Problem Session 4

Topics

- T1 - High dynamic range images
- T2 - SNR Calculations with Burst Imaging
- T3 - SNR Calculations with Flutter Shutter

Inputs

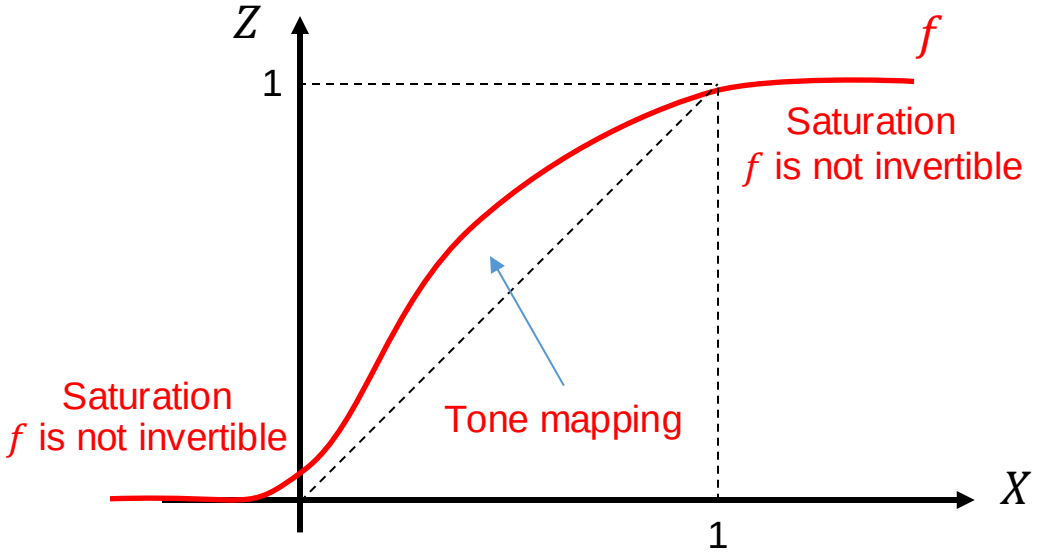
```
# Number of Images
16
# Filename  1/shutter_speed f/stop gain(db) ND_filters
memorial0061.ppm 0.03125 8 0 0
memorial0062.ppm 0.0625  8 0 0
memorial0063.ppm 0.125   8 0 0
memorial0064.ppm 0.25    8 0 0
memorial0065.ppm 0.5     8 0 0
memorial0066.ppm 1       8 0 0
memorial0067.ppm 2       8 0 0
memorial0068.ppm 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
```

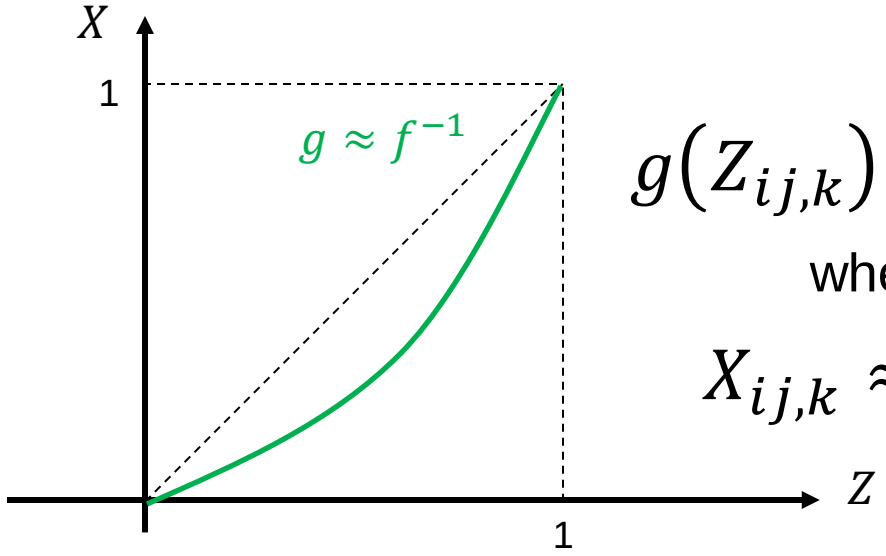
Debevec's Idea

$$E_{ij} \xrightarrow{\text{Exposure}} X_{ij,k} \xrightarrow{\text{Saturation + Tone Mapping}} Z_{ij,k}$$

pixel exposure time

$$X_{ij,k} = E_{ij} t_k \qquad Z_{ij,k} = f(X_{ij,k})$$





$$g(Z_{ij,k}) \approx X_{ij,k}$$

when

$$X_{ij,k} \approx 0.5$$

Debevec's Idea

$$g(Z_{ij,k}) \approx X_{ij,k} = E_{ij} t_k \qquad \text{when} \quad X_{ij,k} \approx 0.5$$

$$\log g(Z_{ij,k}) \approx \log E_{ij} + \log t_k \qquad \text{when} \quad X_{ij,k} \approx 0.5$$

$$\mathcal{L}(E_{ij}) = \sum_{i,j,k} w_{ij,k} \left(\log g(Z_{ij,k}) - \log E_{ij} - \log t_k\right)^2$$

where $\quad w_{ij,k} \ll 1 \Leftrightarrow X_{ij,k} \neq 0.5$

Minimize $\mathcal{L}(E_{ij})$ over E_{ij} $\longrightarrow$ $E_{ij} = \exp\left(\dfrac{\sum_k w_{ij,k}\left(\log \dfrac{g(Z_{ij,k})}{t_k}\right)}{\sum_k w_{ij,k}}\right)$

Debevec’s Paper

Debevec, Paul E., and Jitendra Malik. "Recovering high dynamic range radiance maps from photographs." *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023. 643-652.

