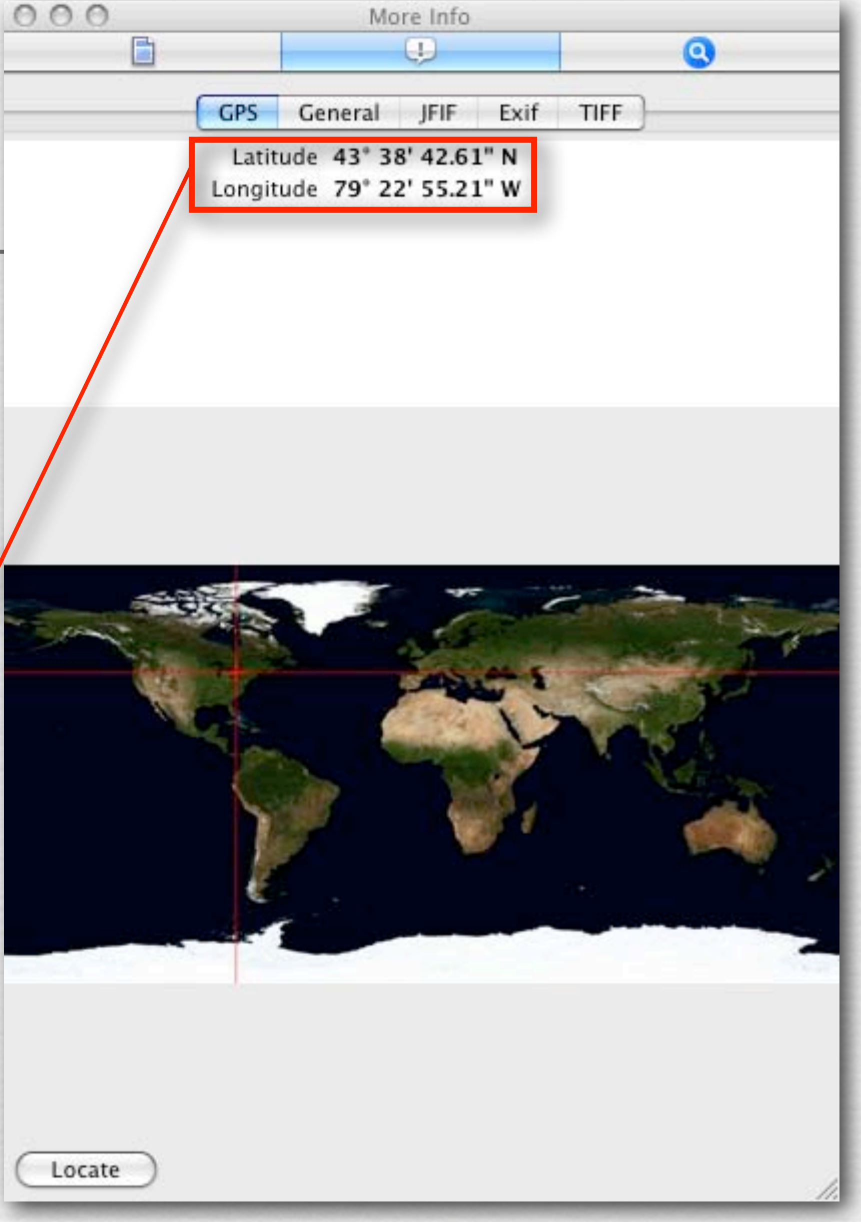
shot with Canon 5D Mark II

```
Focus Distance Upper          : 0.7
Focus Distance Lower          : 0.67
```

