In-camera image processing

CS 178, Spring 2009

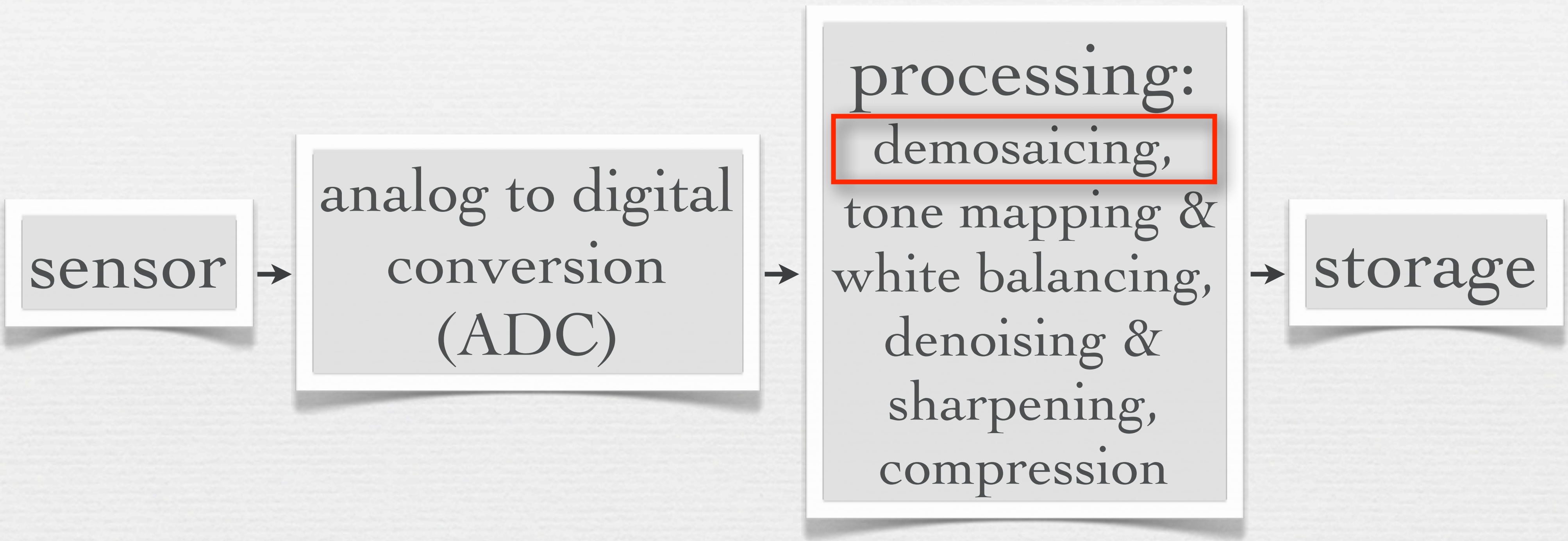
Begun 5/28/09, finished 6/2/09.

Marc Levoy
Computer Science Department
Stanford University

Starting May 30 at SFMOMA

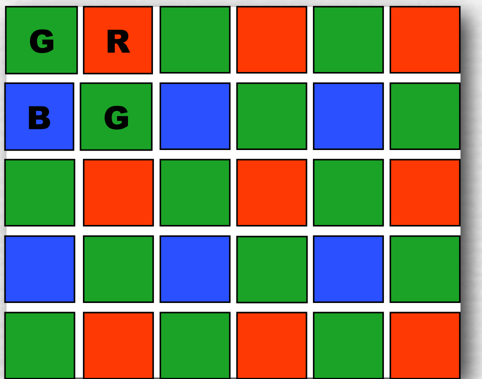
- San Francisco Museum of Modern Art

Camera pixel pipeline

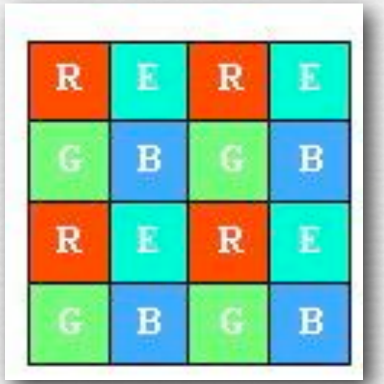


- every camera uses different algorithms
- the processing order may vary
- most of it is proprietary

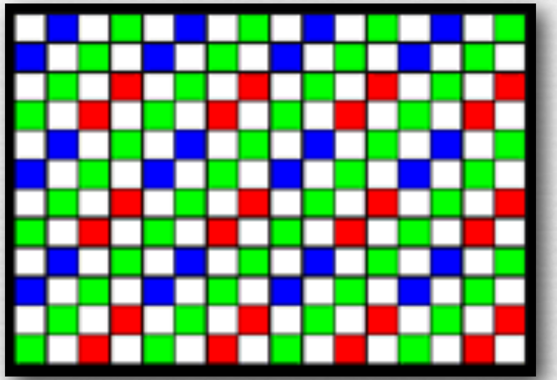
Color filter arrays (review)



Bayer pattern

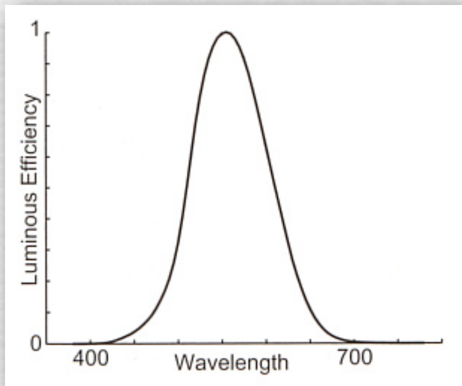


Sony RGB+E
better color


Kodak RGB+C
less noise

- Why more green pixels than red or blue?
  - because humans are most sensitive in the middle of the visible spectrum
  - remember the human luminous efficiency curve

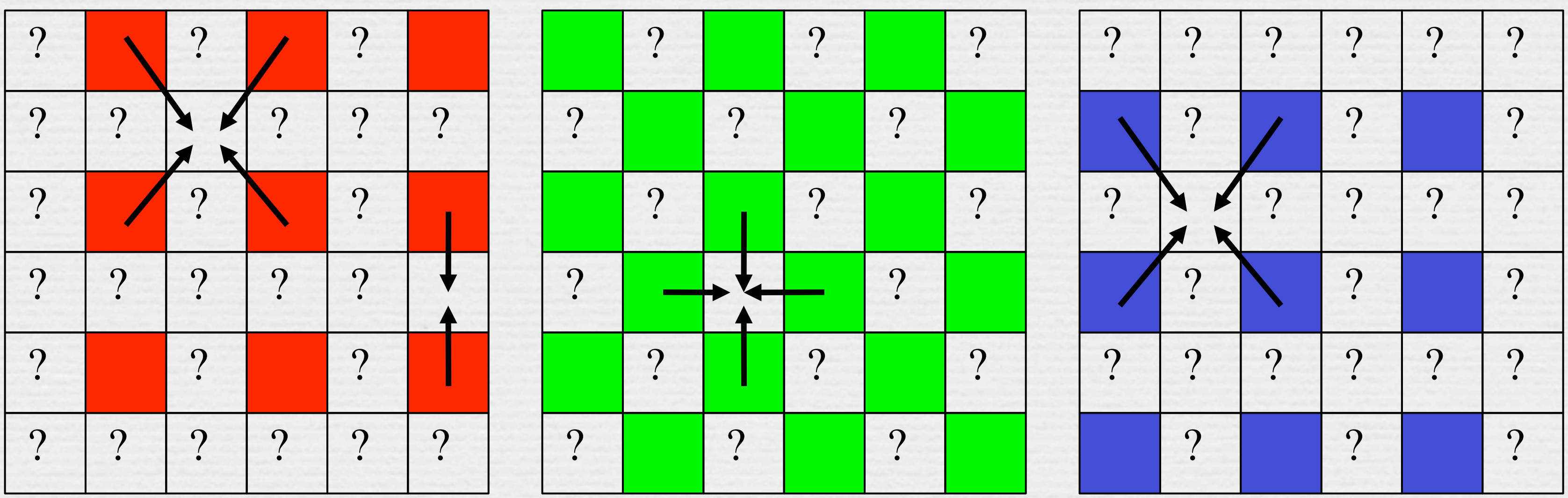
(Stone)



Crop from raw Bayer mosaic image

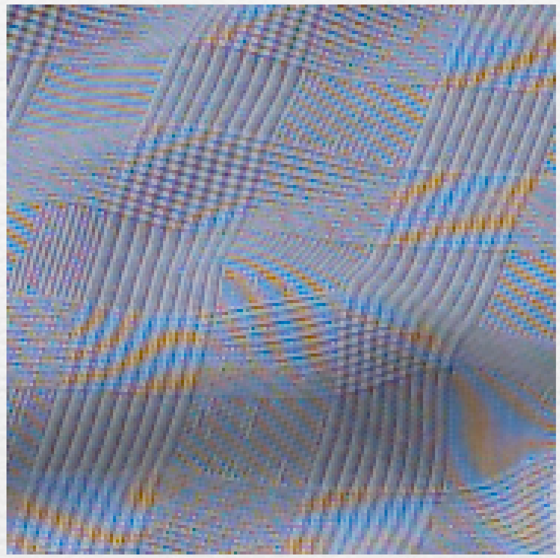
Demosaicing

- linear interpolation
  - average of the 4 nearest neighbors
- smoother kernels are possible
  - e.g. bicubic interpolation (what Photoshop uses by default)
  - but need more neighbors (16 instead of 4)



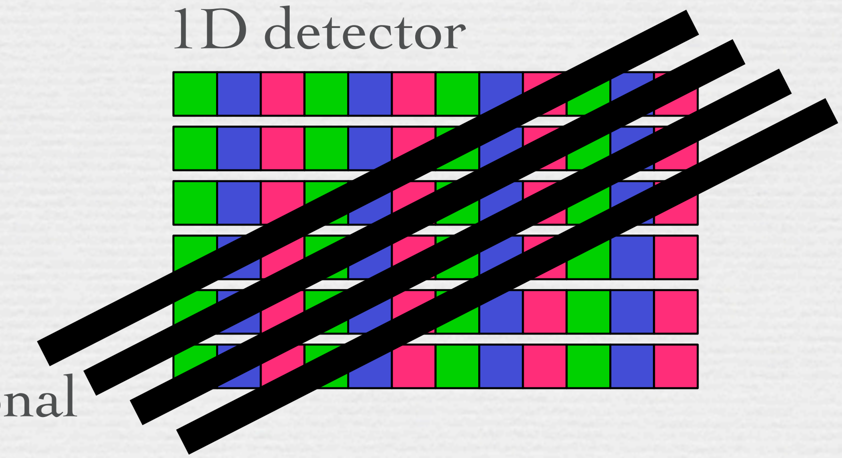
Demosaicing errors

- color fringes or moiré



- the cause of color moiré
  - fine black and white detail in scene is mis-interpreted by interpolation algorithm as color information

simplified 1D detector



fine diagonal B&W stripes

Common solution: low-pass filter chrominance signal

- color artifacts are places where chrominance changes abruptly and only temporarily
- use a median filter on chrominance to remove outlier transient chrominance changes [Freeman 1988]
  - replace the chrominance of each pixel by the median value in a neighborhood
  - this is a non-linear filter