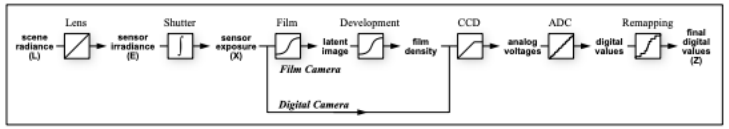


Figure 1: **Image Acquisition Pipeline** *shows how scene radiance becomes pixel values for both film and digital cameras. Unknown nonlinear mappings can occur during exposure, development, scanning, digitization, and remapping. The algorithm in this paper determines the aggregate mapping from scene radiance* L *to pixel values* Z *from a set of differently exposed images.*

the entire dynamic range captured by the original photographs.

1.1 Applications

Our technique of deriving imaging response functions and recovering high dynamic range radiance maps has many possible applications in computer graphics:

Image-based modeling and rendering

Image-based modeling and rendering systems to date (e.g. [11, 15, 2, 3, 12, 6, 17]) make the assumption that all the images are taken with the same exposure settings and film response functions. However, almost any large-scale environment will have some areas that are much brighter than others, making it impossible to adequately photograph the scene using a single exposure setting. In indoor scenes with windows, this situation often arises within the field of view of a single photograph, since the areas visible through the windows can be far brighter than the areas inside the building.

By determining the response functions of the imaging device, the method presented here allows one to correctly fuse pixel data from photographs taken at different exposure settings. As a result, one can properly photograph outdoor areas with short exposures, and indoor areas with longer exposures, without creating inconsistencies in the data set. Furthermore, knowing the response functions can be helpful in merging photographs taken with different imaging systems, such as video cameras, digital cameras, and film cameras with various film stocks and digitization processes.

The area of image-based modeling and rendering is working toward recovering more advanced reflection models (up to complete BRDF's) of the surfaces in the scene (e.g. [21]). These methods, which involve observing surface radiance in various directions under various lighting conditions, require absolute radiance values rather than the nonlinearly mapped pixel values found in conventional images. Just as important, the recovery of high dynamic range images will allow these methods to obtain accurate radiance values from surface specularities and from incident light sources. Such higher radiance values usually become clamped in conventional images.

Image processing

Most image processing operations, such as blurring, edge detection, color correction, and image correspondence, expect pixel values to be proportional to the scene radiance. Because of nonlinear image response, especially at the point of saturation, these operations can produce incorrect results for conventional images.

In computer graphics, one common image processing operation is the application of synthetic motion blur to images. In our results (Section 3), we will show that using true radiance maps produces significantly more realistic motion blur effects for high dynamic range scenes.

Image compositing

Many applications in computer graphics involve compositing image data from images obtained by different processes. For example, a background matte might be shot with a still camera, live action might be shot with a different film stock or scanning process, and CG elements would be produced by rendering algorithms. When there are significant differences in the response curves of these imaging processes, the composite image can be visually unconvincing. The technique presented in this paper provides a convenient and robust method of determining the overall response curve of any imaging process, allowing images from different processes to be used consistently as radiance maps. Furthermore, the recovered response curves can be inverted to render the composite radiance map as if it had been photographed with any of the original imaging processes, or a different imaging process entirely.

A research tool

One goal of computer graphics is to simulate the image formation process in a way that produces results that are consistent with what happens in the real world. Recovering radiance maps of real-world scenes should allow more quantitative evaluations of rendering algorithms to be made in addition to the qualitative scrutiny they traditionally receive. In particular, the method should be useful for developing reflectance and illumination models, and comparing global illumination solutions against ground truth data.

Rendering high dynamic range scenes on conventional display devices is the subject of considerable previous work, including [20, 16, 5, 23]. The work presented in this paper will allow such methods to be tested on real radiance maps in addition to synthetically computed radiance solutions.

1.2 Background

The photochemical processes involved in silver halide photography have been the subject of continued innovation and research ever since the invention of the daguerretype in 1839. [18] and [8] provide a comprehensive treatment of the theory and mechanisms involved. For the newer technology of solid-state imaging with charge coupled devices, [19] is an excellent reference. The technical and artistic problem of representing the dynamic range of a natural scene on the limited range of film has concerned photographers from the early days – [1] presents one of the best known systems to choose shutter speeds, lens apertures, and developing conditions to best coerce the dynamic range of a scene to fit into what is possible on a print. In scientific applications of photography, such as in astronomy, the nonlinear film response has been addressed by suitable calibration procedures. It is our objective instead to develop a simple self-calibrating procedure not requiring calibration charts or photometric measuring devices.

