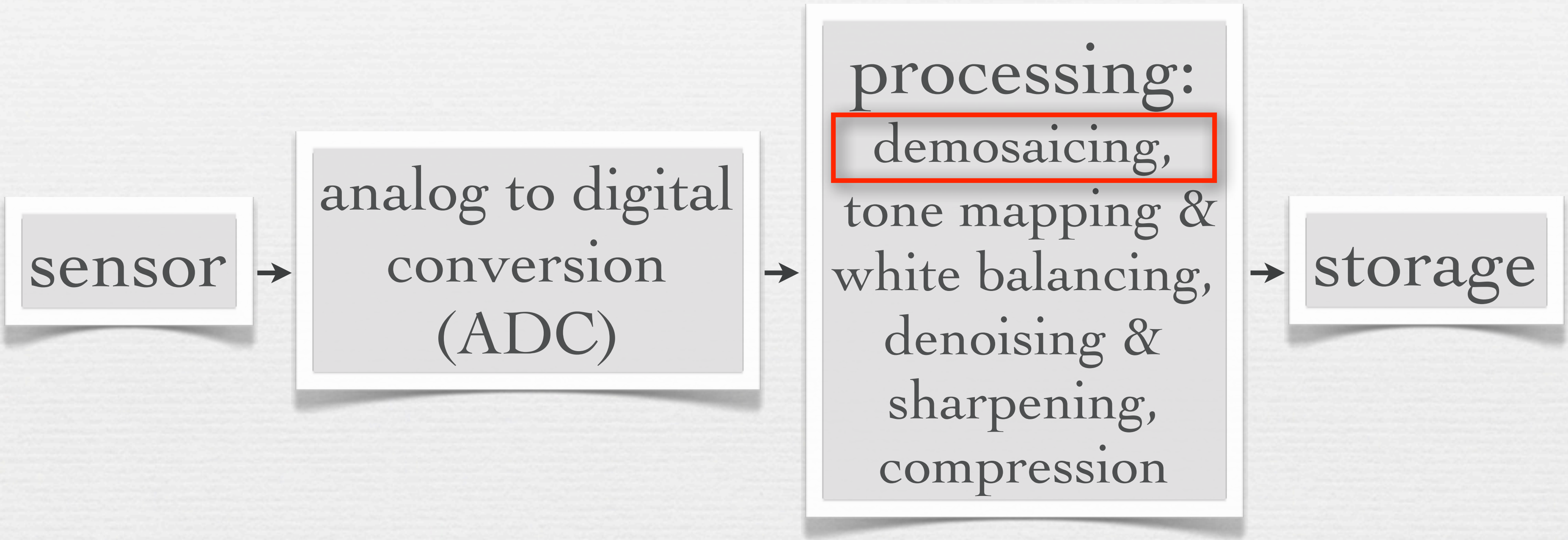
Post-processing pipeline

CS 448A, Winter 2010

Marc Levoy
Computer Science Department
Stanford University

Camera pixel pipeline



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

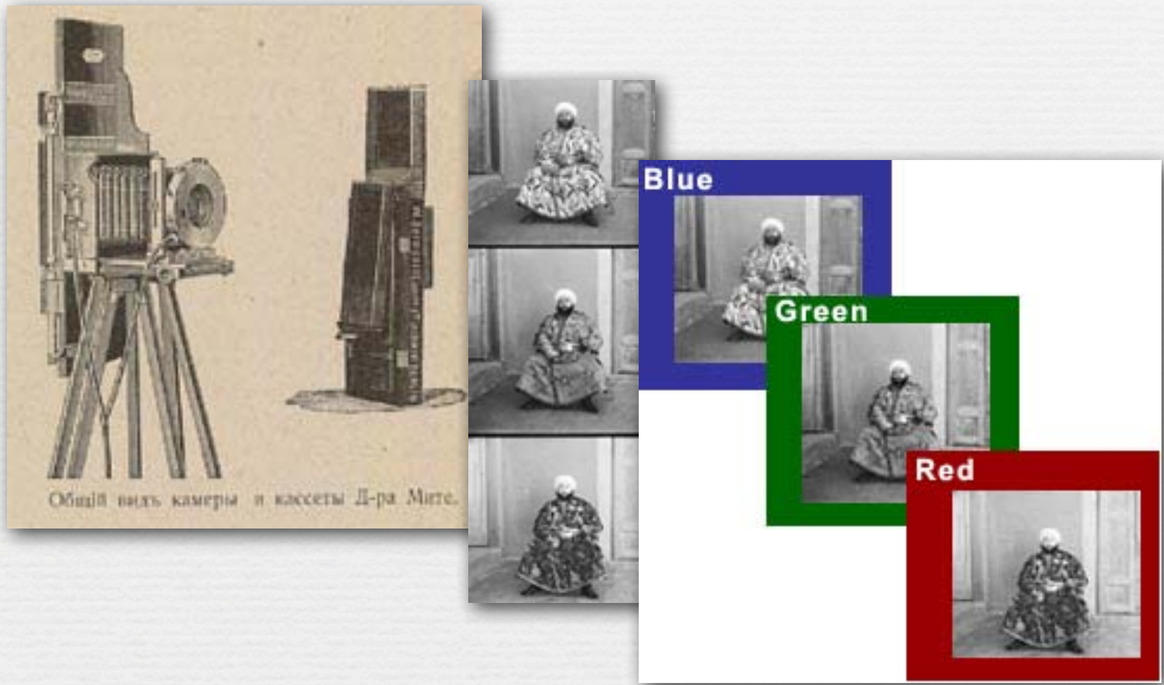
Sensing color

- silicon detects all visible frequencies well
- can't differentiate wavelengths after photon knocks an electron loose
  - all electrons look alike
- must select desired frequencies before light reaches photodetector
  - block using a filter, or separate using a prism or grating
- 3 spectral responses is enough
  - a few consumer cameras record 4
- silicon is also sensitive to near infrared (NIR)
  - most sensors have an IR filter to block it
  - to make a NIR camera, remove this filter

Color sensing technologies

- field-sequential
- 3-sensor
- vertically stacked
- color filter arrays

Sergey Prokudin-Gorsky

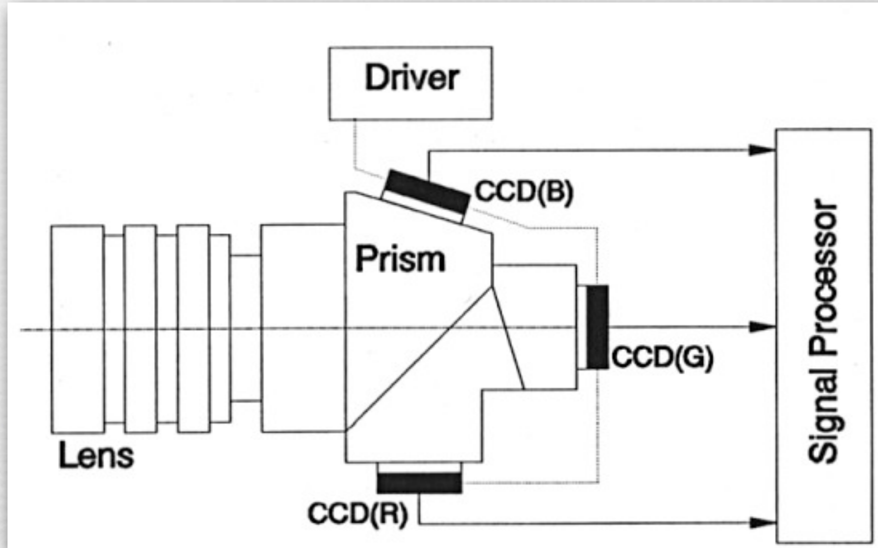


- shot sequentially through R, G, B filters
- simultaneous projection provided good saturation, but available printing technology did not
- digital restoration lets us see them in full glory...

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

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

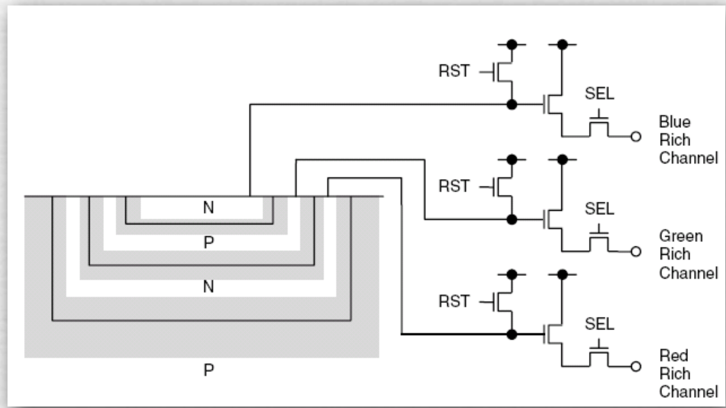
3-CCD cameras



(Theuwissen)

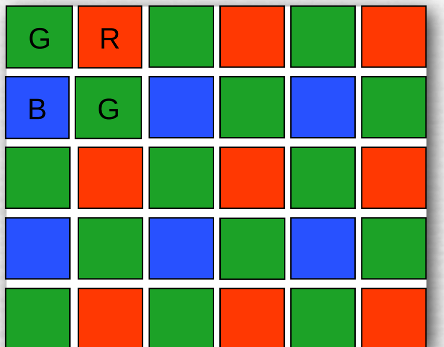
- high-quality video cameras
- prism & dichroic mirrors split the image into 3 colors, each routed to a separate CCD sensor
- no light loss, as compared to filters
- expensive, and complicates lens design

Foveon stacked sensor

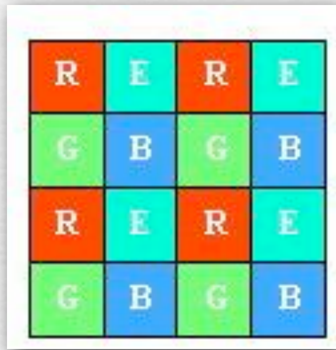


- longer wavelengths penetrate deeper into silicon, so arrange a set of vertically stacked detectors
  - top gets mostly blue, middle gets green, bottom gets red
  - no control over spectral responses, so requires processing
- fewer color artifacts than color filter arrays
  - but possibly worse noise performance

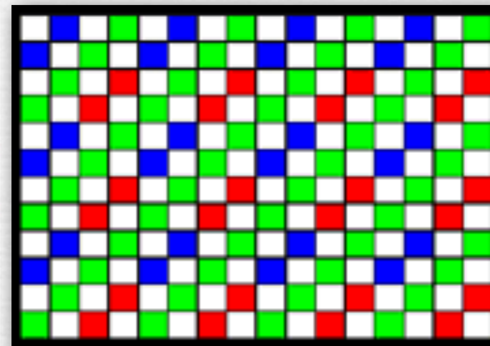
Color filter arrays



Bayer pattern



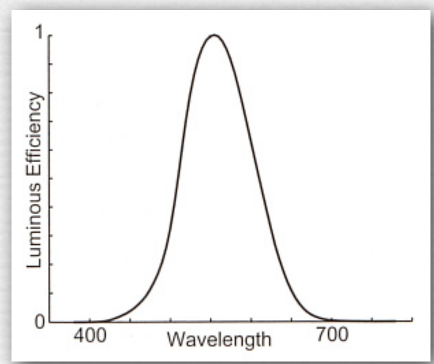
Sony RGB+E
better color



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
  - sensitivity given by the human luminous efficiency curve

(Stone)