Apparent spatial sharpness depends mainly on luminance, not chrominance

original image

(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance

red-green
channel
blurred

(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance

original image

(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance

blue-yellow
channel
blurred

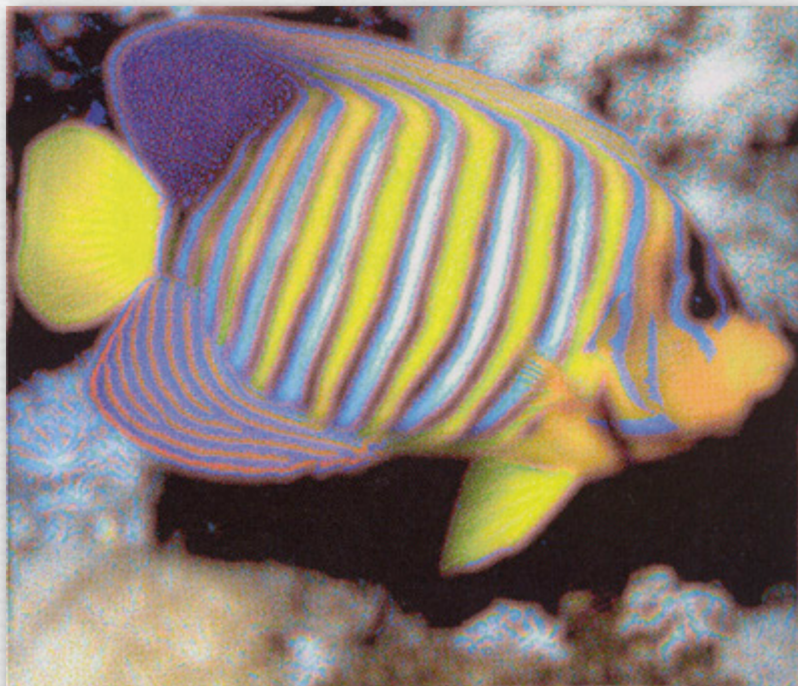
(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance

original image

(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance



(Wandell)

luminance
channel
blurred

Common solution: low-pass filter chrominance signal

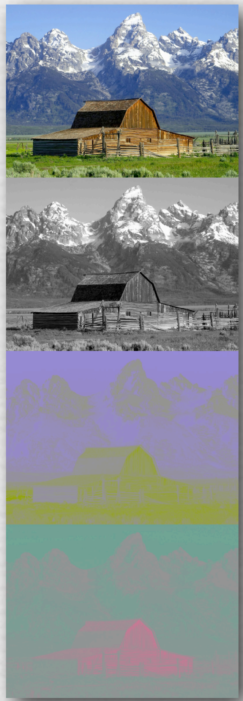
- color artifacts are places where chrominance changes abruptly and only temporarily
- use a median filter on chrominance to remove outlier transient chrominance changes [Freeman 1988]
  - replace the chrominance of each pixel by the median value in a neighborhood
  - this is a non-linear filter
- resulting algorithm
  - 1. apply naive interpolation
  - 2. convert to YCbCr
  - 3. median filter Cr & Cb
  - 4. reconstruct R, G, B from sensor value and filtered Cr & Cb

(wikipedia)

Y

Cb

Cr



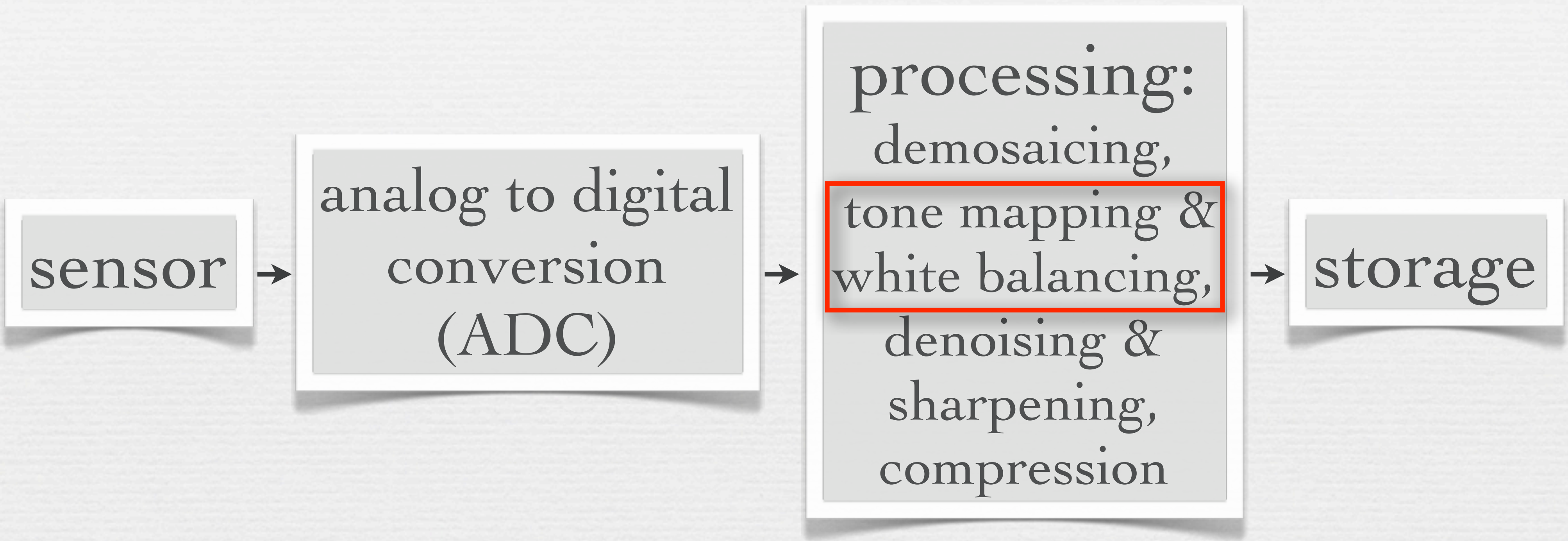
Comparison

linear interpolation

median-filtered interpolation

- take-home lesson: 2/3 of your data is made up!
- there are better and worse ways to do this

Camera pixel pipeline



White balancing (review)

- 1. find the color temperature of the illumination as an (R,G,B)
- 2. scale the RGB values of all pixels in the photograph up or down so that the chosen (R,G,B) becomes (1,1,1)
- the appearance of (1,1,1) depends on the camera's reference white
  - for sRGB cameras, chosen (R,G,B) will match D65 (6500°K)

(Marc Levoy)

Contrast

I believe I mis-pronounced acutance in class. The accent should be on the "u", i.e. ak-kew´-tance.

- *dynamic range* is max / min intensity the camera can record

$$DR = \frac{\text{max output swing}}{\text{noise in the dark}} = \frac{\text{saturation level} - D\,t}{\sqrt{D\,t + N_r^2}}$$

- *contrast ratio* is max / min intensity in an image
  - if I_{min} = 0 and I_{max} = 255, then contrast = 256:1
- *image contrast* is luminance difference / average luminance; for N pixels with intensities $I_1,\ldots,I_N$, one formulation is

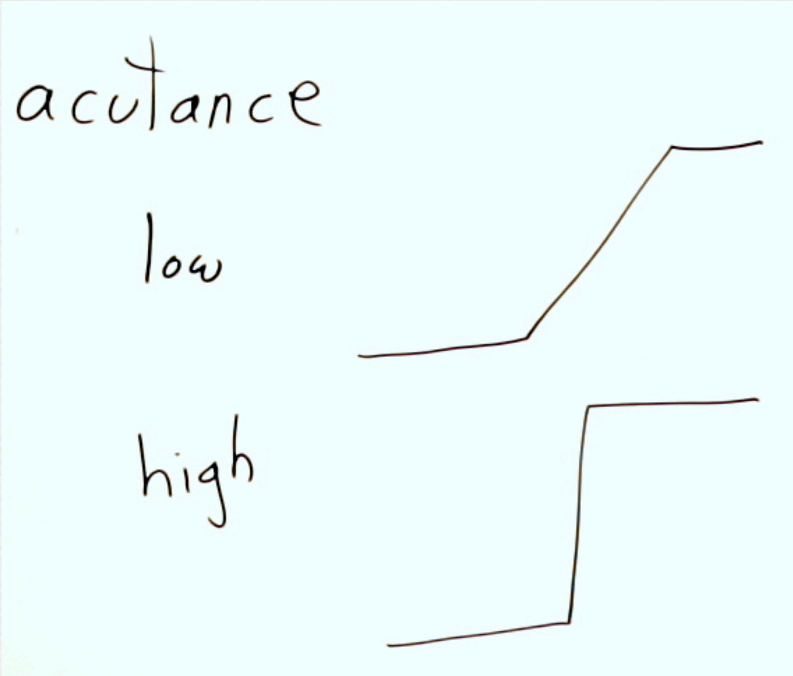
$$\text{RMS contrast} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(I_i - \mu)^2}$$

(i.e. standard deviation of image intensity)

- *acutance* is edge contrast, as measured by intensity gradient
  - unsharp masking increases acutance