In previous work, [13] used multiple flux integration times of a CCD array to acquire extended dynamic range images. Since direct CCD outputs were available, the work did not need to deal with the

Step 1: Linearize

Linearize the images using gamma of 2.2

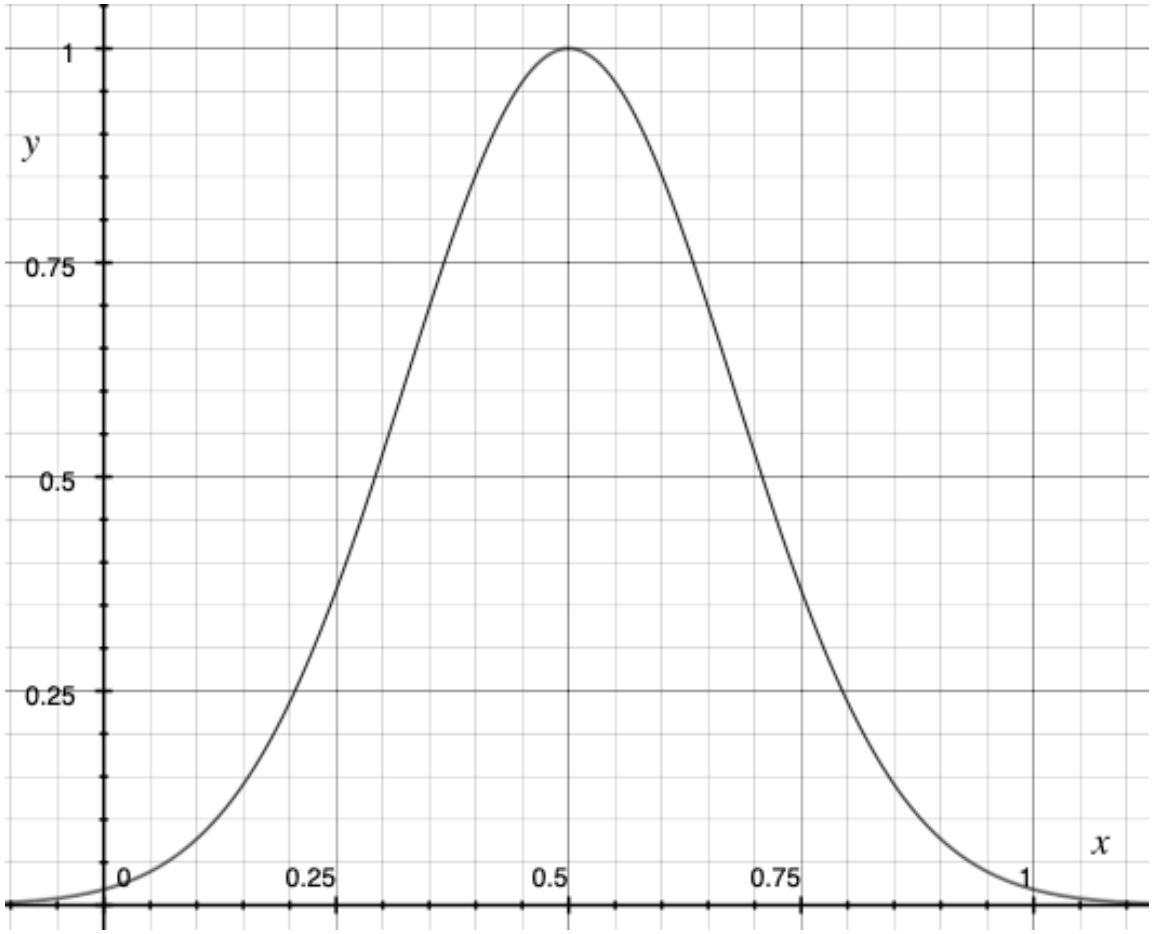
$$X_{ij,k} = Z_{ij,k}^{2.2}$$

Step 2: Compute Weights

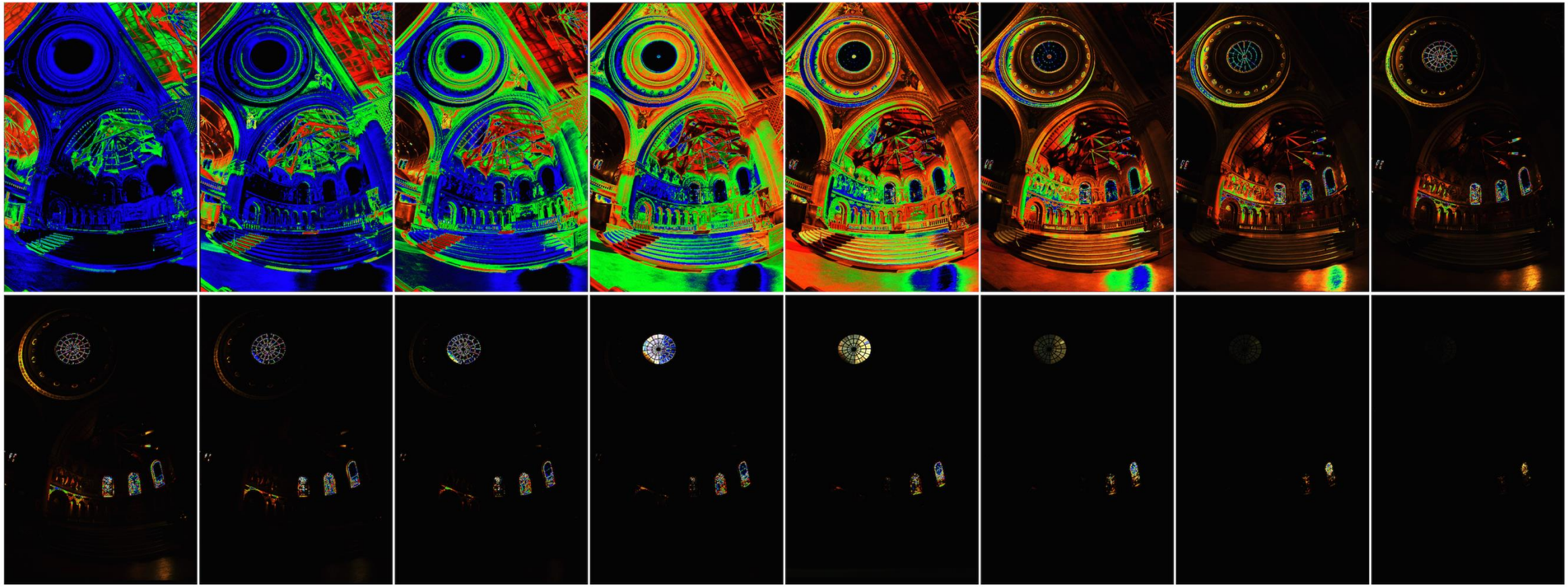
Linearized image
(γ=2.2)

Center of the dynamic range
(pixels are in [0,1])