Example of Bayer mosaic image

Small fan at Stanford women's soccer game

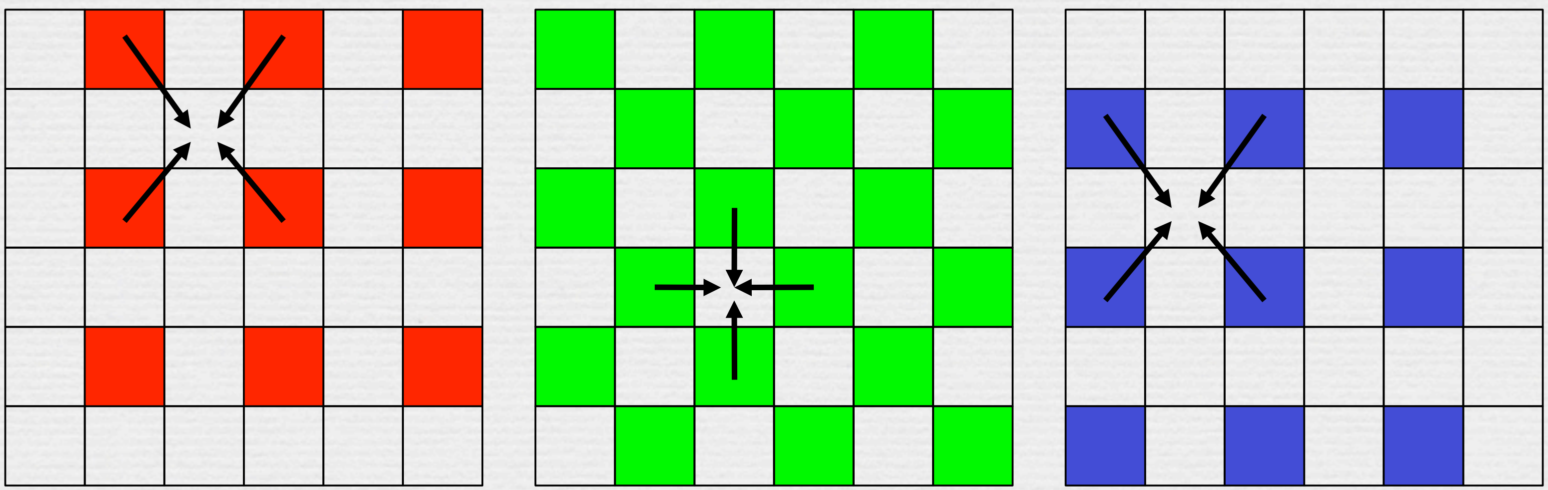
(Canon 1D III)

Example of Bayer mosaic image

Before demosaicking (`dcraw -d`)

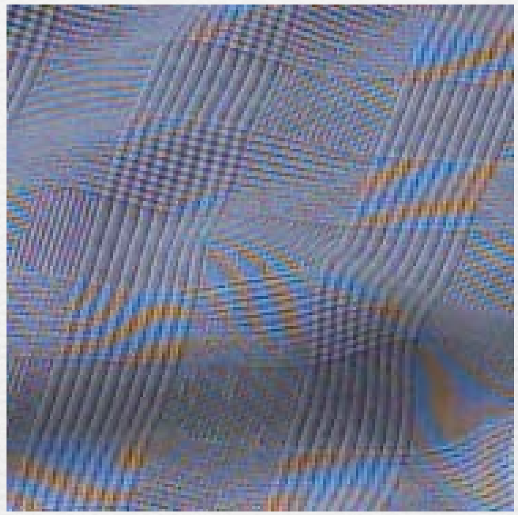
Demosaicing

- 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)

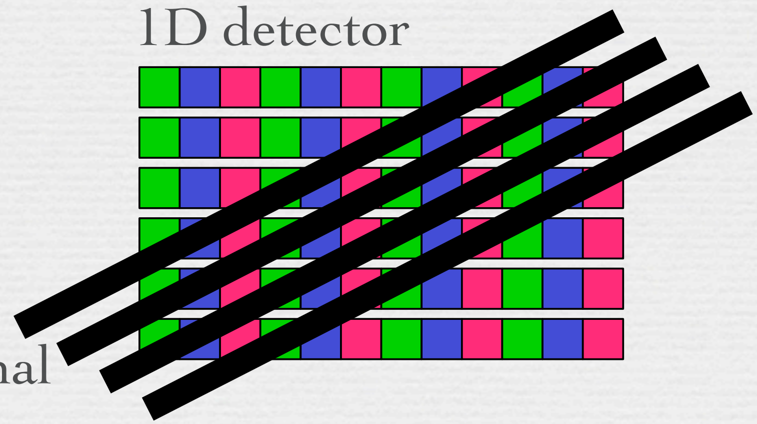


Demosaicing errors

- 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



Common solution:
low-pass filter chrominance signal

- color artifacts are places where chrominance changes abruptly but only transiently

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)

Y

Cb

Cr

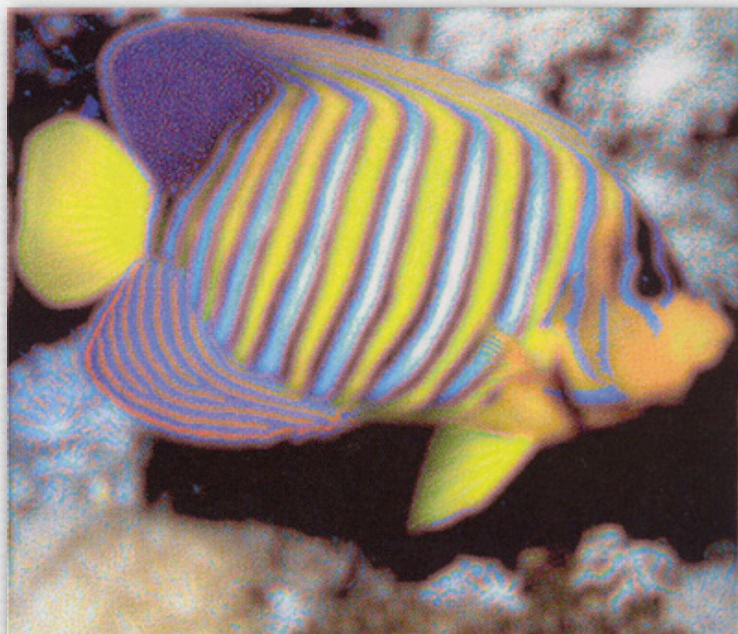
Apparent spatial sharpness depends mainly on luminance, not chrominance

original image

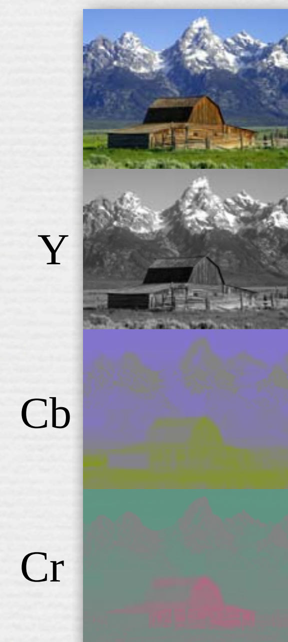
(Wandell)

Apparent spatial sharpness depends mainly on luminance, not chrominance

luminance
channel
blurred



(Wandell)



Common solution:
low-pass filter chrominance signal

- color artifacts are places where chrominance changes abruptly but only transiently
- 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

spike noise
is removed

monotonic edges
remain unchanged

Common solution: low-pass filter chrominance signal

- color artifacts are places where chrominance changes abruptly but only transiently
- 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
- summary of 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)

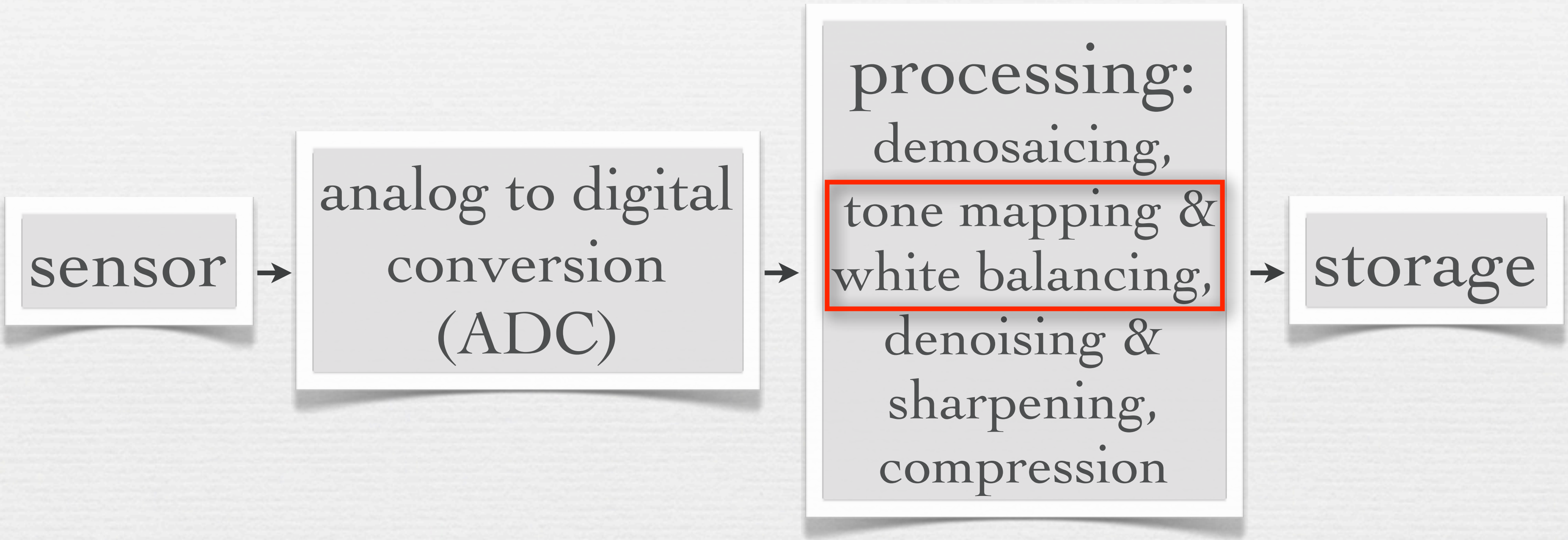
Comparison

linear interpolation

median-filtered interpolation

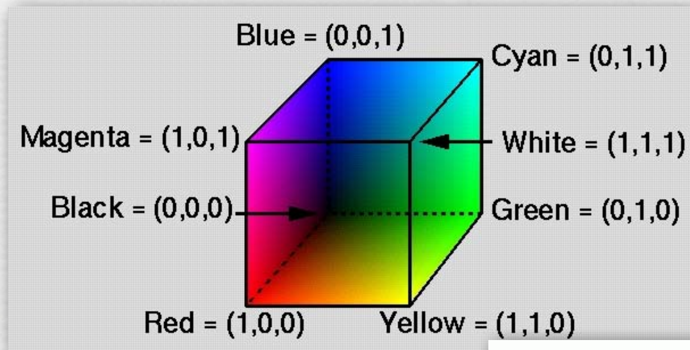
- take-home lesson: 2/3 of your data is made up!
- there are better and worse ways to do this

Camera pixel pipeline



Summary of chromaticity diagrams (1 of 2)

- choose three primaries R,G,B, pure wavelengths or not
- adjust R=1,G=1,B=1 to obtain a desired *reference white*
- this yields an *RGB cube*



