Computational Photography

Prof. Feng Liu

Spring 2022

http://www.cs.pdx.edu/~fliu/courses/cs510/

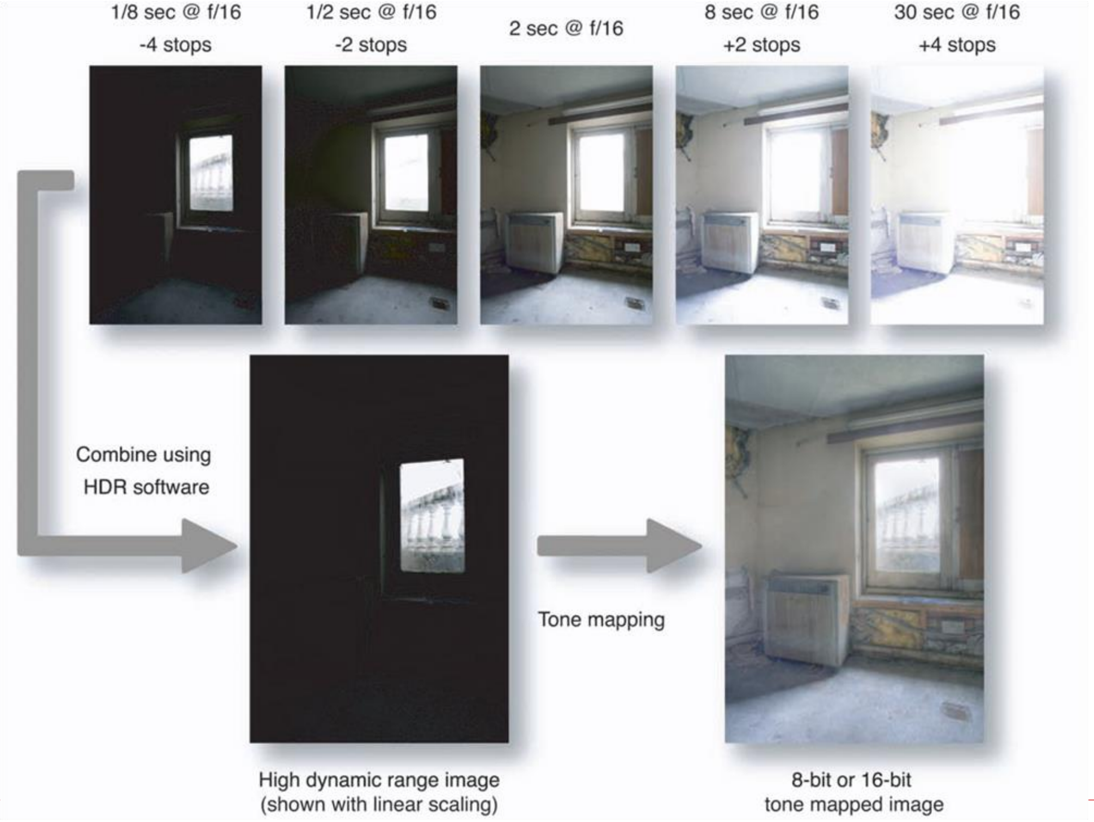
04/19/2022

Last Time

- Re-lighting
  - Tone mapping

Today

- Re-lighting: high dynamic range imaging
- Full presentation schedule is online



High dynamic range imaging

Digital Visual Effects

Yung-Yu Chuang

with slides by Fredo Durand, Brian Curless, Steve Seitz, Paul Debevec and Alexei Efros