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



Step 2: Compute Weights

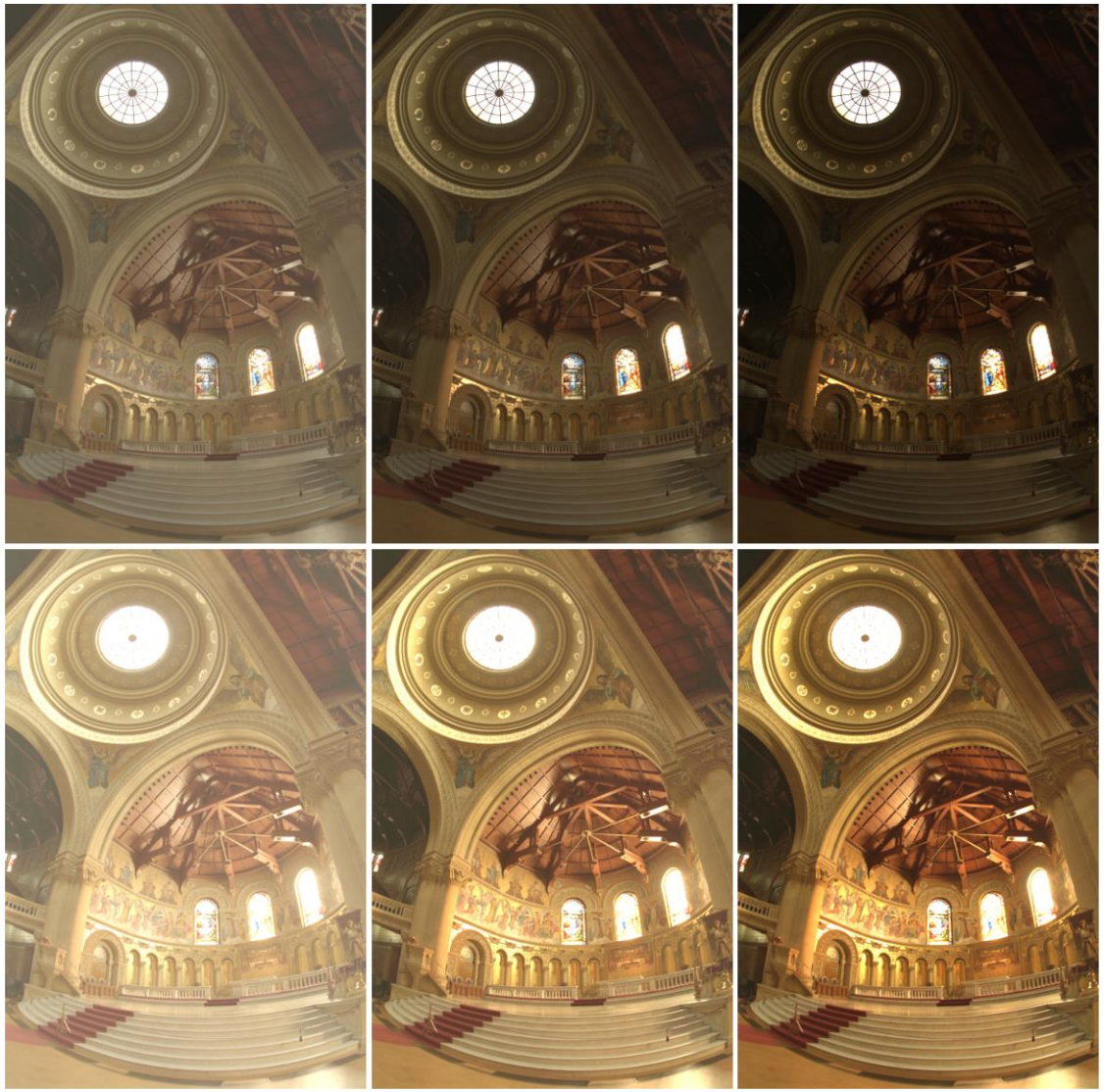


Step 3: “Average” Images

$$E_{ij} = \exp\left(\frac{\sum_k w_{ij,k}\left(\log\frac{X_{ij,k}}{t_k}\right)}{\sum_k w_{ij,k}}\right)$$

Step 4: Tonemapping

$$\hat{I} = \mathrm{clamp}\big((s \cdot E)^{\gamma}\big)$$



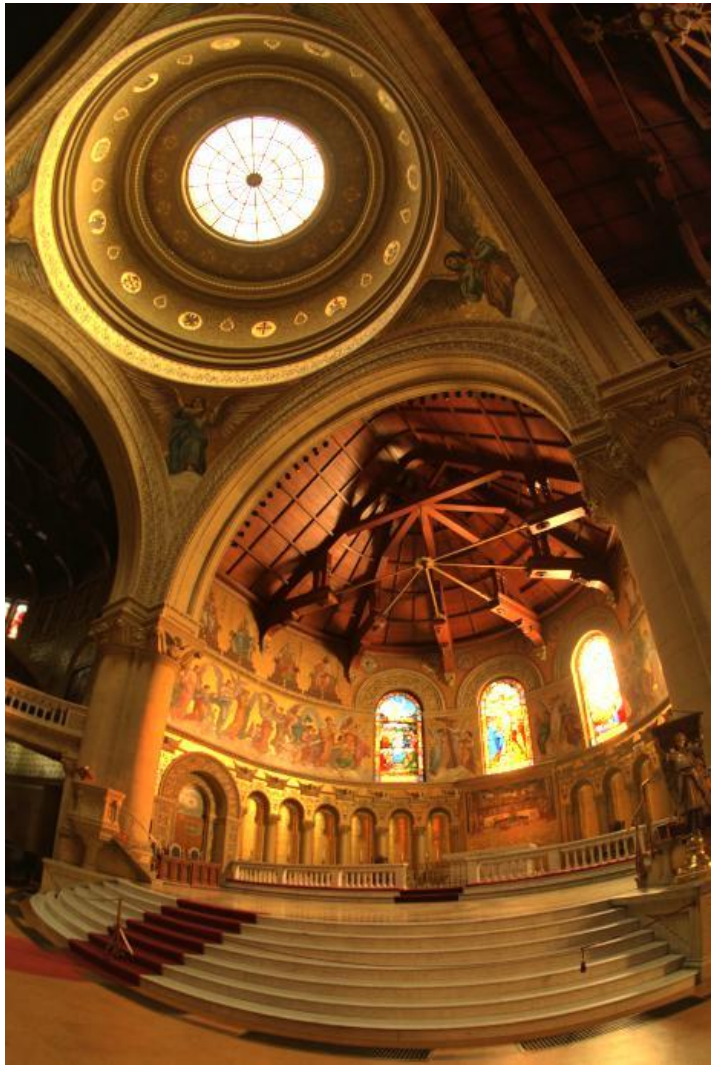
$s = 0.1$

$s = 1.0$

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

Step 4: Tonemapping

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

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

- 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

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

- Adding independent Poisson distributions

$$Pois(\lambda_1) + Pois(\lambda_2) \sim Pois(\lambda_1 + \lambda_2)$$

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

Task 3: Flutter Shutter

	Consumer camera Everyday use	CMOS camera Photon-limited application
Flutter Shutter		
Burst		