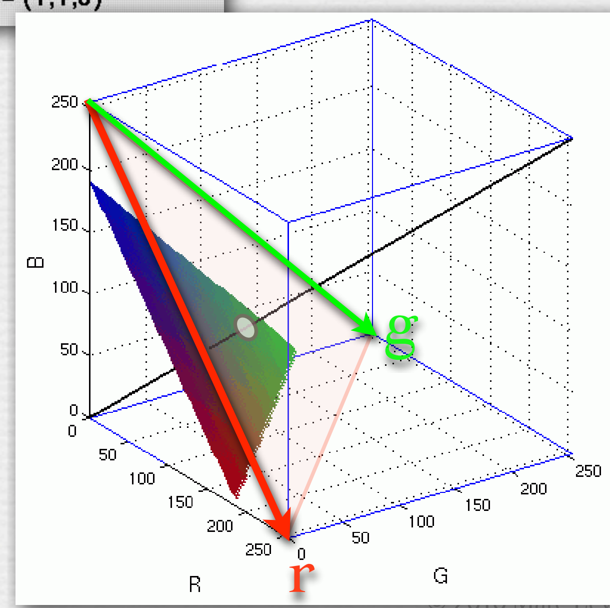
$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

(FLASH DEMO)

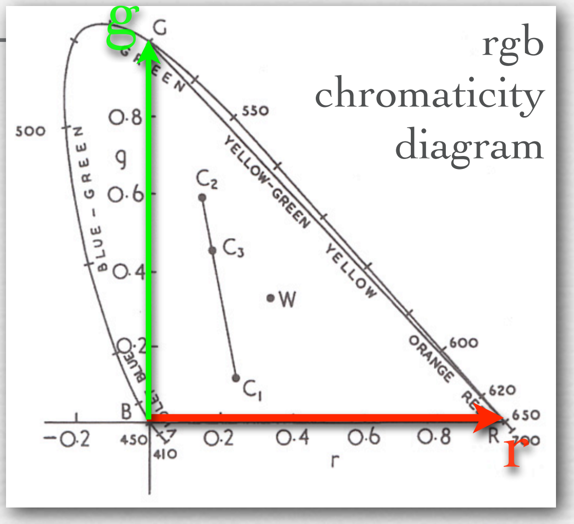
http://graphics.stanford.edu/courses/cs178/applets/threedgamut.html

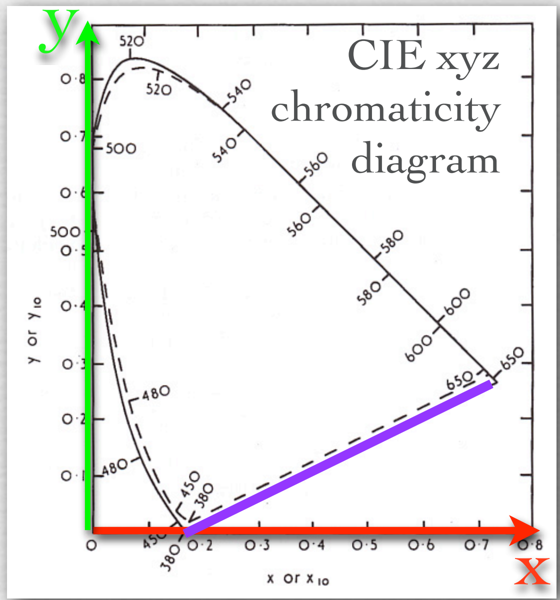
- one may factor the brightness out of any point in the cube by drawing a line to the origin and intersecting this line with the triangle made by corners Red, Green, Blue
- all points on this triangle, which are addressable by two coordinates, have the same brightness but differing *chromaticity*



Summary of chromaticity diagrams (2 of 2)

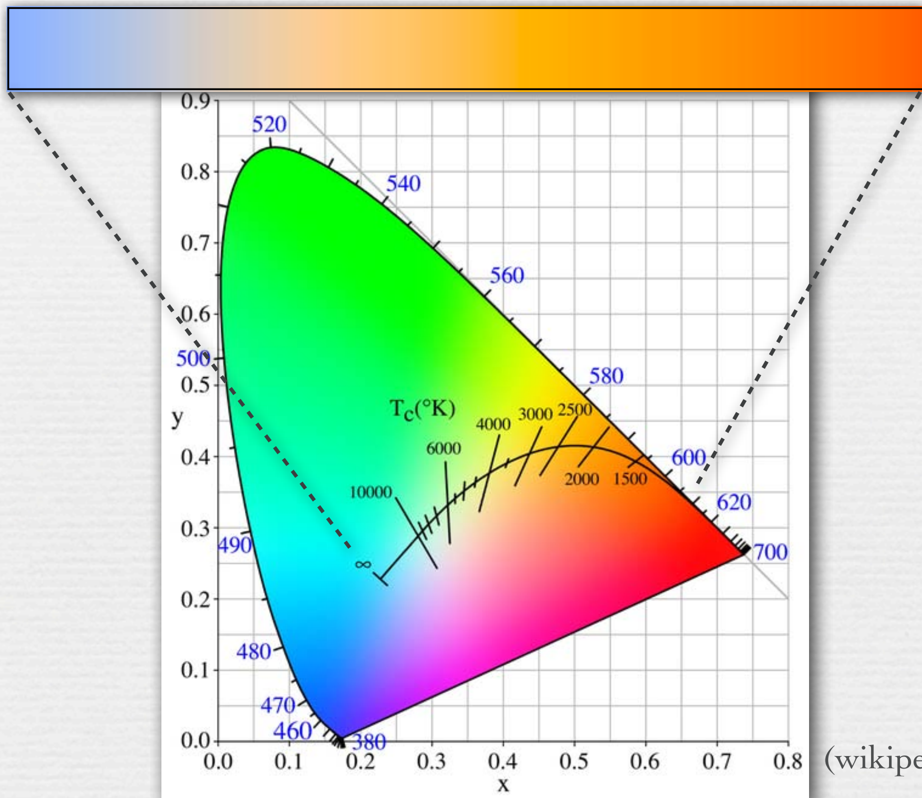
- this triangle is called the *rgb chromaticity diagram* for the chosen RGB primaries
  - mixtures of colors lie along straight lines
  - neutral (black to white) lies at (⅓, ⅓)
  - r>0, g>0 does not enclose spectral locus
- the same construction can be performed using <u>any</u> set of 3 vectors as primaries, even physically impossible ones
- the CIE has defined a set of primaries XYZ, and the associated *xyz chromaticity diagram*
  - x>0, y>0 <u>does</u> enclose spectral locus
  - one can connect red and green on the locus with a *line of extra-spectral purples*
  - x,y is a standardized way to denote colors





(Hunt)

Application of chromaticity diagrams #1: color temperature and white balancing



(wikipedia)

<u>correlated color temperatures</u>

3200°K incandescent light
4000°K cool white fluorescent
5000°K equal energy white (D50, E)
6000°K midday sun, photo flash
6500°K overcast, television (D65)
7500°K northern blue sky

- the apparent colors emitted by a *black-body radiator* heated to different temperatures fall on a curve in the chromaticity diagram
- for non-blackbody sources, the nearest point on the curve is called the *correlated color temperature*

White balancing in digital photography

- 1. choose an object in the photograph you think is neutral (somewhere between black and white) in the real world
- 2. compute scale factors (S_R,S_G,S_B) that map the object's (R,G,B) to neutral (R=G=B), i.e. $S_R = \frac{1}{3}(R+G+B) / R$, etc.
- 3. apply the same scaling to all pixels in the sensed image
- the eventual appearance of R=G=B, hence of your chosen object, depends on the color space of the camera
  - the color space of most digital cameras is sRGB
  - the reference white for sRGB is D65 (6500°K)
- thus, white balancing on an sRGB camera forces your chosen object to appear 6500°K (blueish white)
- if you trust your object to be neutral, this procedure is equivalent to finding the color temperature of the illumination

Finding the color temperature of the illumination

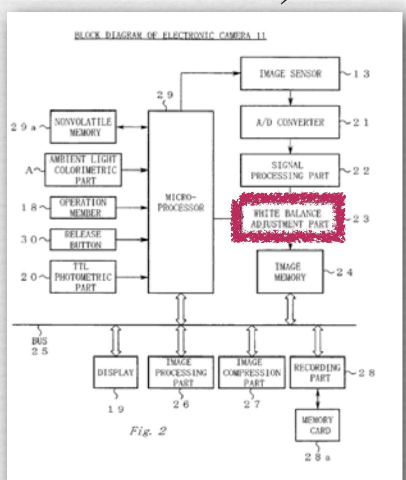
- Auto White Balance (AWB)
  - gray world: assume the average color of a scene is gray, so force the average color to be gray - often inappropriate

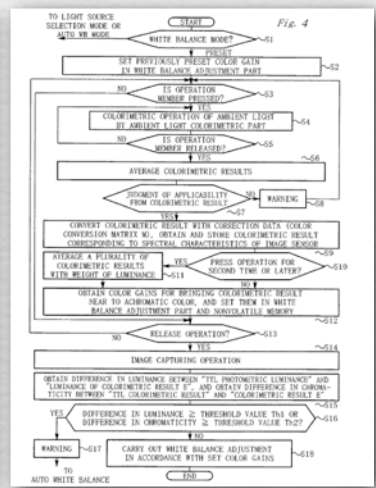
(Marc Levoy)

average (R, G, B) = (100%, 81%, 73%) ➔ (100%, 100% 100%)

Finding the color temperature of the illumination

- Auto White Balance (AWB)
  - gray world: assume the average color of a scene is gray, so <u>force</u> the average color to be gray - often inappropriate
  - assume the brightest pixel (after demosaicing) is a specular highlight and therefore should be white
    - fails if pixel is saturated
    - fails if object is metallic - gold has gold-colored highlights
  - find a neutral-colored object in the scene
    - but how??

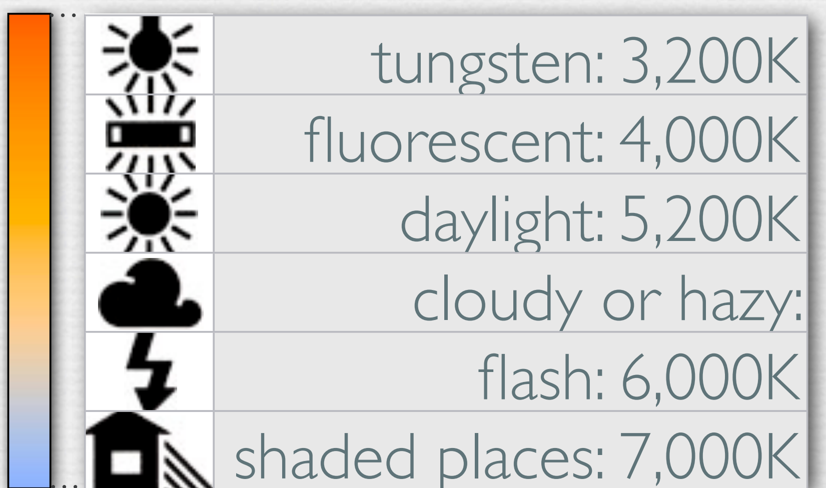


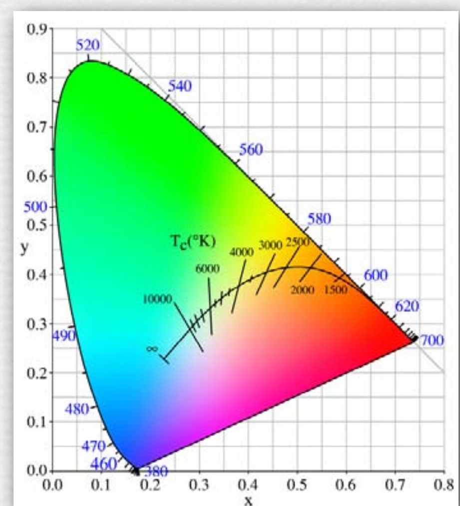


(Nikon patent)

Finding the color temperature of the illumination

- Auto White Balance (AWB)
- manually specify the illumination's color temperature
  - each color temperature corresponds to a unique (x,y)
  - for a given camera, one can measure the (R,G,B) values recorded when a neutral object is illuminated by this (x,y)
  - compute scale factors (S_R,S_G,S_B) that map this (R,G,B) to neutral (R=G=B); apply this scaling to all pixels as before