EXIF data

Mac Preview

shot with iPhone 3G

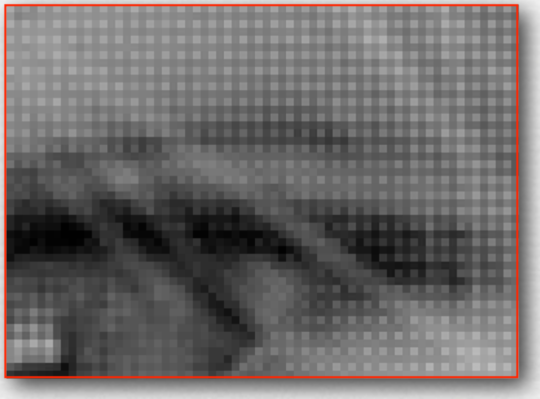


Latitude 43° 38' 42.61" N
Longitude 79° 22' 55.21" W

Latitude 43° 38' 42.61" N
Longitude 79° 22' 55.21" W

RAW files

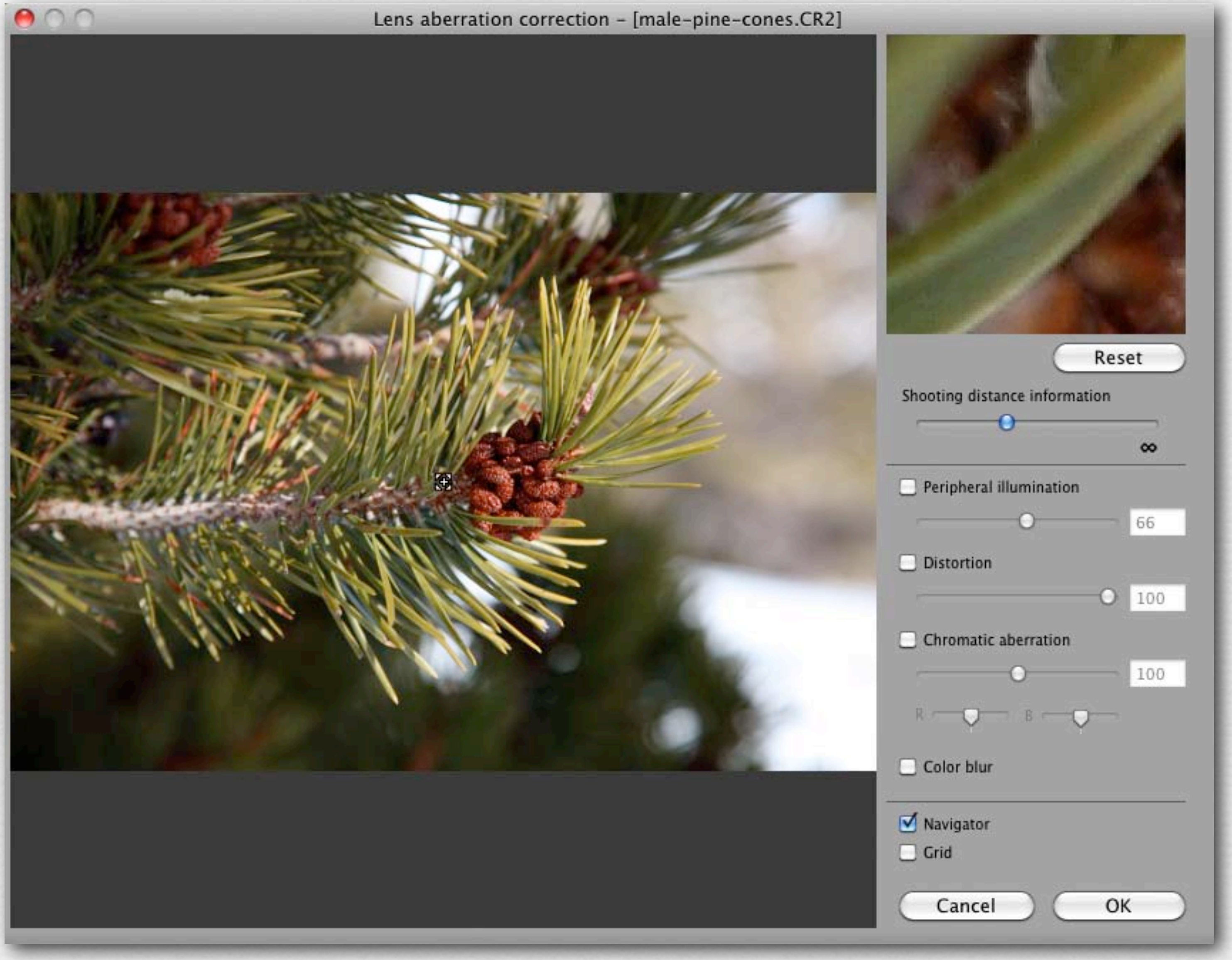
- ✦ minimally processed images, not even demosaiced
- ✦ uncompressed or losslessly compressed
- ✦ includes metadata, possibly encrypted
- ✦ file format varies by manufacturer



- ✦ example extensions: .CR2, .NEF, .RW2, .ARW
- ✦ processed and converted to a JPEG file using
  - • proprietary software (e.g. Canon Digital Photo Professional)
  - • Photoshop or Lightroom (if they support your camera)
  - • freeware programs like `dcraw`
  - • or in your camera (every time you store a JPEG)
  - • but their processing algorithms are all different!