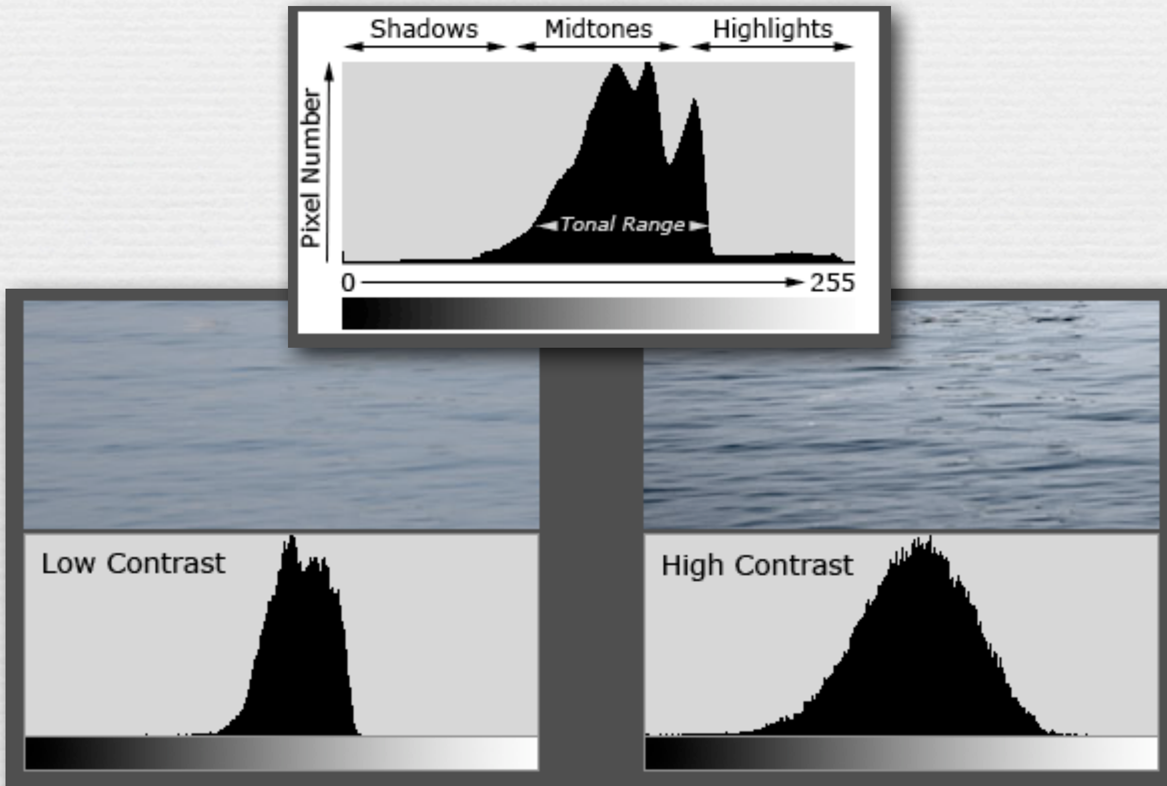
Acutance

- one possible definition is given in wikipedia at http://en.wikipedia.org/wiki/Acutance, illustrated by our flagship photo (Dorothea Lange's Migrant Mother), in fact!



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

- manual editing
  - store image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.



(cambridgeincoulour.com)

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

- manual editing
  - store image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.
- gamma transform
  - output = input$^{\gamma}$ (for $0 \leq I_i \leq 1$)
  - simple but crude

original

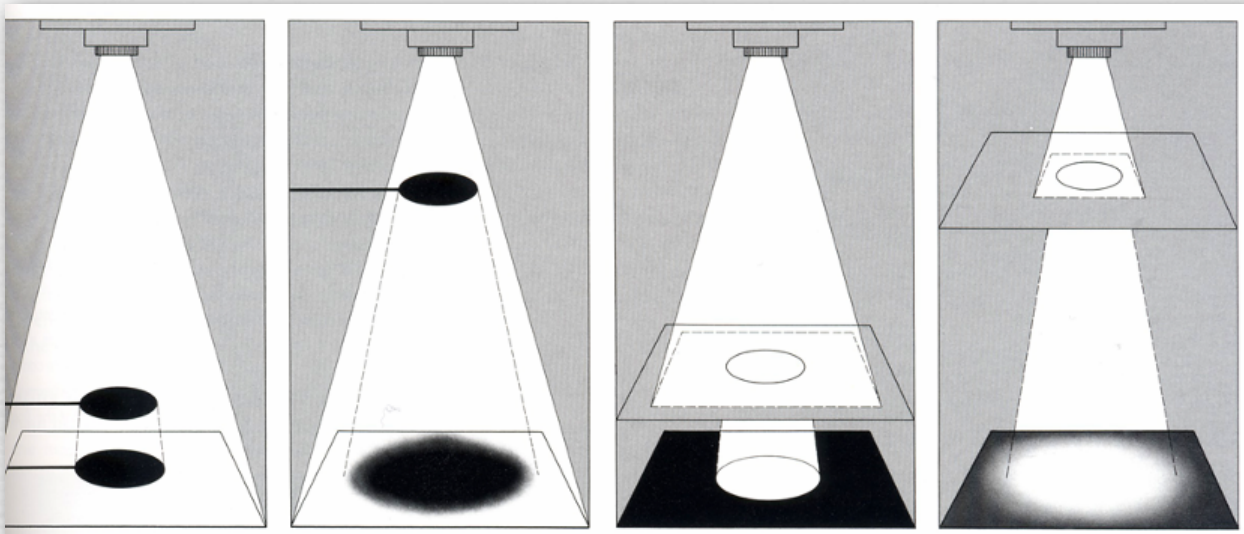
$\gamma = 0.5$

$\gamma = 2.0$

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

- manual editing
  - store image in RAW mode, then fiddle with histogram in Photoshop, `dcraw`, Canon Digital Photo Professional, etc.
- gamma transform
  - output = input$^{\gamma}$ (for $0 \le I_i \le 1$)
  - simple but crude
- histogram equalization

Traditional dodging and burning

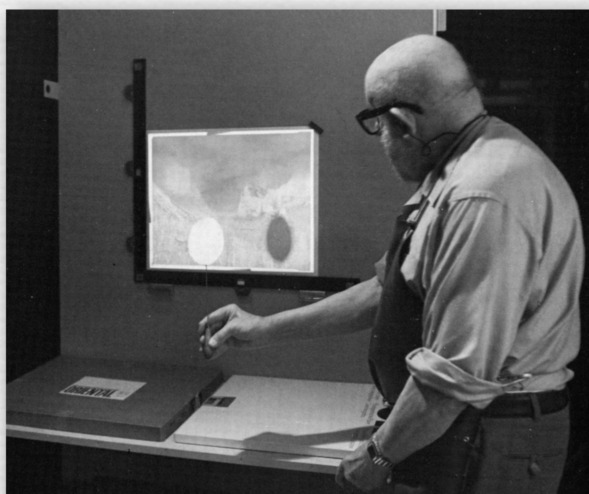


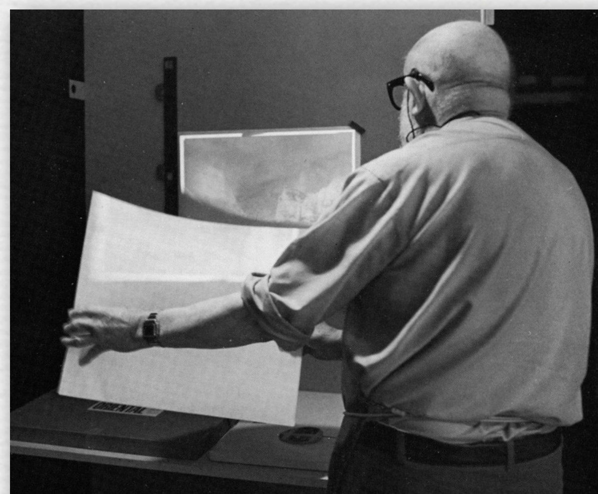
(Rudman)

dodging
(leaves print lighter)

burning
(makes print darker)

Ansel Adams in
his darkroom





(Adams)

straight
print

Ansel Adams, Clearing Winter Storm, 1942

toned
print

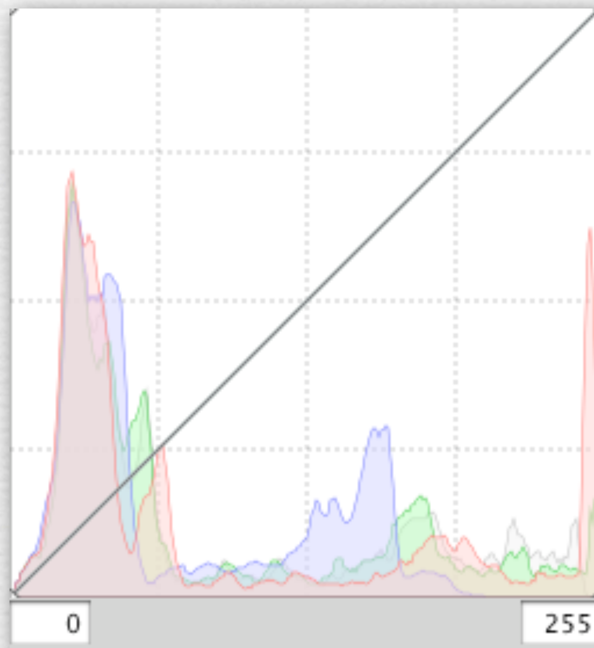
Ansel Adams, Clearing Winter Storm, 1942

Histogram equalization

1. convert image to L*a*b* in range [0,1]

2. calculate histogram of L* channel $pdf(i) = \frac{N_i}{N}$,
where N_i is the number of pixels of intensity i,
and N is the total number of pixels

image

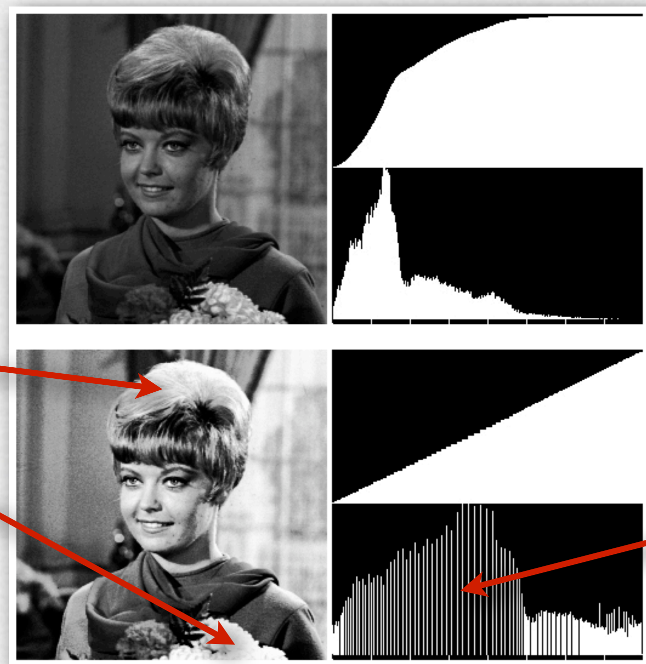


histogram

Since composing this slide, I found a web site with a better example on it:
http://www.generation5.org/content/2004/histogramEqualization.asp

Histogram equalization

1. convert image to L*a*b* in range [0,1]
2. calculate histogram of L* channel $pdf(i) = \frac{N_i}{N}$,
where N_i is the number of pixels of intensity i,
and N is the total number of pixels
3. calculate cumulative density function $cdf(i) = \sum_{j=0}^{i} pdf(j)$
4. re-map each pixel using $I_{out} = cdf(I_{in}) \times 255 / N$ (for 8-bit pixels)



High dynamic range imaging (review)

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

(Marc Levoy)

(Marc Levoy)

(Marc Levoy)

Cathedral, Valencia

(Marc Levoy)

tone mapping
by exposure and
gamma

Photoshop's "exposure and gamma" tone mapping method is more complicated than the simple gamma transform I showed a few slides back. To be honest, I don't know how it works (and it's proprietary). The exposure slider is clearly establishing a narrow range of intensities (within the high-dynamic range) that will then be subject to a gamma transform. Intensities outside this narrow range should be mapped to black or white, more or less, but Photoshop appears to feather this effect in some way, to avoid harsh clipping of the tonal range. If someone knows what they've implemented, please tell me.