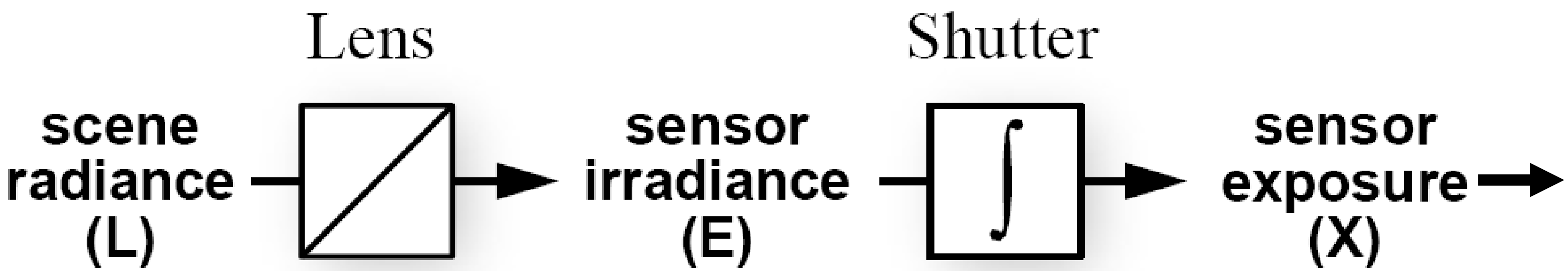
The world is high dynamic range

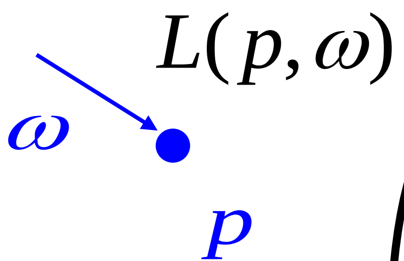
The world is high dynamic range

Camera is an imperfect device

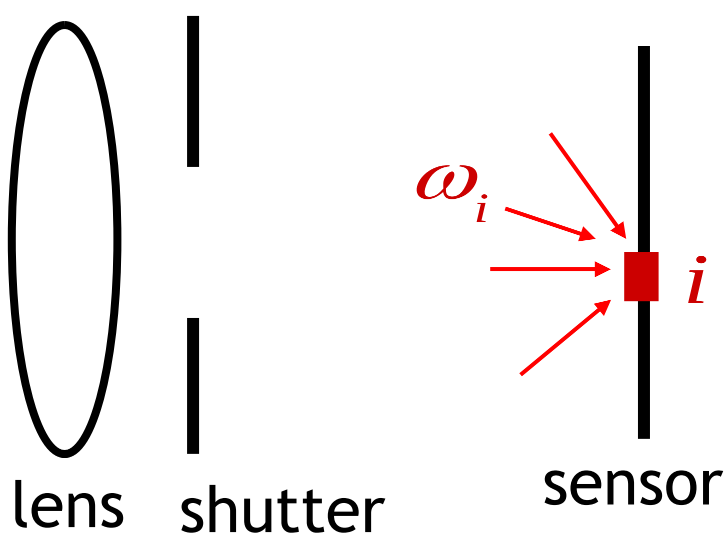
- Camera is an imperfect device for measuring the radiance distribution of a scene because it cannot capture the full spectral content and dynamic range.
- Limitations in sensor design prevent cameras from capturing all information passed by lens.

