Noise and ISO

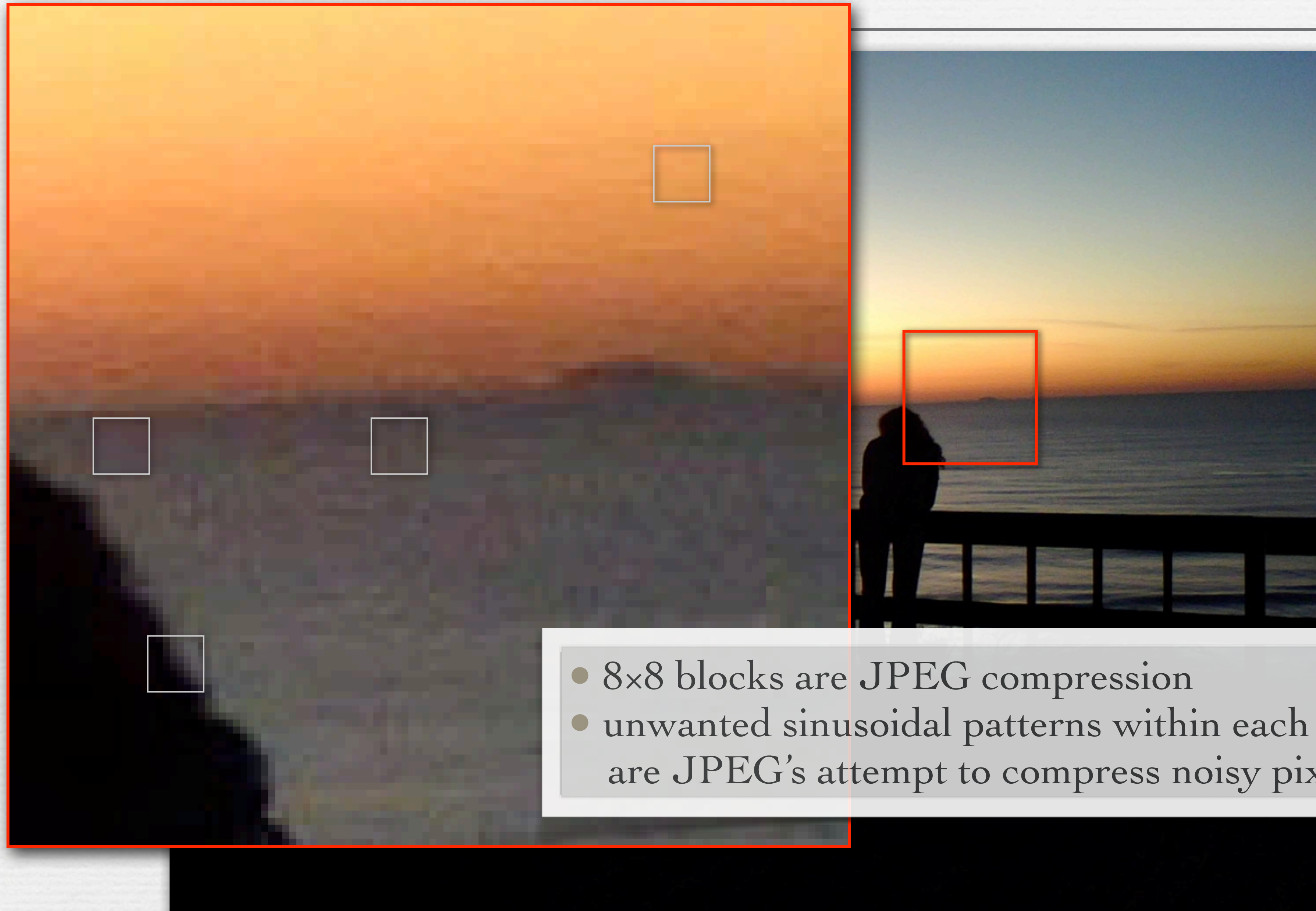
CS 178, Spring 2009

Marc Levoy
Computer Science Department
Stanford University

Outline

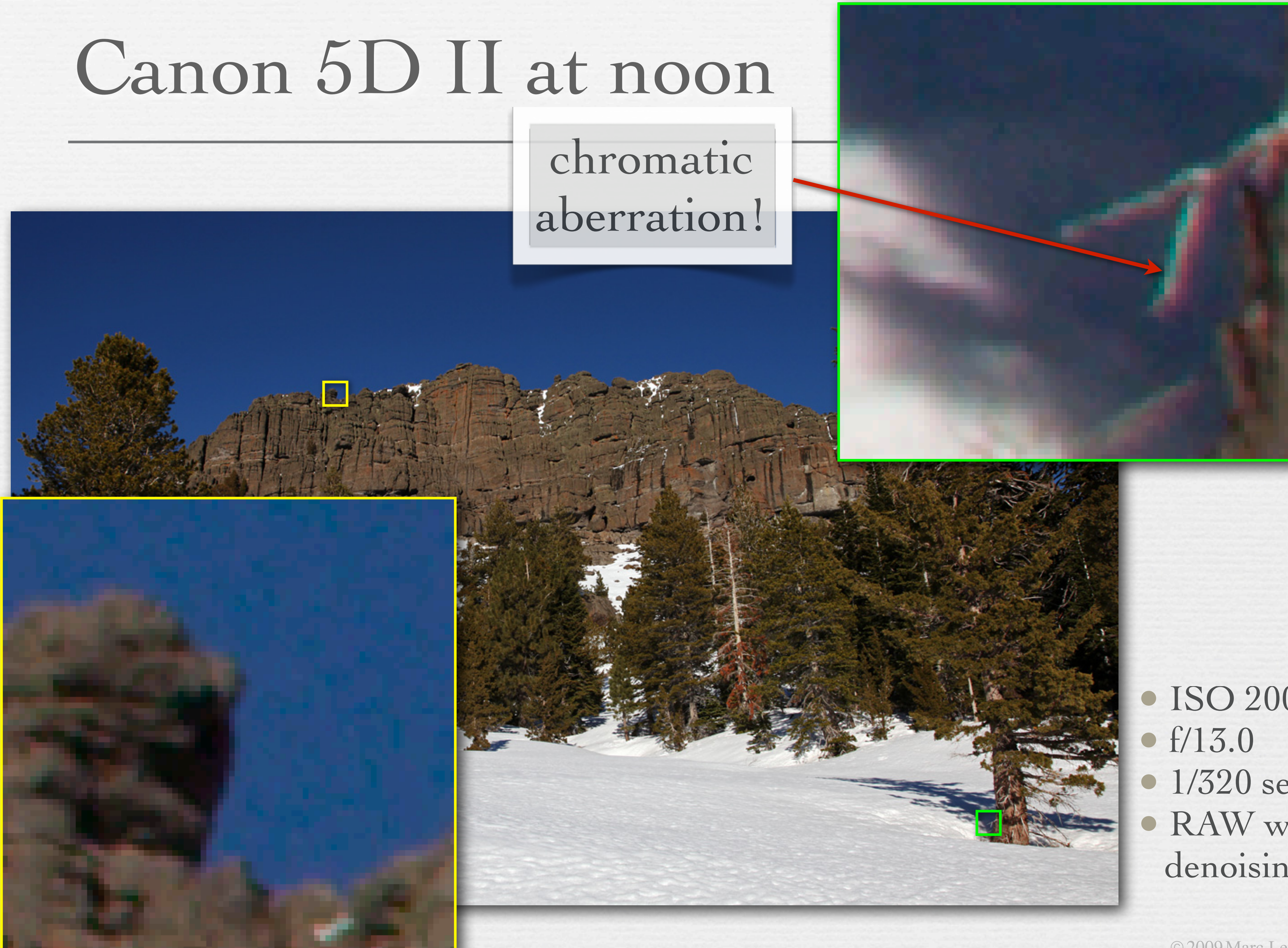
- examples of camera sensor noise
  - don't confuse it with JPEG compression artifacts
- probability, mean, variance, signal-to-noise ratio
- laundry list of noise sources
  - photon shot noise, dark current,
    hot pixels, fixed pattern noise, read noise
- SNR (again), quantization, dynamic range, bits per pixel
- ISO
- denoising
  - including aligning and averaging multiple shots
- night photography (Jesse Levinson)

Nokia N95 cell phone at dusk



- 8×8 blocks are JPEG compression
- unwanted sinusoidal patterns within each block are JPEG's attempt to compress noisy pixels

