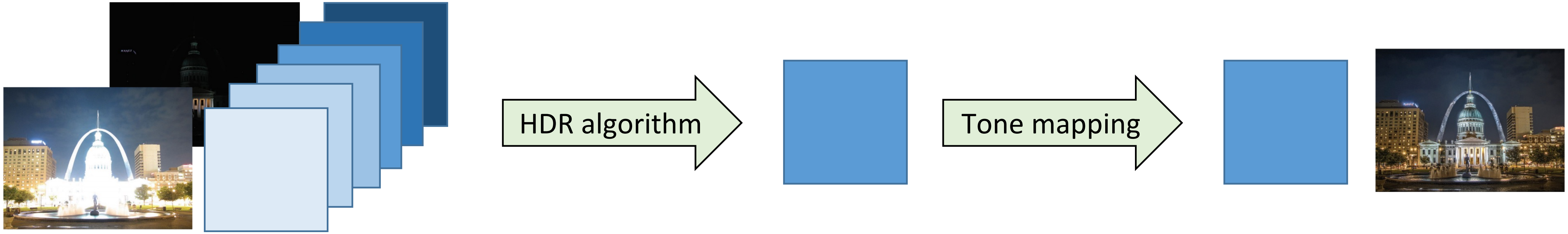
Problem Session 4

Topics

- High dynamic range images
  - Debevec's Method
  - Tone mapping
- SNR Calculations
  - Burst Imaging
  - Flutter Shutter

HDR & Tone mapping



Several 8-bit images
acquired at different
exposures ([0 255])

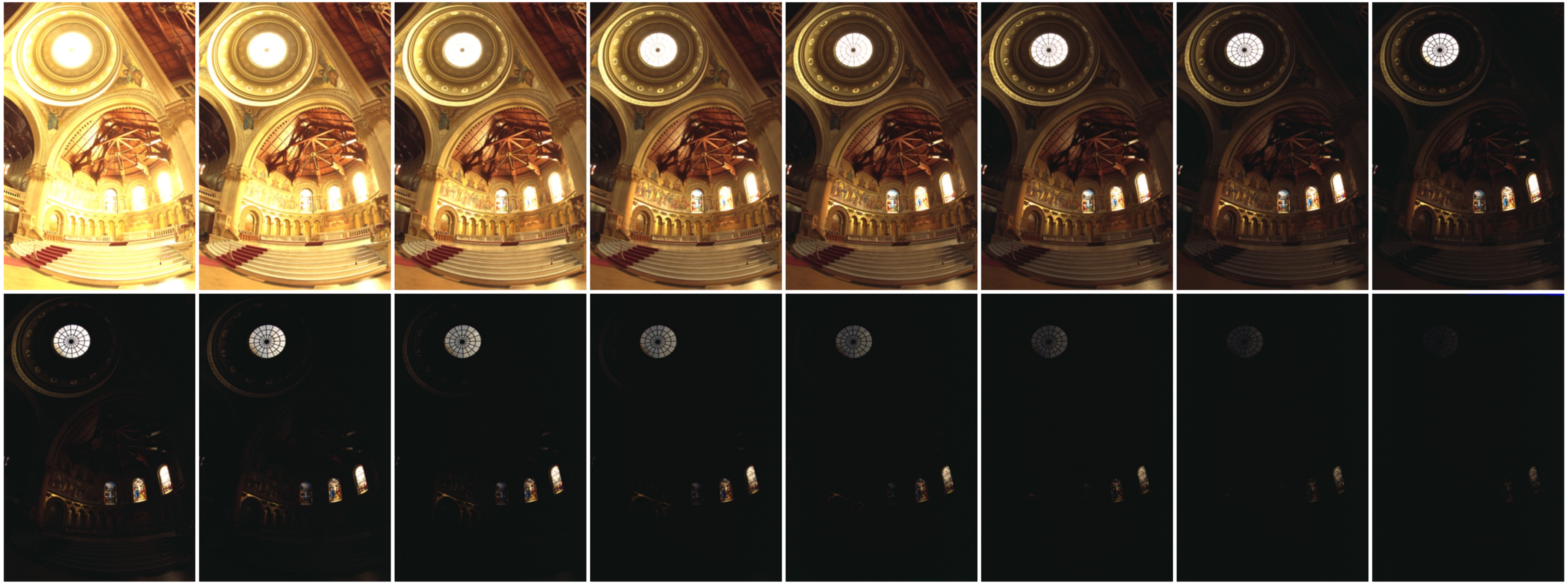
64-bit image
That has all
important
information