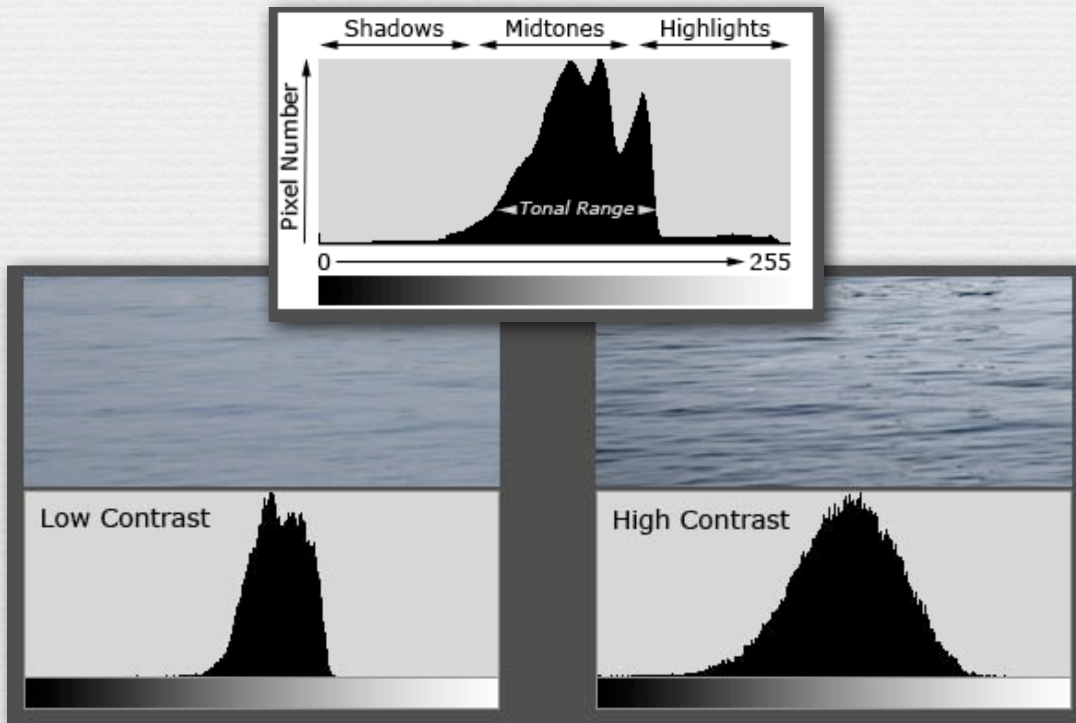
Incorrectly chosen white balance

(Eddy Talvala)

- scene was photographed in sunlight, then re-balanced as if it had been photographed under something warmer, like tungsten
  - re-balancer assumed illumination was very reddish, so it boosted blues
  - same thing would have happened if originally shot with tungsten WB

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 $0 \le I_i \le 1$)
  - simple but crude

original

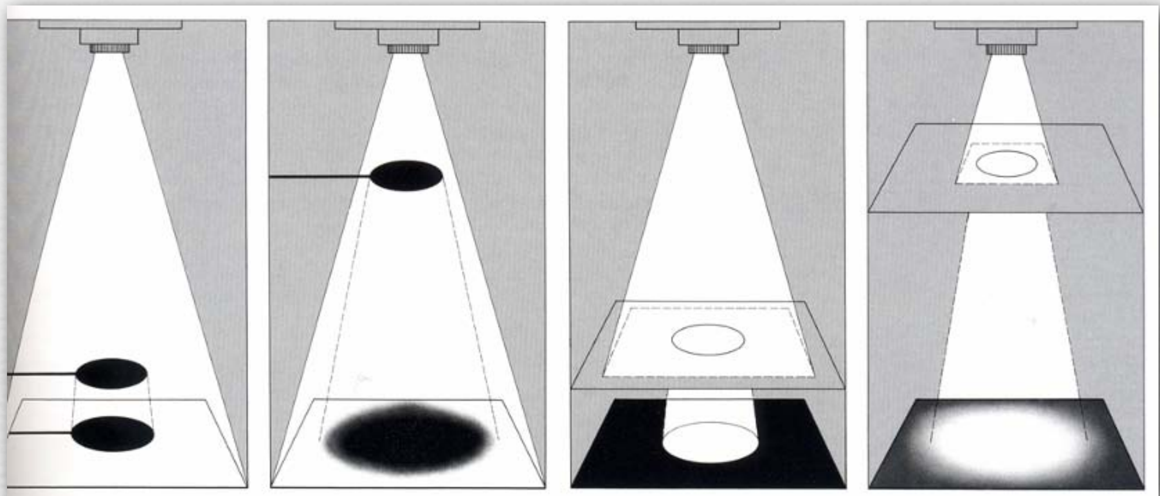
$\gamma = 0.5$

$\gamma = 2.0$

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 $0 \leq I_i \leq 1$)
  - simple but crude
- global versus local transformations

Traditional dodging and burning

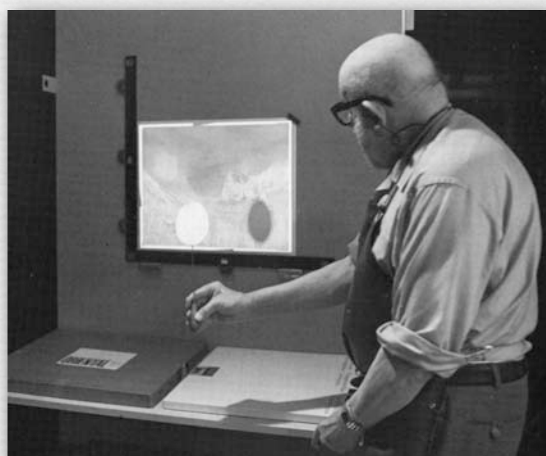


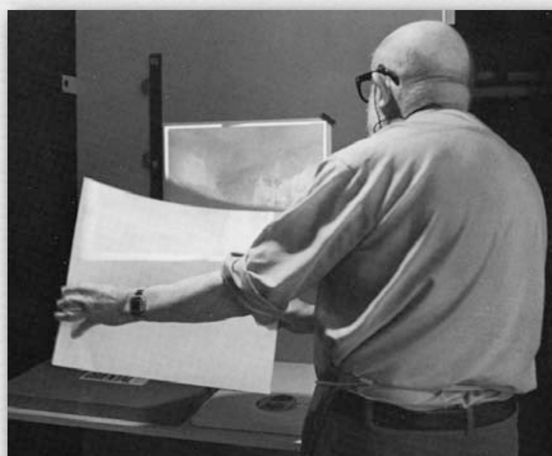
(Rudman)

dodging
(leaves print lighter)

burning
(makes print darker)

Ansel Adams in his darkroom





(Adams)

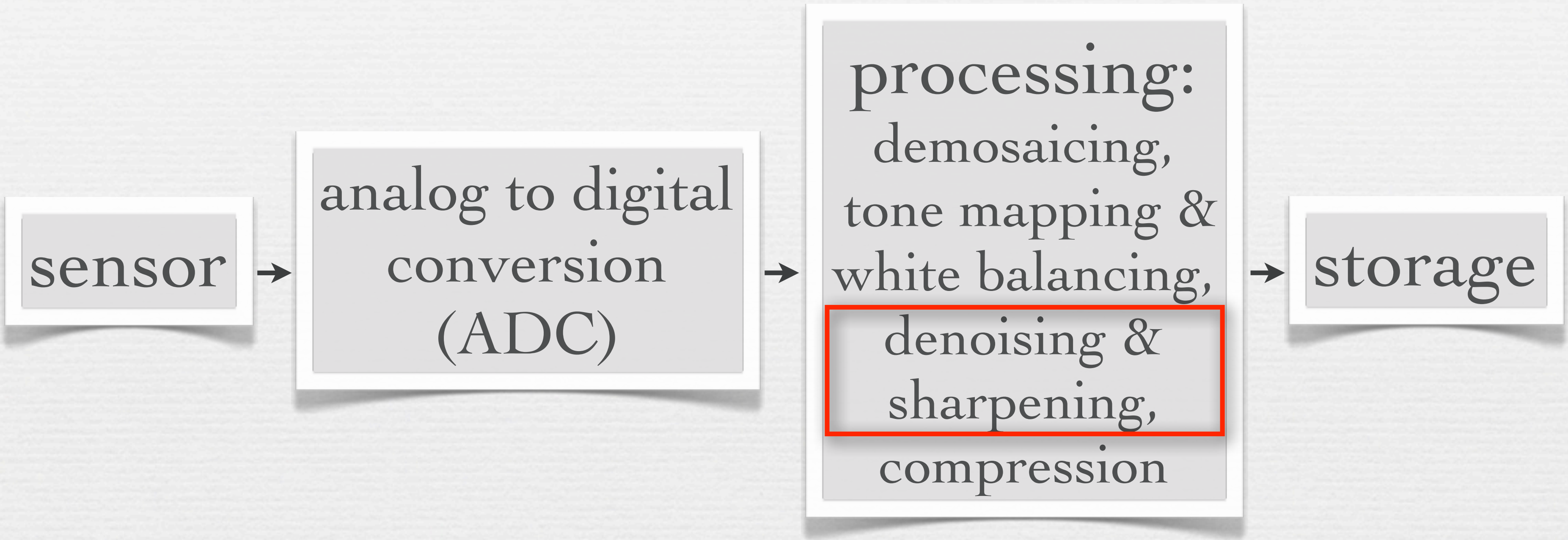
straight
print

Ansel Adams, Clearing Winter Storm, 1942

toned
print

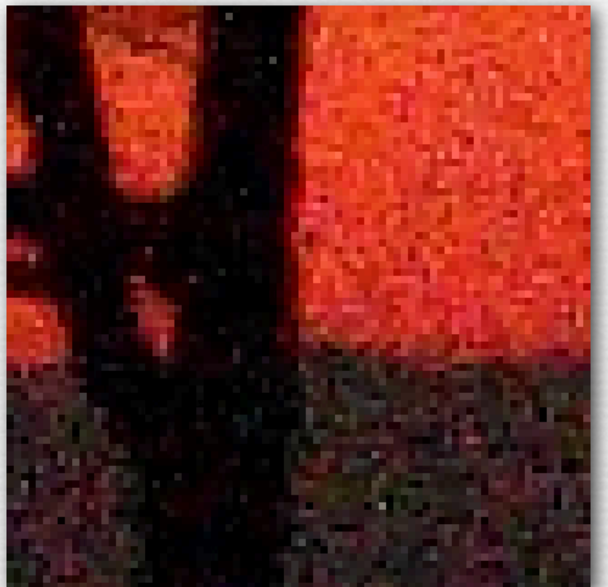
Ansel Adams, Clearing Winter Storm, 1942