Canon 5D II at noon



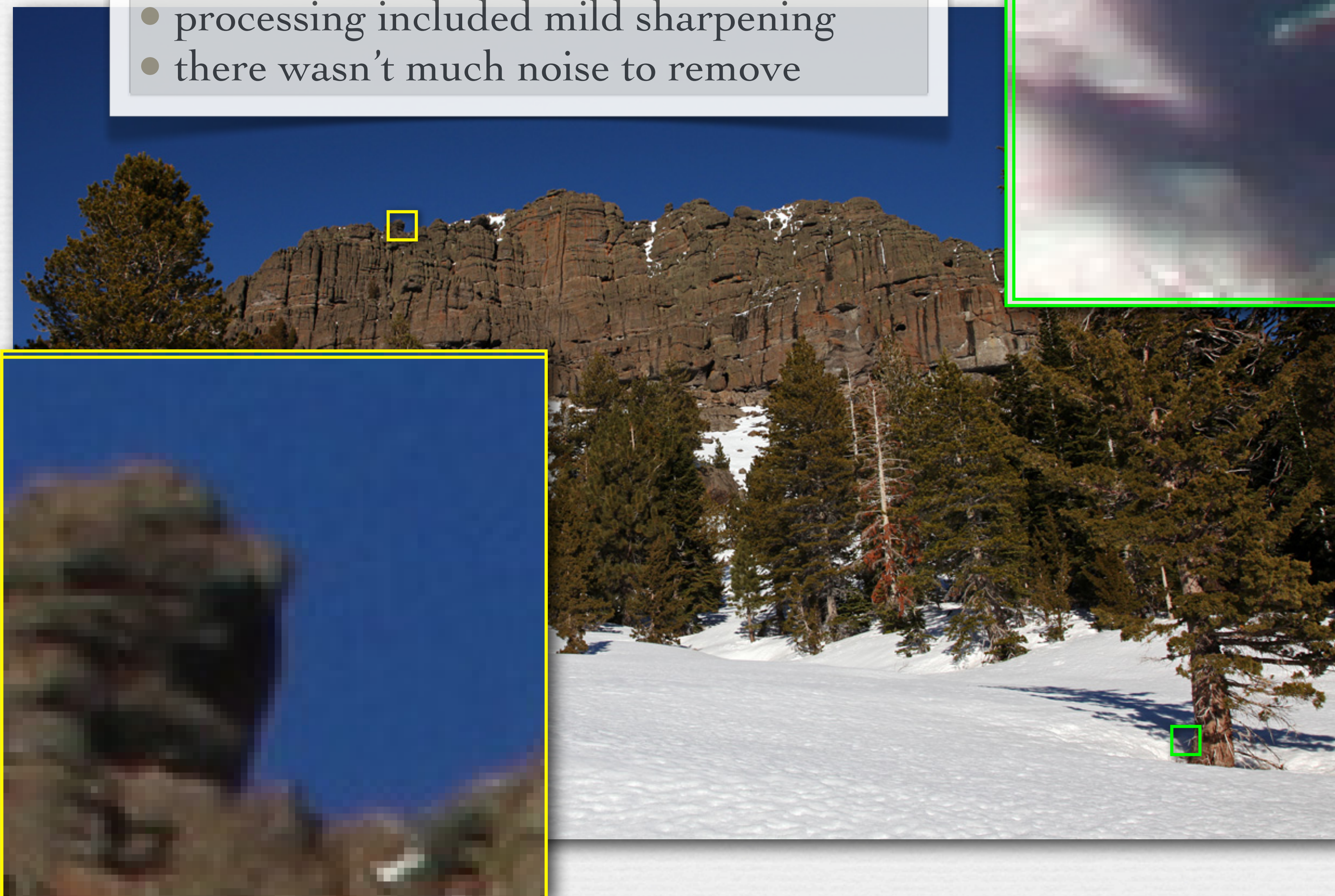
- ISO 200
- f/13.0
- 1/320 sec
- RAW w/o denoising