Cathedral,
Valencia

(Marc Levoy)

tone mapping
by histogram
equalization

Cathedral,
Valencia

(Marc Levoy)

Tone mapping techniques
(slides from Fredo Durand)

- image has 10,000:1 dynamic range, projector has ~200:1
- how can we compress the image's dynamic range?

Global tone mapping operators

- gamma compression applied independently on R,G,B

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

- colors become washed out

input

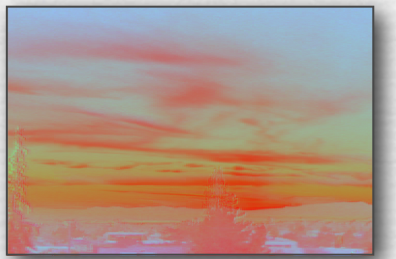
output

Global tone mapping operators

- gamma compression on intensity only
- colors are preserved, but become garish if you try to substantially enhance dark areas

luminance

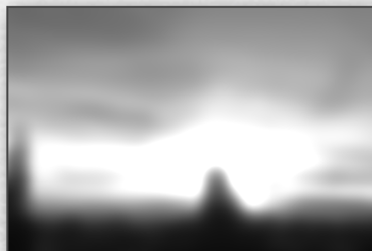
color



Local tone mapping operators

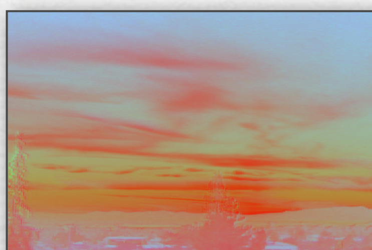
- reduce contrast of low frequencies, while preserving high frequencies [Oppenheim 1968, Chiu et al. 1993]
- produces halos!

low frequency



high frequency

color



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]

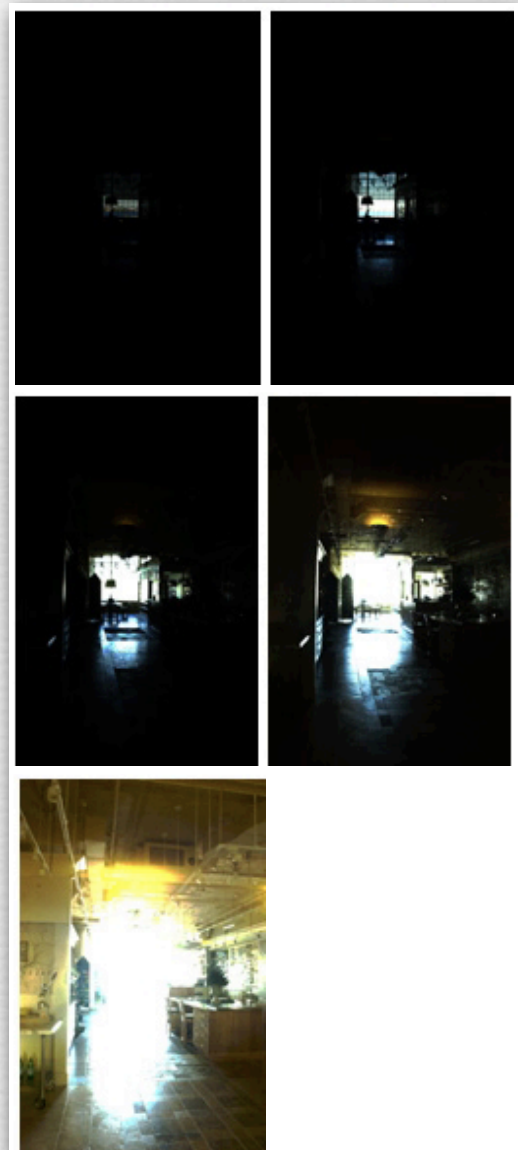
large scale

detail

color

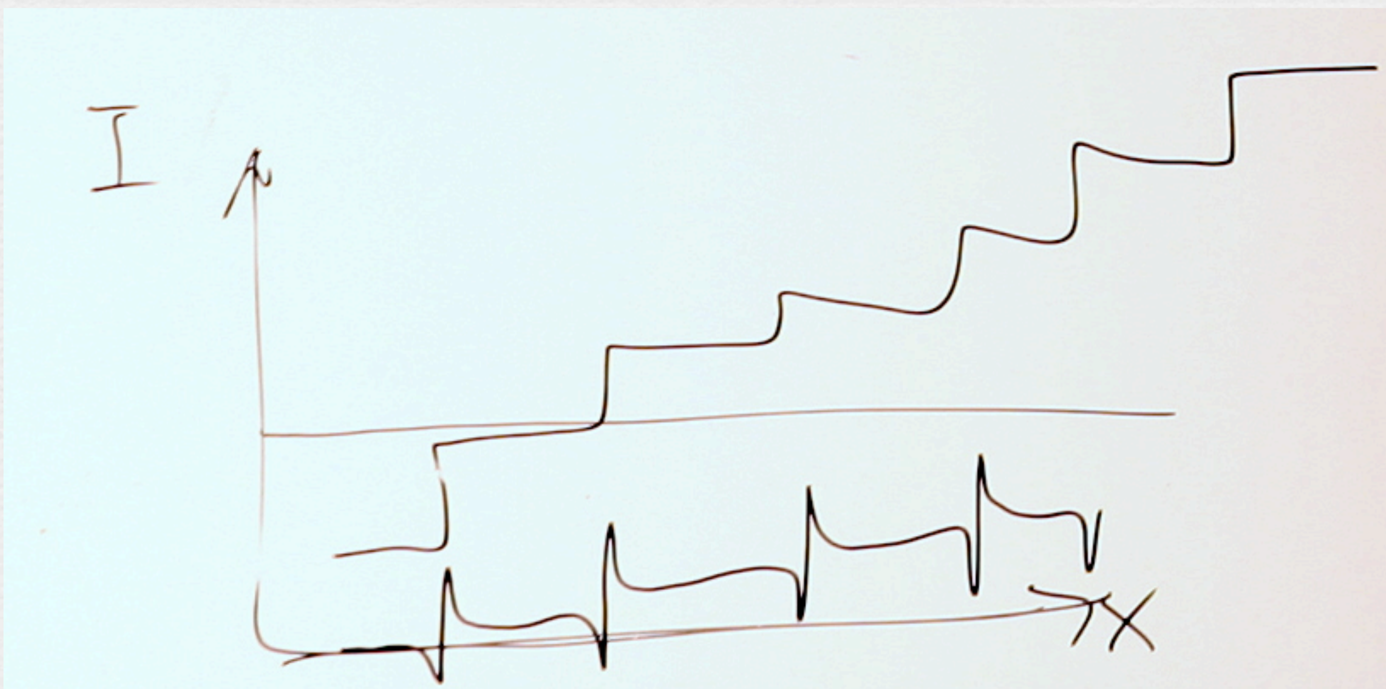
Tone mapping using bilateral filters

[Durand and Dorsey SIGGRAPH 2002]

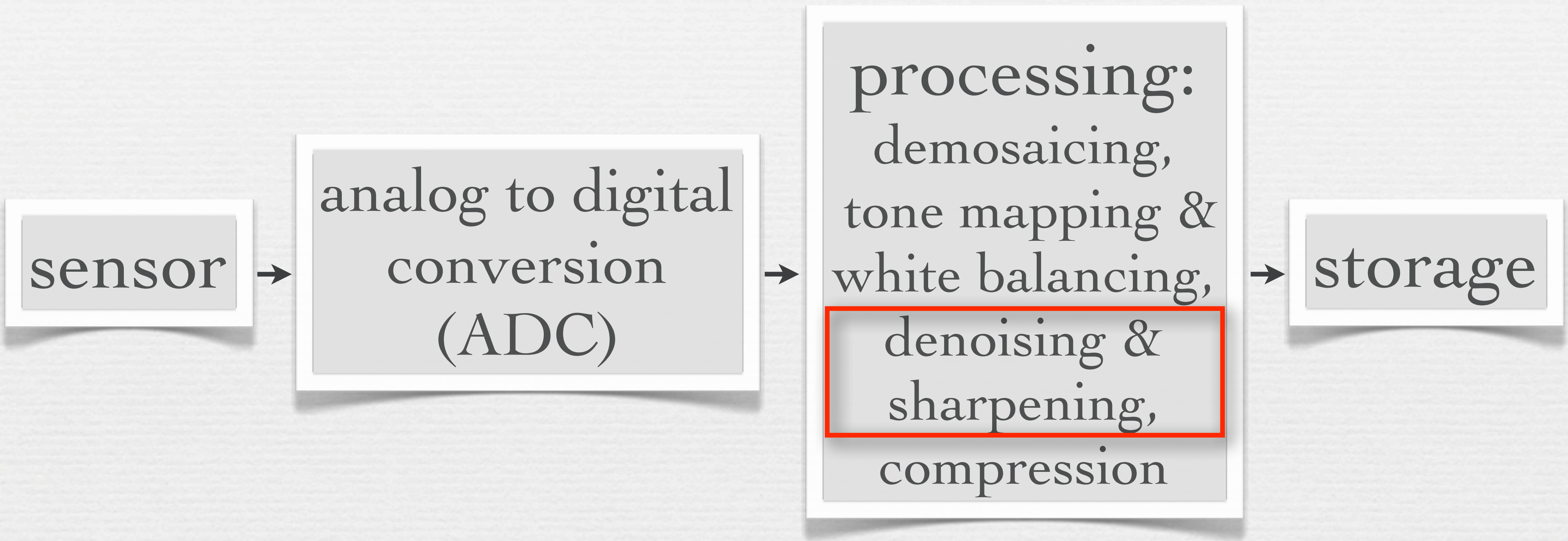


Why might tone mapping look cartoony?