Camera pixel pipeline

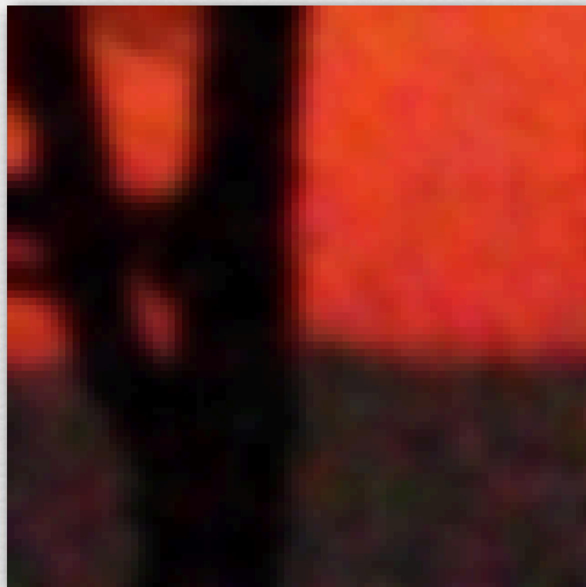


Canon 5D II at dusk

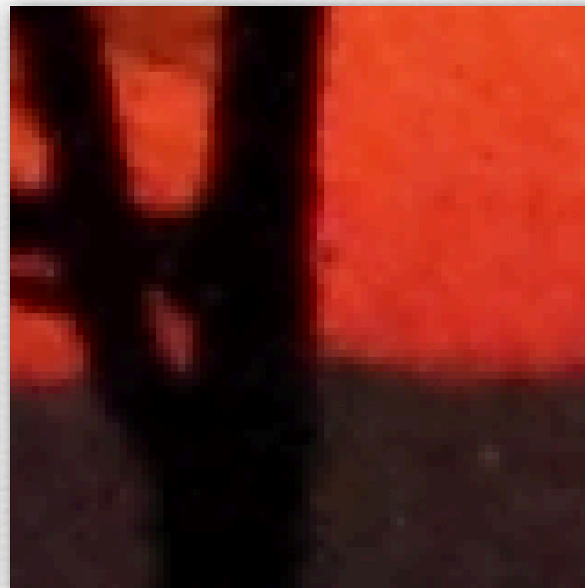
Denoising



RAW (ISO 6400)



Gaussian blur, radius = 1.3

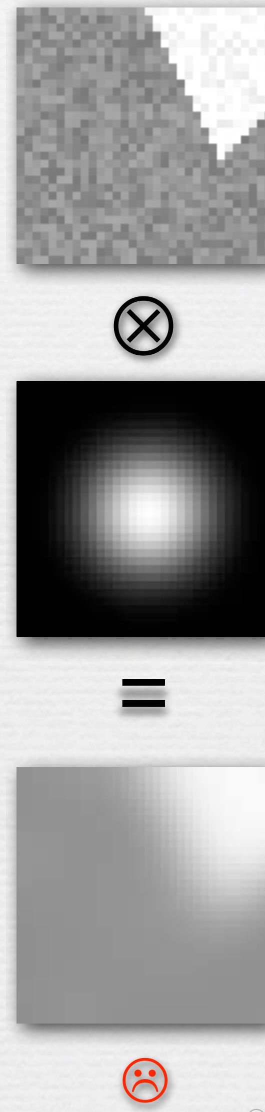


Canon denoising

- 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...

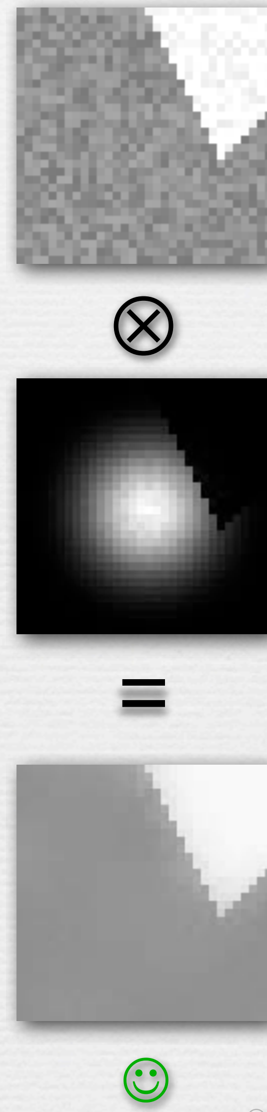
Bilateral filtering [Tomasi ICCV 1998]

- assume the image is piecewise constant with noise added
- therefore, nearby pixels are probably a different noisy measurement of the same data
- blurring doesn't work
- we should do a weighted blur where the weight is the probability a pixel is from the same piece of the scene

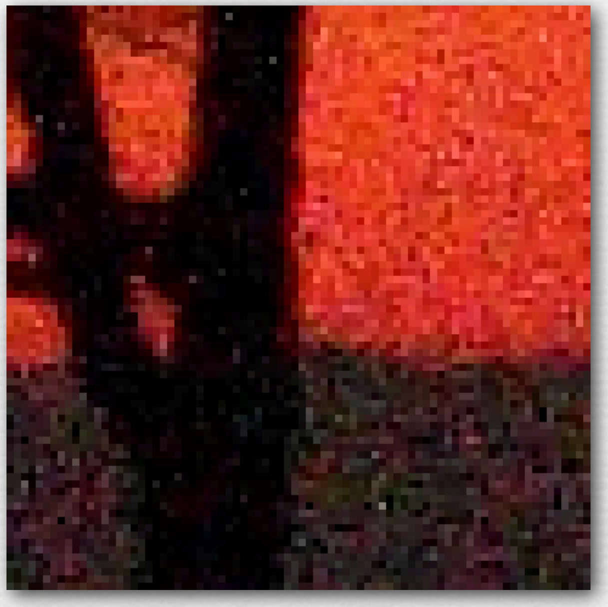


Bilateral filtering

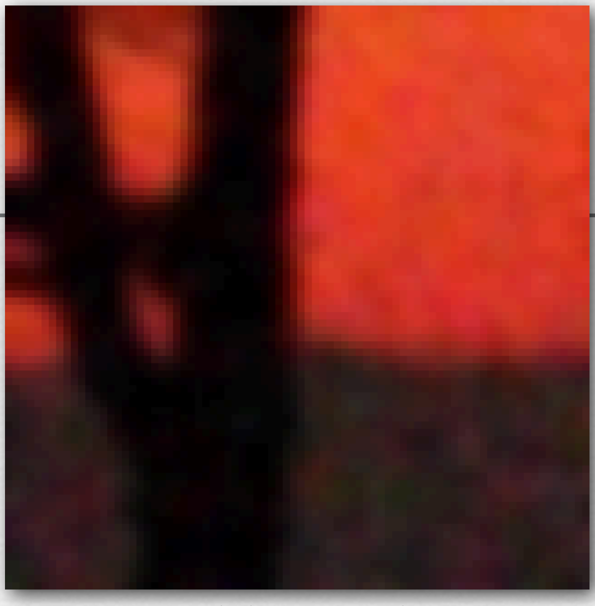
- if the pixels are similar in intensity, the probability they are from the same piece of the scene is high
- so perform a convolution where the weight assigned to nearby pixels falls off
  - with increasing *(x,y)* distance from the pixel being blurred
  - with increasing intensity difference from the pixel being blurred
- i.e. blur in the *domain* and *range* dimensions!



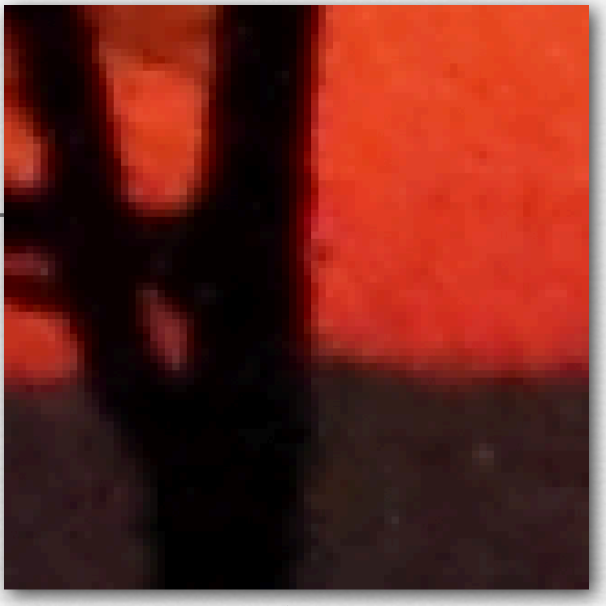
Denoising



RAW (ISO 6400)