RAW file processor

Lens aberration correction panel in Canon Digital Photo Professional



Lens aberration correction – [male-pine-cones.CR2]

Shooting distance information ∞

□ Peripheral illumination 66

□ Distortion 100

□ Chromatic aberration 100

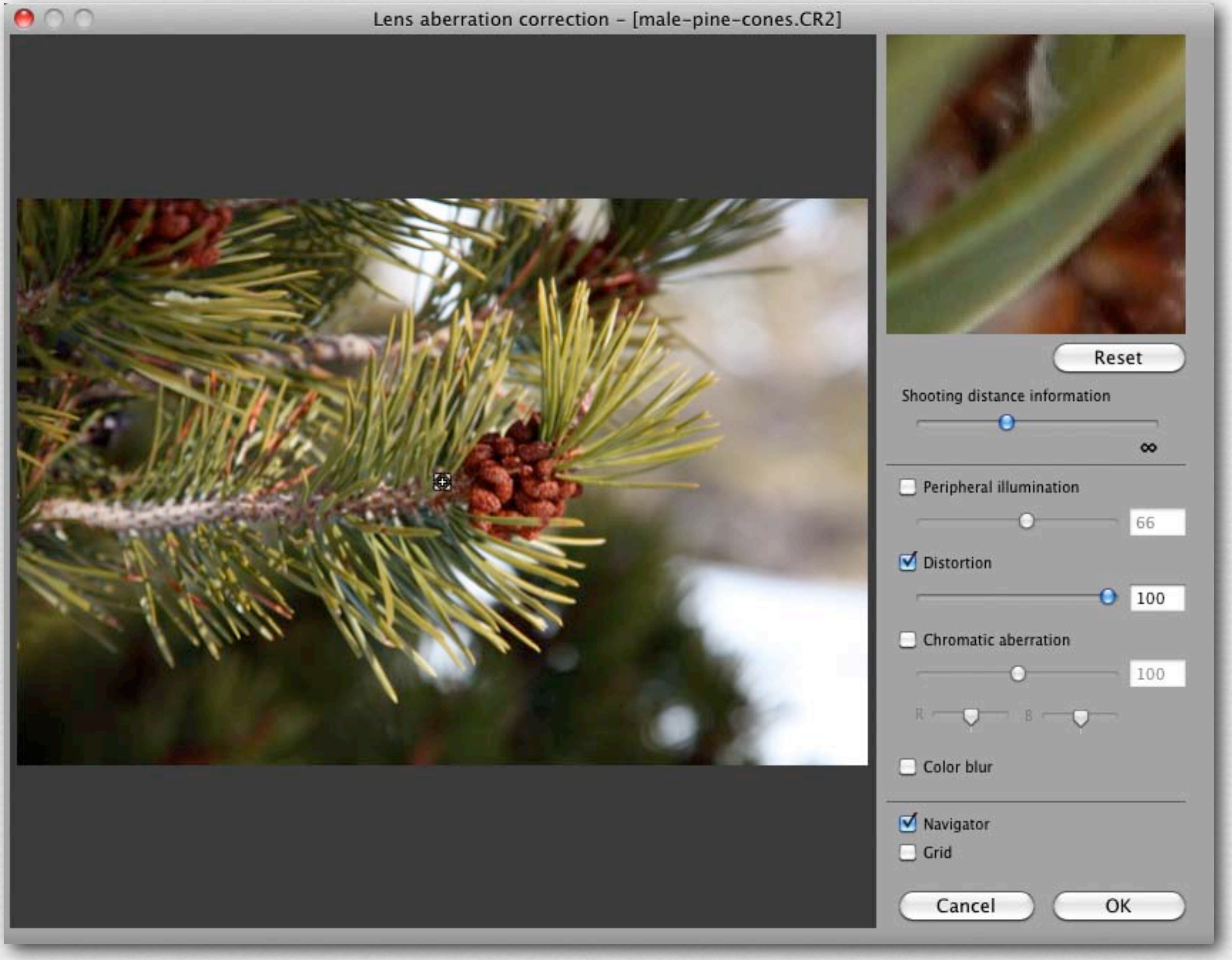
□ Color blur

☑ Navigator

□ Grid

RAW file processor

Lens aberration correction panel in Canon Digital Photo Professional



Lens aberration correction – [male-pine-cones.CR2]