8-bit image
That has all important
information and can be
viewed on a regular screen

Task 1: High dynamic range

(Debevec's method)

```
# Number of Images
16
# Filename  1/shutter_speed f/stop gain(db) ND_filters
memorial0061.ppm 0.03125 8 0 0
memorial0062.ppm 0.0625  8 0 0
memorial0063.ppm 0.125   8 0 0
memorial0064.ppm 0.25    8 0 0
memorial0065.ppm 0.5     8 0 0
memorial0066.ppm 1       8 0 0
memorial0067.ppm 2       8 0 0
memorial0068.ppm 4       8 0 0
memorial0069.ppm 8       8 0 0
memorial0070.ppm 16      8 0 0
memorial0071.ppm 32      8 0 0
memorial0072.ppm 64      8 0 0
memorial0073.ppm 128     8 0 0
memorial0074.ppm 256     8 0 0
memorial0075.ppm 512     8 0 0
memorial0076.ppm 1024    8 0 0
```



Task 1: High dynamic range

Linearize the images using gamma of 2.2

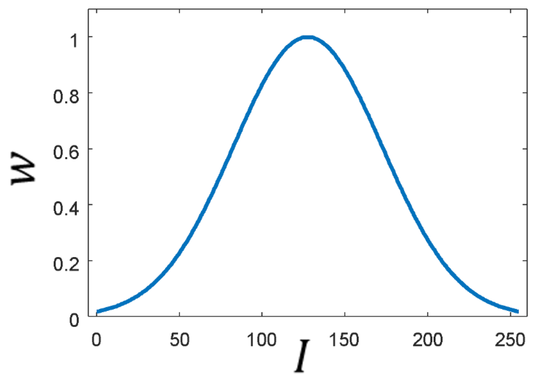
Task 1: High dynamic range image

Computing the weights: we want to give a higher weight to pixels that are close to the center of the dynamic range at each exposure k.

$$w_{k,ij} = \exp\left(-4\frac{(I_{k,ij} - 0.5)^2}{0.5^2}\right)$$

Center of the dynamic range (127.5 if pixels are [0,255])