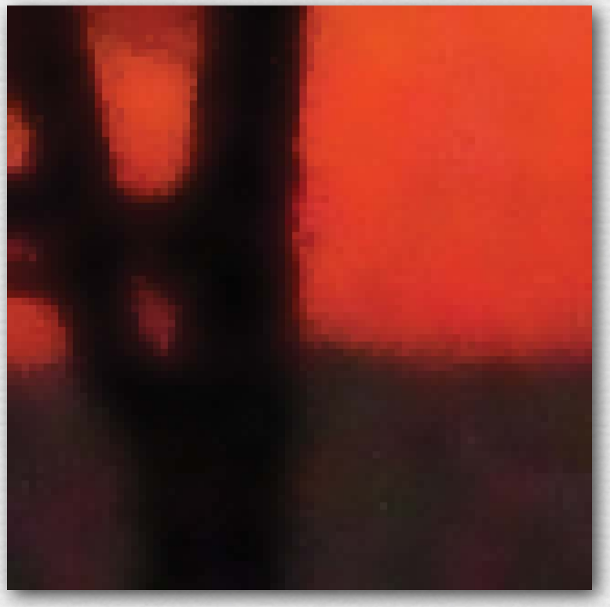


Gaussian blur, radius = 1.3



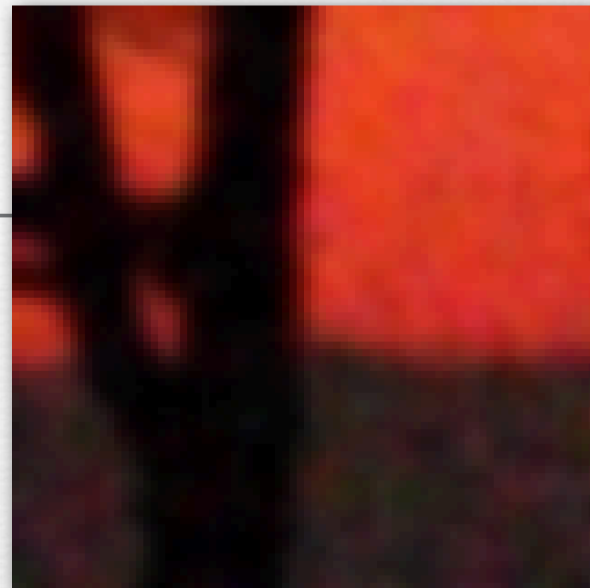
Canon denoising

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

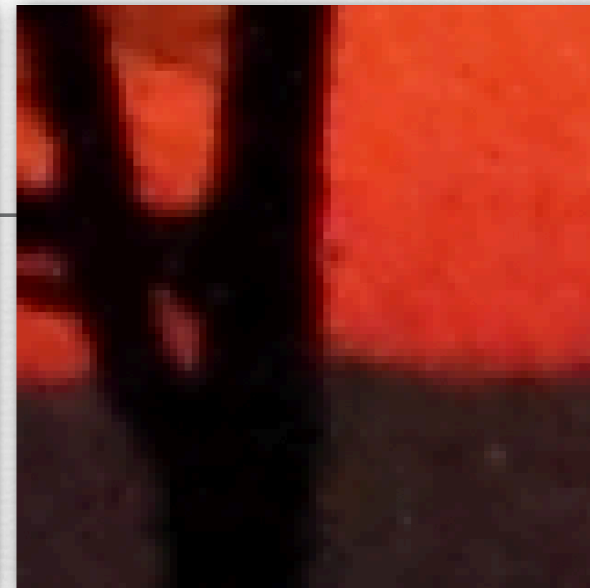


bilateral filtering

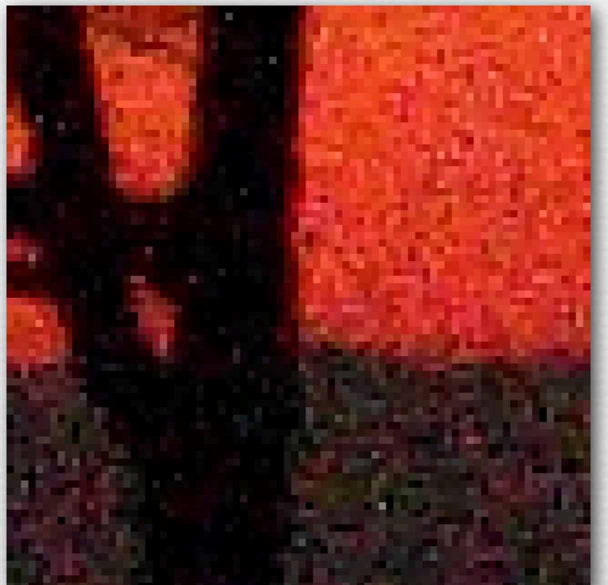
Denoising



Gaussian blur, radius = 1.3

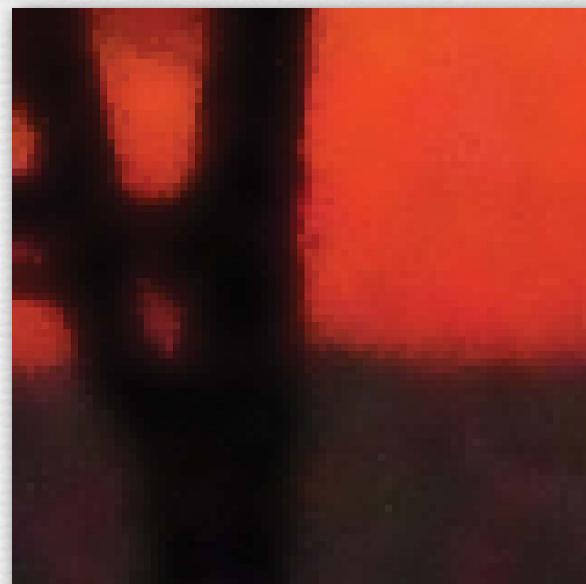


Canon denoising



RAW (ISO 6400)

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



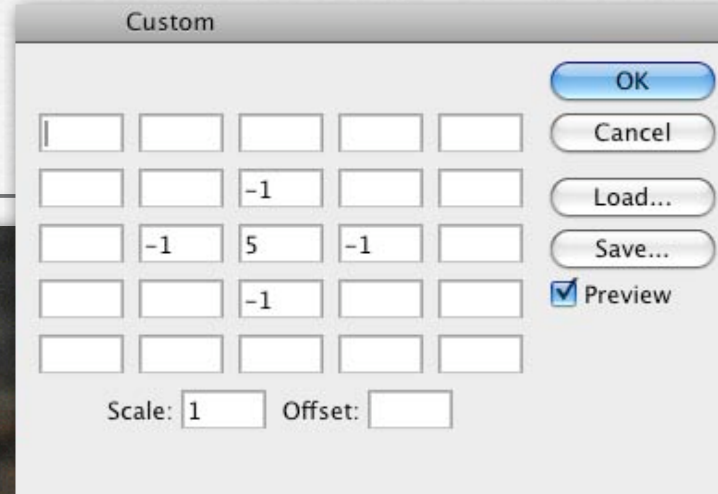
bilateral filtering

Sharpening

original

(Marc Levoy)

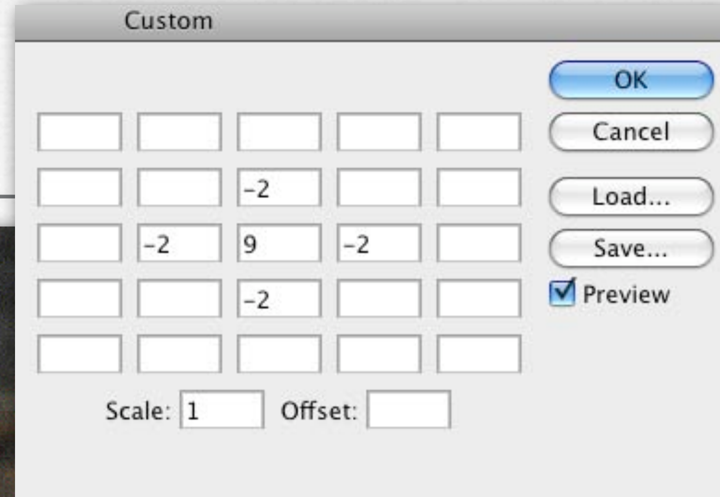
Sharpening



Filter/Other/Custom
in Photoshop CS4

(Marc Levoy)

Sharpening



Filter/Other/Custom
in Photoshop CS4

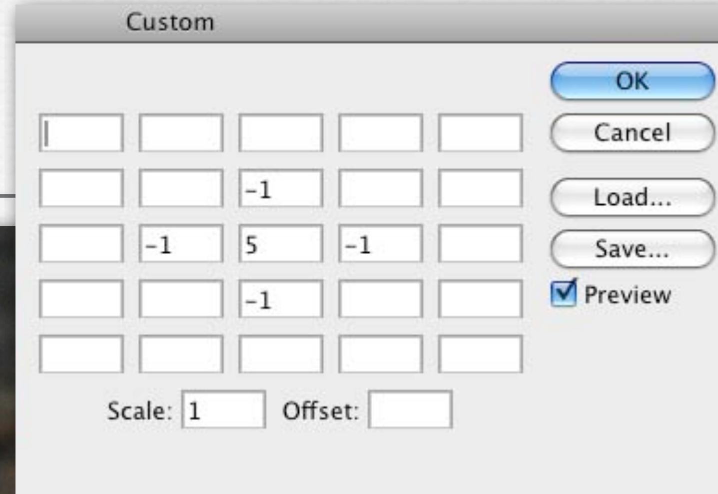
(Marc Levoy)

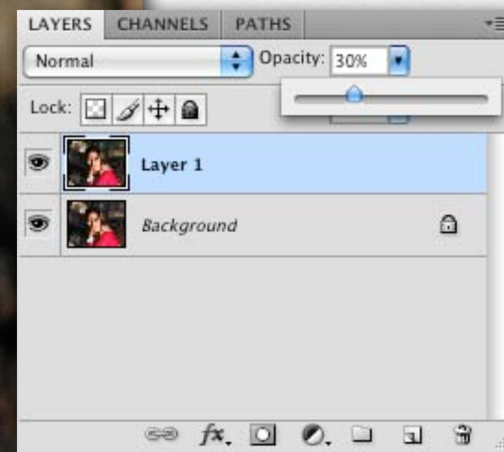
Sharpening

original

(Marc Levoy)

Sharpening

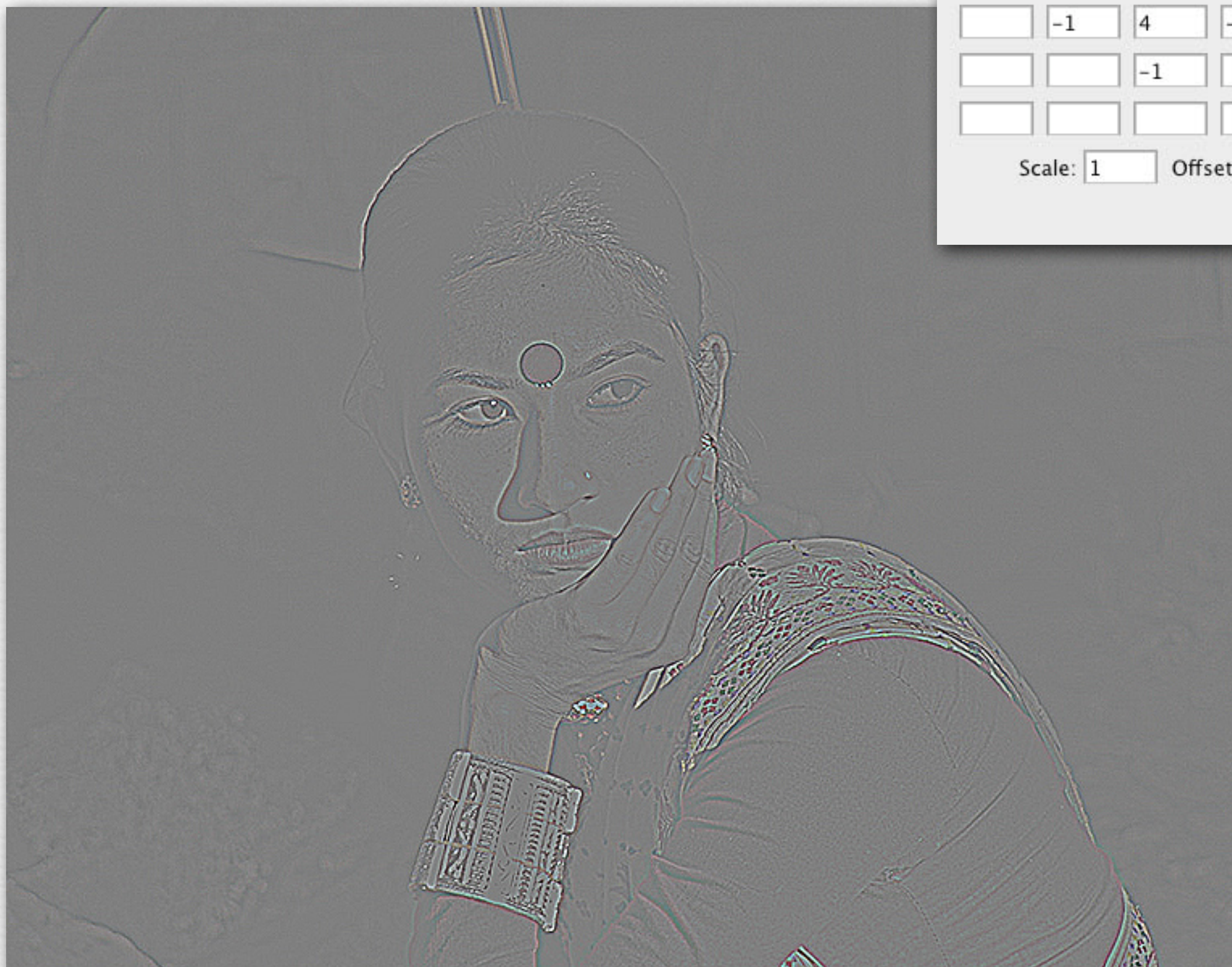


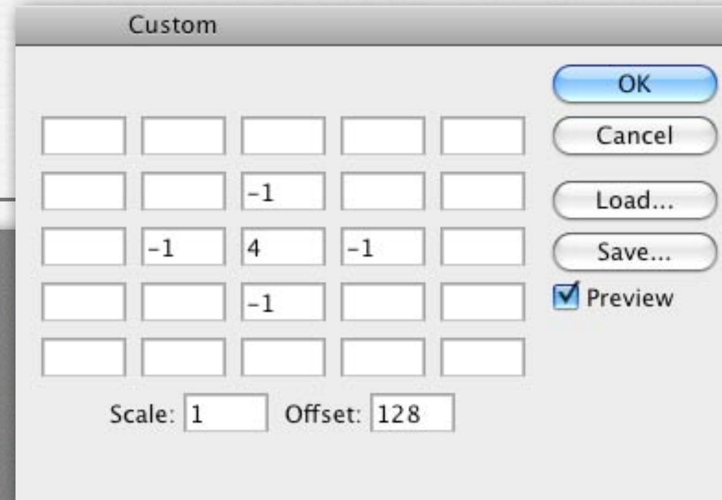


1st layer is original,
2nd layer is sharpened,
blend w. 30% opacity

(Marc Levoy)