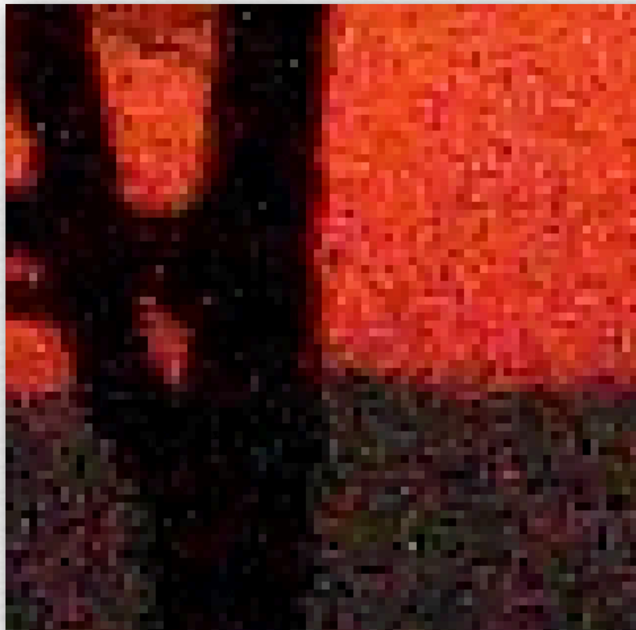
- a step wedge (at top) is converted by tone mapping to the plot at bottom
  - the human eye does this internally, but that doesn't necessarily mean that we want to present an image like this to the human eye!



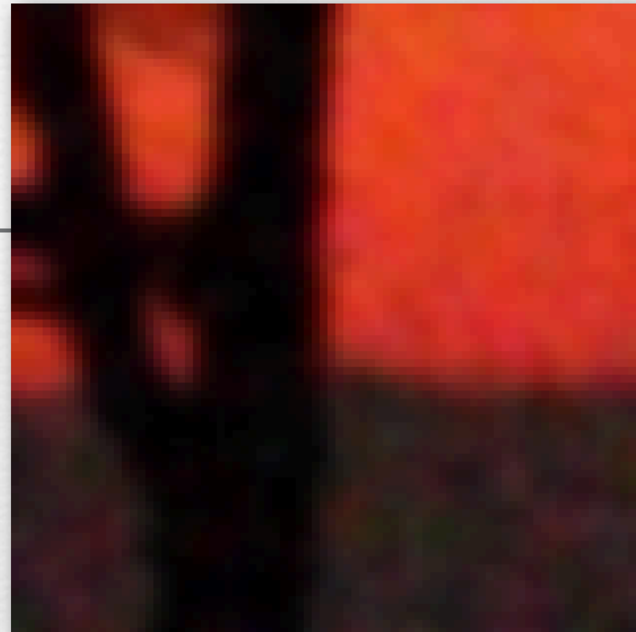
Camera pixel pipeline



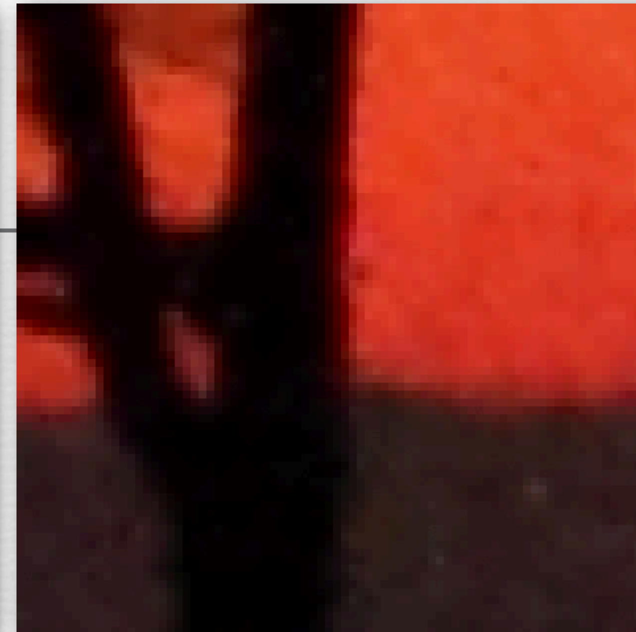
Denoising



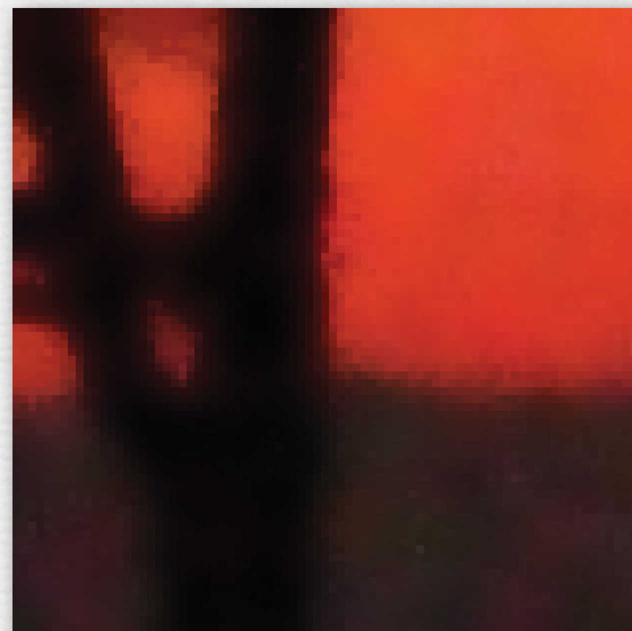
RAW (ISO 6400)



Gaussian blur, radius = 1.3



Canon denoising



bilateral filtering

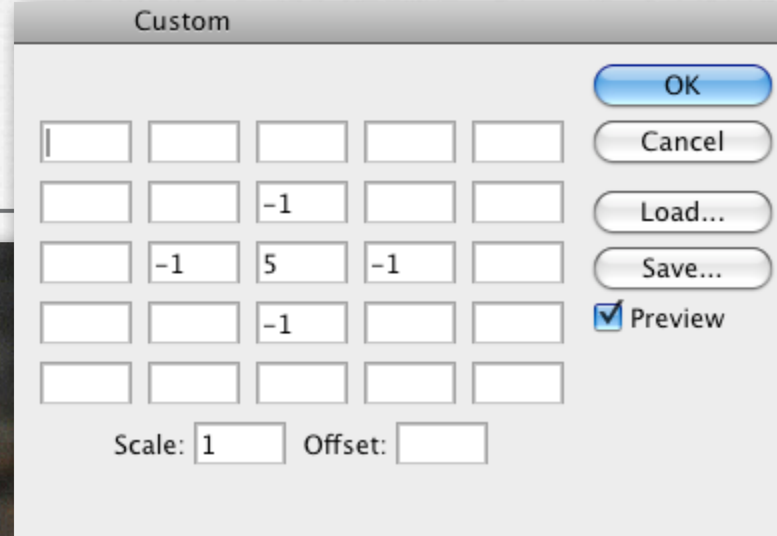
- bilateral filtering removes sensor noise without blurring edges
- can be applied more (or less) strongly to chrominance than luminance
- can be combined with demosaicing

Sharpening

original

(Marc Levoy)

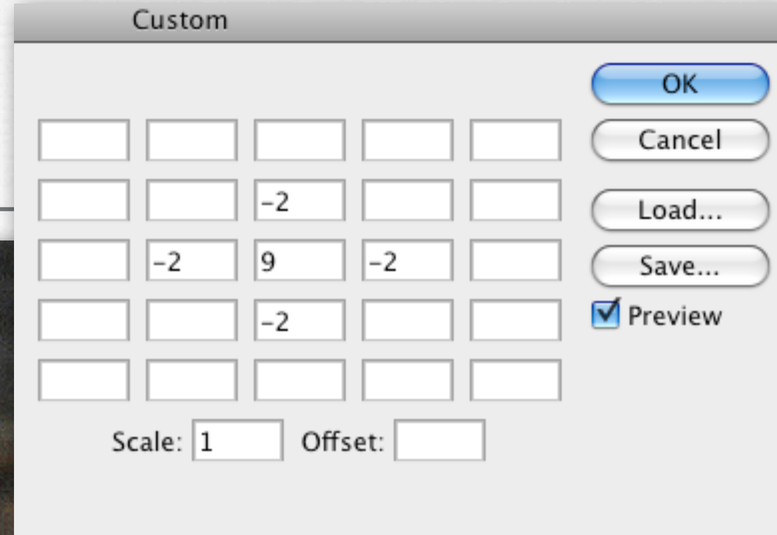
Sharpening



Filter/Other/Custom in Photoshop CS4

(Marc Levoy)

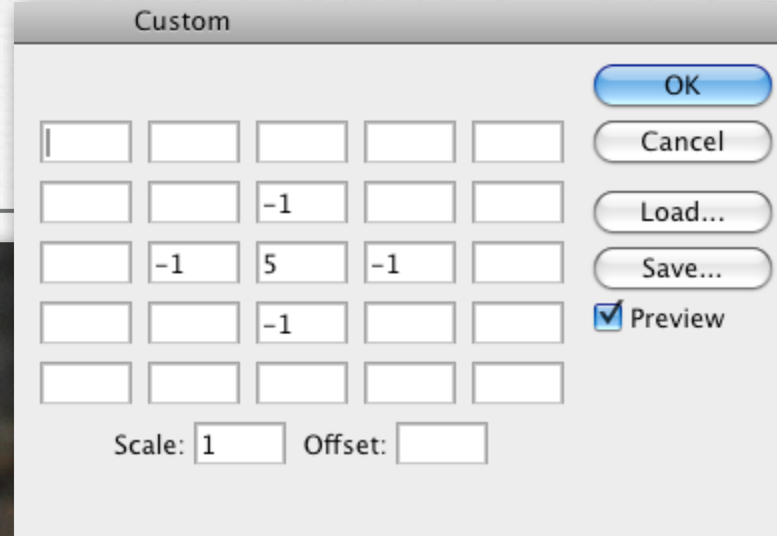
Sharpening

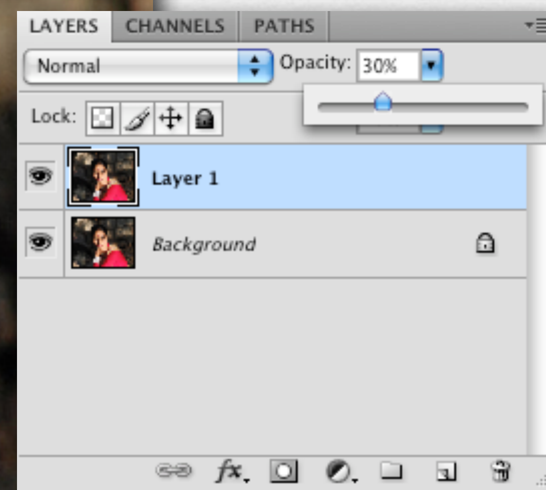


Filter/Other/Custom in Photoshop CS4

(Marc Levoy)