Shooting distance information

∞

☐ Peripheral illumination 66

☑ Distortion 100

☐ Chromatic aberration 100

R B

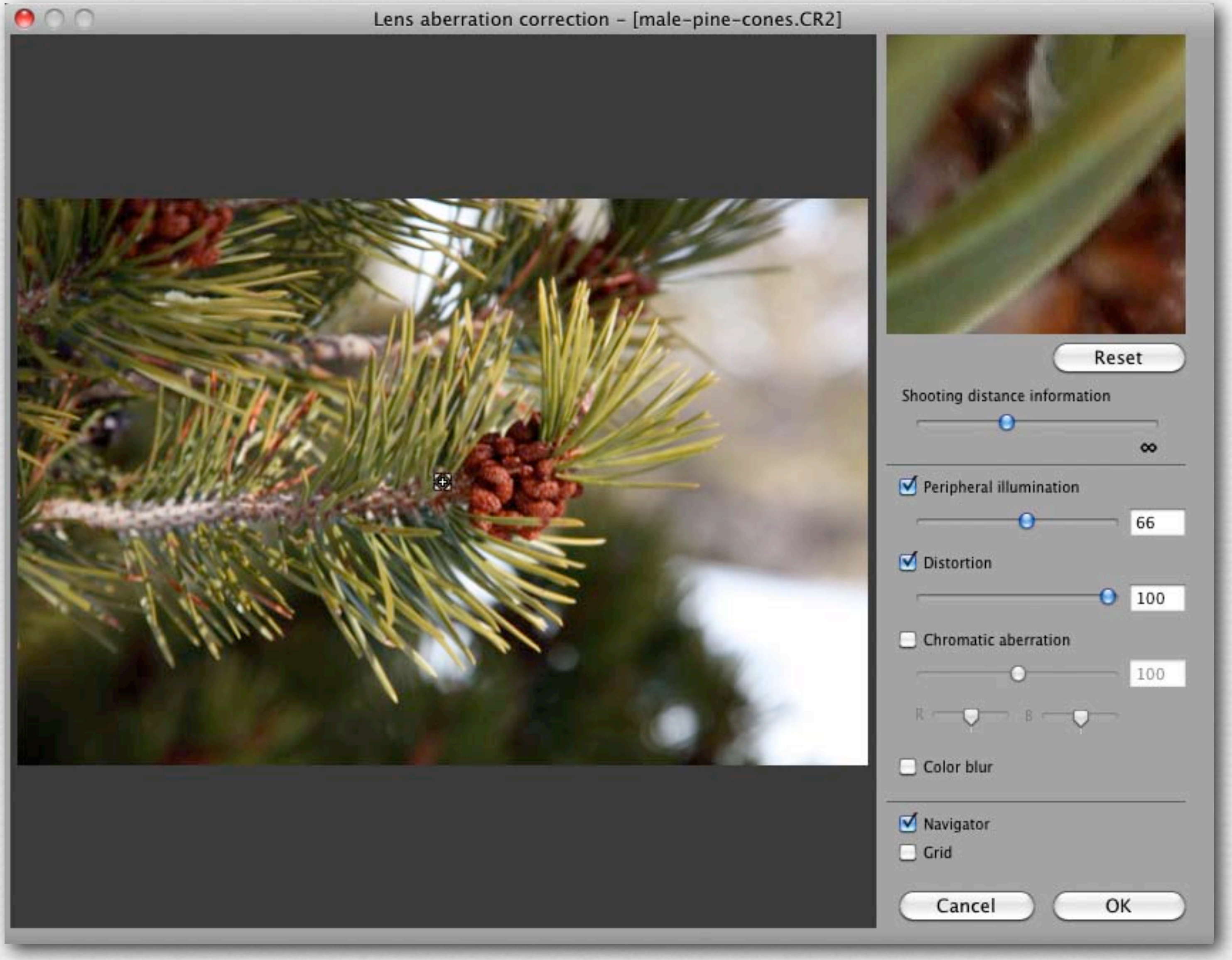
☐ Color blur

☑ Navigator

☐ Grid

RAW file processor

Lens aberration correction panel in Canon Digital Photo Professional



Lens aberration correction – [male-pine-cones.CR2]