Calculate SNR and discuss your results

Task 3: SNR calculations

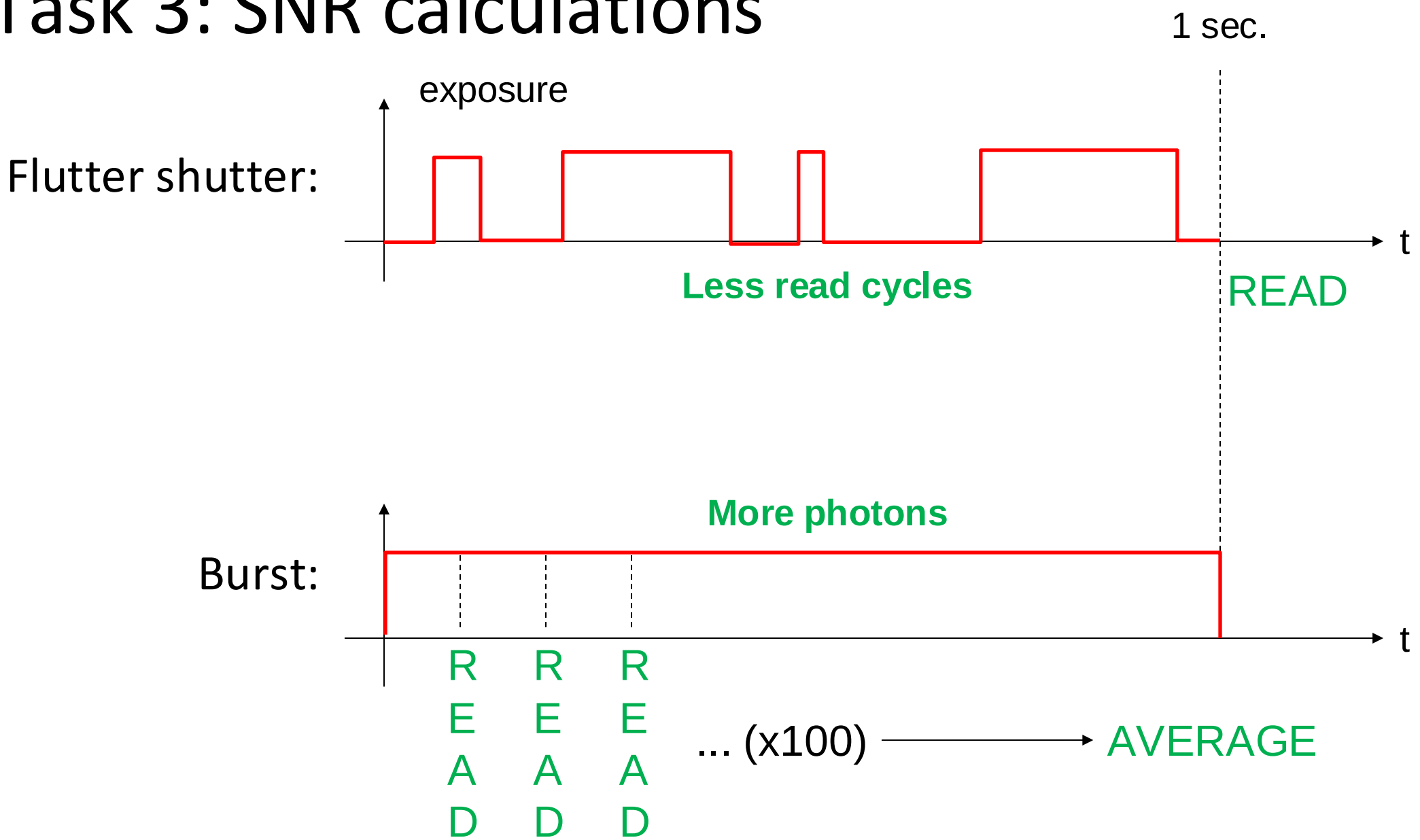
Consumer camera in everyday conditions:

- Many photons (no shot noise)
- Room temperature (high read noise)

sCMOS microscope sensor in controlled conditions:

- Few photons (high shot noise)
- Cooled sensor (no read noise)

Task 3: SNR calculations



Task 3: SNR calculations

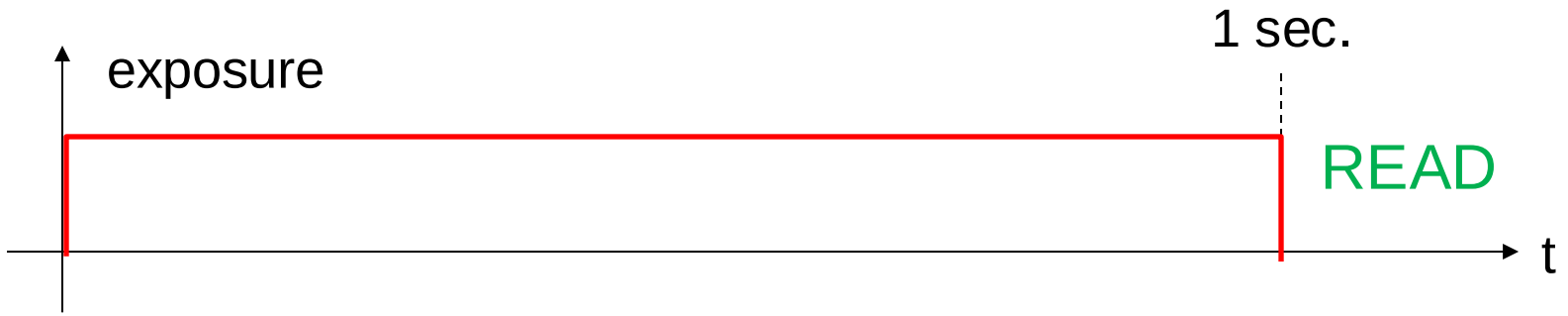
What behavior do you expect?