Canon 5D II at noon

- ISO 200
- f/13.0
- 1/320 sec
- Canon processed

Canon 5D II at noon

- processing included mild sharpening
- there wasn't much noise to remove

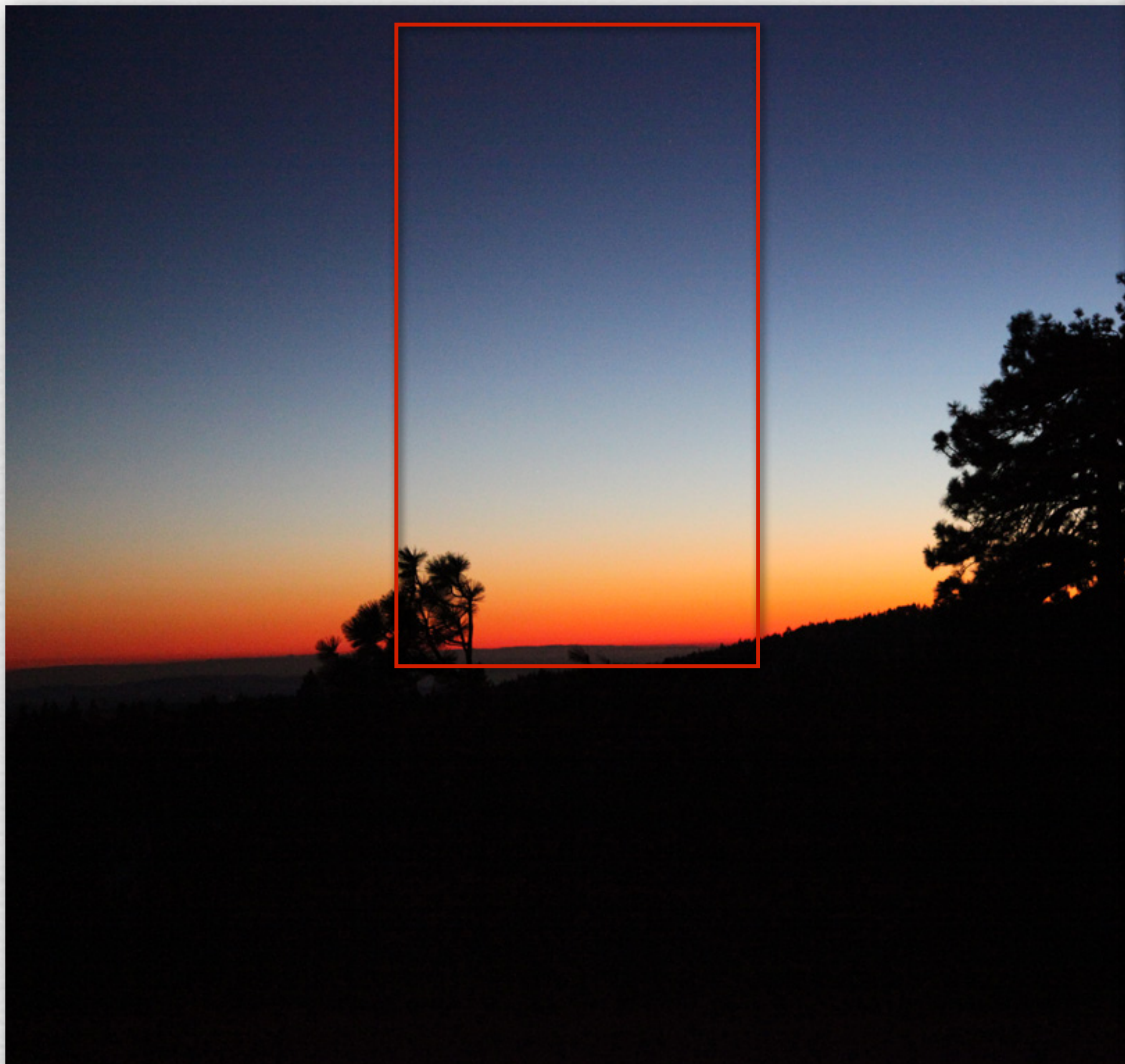


- ISO 200
- f/13.0
- 1/320 sec
- Canon processed

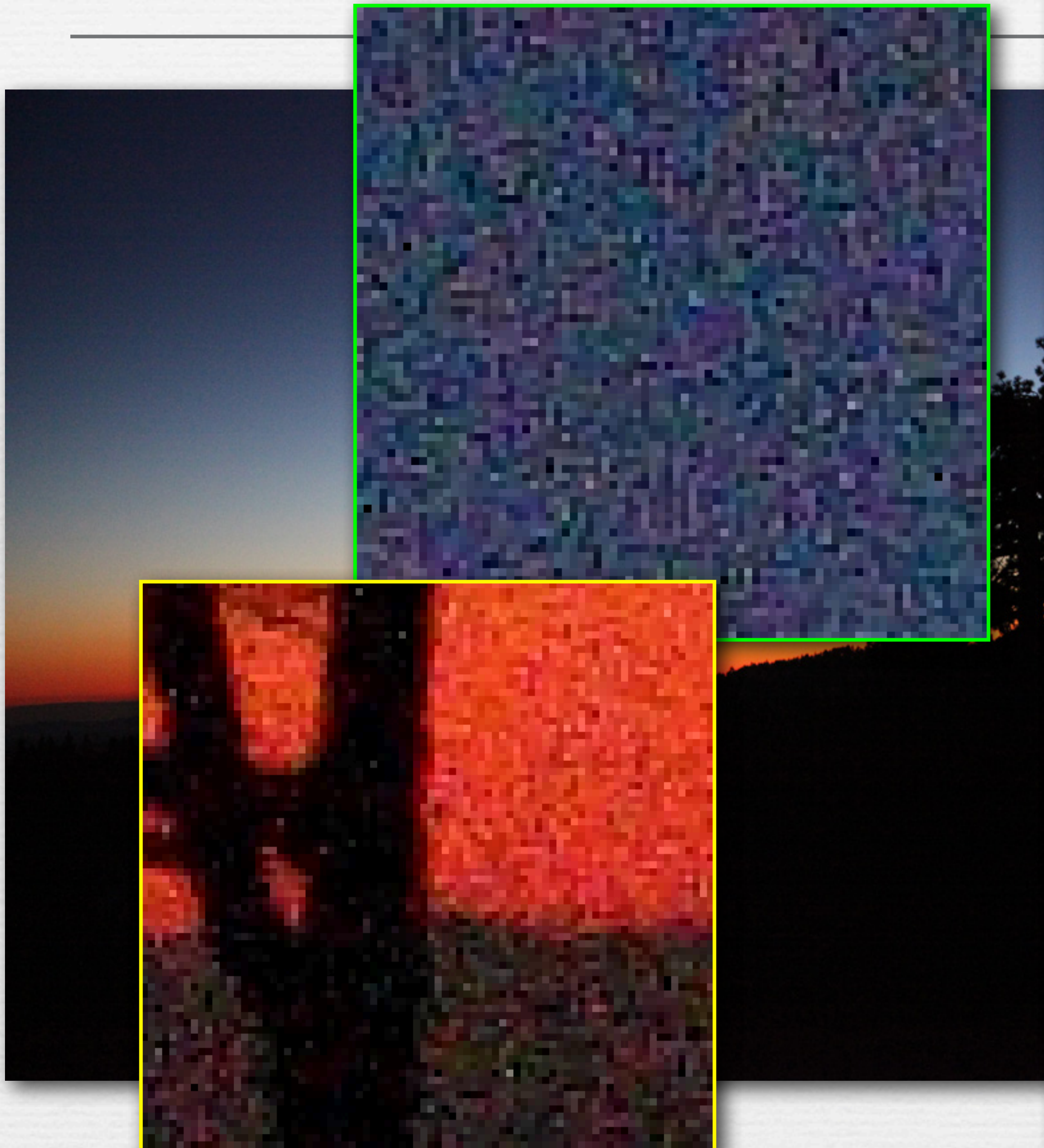
Canon 5D II at dusk

- ISO 6400
- f/4.0
- 1/13 sec
- RAW w/o denoising

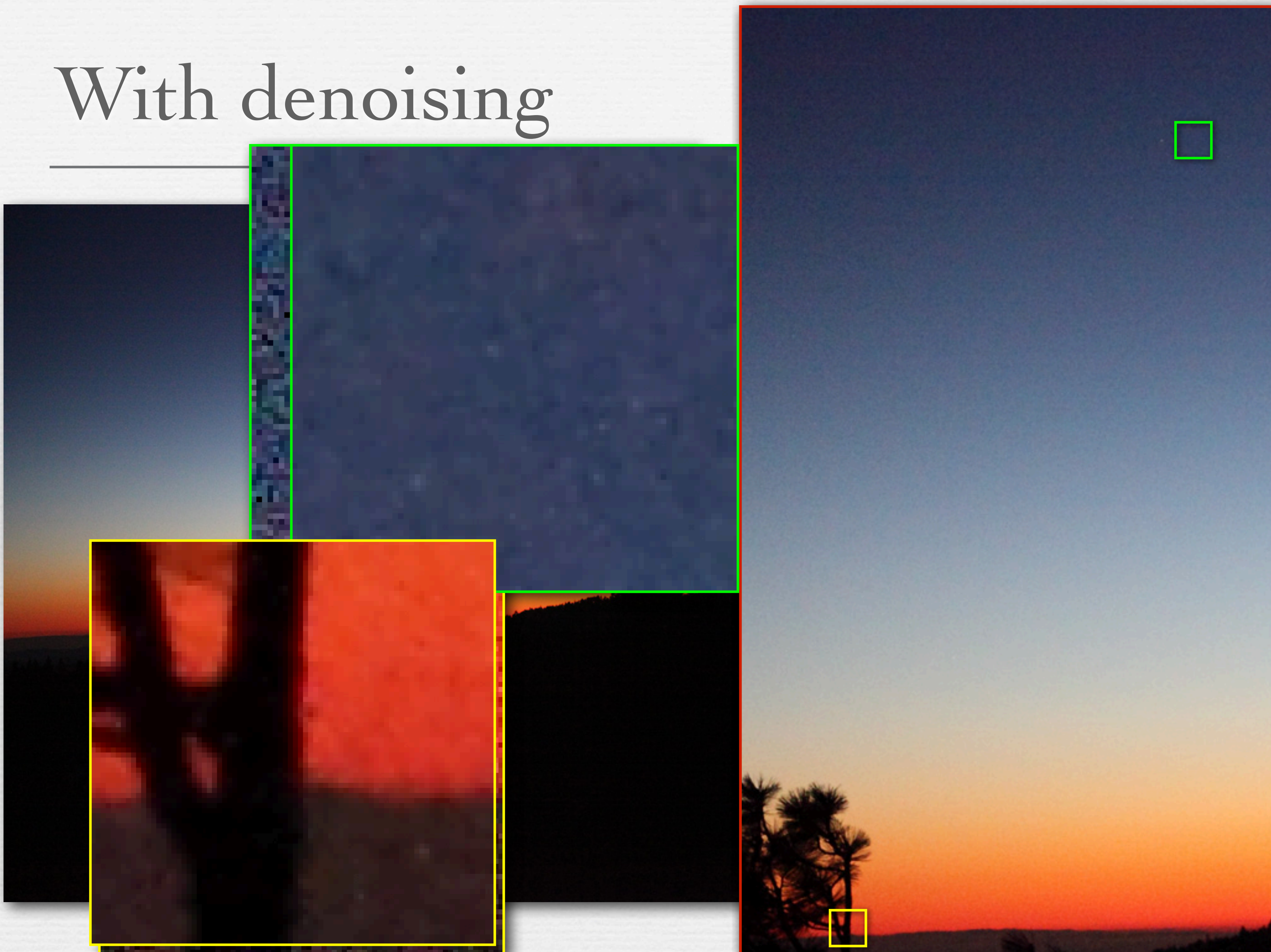
Canon 5D II at dusk



Canon 5D II at dusk



With denoising



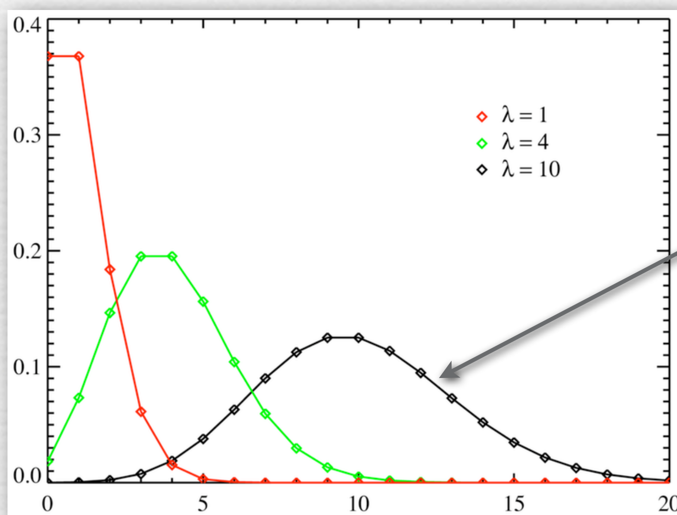
Photon shot noise

- the number of photons arriving during an exposure varies from exposure to exposure and from pixel to pixel
- this number is governed by the Poisson distribution

Poisson distribution

- expresses the probability that a certain number of events will occur during an interval of time
- applicable to rare events that occur
  - with a known average rate, and
  - independently of the time since the last event
- if on average λ events occur in an interval of time, the probability *p* that *k* events occur instead is

$$p(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$



probability density function

Mean and variance

- the mean of a probability density function is

$$\mu = \int x\, p(x)\, dx$$

- the variance of a probability density function is

$$\sigma^2 = \int (x-\mu)^2\, p(x)\, dx$$

- the mean and variance of the Poisson distribution are

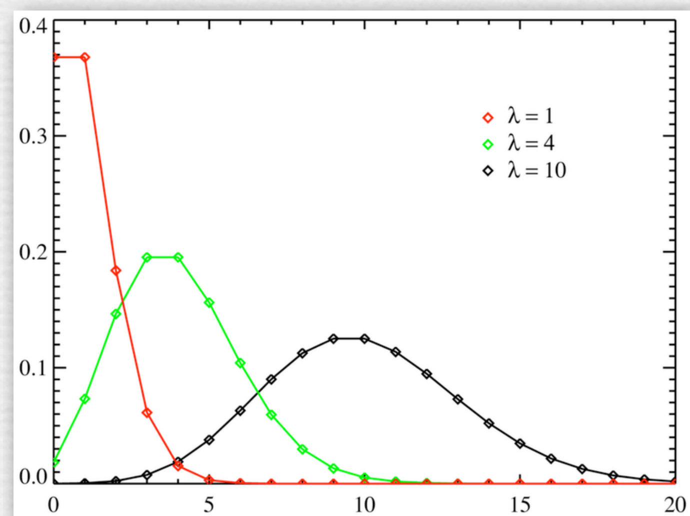
$$\mu = \lambda$$

$$\sigma^2 = \lambda$$

- the standard deviation is

$$\sigma = \sqrt{\lambda}$$

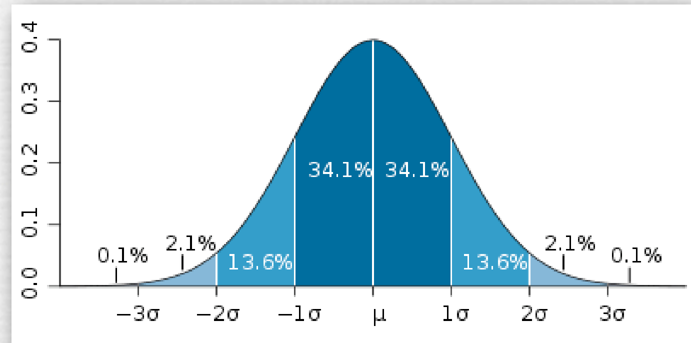
Deviation grows slower than the average.