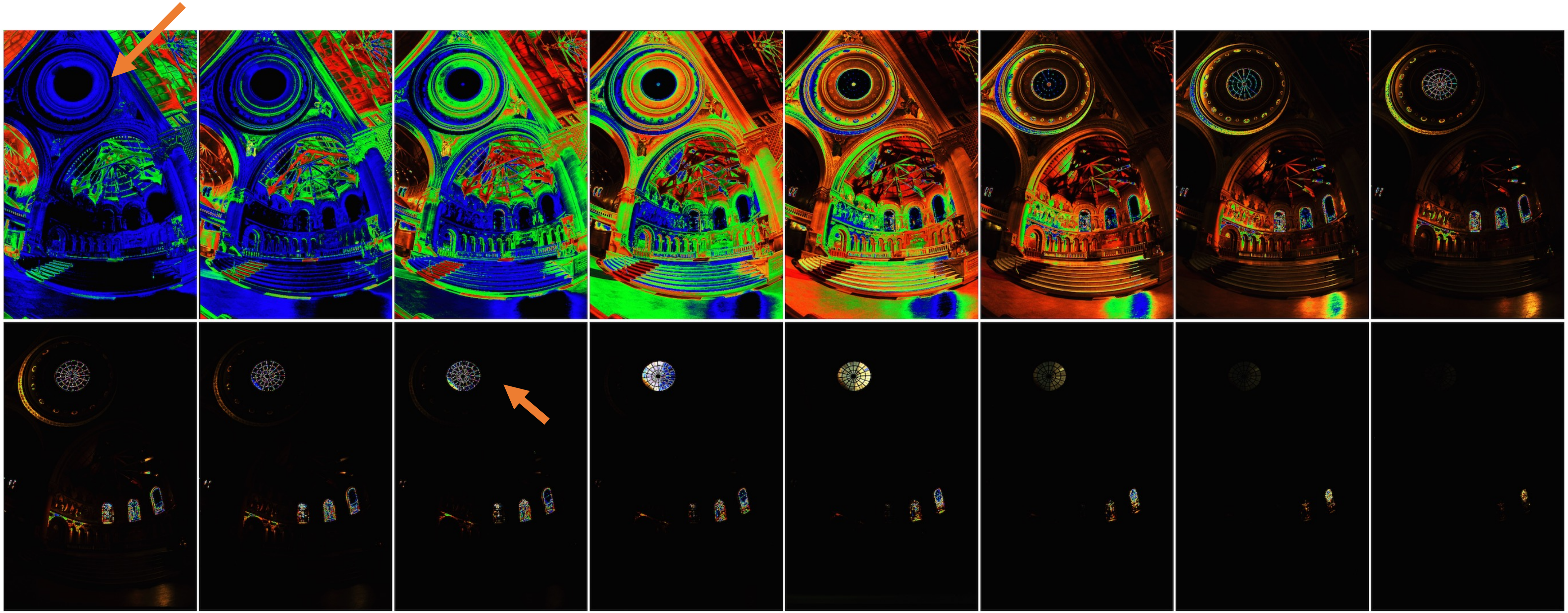
Linearized image (γ=2.2)



Compute this per color channel

Task 1: High dynamic range

Weights using Debevec's method (all 3 color channels)



Task 1: High dynamic range image

Find a good estimation, $\widehat{X}$, for the "true image", X, using an optimization problem:

Perceptually similar

Minimize the difference, in log scale, between your result, $\hat{X}$, at different exposures t_k and the acquired images at different exposures (I_{lin_k}). Multiplied by weight to indicate what's more important.

$$\underset{\widehat{X}}{\text{minimze}}\ O = \sum_k w_k \left(\log(I_{lin_k}) - \log(t_k \widehat{X})\right)^2$$

Calculating the derivative of O, we get:

hdr

$$\hat{X} = \exp\left(\frac{\sum_k w_k \left(\log(I_{lin_k}) - \log(t_k)\right)}{\sum_k w_k}\right)$$

scale

Task 1: High dynamic range - Tonemapping

- After exp, scaling, cropping, try your own scaling+gamma correction
- Choose scale and gamma yourself, report chosen parameters and resulting image
    - Don't scale+shift to fill the range [0, 1], scale only

$$I_{HDR} = (s \times I_{HDR\,linear})^{\gamma}$$

- Also try Drago's tonemapping from opencv

```
# Normalize
hdr = np.exp(hdr / scale)
hdr *= 0.8371896/np.mean(hdr)  # th
# convert to 32 bit floating point
hdr = np.float32(hdr)

# crop boundary - image data here a
hdr = hdr[29:720, 19:480, :]
```