Camera pipeline





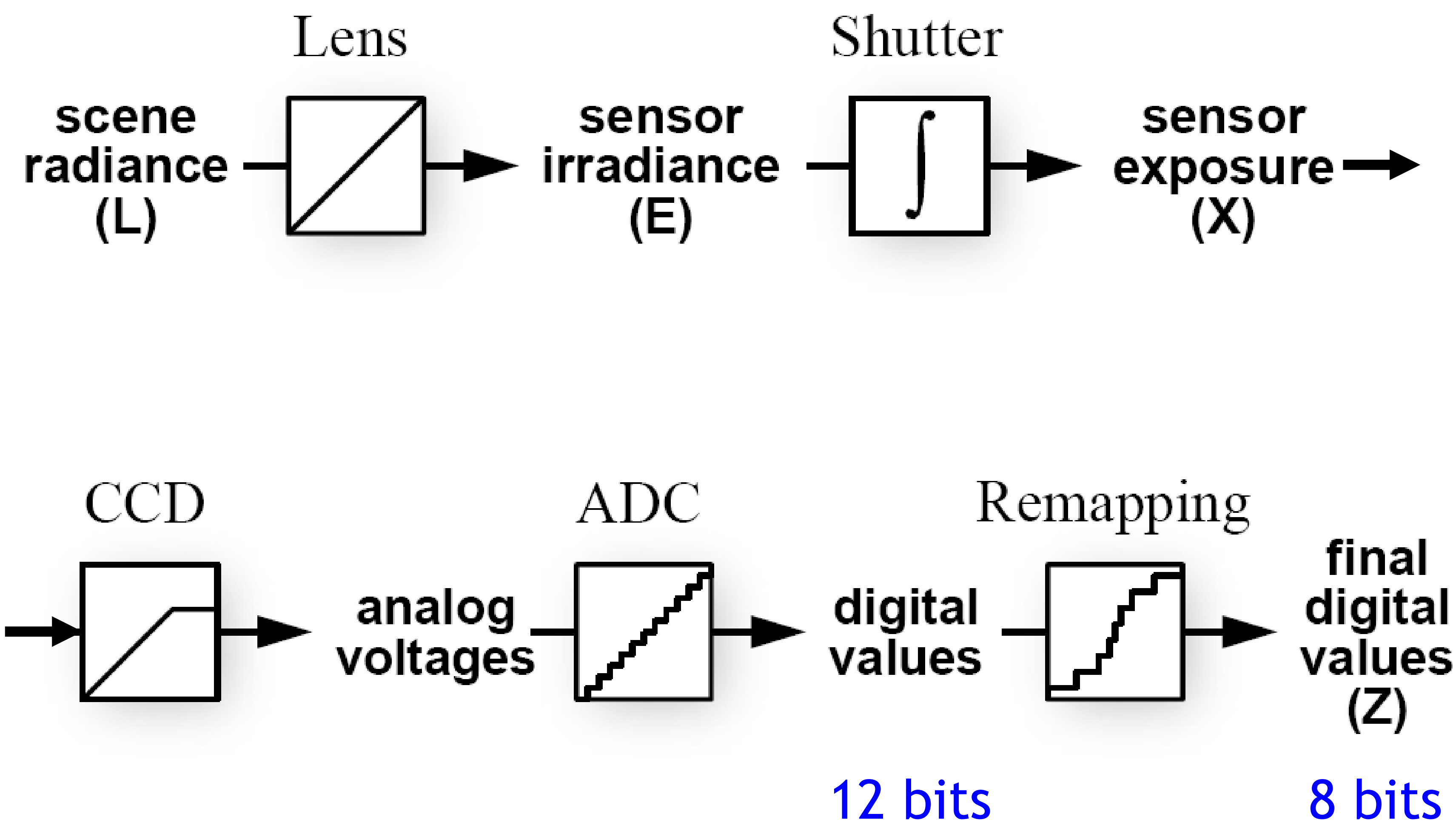
Assume a static scene, Thus, L is not a function of time.