Poisson versus Normal distribution

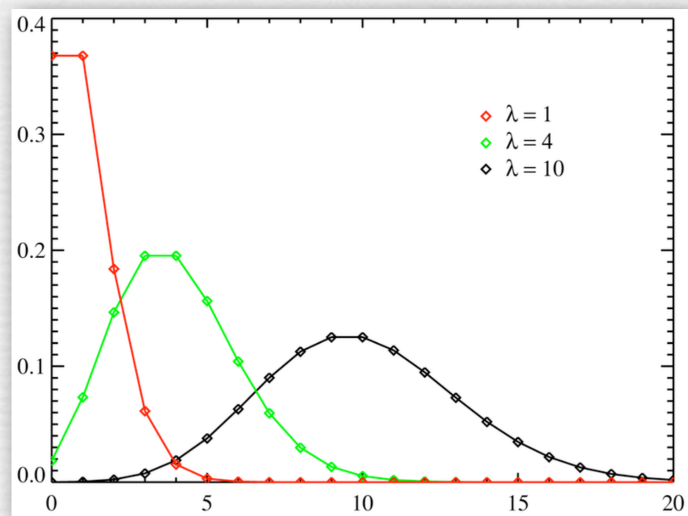
- for large λ (>1000), the Poisson distribution is well approximated by a Gaussian (i.e. Normal) distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



- in a Normal distribution, ~68% of the area falls within one standard deviation of the mean

- for a Poisson distribution with λ = 10
  - on average 10 events will occur during the interval of time
  - there is a 68% chance the number of events will fall in the range 10 ± √10



Signal-to-noise ratio (SNR)

$$SNR = \frac{\text{mean pixel value}}{\text{standard deviation of pixel value}} = \frac{\mu}{\sigma}$$

$$SNR\ (\text{dB}) = 20 \log_{10}\left(\frac{\mu}{\sigma}\right)$$

- example
  - if SNR improves from 100:1 to 200:1,
    it improves $20 \log_{10}(200) - 20 \log_{10}(100) = +6$ dB

Photon shot noise, 2^{nd} try

- photons arrive in a Poisson distribution

$$\mu = \lambda$$
$$\sigma = \sqrt{\lambda}$$

- so

$$SNR = \frac{\mu}{\sigma} = \sqrt{\lambda}$$

- shot noise scales as square root of number of photons
- examples
  - doubling the width and height of a pixel increases its area by 4×, hence # of photons by 4×, hence SNR by 2× or +6 dB
  - opening the aperture by 1 f/stop increases the # of photons by 2×, hence SNR by √2 or +3 dB

Empirical example

- Kodak Q14 test chart

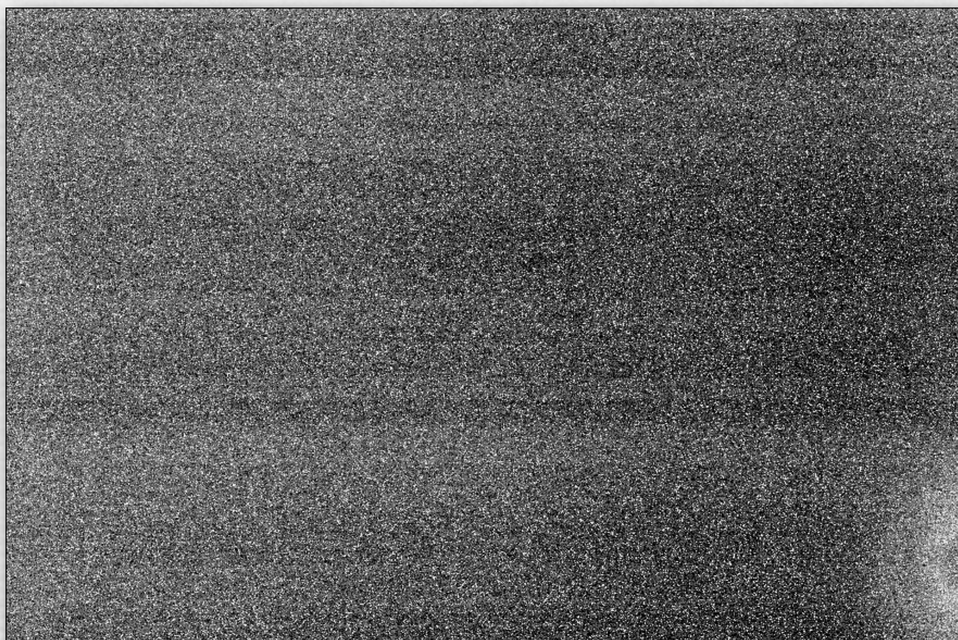
- Canon 10D, ISO 1600, crop from recorded image

- SNR improves with increasing signal

(http://www.imatest.com/docs/noise.html)

Dark current

- electrons dislodged by random thermal activity
- increases linearly with exposure time
- increases exponentially with temperature
- varies across sensor, and includes its own shot noise

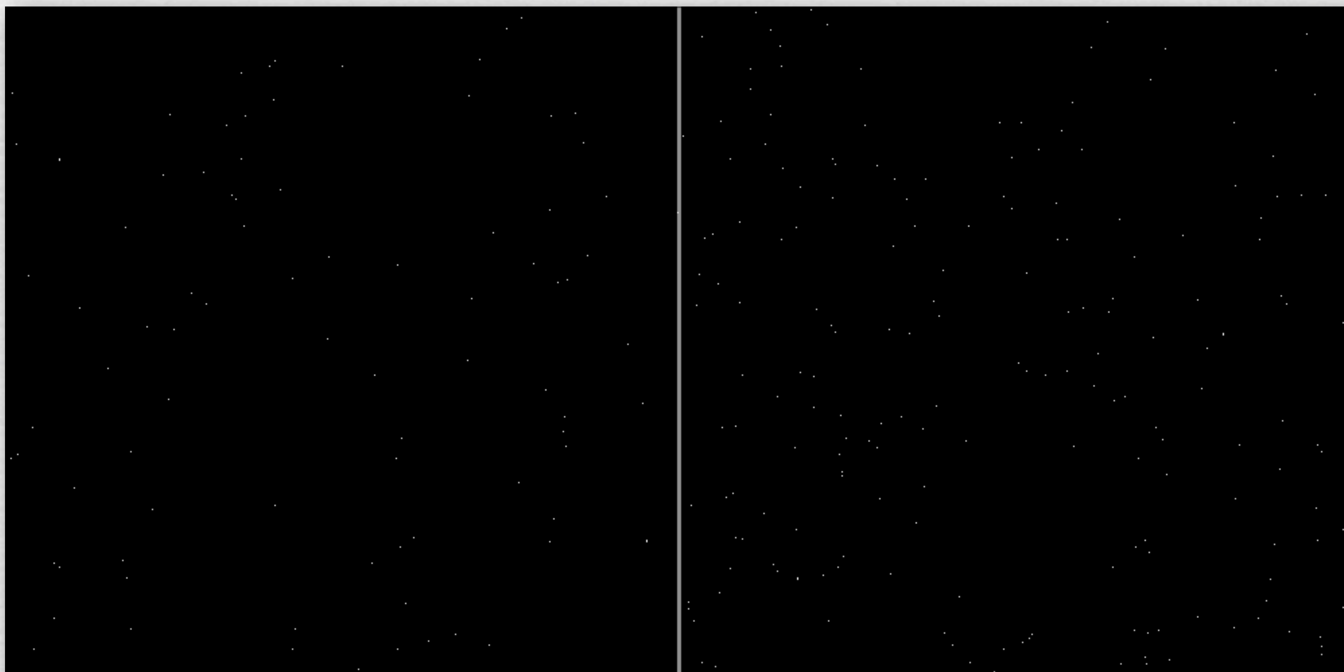


(http://theory.uchicago.edu/~ejm/pix/20d/tests/noise/)