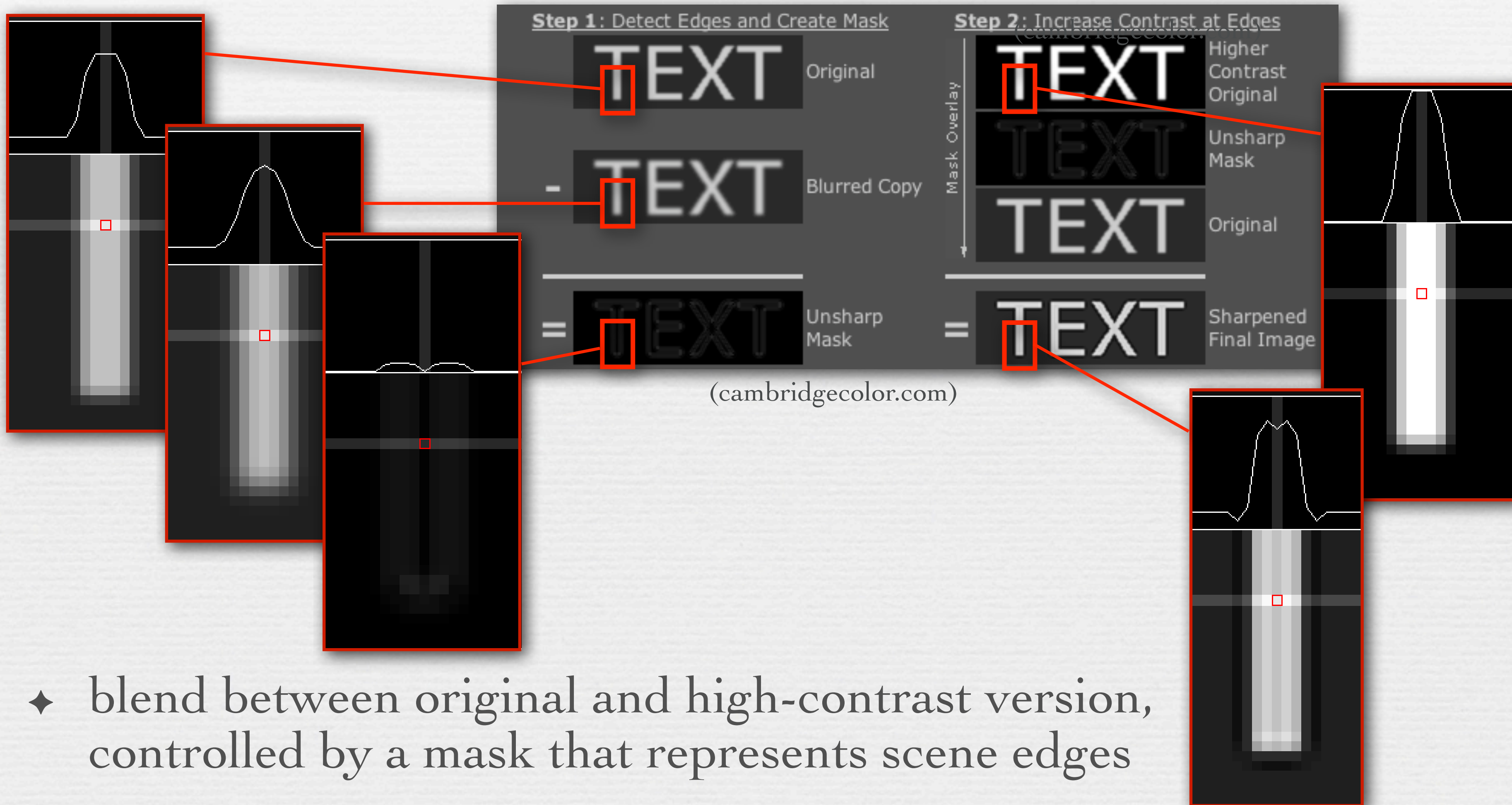
Sharpening





Filter/Other/Custom
in Photoshop CS4

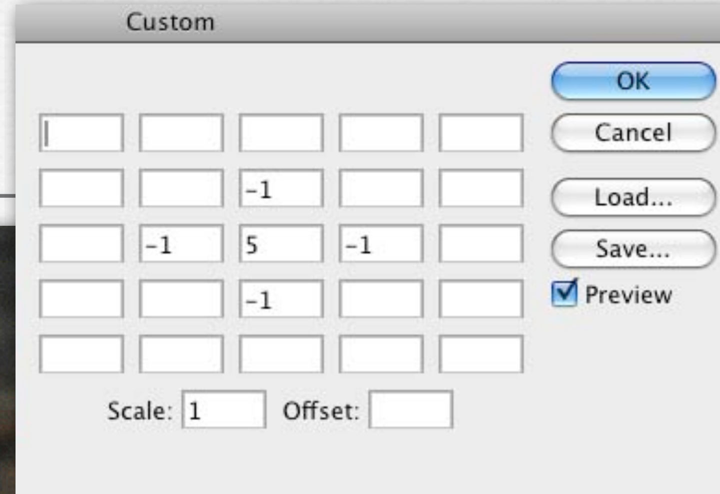
Unsharp masking



(cambridgecolor.com)

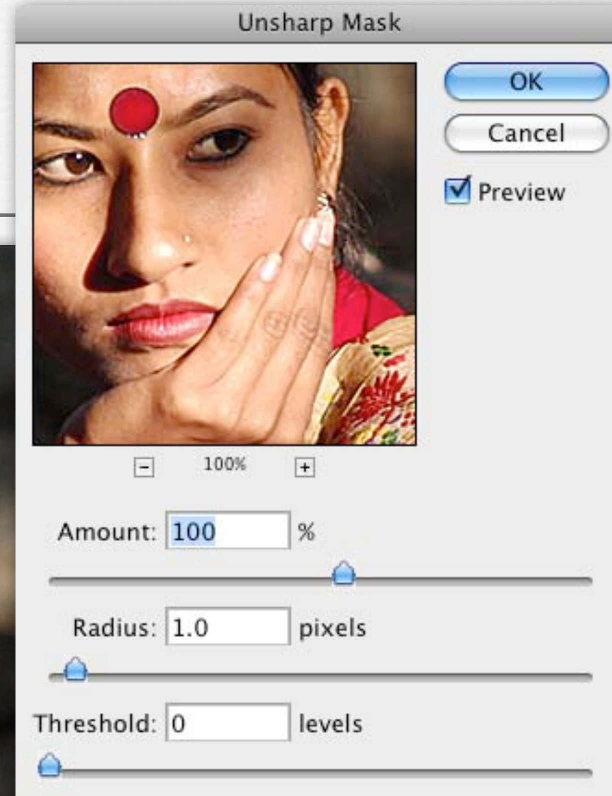
- blend between original and high-contrast version, controlled by a mask that represents scene edges
- dropping (thresholding) the darkest mask pixels avoids sharpening noise, and makes the filter non-linear

Sharpening



Filter/Other/Custom
in Photoshop CS4

Sharpening

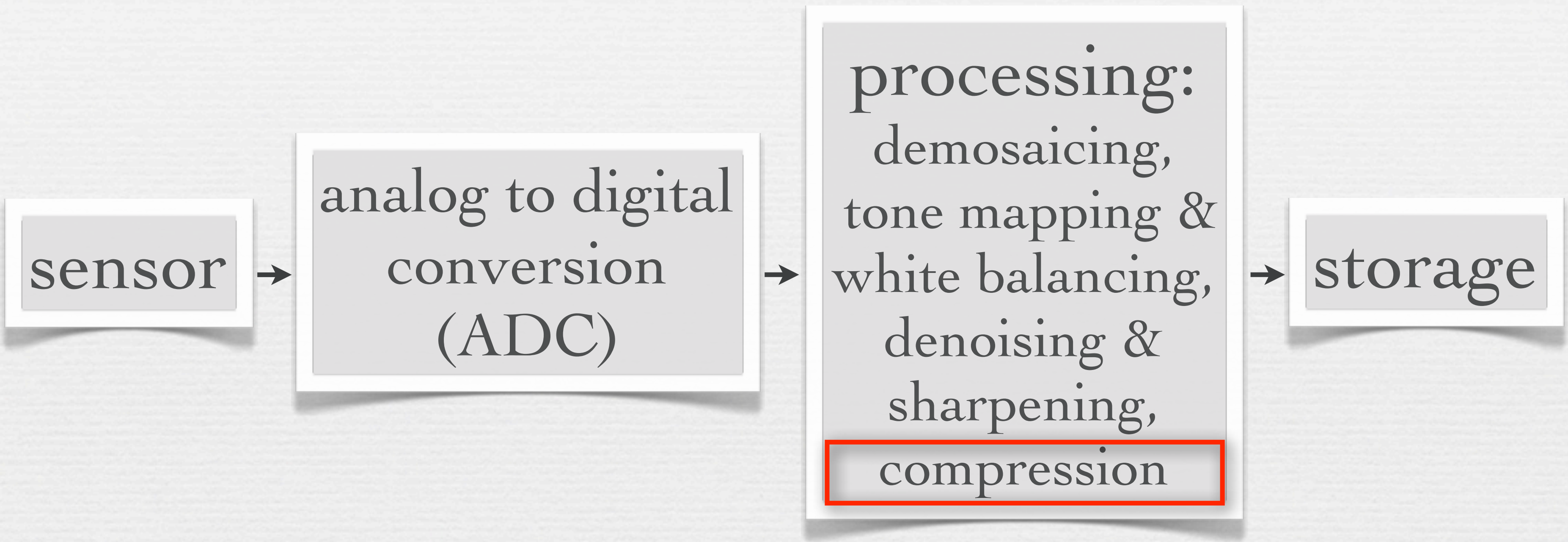


Filter/Sharpen/
Unsharp Mask in CS4

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

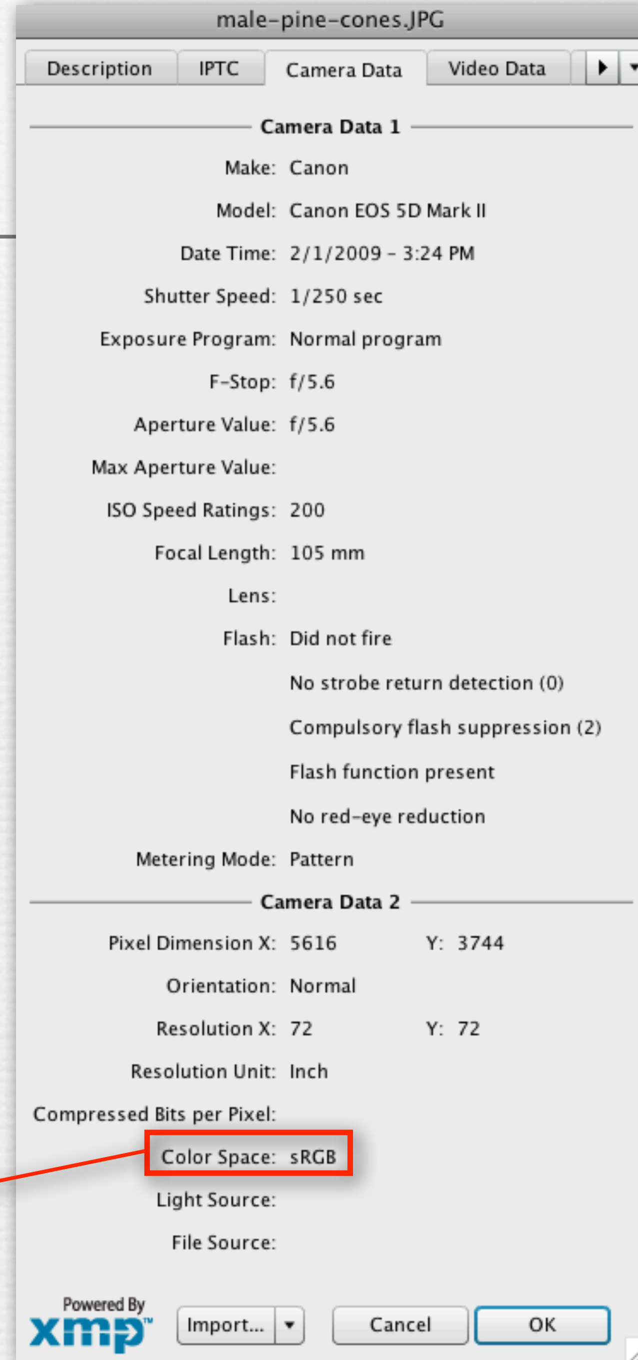
- Exchangeable Image File Format
  - created by Japan Electronic Industries Development Assoc.
- used by nearly every digital camera manufactured today
  - actually a file format
  - JPEG or TIFF file + metadata about the camera and shot
  - .JPG or .JPEG extension is used