Shooting distance information ∞

☑ Peripheral illumination 66

☑ Distortion 100

□ Chromatic aberration 100

R B

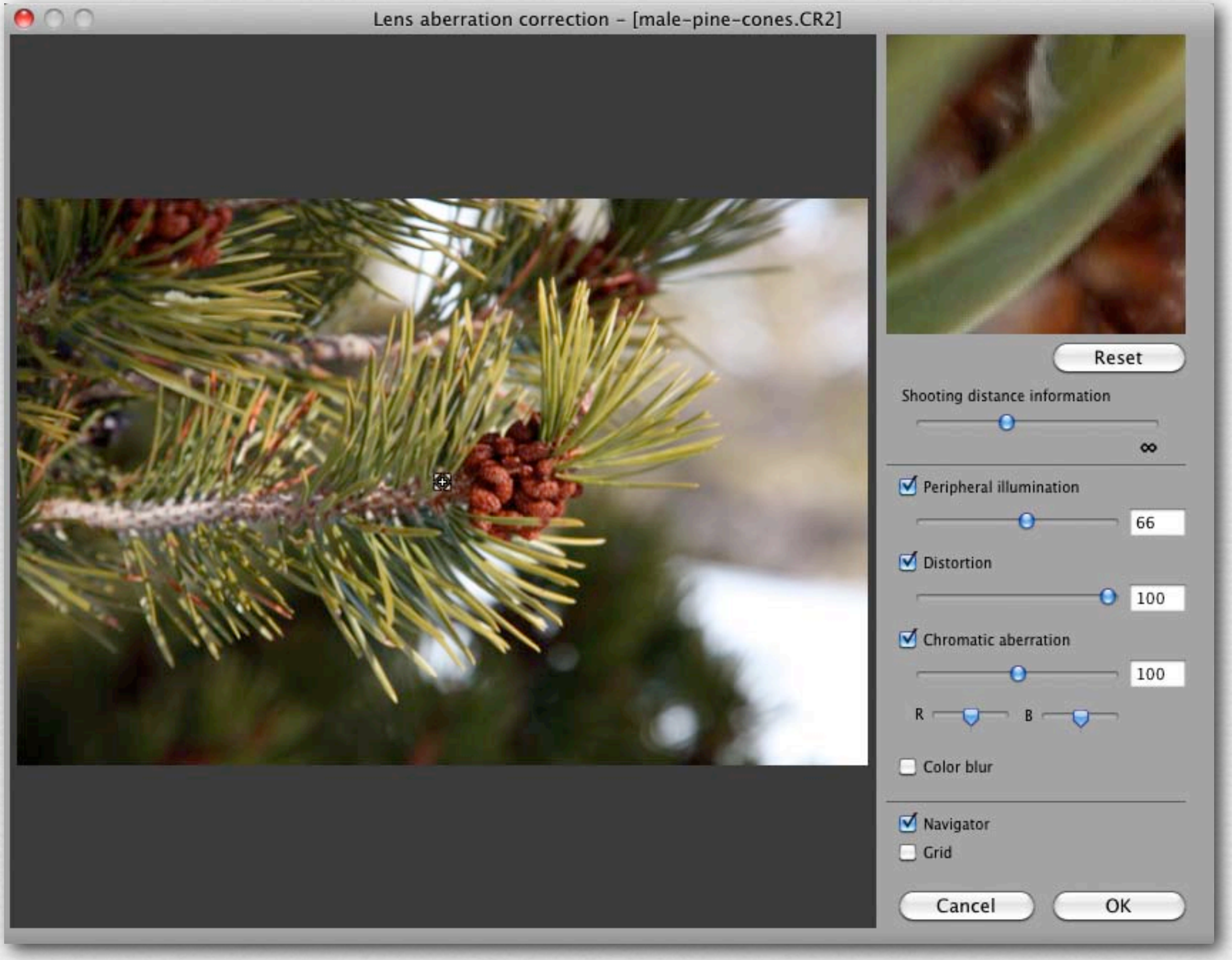
□ Color blur

☑ Navigator

□ Grid

RAW file processor

Lens aberration correction panel in Canon Digital Photo Professional



Lens aberration correction – [male-pine-cones.CR2]

Shooting distance information

☑ Peripheral illumination 66

☑ Distortion 100

☑ Chromatic aberration 100

□ Color blur

☑ Navigator

□ Grid

JPEG image compression

- compression (in camera)

- input is Y'CbCr
- Cb and Cr typically downsampled by 2× in X and Y
- each component is compressed separately