Canon 20D, 612 sec exposure

Hot pixels

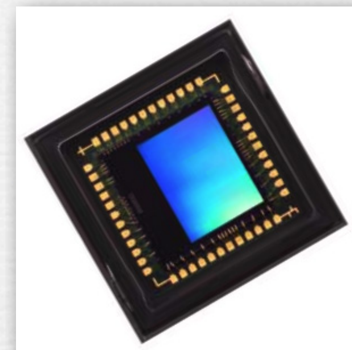
- electrons leaking into well due to manufacturing defects
- increases linearly with exposure time
- increases with temperature, but hard to model
- changes over time, and every camera has them



Canon 20D, 15 sec and 30 sec exposures

Fixing dark current and hot pixels

- example
  - Aptina MT9P031 (in Nokia N95 cell phone)
  - full well capacity = ~8500 electrons
  - dark current = 25 electrons/pix/sec at 55°C

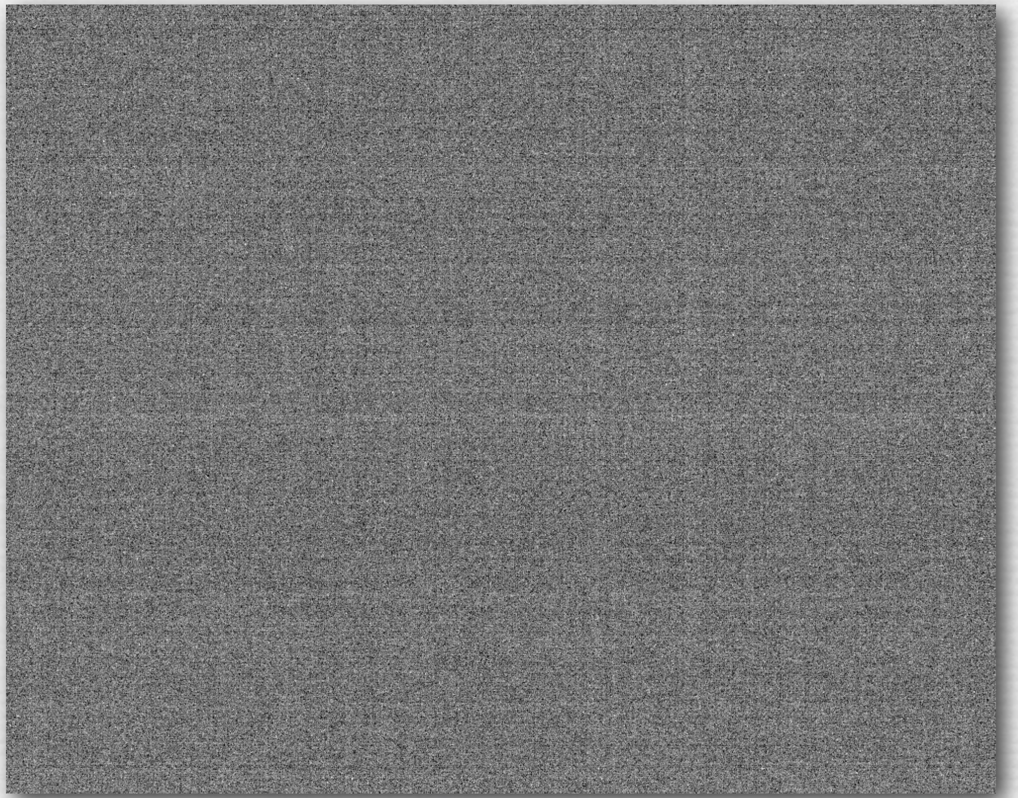


- solution #1: chill the sensor
  - Retiga 4000R bioimaging camera
  - Peltier cooled 25°C below ambient
  - full well capacity = 40,000 electrons
  - dark current = 1.64 electrons/pix/sec

- solution #2: dark frame subtraction
  - available on high-end SLRs

Fixed pattern noise (FPN)

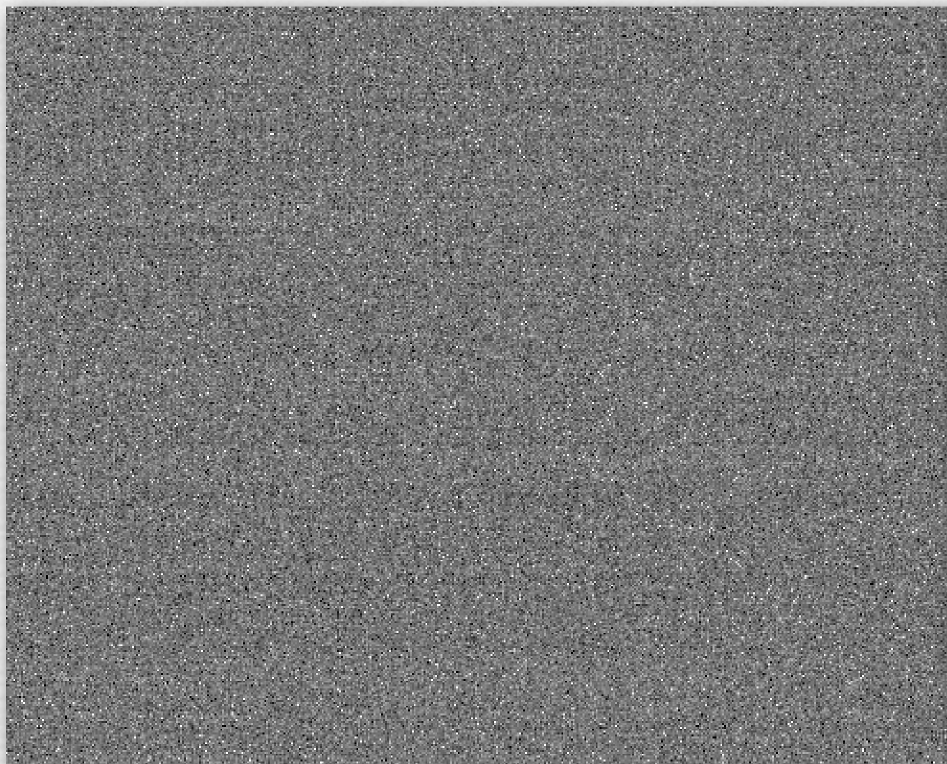
- manufacturing variations across pixels, columns, blocks
- mainly in CMOS sensors
- doesn't change over time, so read once and subtract



Canon 20D, ISO 800, cropped

Read noise

- thermal noise in readout circuitry
- again, mainly in CMOS sensors
- not fixed pattern, so only solution is cooling



Canon 1Ds Mark III, cropped

Signal-to-noise ratio, 2nd try

$$SNR = \frac{\text{mean pixel value}}{\text{standard deviation of pixel value}} = \frac{\mu}{\sigma}$$

$$= \frac{P\,Q_e\,t}{\sqrt{P\,Q_e\,t + D\,t + N_r^2}}$$

- where

P = incident photon flux (photons/pixel/sec)

Q_e = quantum efficiency

t = exposure time (sec)

D = dark current (electrons/pixel/sec), including hot pixels

N_r = read noise (rms electrons/pixel), including fixed pattern noise

(formula from http://learn.hamamatsu.com/articles/ccdsnr.html)

Signal-to-noise ratio, 2nd try

$$SNR = \frac{\text{mean pixel value}}{\text{standard deviation of pixel value}}$$

$$= \frac{P\,Q_e\,t}{\sqrt{P\,Q_e\,t + D\,t + N_r^2}}$$

- examples
  - Retiga 4000R = (1000 × 55%) / √(1000 × 55% + 1.64 + 12^2) = 20.8:1 assuming 1000 photons/pixel/sec for 1 second
  - Aptina MT9P031 = (1000÷11×69%) / √(1000÷11×69% + 25 + 2.6^2) = 6.5:1 assuming pixels are 1/11 as large as Retiga's

Dynamic range

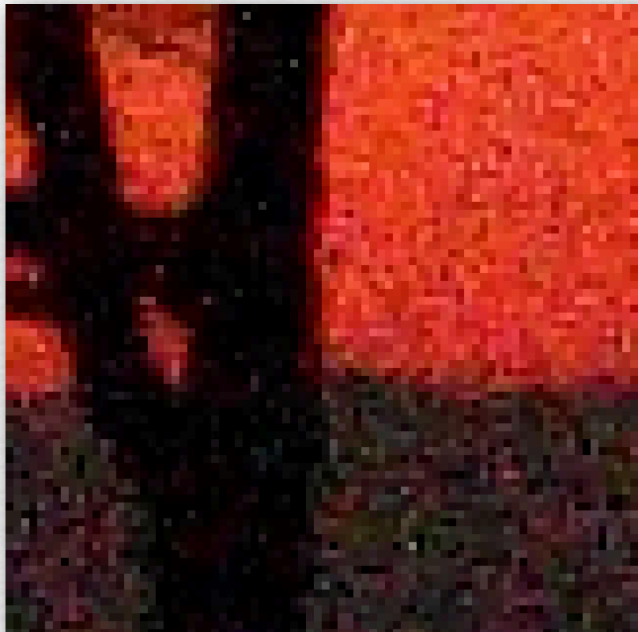
$$DR = \frac{\text{max output swing}}{\text{noise in the dark}} = \frac{\text{saturation level - } D\,t}{\sqrt{D\,t + N_r^2}}$$