EXIF data

File/File Info in Photoshop CS4

(Marc Levoy)

shot with Canon 5D Mark II



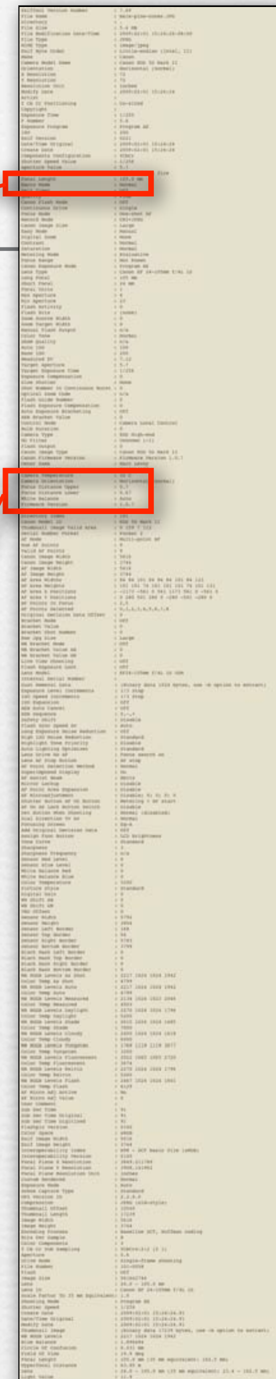
Color Space: sRGB

EXIF data

exiftool

Focal Length : 105.0 mm

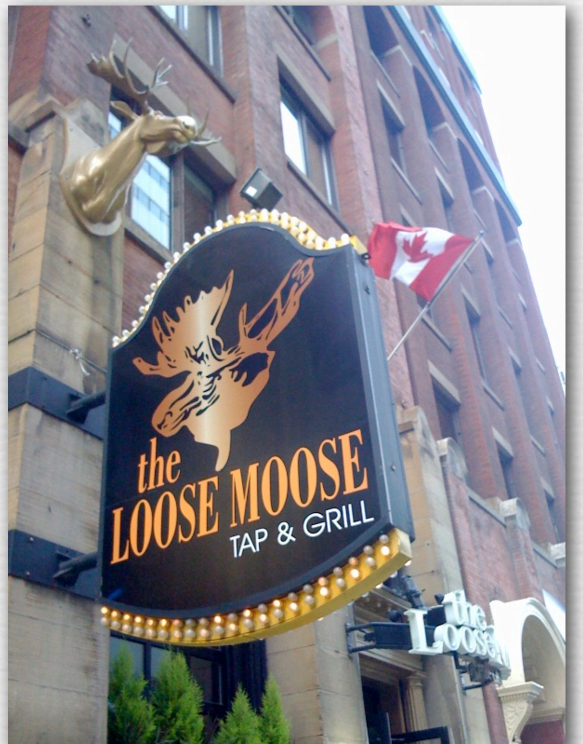
shot with Canon 5D Mark II



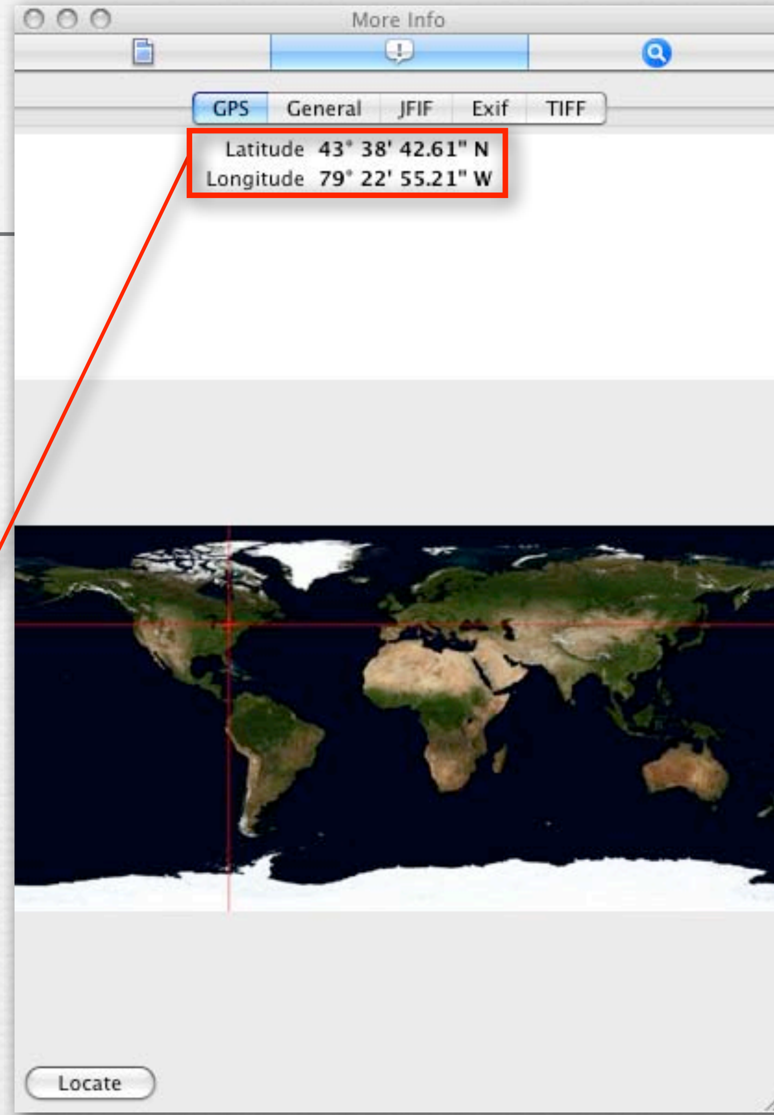
Focus Distance Upper : 0.7
Focus Distance Lower : 0.67

EXIF data

Mac
Preview



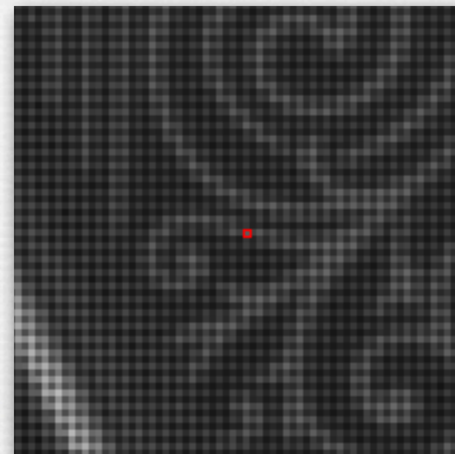
shot with iPhone 3G



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

RAW files

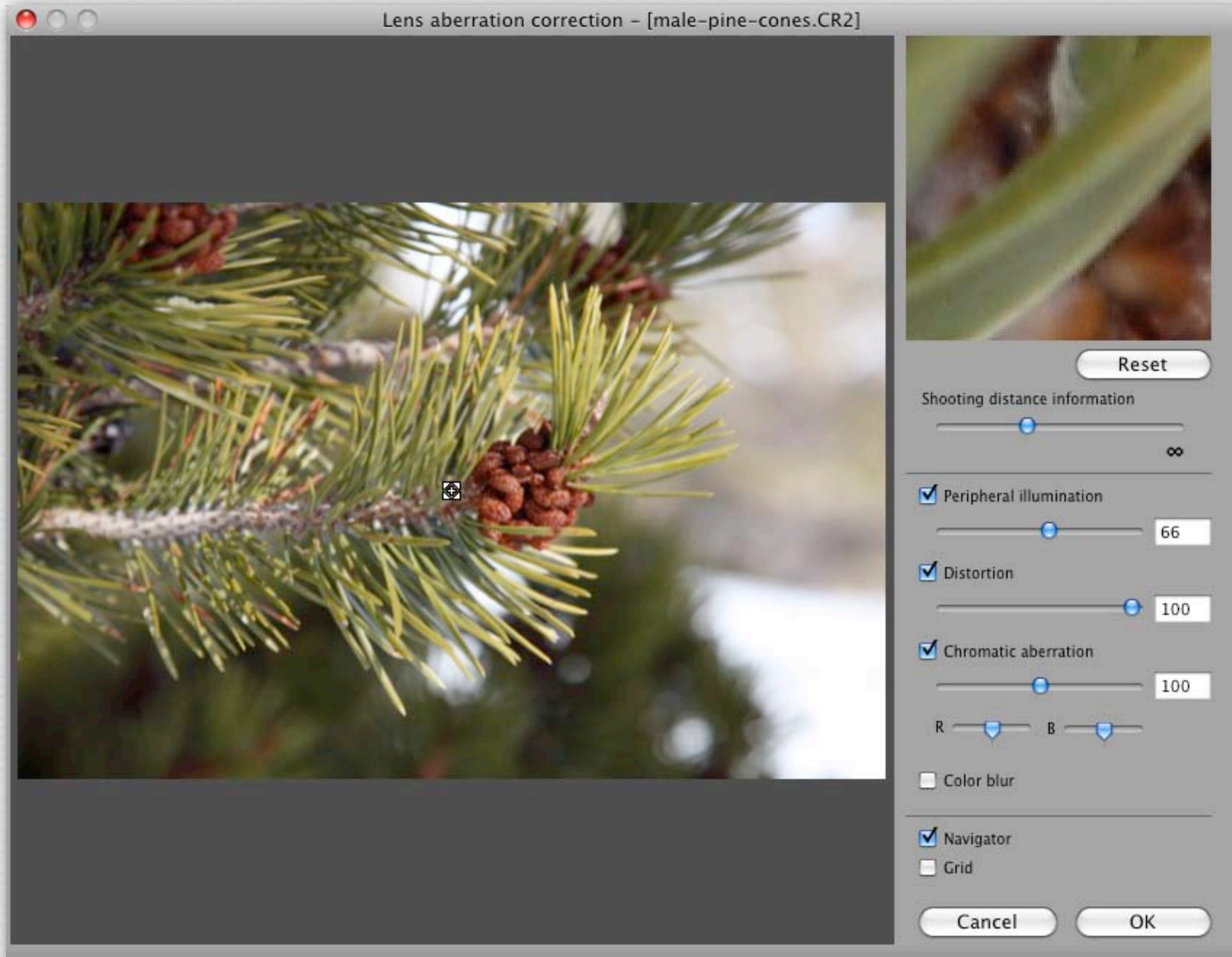
- minimally processed images, not even demosaiced
- uncompressed or losslessly compressed
- includes metadata, possibly encrypted
- file format varies by manufacturer
- example extensions: .CR2, .NEF, .RW2



- processed and converted to a JPEG file using
  - proprietary software (e.g. Canon Digital Photo Professional)
  - Photoshop or Lightroom (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



Slide credits

- Fredo Durand

- Stone, M., *A Field Guide to Digital Color*, A.K. Peters, 2003.
- Wandell, B., *Foundations of Vision*, Sinauer, 1995.
- Rudman, T., *Photographer's Master Printing Course*, Focal Press, 1998.
- Adams, A., *The Print*, Little, Brown and Co., 1980.