Sharpening

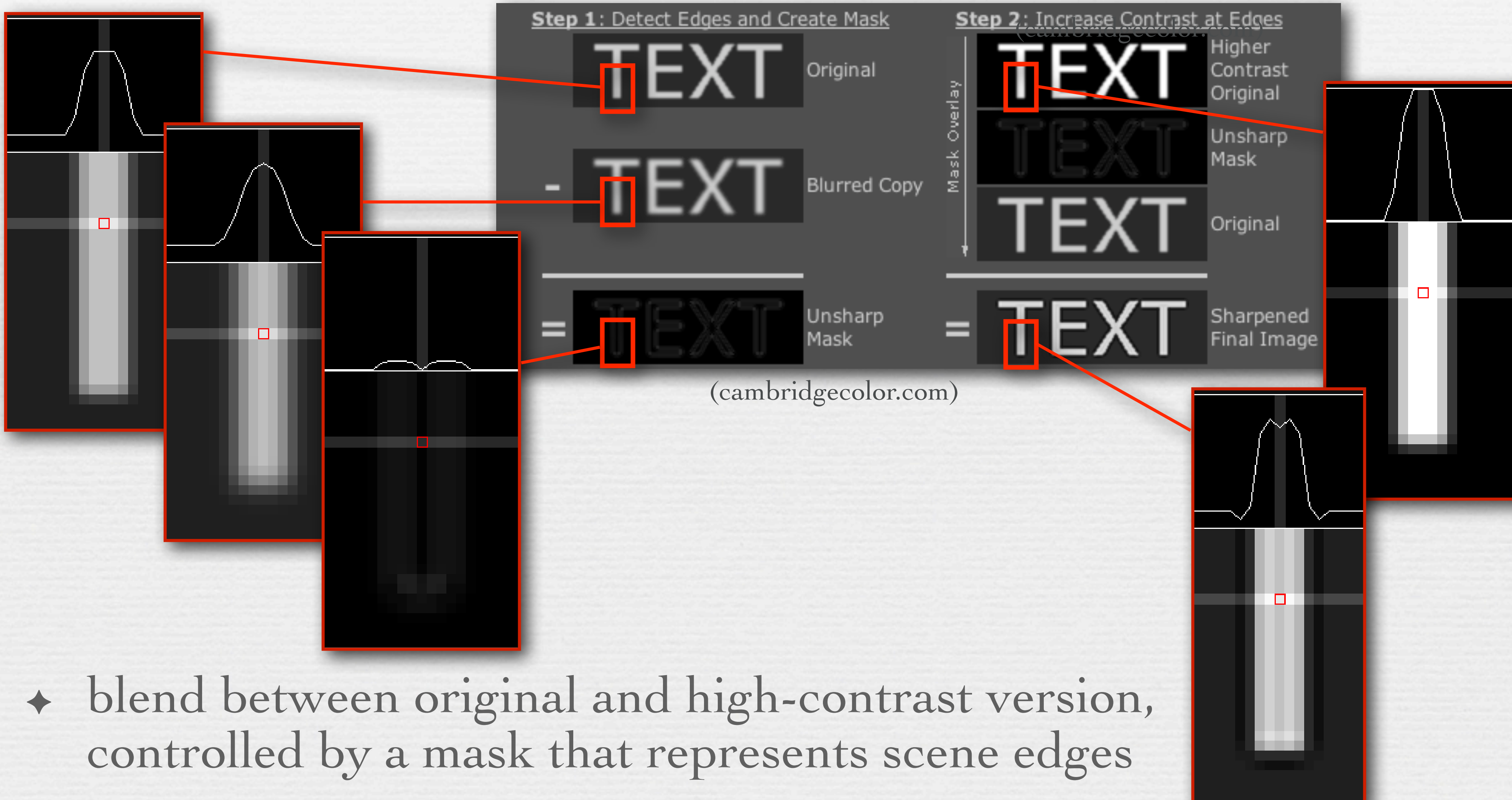




1st layer is original,
2nd layer is sharpened,
blend w. 30% opacity

(Marc Levoy)

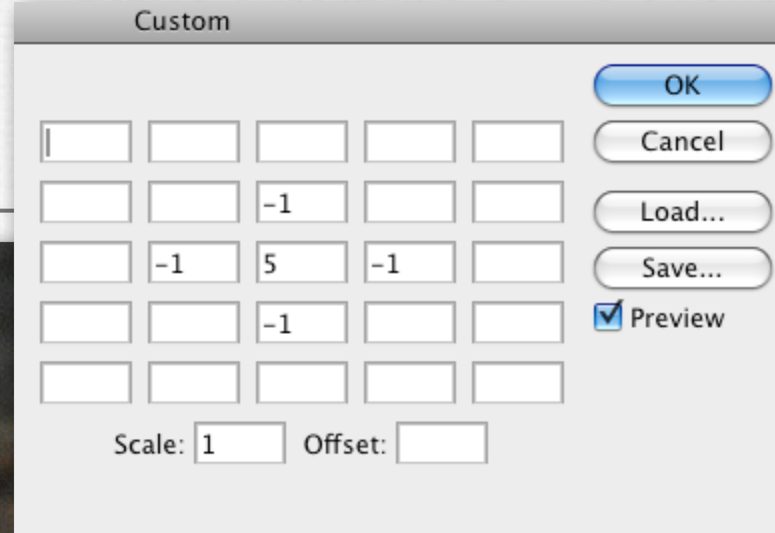
Unsharp masking



(cambridgecolor.com)

- 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

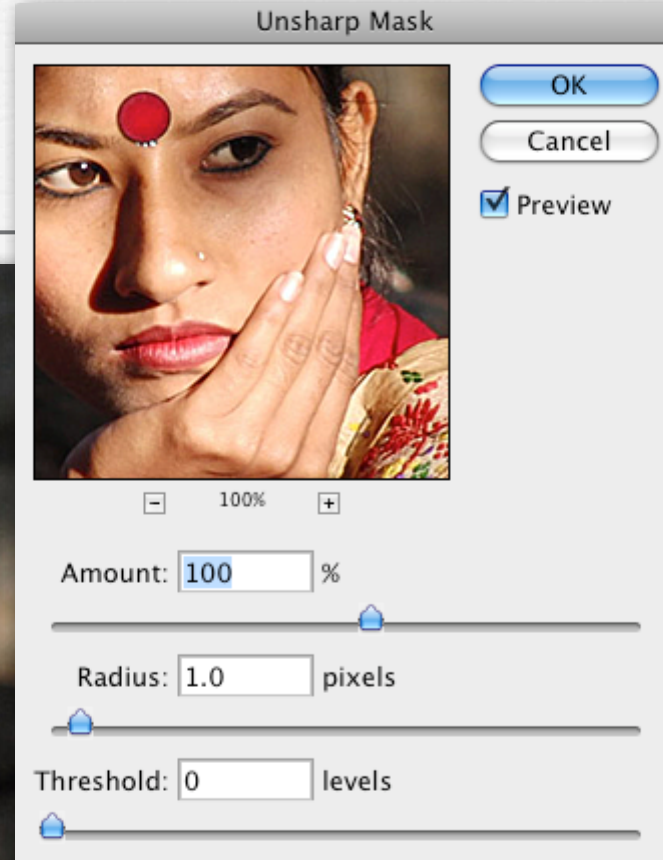
Sharpening



Filter/Other/Custom in Photoshop CS4

(Marc Levoy)

Sharpening

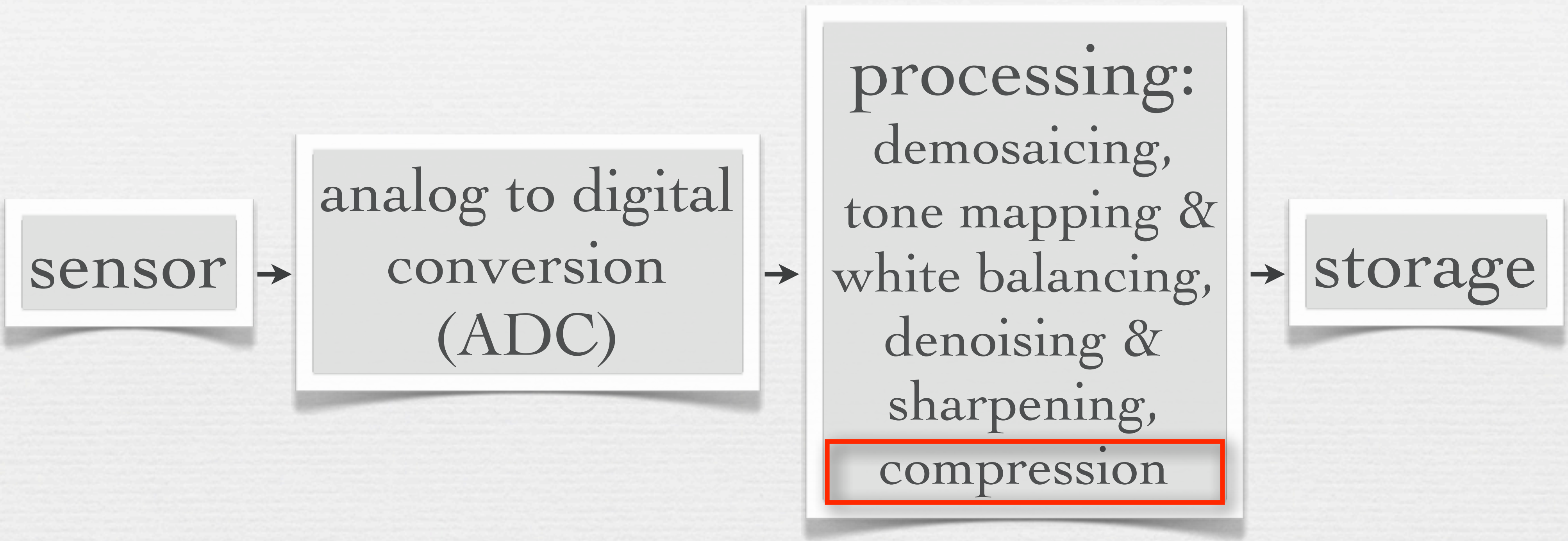


Filter/Sharpen/
Unsharp Mask in CS4

Sharpening

original

Camera pixel pipeline



JPEG files

- Joint Photographic Experts Group
  - organized 1986, standard adopted 1994
- defines how an image is to be compressed into a stream of bytes (codec) 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 each time you open → edit → save
  - especially if you crop or shift pixels (hence block boundaries)
  - for lossless compression, use PNG or TIFF