Task 1: High dynamic range

Adjusting γ and s

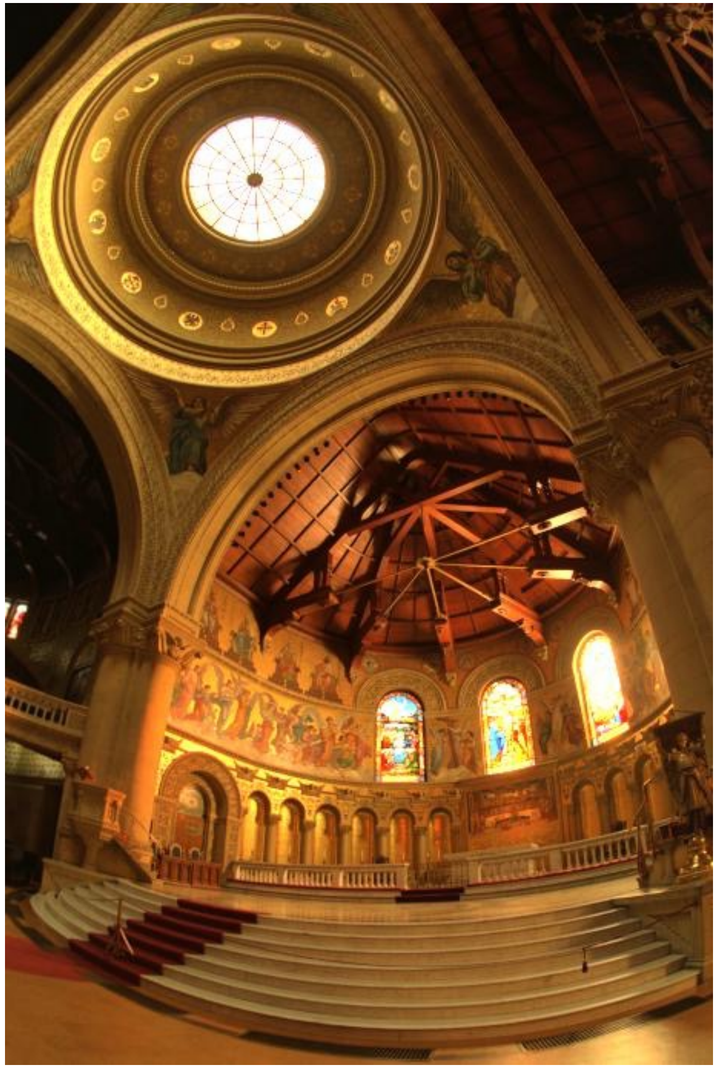
$s = 0.1$

$s = 1.0$

$\gamma = \frac{1}{5}$ $\gamma = \frac{1}{3}$ $\gamma = \frac{1}{2.2}$

Task 1: High dynamic range

Generate a tonemapped image using `cv2.createTonemapDrago`



Task 2/3: Denoising and SNR

- Read noise
  - Noise from heat in pixel hardware when image is captured
  - Signal-independent
  - Typically modelled as a **Gaussian distribution**
- Shot noise
  - Noise from statistics of light-pixel interactions
  - Signal-dependent
  - Typically modelled as a **Poisson distribution**

Task 2: Burst SNR calculations

$$SNR = \frac{\mu}{\sigma} = \frac{mean\ number\ of\ photons}{standard\ deviation\ of\ noise}$$

- Adding k signals of strength μ => mean becomes $k\mu$
- Scaling Gaussian/Poisson variable by k => variance becomes $k^2\sigma^2$
- Adding independent Gaussian distributions

Variance = (standard deviation)^2

$$G(\mu_1, \sigma_1^2) + G(\mu_2, \sigma_2^2) \sim G(\mu_1 + \mu_2, \underbrace{\sigma_1^2 + \sigma_2^2})$$

- Adding independent Poisson distributions

Both mean and variance

$$Pois(\lambda_1) + Pois(\lambda_2) \sim Pois(\underbrace{\lambda_1 + \lambda_2})$$

- Both signal and noise increase with the number of photons.
  - However, signal increases faster than noise!

Task 3: Flutter Shutter

You're asked to compare the SNR of two imaging setups:

- Consumer camera in everyday use
- sCMOS in microscopy

At different acquisition modes:

- Flutter shutter
- Burst

Calculate the SNR and discuss your results.