"Less read cycles" (flutter): preferred when read noise-limited

"More photons" (burst): preferred when shot noise-limited

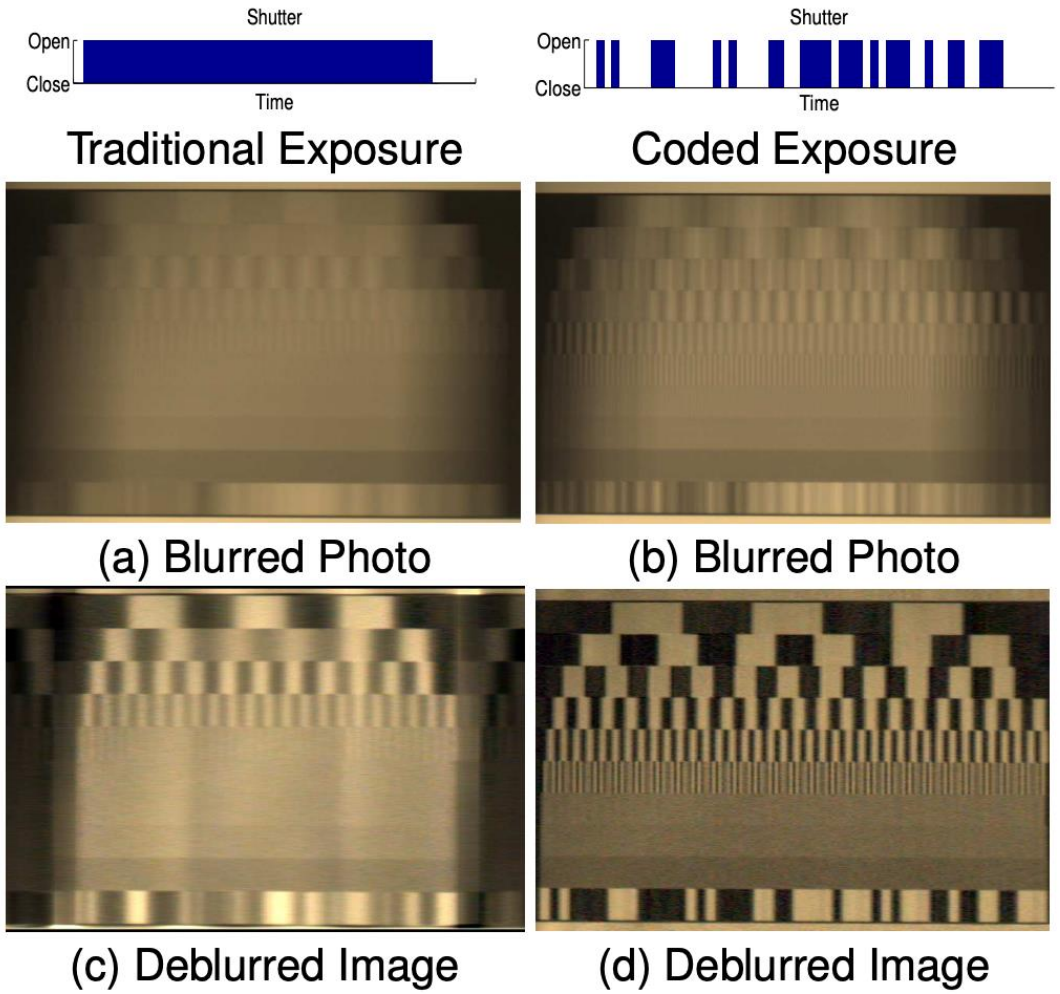
- For burst imaging, use the formulas from Task 2
- For the flutter shutter, divide the mean number of photons by the standard deviation of the noise.

Why not use?