EXIF data

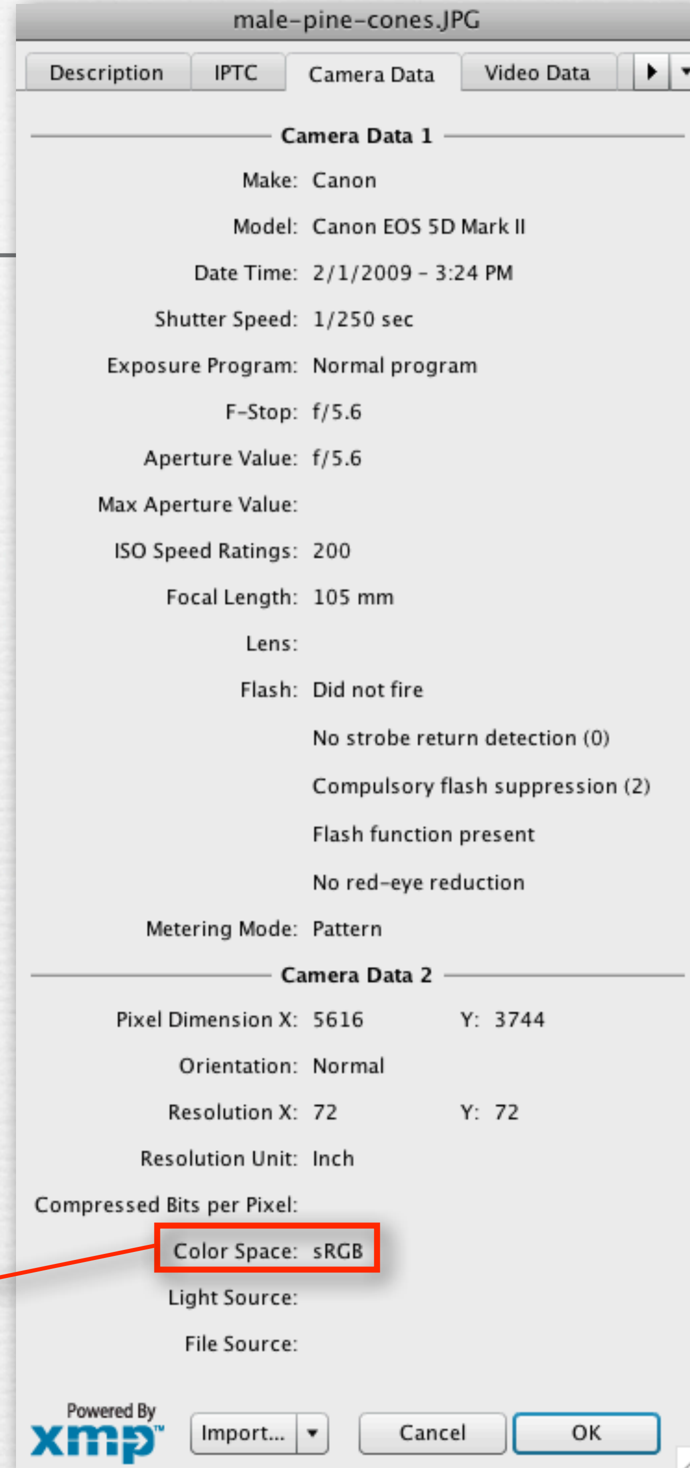
- Exchangeable Image File Format
  - created by Japan Electronic Industries Development Assoc.
- used by nearly every digital camera manufactured today
  - actually a file format
  - JPEG or TIFF file + metadata about the camera and shot
  - .JPG or .JPEG extension is used

EXIF data

File/File Info in Photoshop CS4

(Marc Levoy)