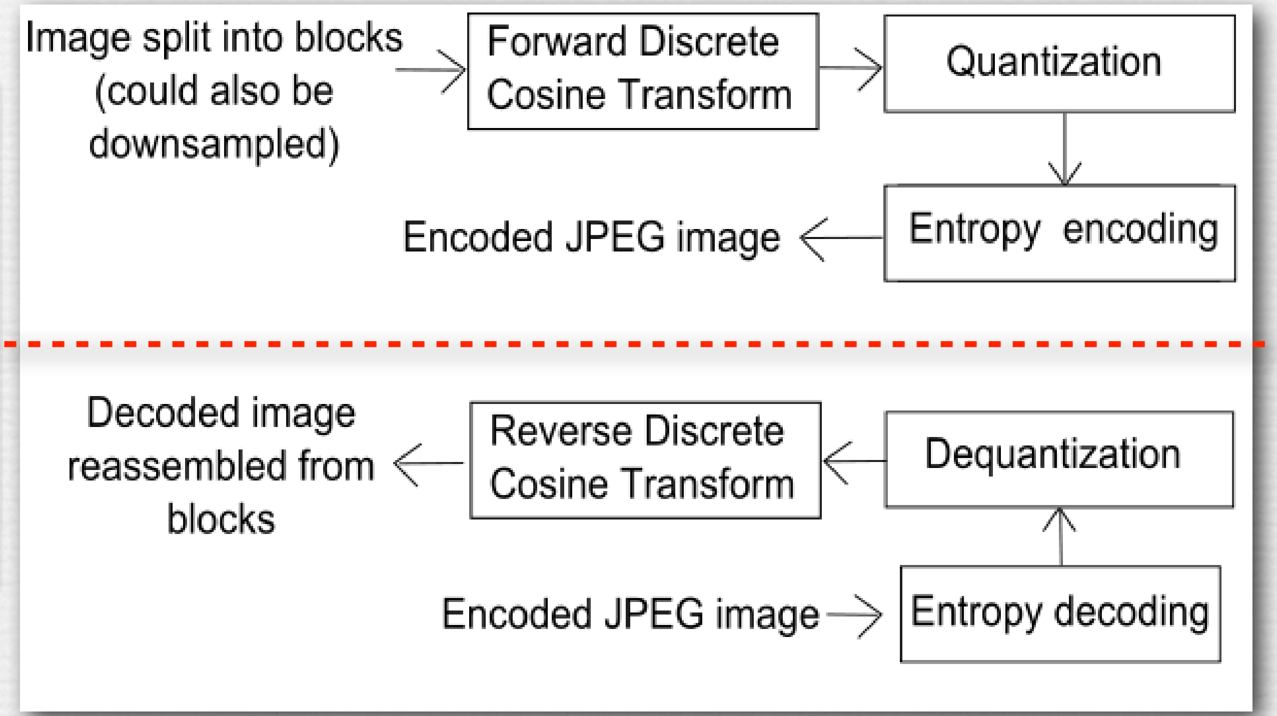
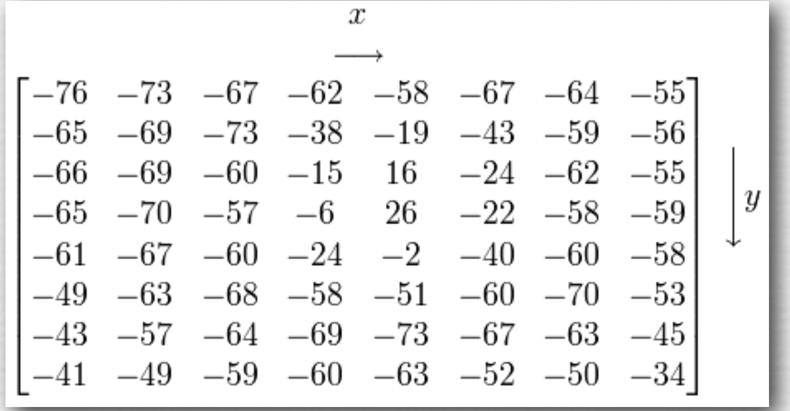


Image split into blocks (could also be downsampled)
Forward Discrete Cosine Transform
Quantization
Entropy encoding
Encoded JPEG image
Decoded image reassembled from blocks
Reverse Discrete Cosine Transform
Dequantization
Encoded JPEG image
Entropy decoding

(wikipedia)

- decompression (for display)

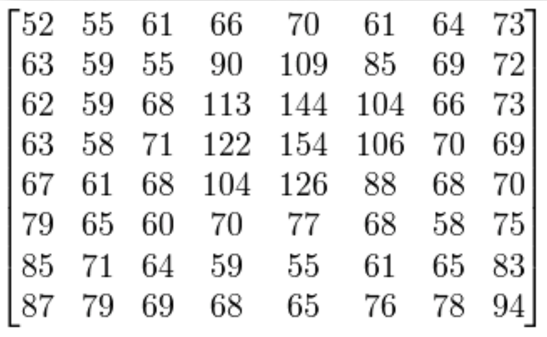
JPEG image compression



$$\begin{array}{cc}\overset{x}{\longrightarrow} & \\ \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} & \Big\downarrow y \end{array}$$

zero-centered image

$\longleftarrow$

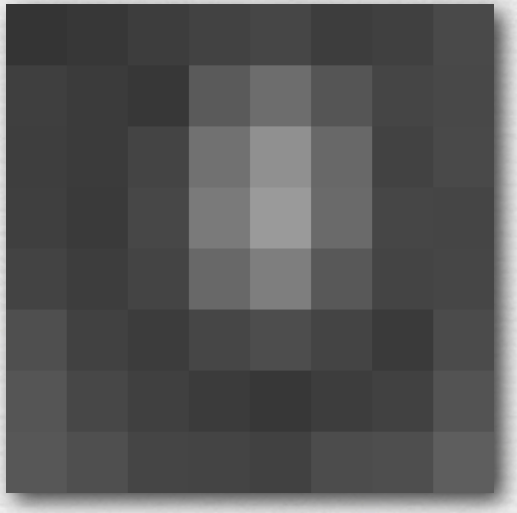


$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

8-bit image

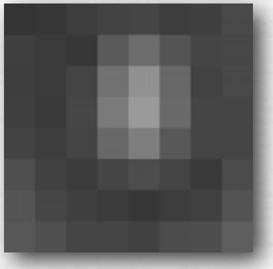
$\uparrow$

- step #1: split into 8×8 pixel blocks
- step #2: quantize to 8 bits / pixel
- step #3: convert to zero-centered