Task 3: SNR calculations

Consumer camera in everyday conditions:

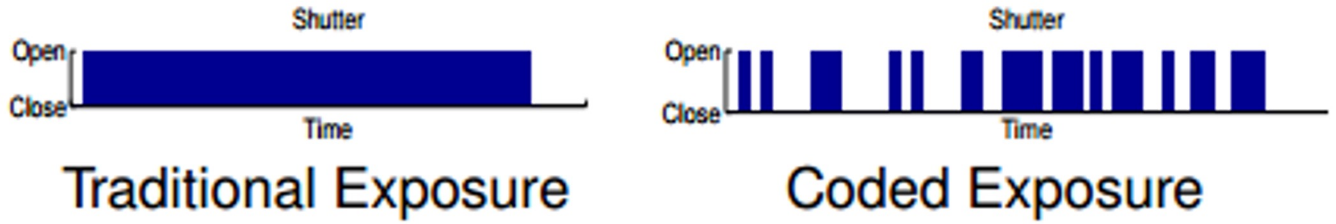
- Lots of photons (low shot noise)
- Room temperature (high read noise)

sCMOS microscope sensor in controlled conditions:

- Few photons (high shot noise)
- Cooled sensor (low read noise)

Task 3: SNR calculations

Flutter shutter: temporally modulated aperture pattern. Used, for example, for better motion deblurring (see *"Coded Exposure Photography: Motion Deblurring using Fluttered Shutter",* Raskar et al.). The result is a single image.



Burst: acquires multiple short-exposure images. (for this problem, assume no delay between exposures)

Tradeoff: Number of exposures vs. number of photons

Task 3: SNR calculations

For each of the four cases, calculate:

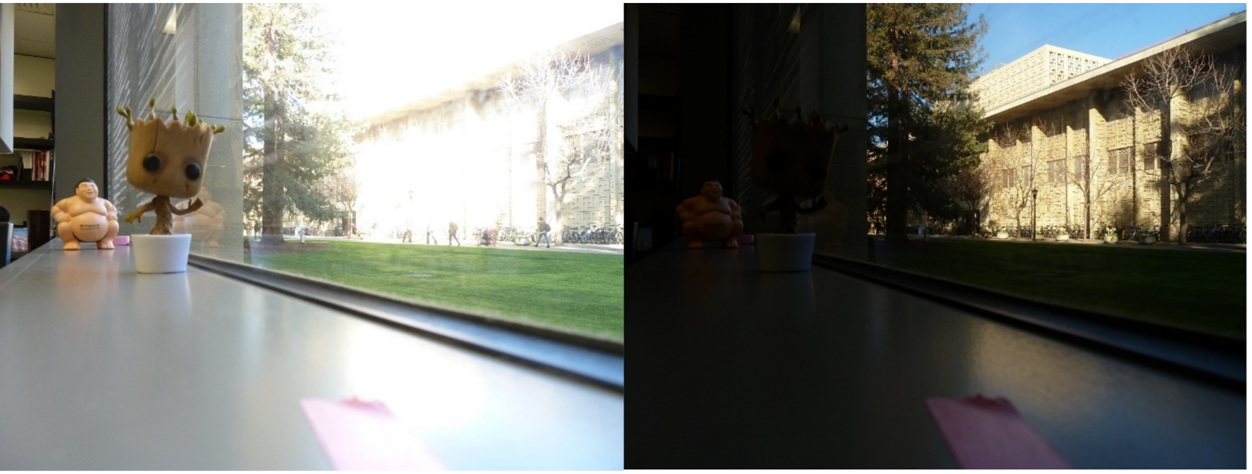
- The average number of photons (the signal)
- The standard deviation of the noise
- and divide them