- examples
  - Retiga 4000R = (40,000 - 1.64) / √(1.64 + 12^2) electrons
    = 3,313:1 (11.7 bits) for a 1 second exposure, and
    = 3,333:1 (11.7 bits) for a 1/60 second exposure
  - Aptina MT9P031 = (8500 - 25) / √(25 + 2.6^2)
    = 1500:1 (10.5 bits) for a 1 second exposure, but
    = 3200:1 (11.6 bits) for a 1/60 second exposure
- determines useful ADC precision
- after gamma correction (for JPEG), you only see ~8 bits

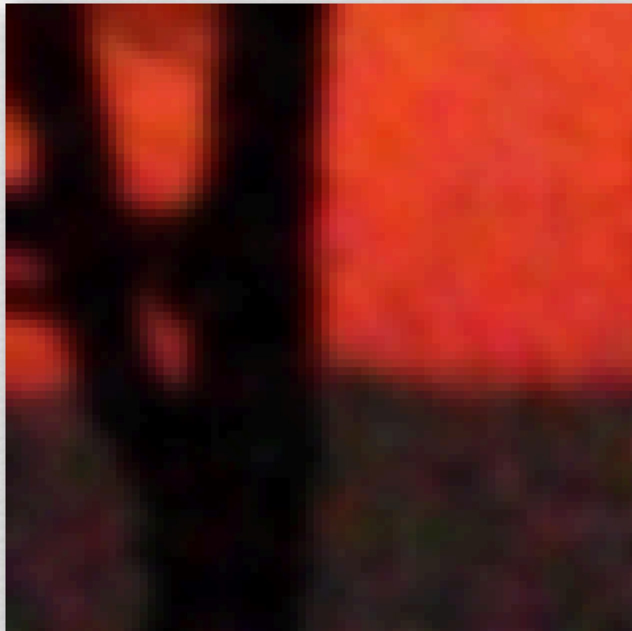
ISO

- amplifies signal before analog-to-digital conversion
  - avoids losing low signal due to quantization and any noise introduced after quantization (yes, there is some)
  - doubling ISO doubles the signal, which is linear with light, so equivalent to doubling exposure time, or minus 1 f/stop
- maximum ISO on Canon is 6400
  - higher ISOs implemented using multiplication after ADC?
- raising ISO improves SNR relative to multiplication after ADC, or equivalently, brightening in Photoshop
- but raising exposure time improves SNR faster, so
- maximize exposure time to the limits imposed by object motion, camera shake, or sensor saturation, then maximize ISO to the limit imposed by ADC saturation

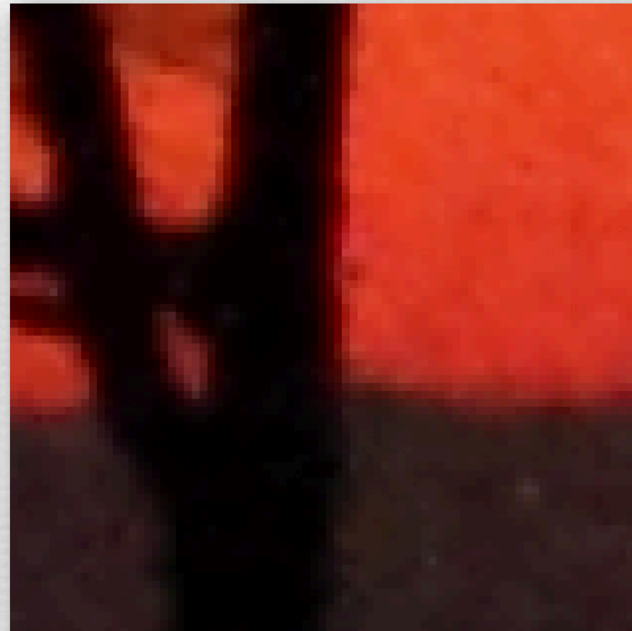
Denoising



RAW (ISO 6400)



Gaussian blur, radius = 1.3

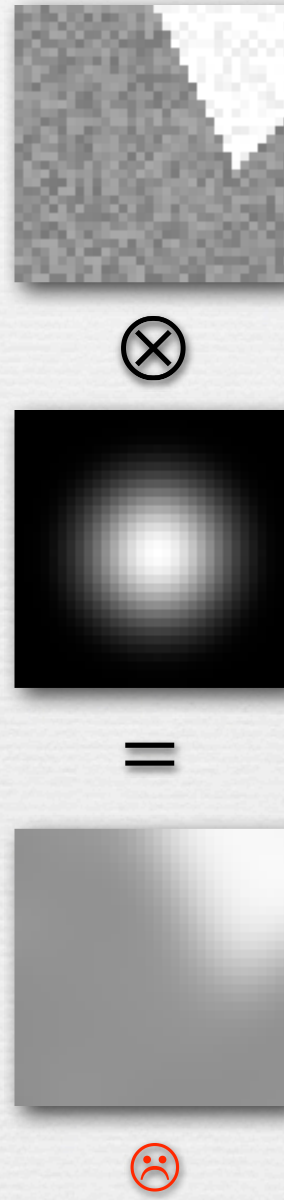


Canon denoising

- 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
- bluring 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

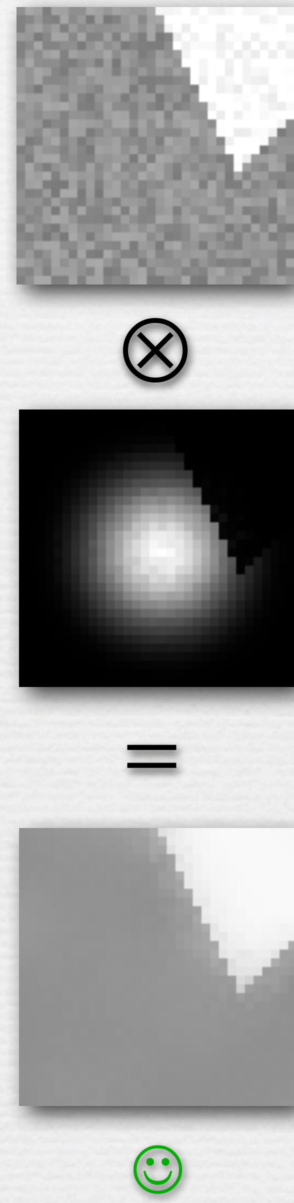


Bilateral filtering [Tomasi ICCV 1998]

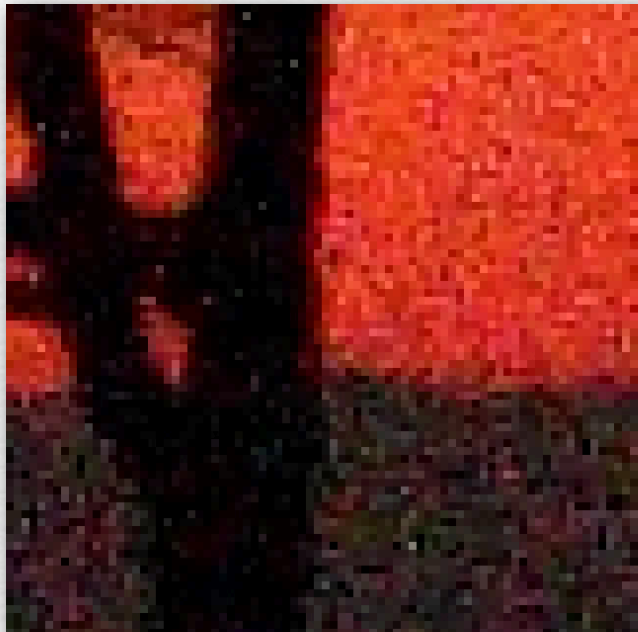
- if the pixels are similar in intensity, the probability they are from the same piece of the scene is high

- 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

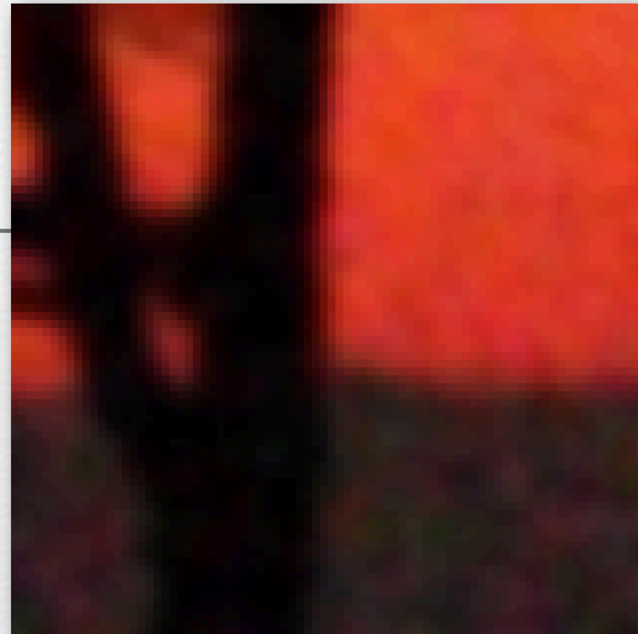
- i.e. blur in the *domain* and *range* dimensions!