8×8 pixel block

JPEG image compression



$$\begin{array}{cc} \overset{x}{\longrightarrow} & \\ \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} & \bigg\downarrow y \end{array}$$

zero-centered image

$$\longrightarrow$$

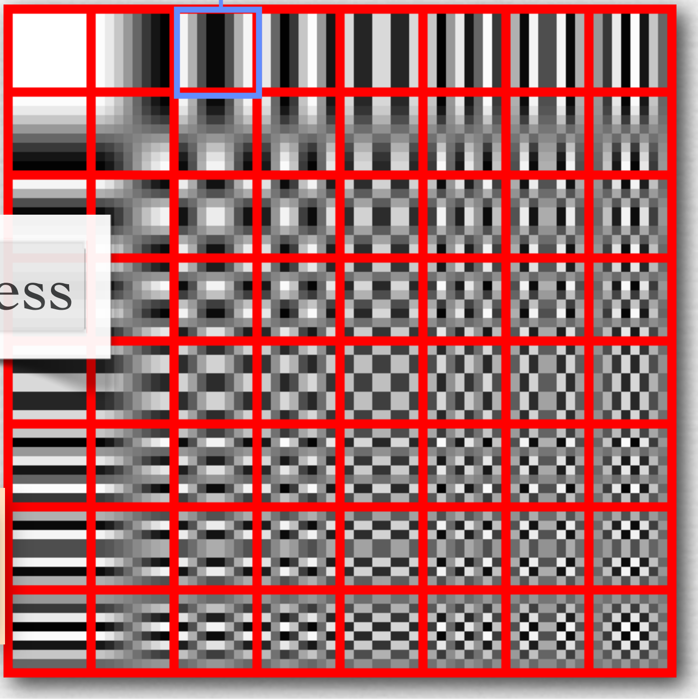
$$\begin{array}{cc} \overset{u}{\longrightarrow} & \\ \begin{bmatrix} -415 & -30 & \boxed{-61} & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} & \bigg\downarrow v \end{array}$$

discrete cosine transform (DCT)

- ✦ any 8×8 pixel zero-centered image can be represented by a weighted sum of the 64 8×8 pixel *basis functions* shown at right

- ✦ step #4: compute the weighting for each basis function using:

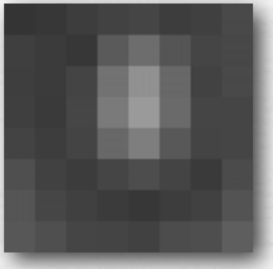
$$G_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right]\cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$



lossless

JPEG n

An observant student asked why the quantization table (bin size) is not a symmetric matrix, i.e. why is its lower-left corner not equal to its upper-right corner? I traced this particular table, which is frequently cited in the literature (and wikipedia) back to a psychophysical study performed in 1984 on a CRT display whose MTFs differed in the horizontal and vertical direction. For today's LCD displays, a symmetric matrix might be preferred.



$$\left(\begin{bmatrix} 16 & 11 & \boxed{10} & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}\right)$$

bin size for each coefficient

$$\overset{u}{\longrightarrow}$$

$$\begin{bmatrix} -415 & -30 & \boxed{-61} & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} \Big\downarrow v$$

discrete cosine transform (DCT)

$$\downarrow$$

- the human visual system is more sensitive to low & mid frequencies than very high frequencies, so quantize the latter coarsely

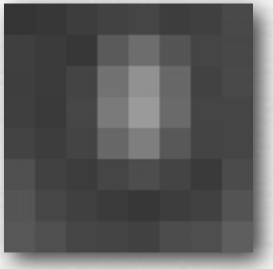
lossy

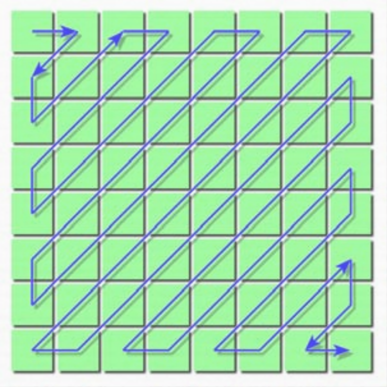
- step #5: quantize the DCT coefficients using bins whose size increases with frequency

$$\begin{bmatrix} -26 & -3 & \boxed{-6} & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

quantized DCT coefficients

JPEG image compression





zig-zag ordering

```
-26
 -3   0
 -3  -2  -6
  2  -4   1  -4
  1   1   5   1   2
 -1   1  -1   2   0   0
  0   0   0  -1  -1 EOB
```