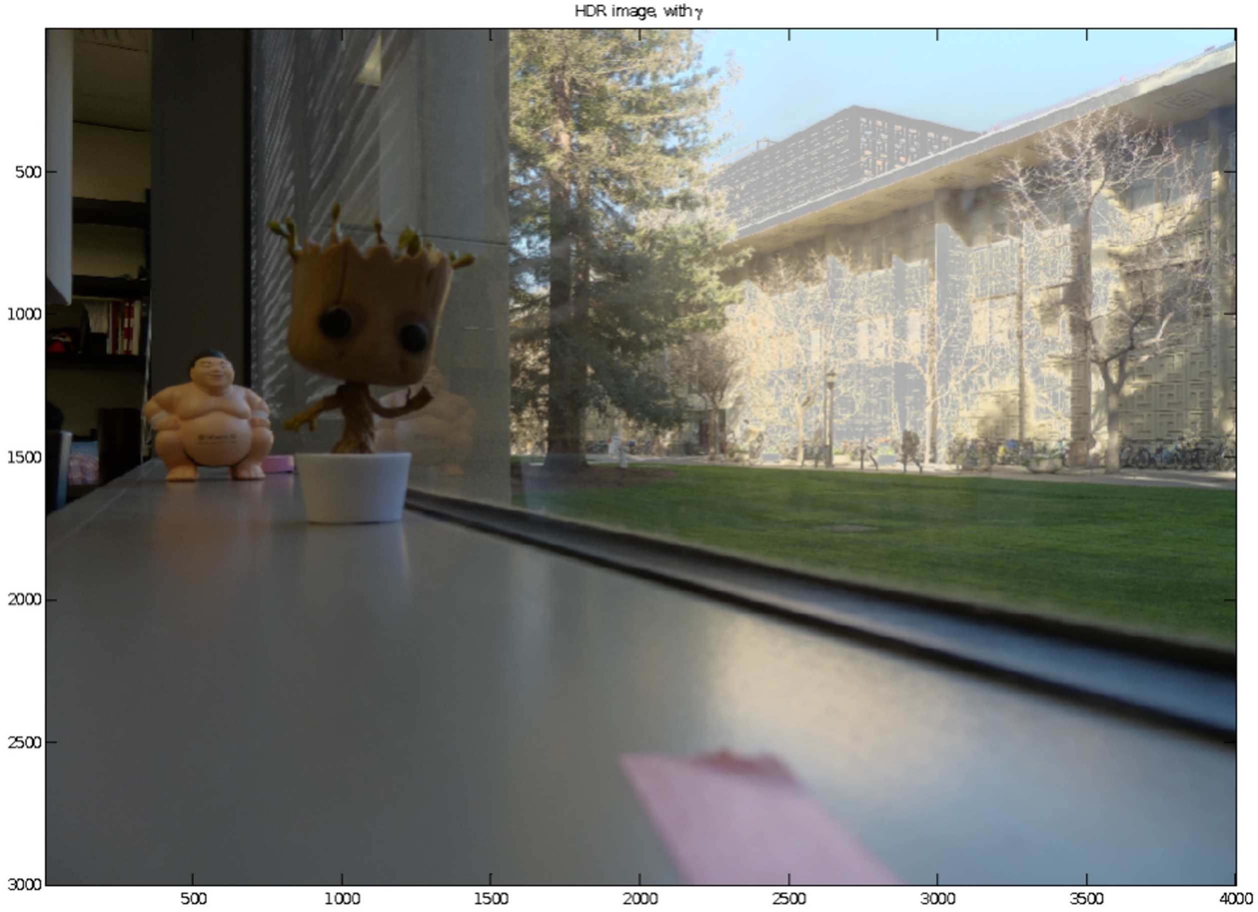
Don’t forget to describe your results and conclusions.

Intuitively, what general behavior do you expect?