shot with Canon 5D Mark II



Color Space: sRGB

EXIF data

exiftool

```
Focal Length                    : 105.0 mm
```

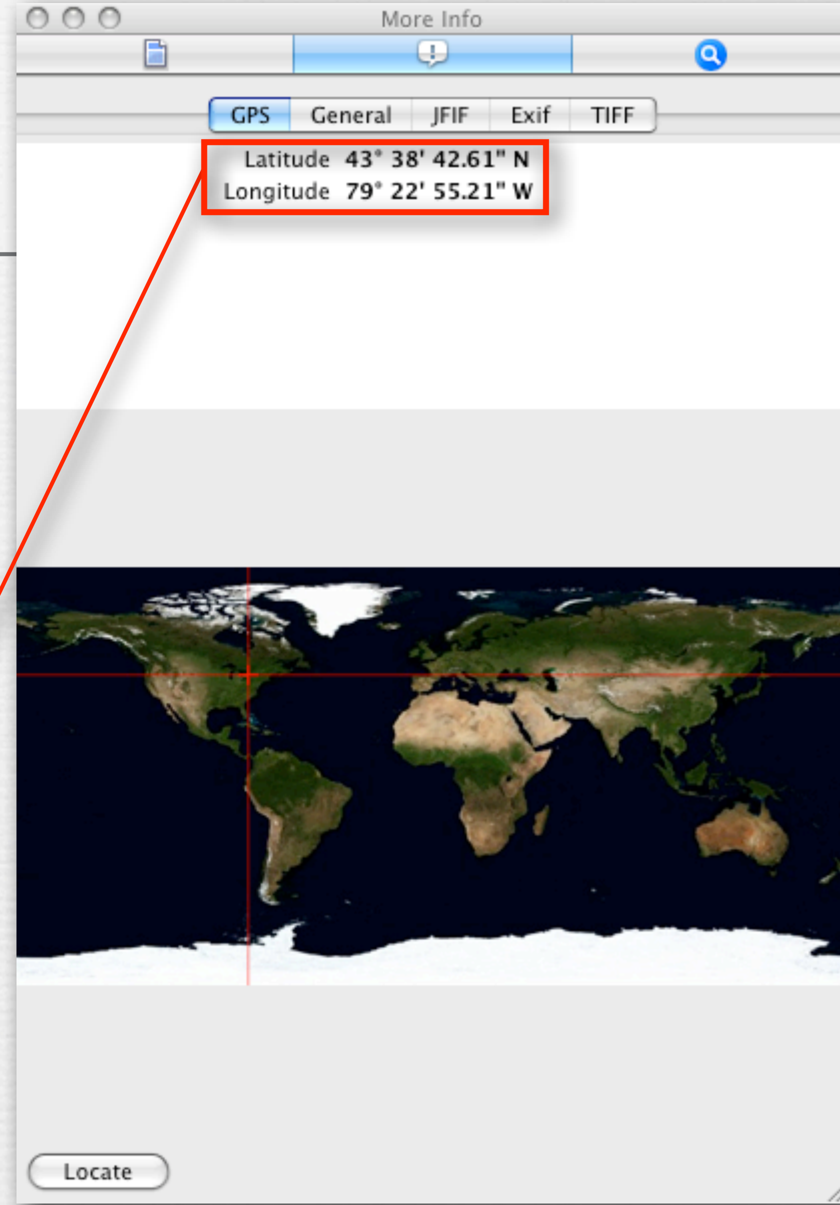
shot with Canon 5D Mark II

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

EXIF data

Mac Preview

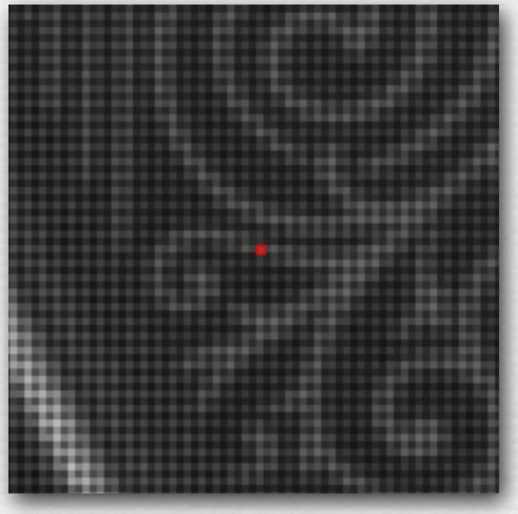
shot with iPhone 3G



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

RAW files

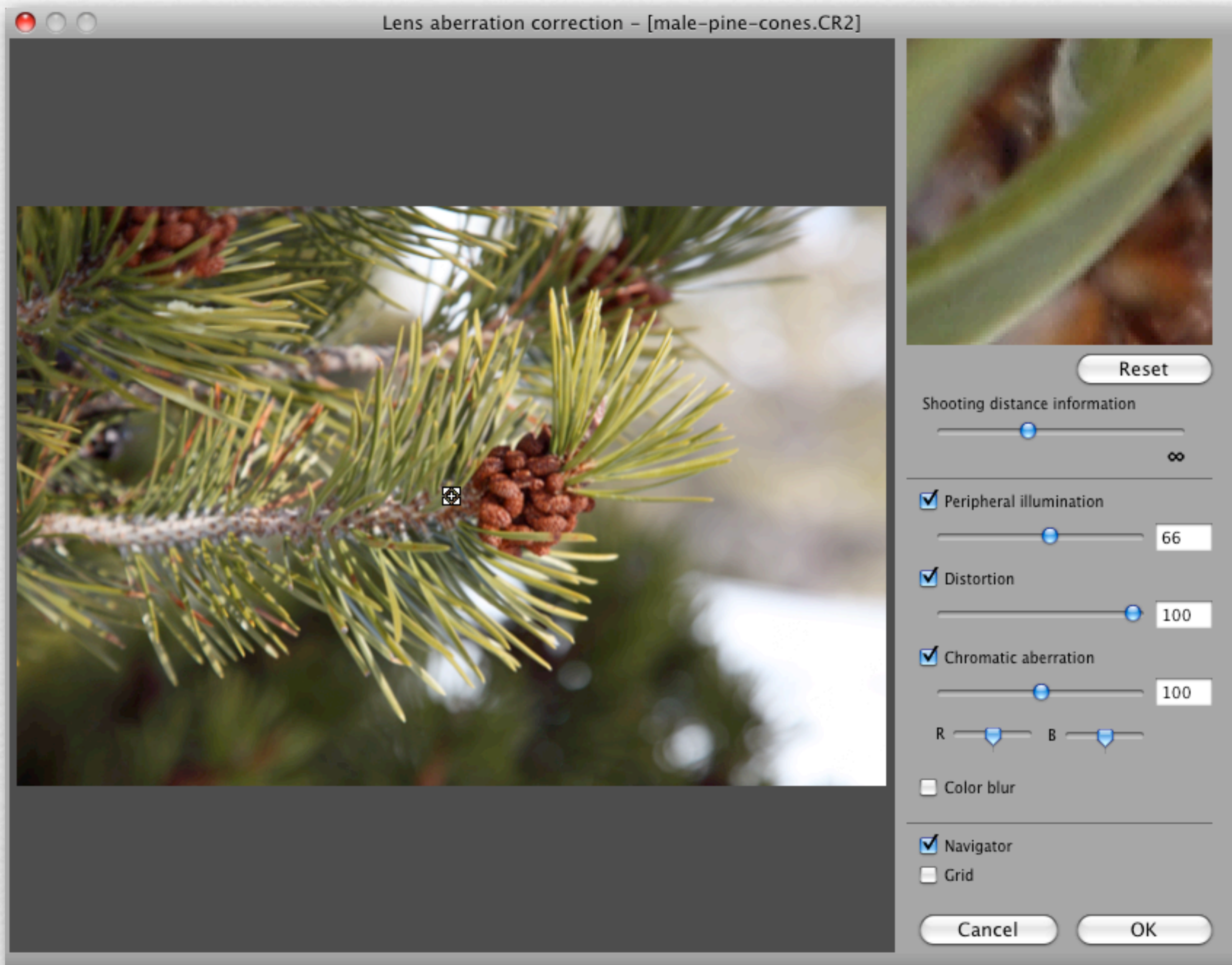
- minimally processed images, not even demosaiced
- uncompressed or losslessly compressed
- includes metadata, possibly encrypted
- file format varies by manufacturer
- example extensions: .CR2, .NEF



- processed and converted to a JPEG file using
  - proprietary software (e.g. Canon Digital Photo Professional)
  - Photoshop (if you're lucky)
  - freeware programs like `dcraw`
  - but their processing algorithms are all different!

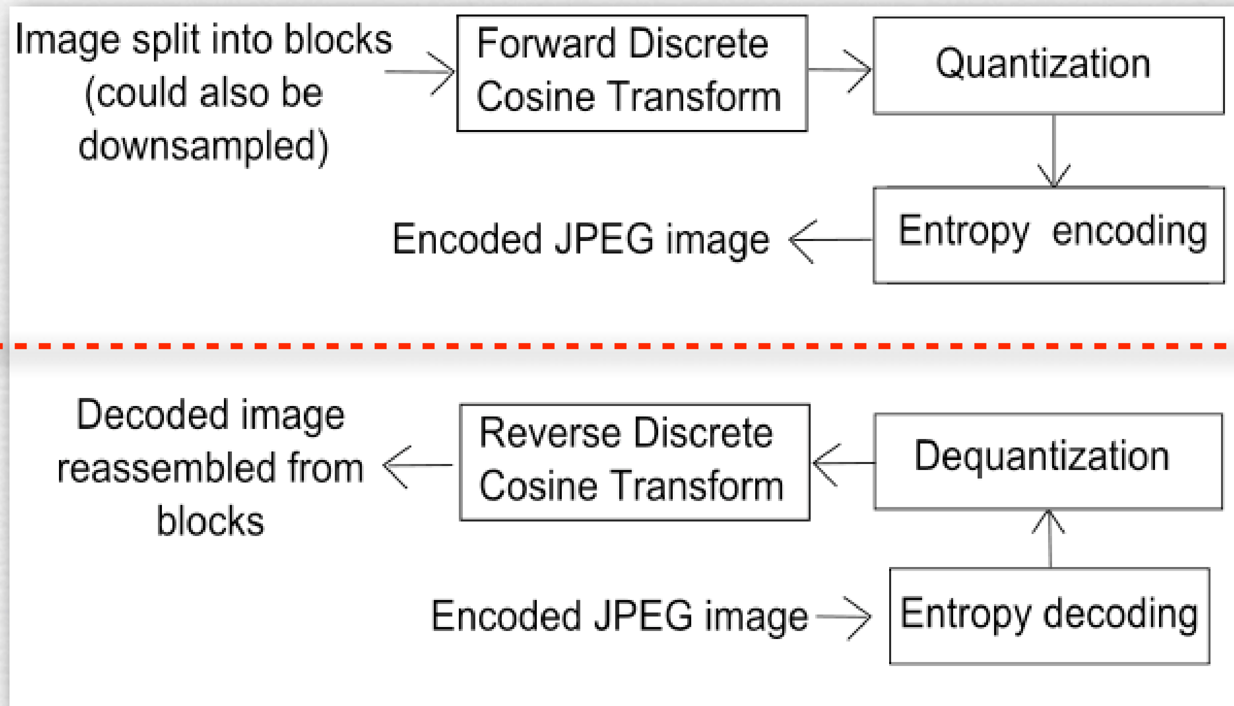
RAW file processor

Lens aberration correction panel in Canon Digital Photo Professional



JPEG image compression

- compression (in camera)



(wikipedia)

- decompression (for display)

JPEG image compression

$x \longrightarrow$, $y \downarrow$

−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

zero-mean image

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

8-bit image

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

8×8 pixel block

JPEG image compression

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

zero-mean image

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

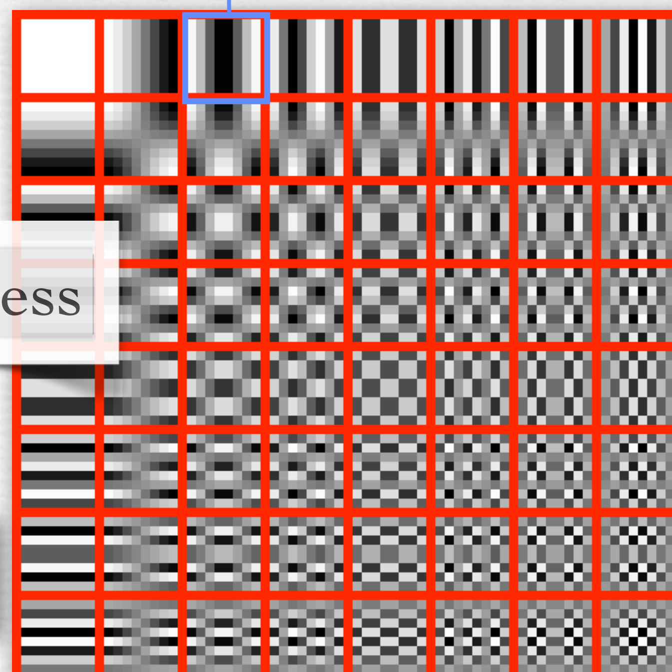
discrete cosine transform (DCT)

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

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

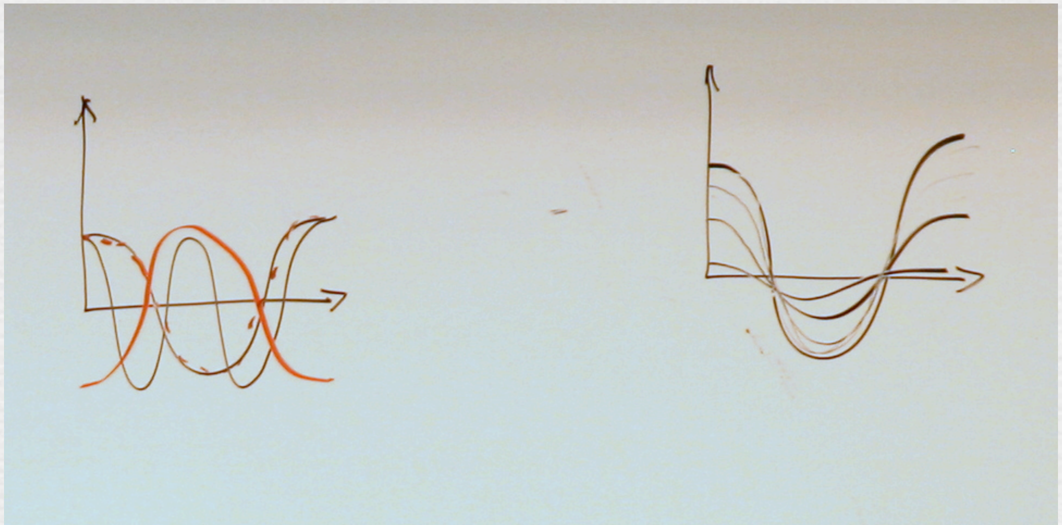
$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

lossless



Cosine basis functions

- at left (in black) are two cosine basis functions, of different frequencies (1 cycle across block width, and 2)
  - these are 1D; JPEG compression uses 2D functions
- at right are vertical scalings of one of these cosine basis functions; the amount of scaling is given by the coefficient computed in step #4 (previous slide)
  - if coefficient is negative, basis function is flipped vertically, as shown by red curve at left



JPEG image compression

16	11	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

bin size for each coefficient

$u \longrightarrow$, $v \downarrow$

−415	−30	−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

discrete cosine transform (DCT)

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

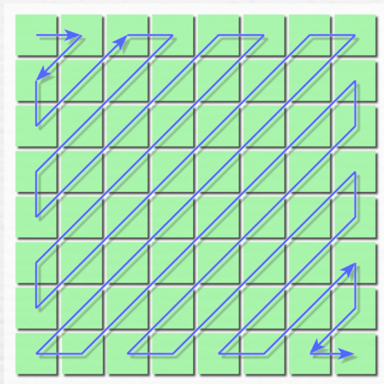
lossy

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

−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

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	Q = 25	Q = 1
		
2.6 : 1	23 : 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

Slide credits

- Fredo Durand
- Jennifer Dolson

- Stone, M., *A Field Guide to Digital Color*, A.K. Peters, 2003.
- 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.