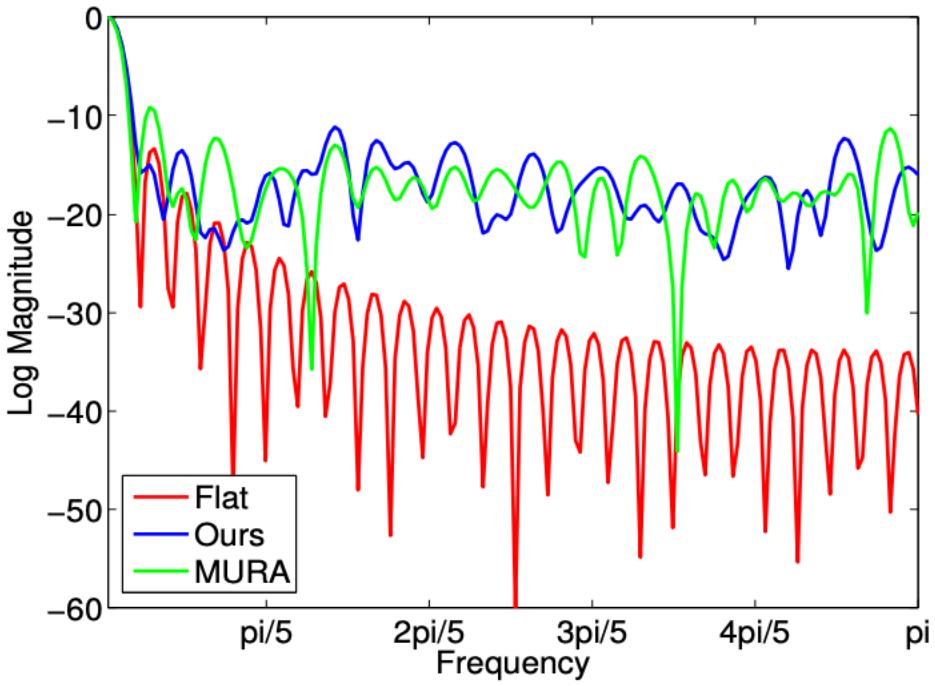


Deblurring becomes ill-posed

Coded Apertures



Fourier transform of the blur kernel

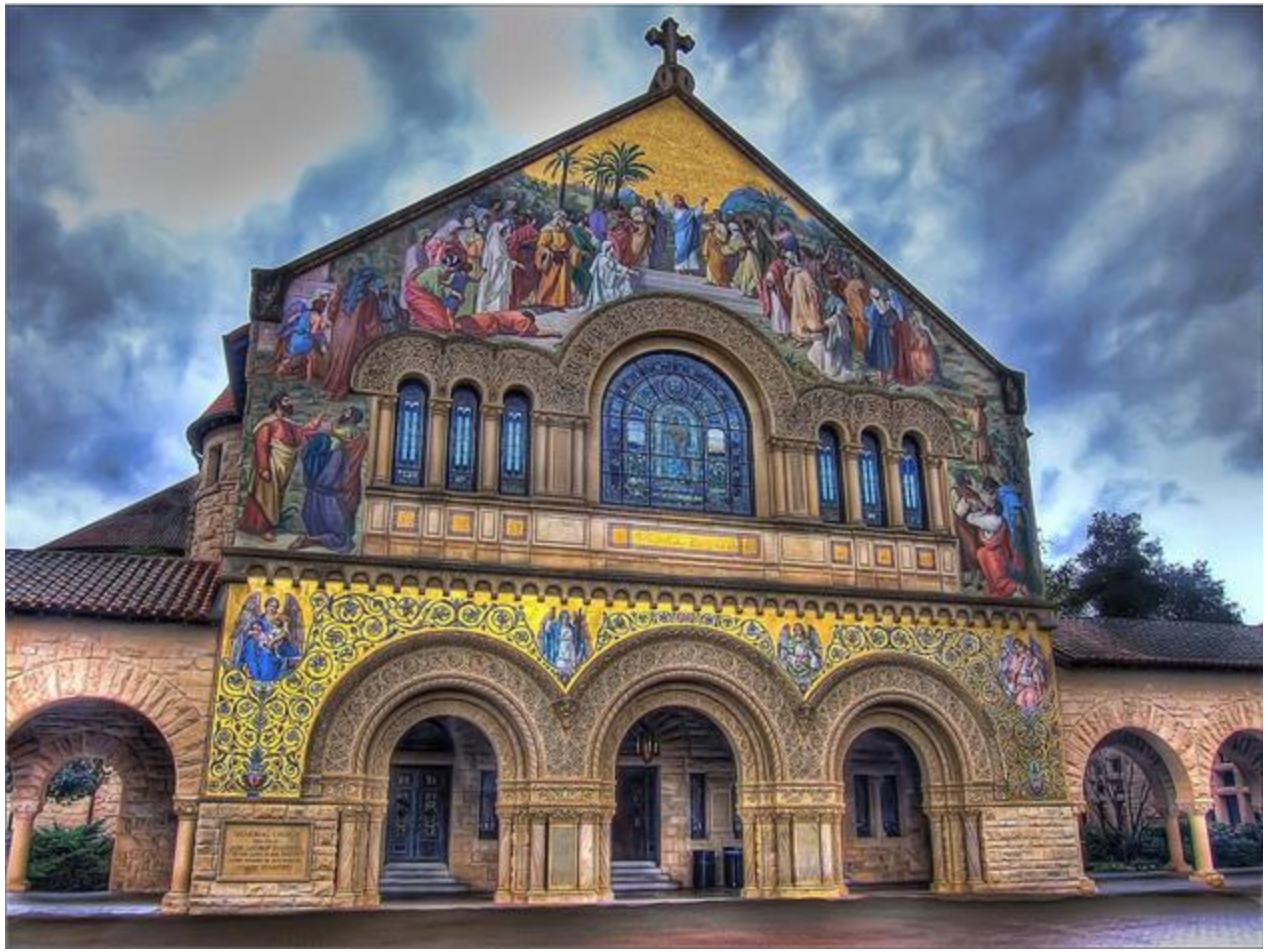


The coded aperture gives an invertible blur kernel

Raskar, Ramesh, Amit Agrawal, and Jack Tumblin. "Coded exposure photography: motion deblurring using fluttered shutter." Acm Siggraph 2006 Papers. 2006. 795-804.

Have a nice weekend!

And good luck with the homework!



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