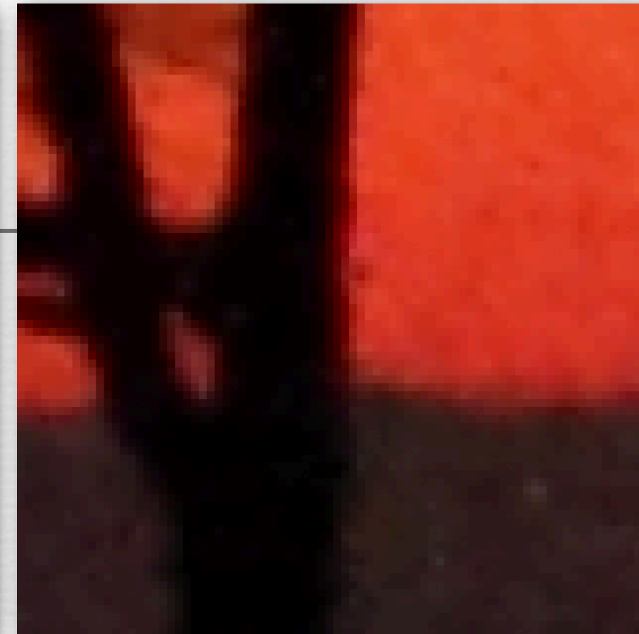
Denoising



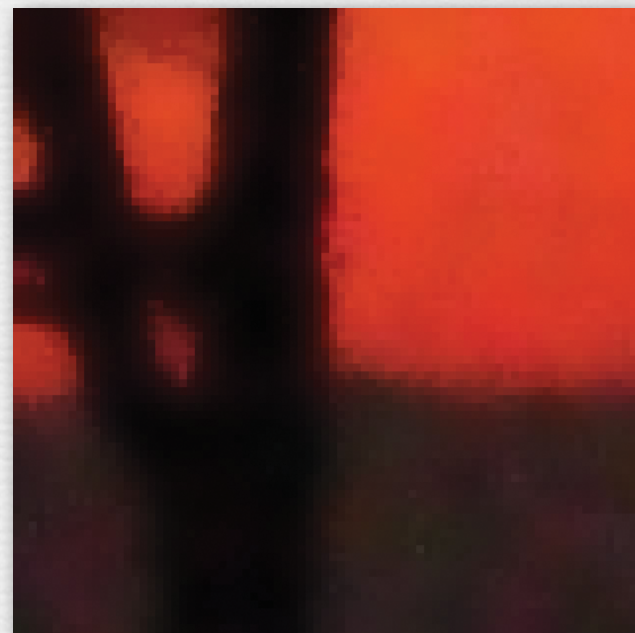
RAW (ISO 6400)