Bonus: your own HDR image

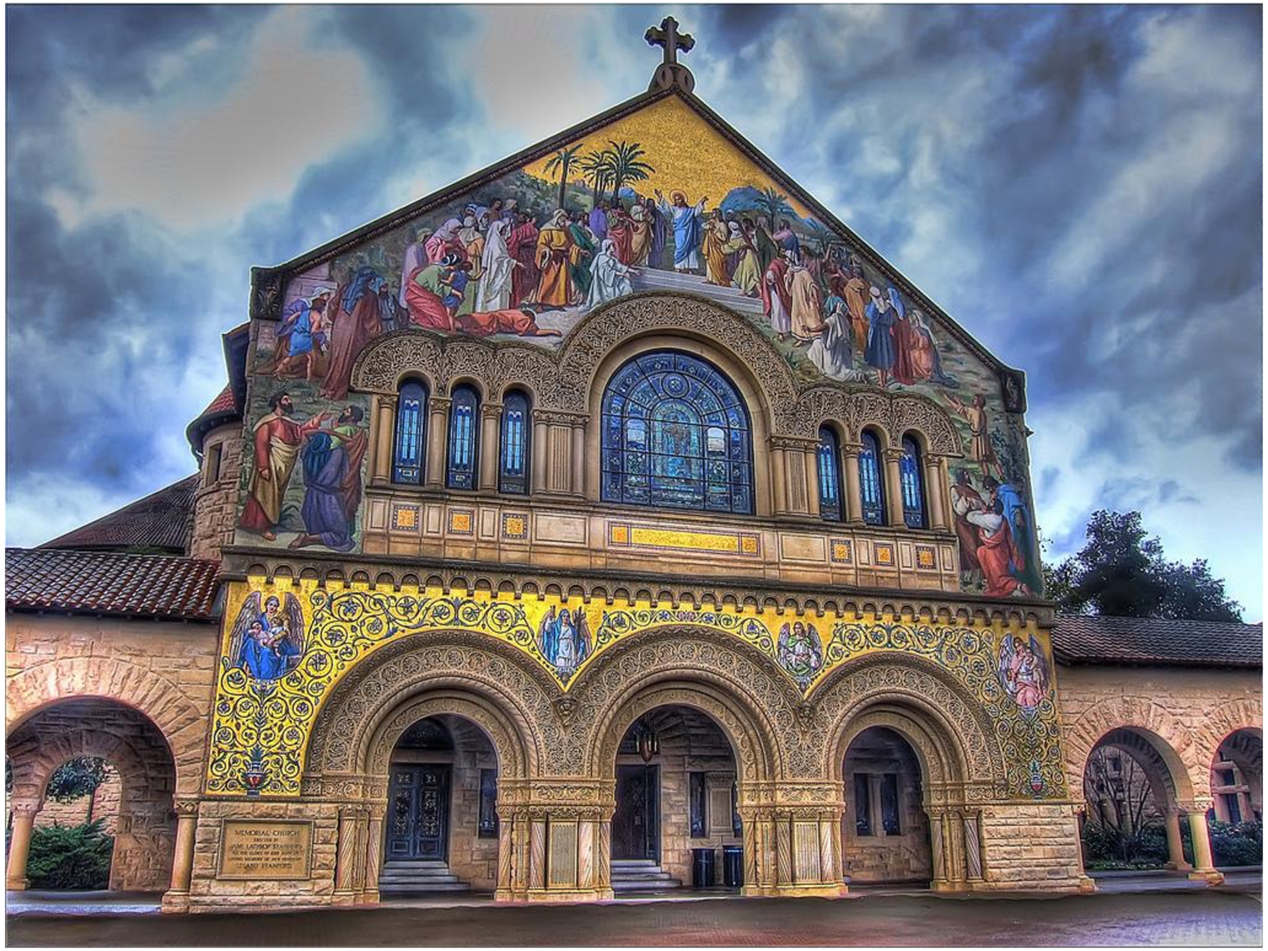
Around ~10 images with different exposures





Have a nice weekend!

And good luck with the homework!



Stanford Memorial Church (HDR) https://www.flickr.com/photos/scottloftesness/4334766965