- ✦ step #6: arrange the non-zero coefficients in zig-zag order

lossless

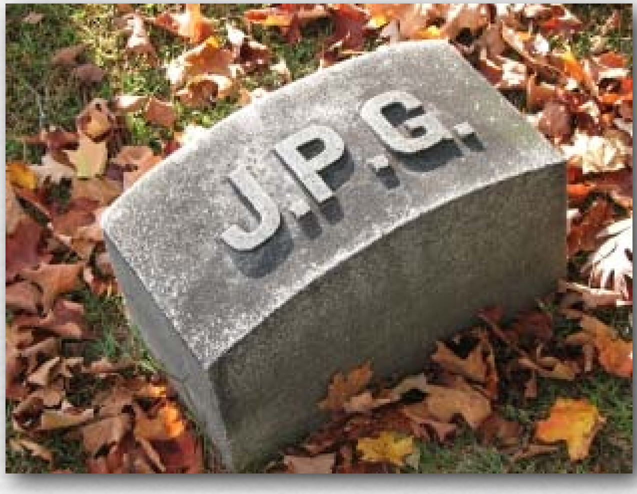
- ✦ step #7: use run-length encoding to remove repeated elements
- ✦ step #8: apply Huffman coding to reduce number of bits needed for each coefficient

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

quantized DCT coefficients

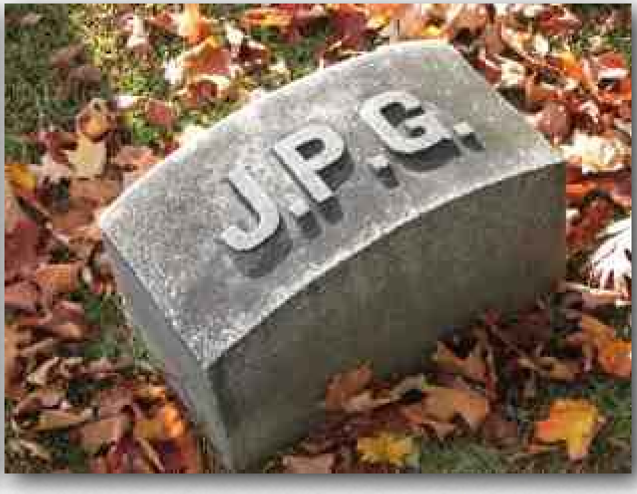
JPEG image compression

Q = 100



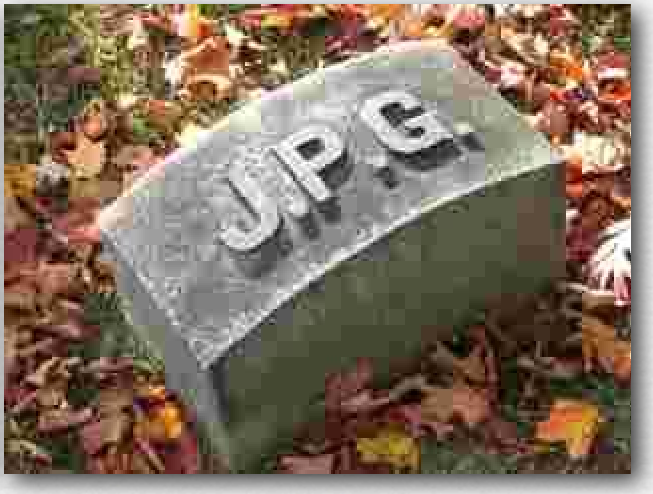
2.6 : 1

Q = 25



23 : 1

Q = 1



144 : 1

↓

144:1 looks fine if it's
displayed small enough

- not easily comparable to Photoshop quality numbers, since Adobe uses its own (proprietary) encoder

Recap

- ✦ RAW files is the direct output of the camera sensor
  - not demosaiced, 16 bits per pixel, losslessly compressed
  - contains metadata, usually proprietary
- ✦ JPEG files are a standard format for storing images
  - typically 8 bits per pixel, lossy compression
  - contains metadata in EXIF format
- ✦ JPEG's compression format is designed to discard details
  - images are partitioned into blocks of 8 × 8 pixels
  - each block is represented by a weighted sum of cosinusoids (DCT)
  - the coefficients of high frequency cosinusoids are heavily quantized, which reduces # of bits, hence file size, but also loses images quality
  - these coefficients are losslessly compressed using Huffman coding

Questions?

Slide credits

- Fredo Durand

- Wandell, B., *Foundations of Vision*, Sinauer, 1995.
- Tanser and Kleiner, *Gardner's Art Through the Ages* (10th ed.), Harcourt Brace, 1996.
- Rudman, T., *Photographer's Master Printing Course*, Focal Press, 1998.
- Adams, A., *The Print*, Little, Brown and Co., 1980.
- Goldstein, B.E., *Sensation and Perception,* Wadsworth, 1999.
- Wolfe, J.M., *Sensation and Perception,* Sinauer, 2006.