Gaussian blur, radius = 1.3



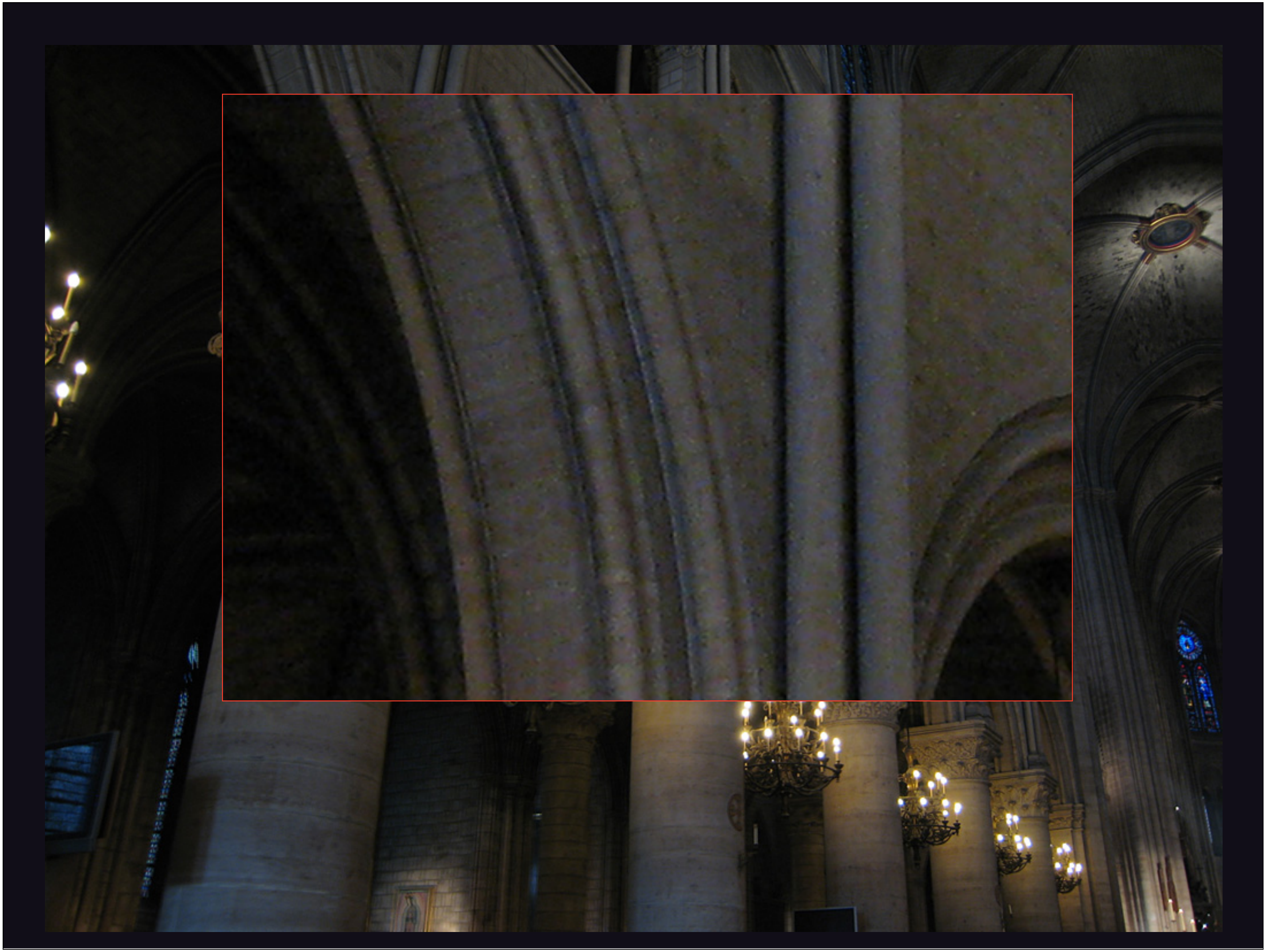
Canon denoising

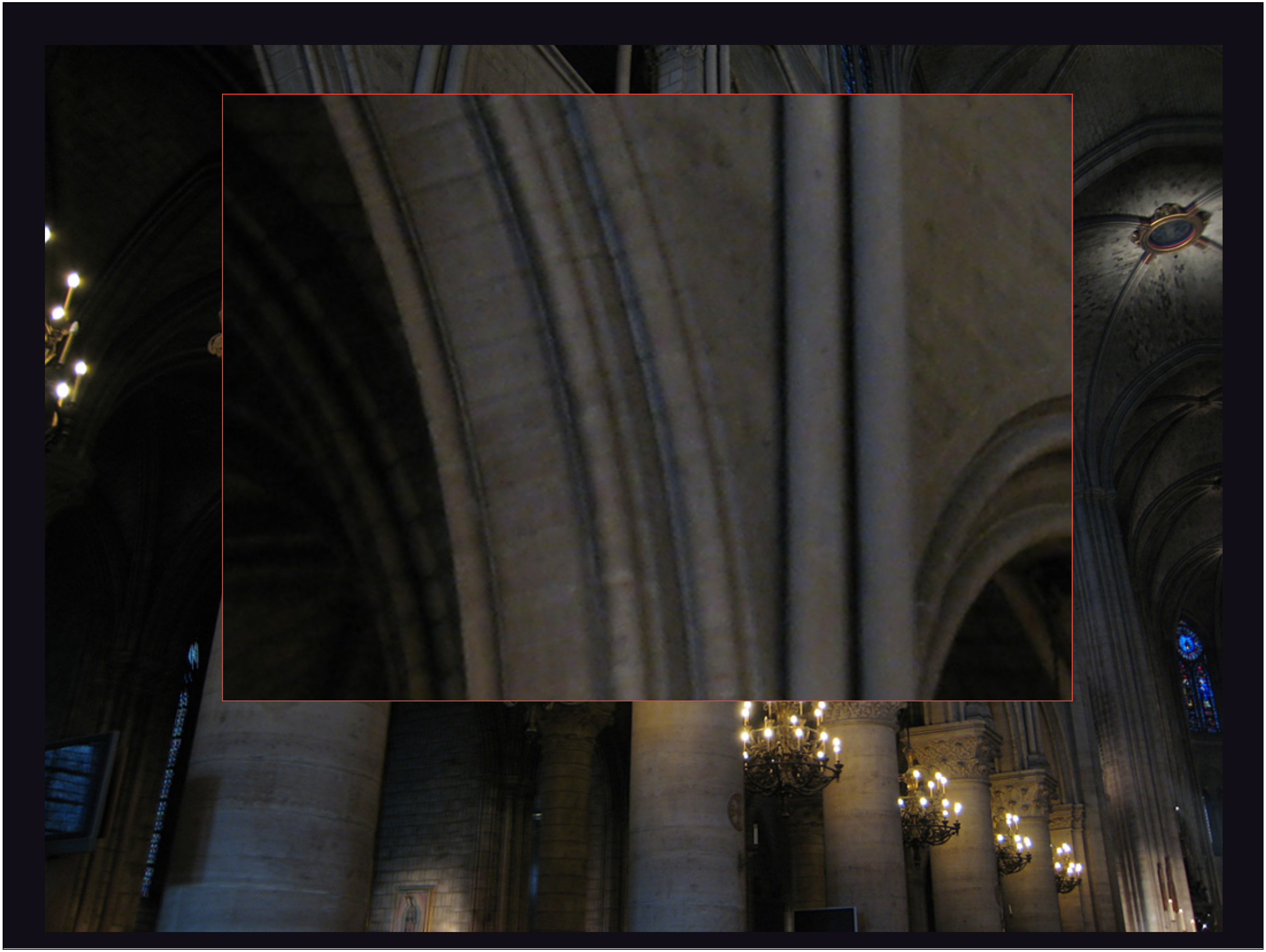


bilateral filtering

- bilateral filtering can easily be extended to RGB
- active area of research...

Averaging several short-exposure, high-ISO shots to avoid camera shake & reduce noise





Aligning a burst of short-exposure, high-ISO shots using the Casio EX-F1

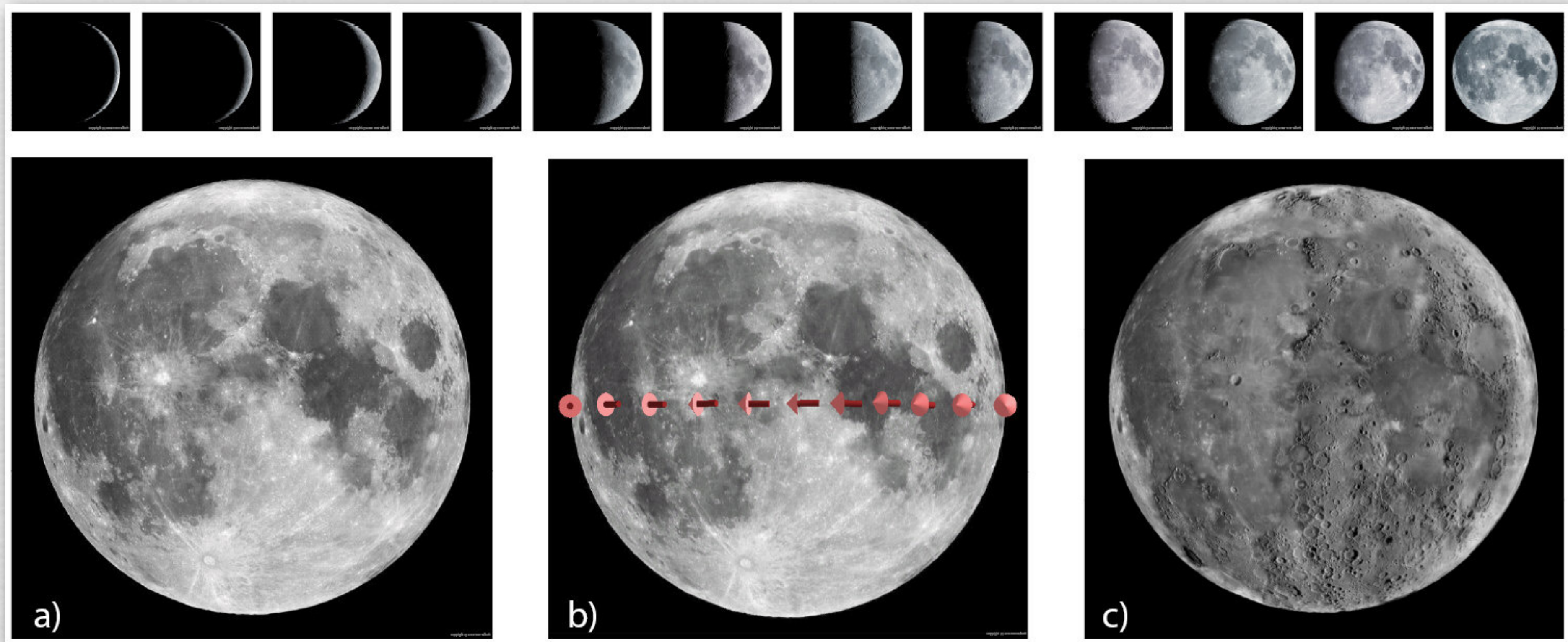
1/3 sec

burst at 60fps

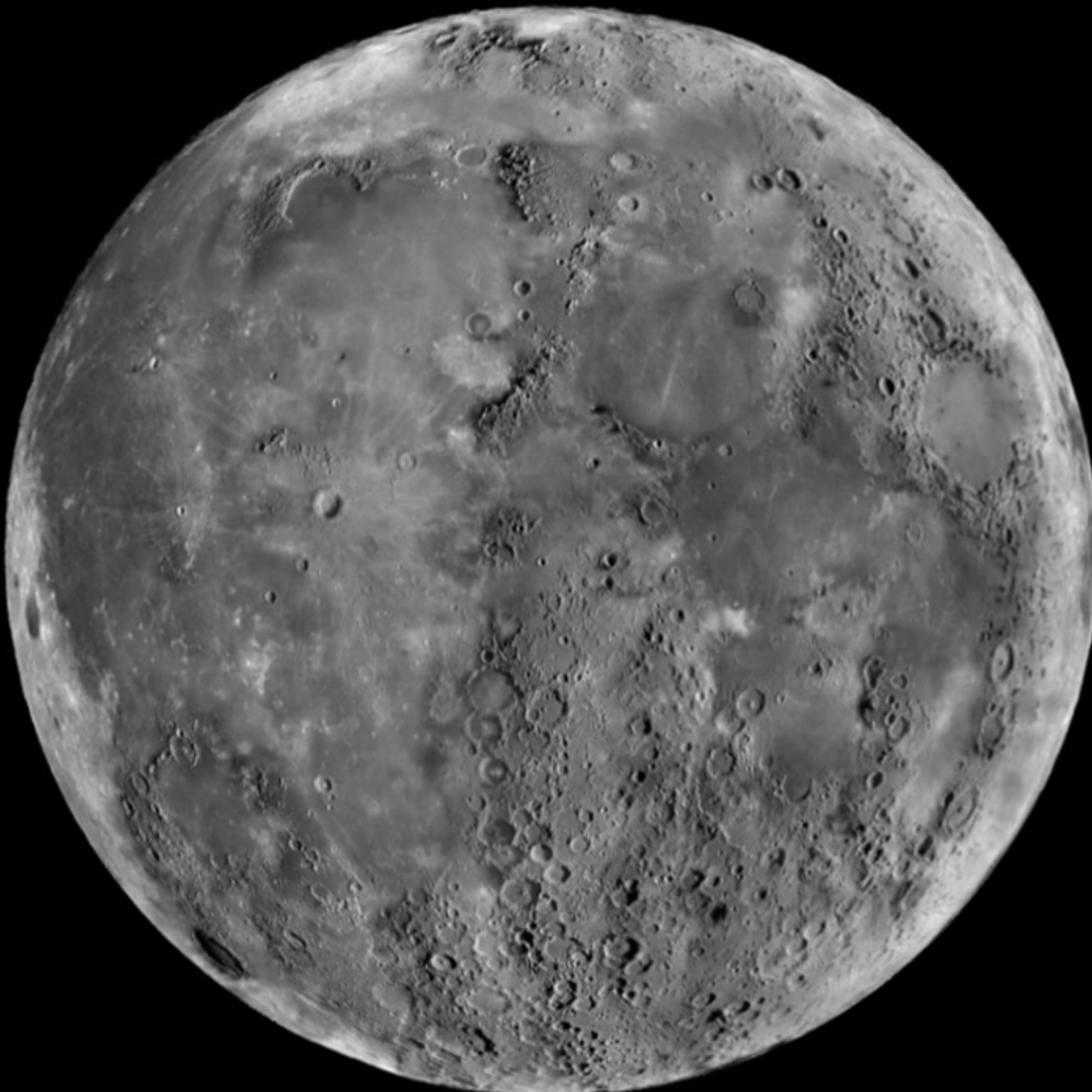
Aligning on a foreground object using the Casio EX-F1

Nonphotorealistic astrophotography

[Akers IEEE Visualization 2003]



- extract strip near illumination horizon from each image
- blend together to produce a single image where every point on the moon's surface exhibits grazing illumination



David Akers, Relighting of Moon, 2003

Slide credits

- Eddy Talvala

- Filippov, A., *How many bits are really needed in the image pixels?* (sic), http://www.linuxdevices.com/articles/AT9913651997.html