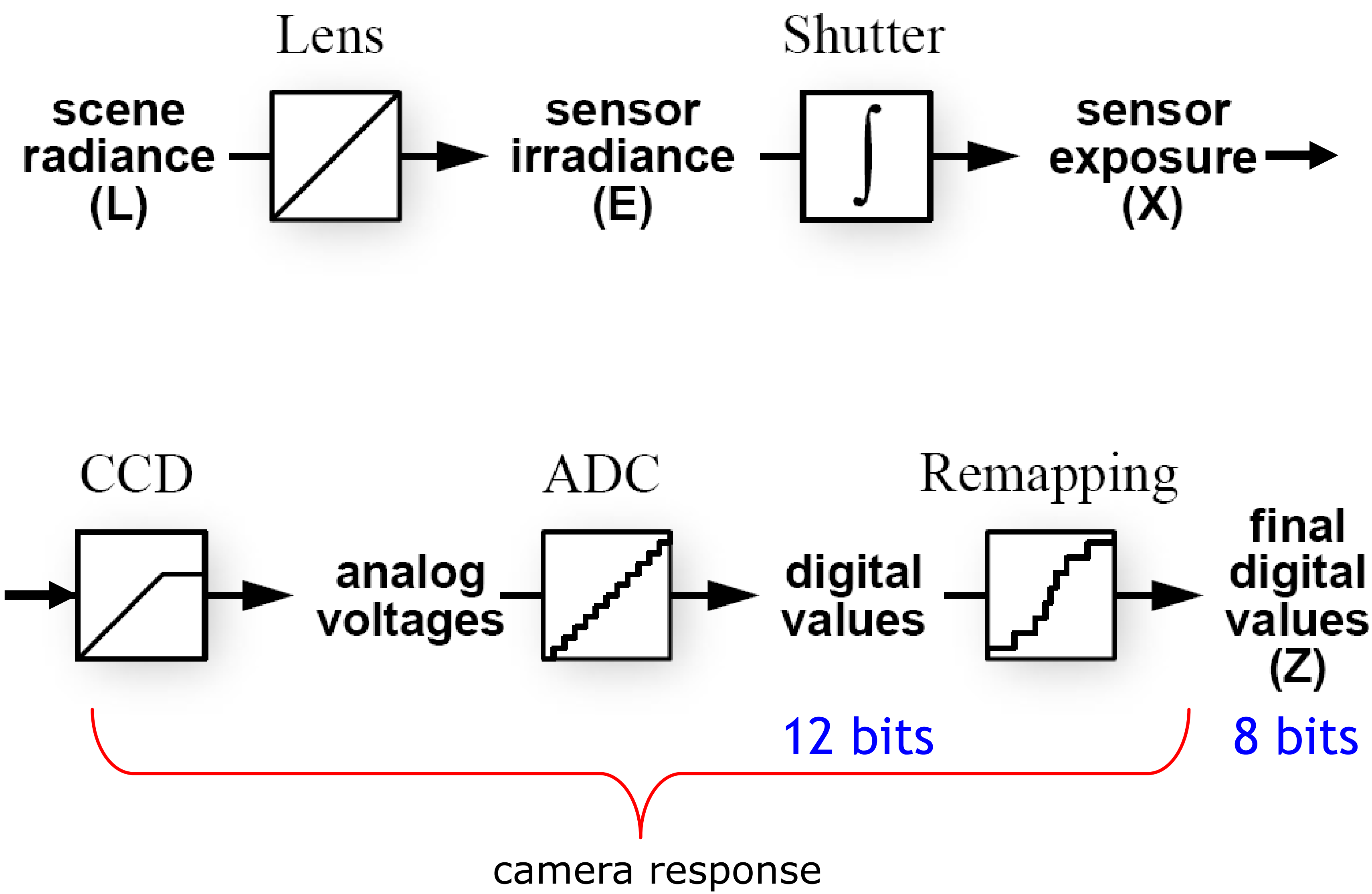
$$E_i = \int_{\Omega} L(i,\omega_i)\,d\omega$$

$$X_i = \int_{t=0}^{\Delta t} E_i\,dt = E_i \Delta t$$

Camera pipeline

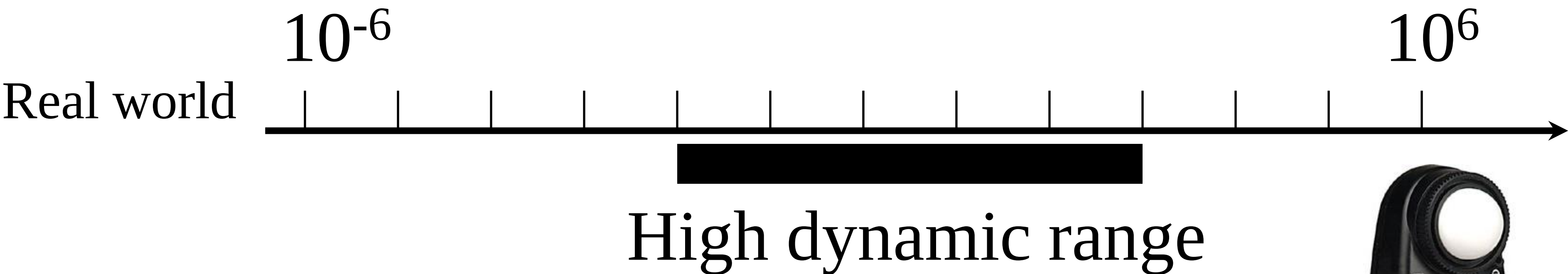


Camera pipeline



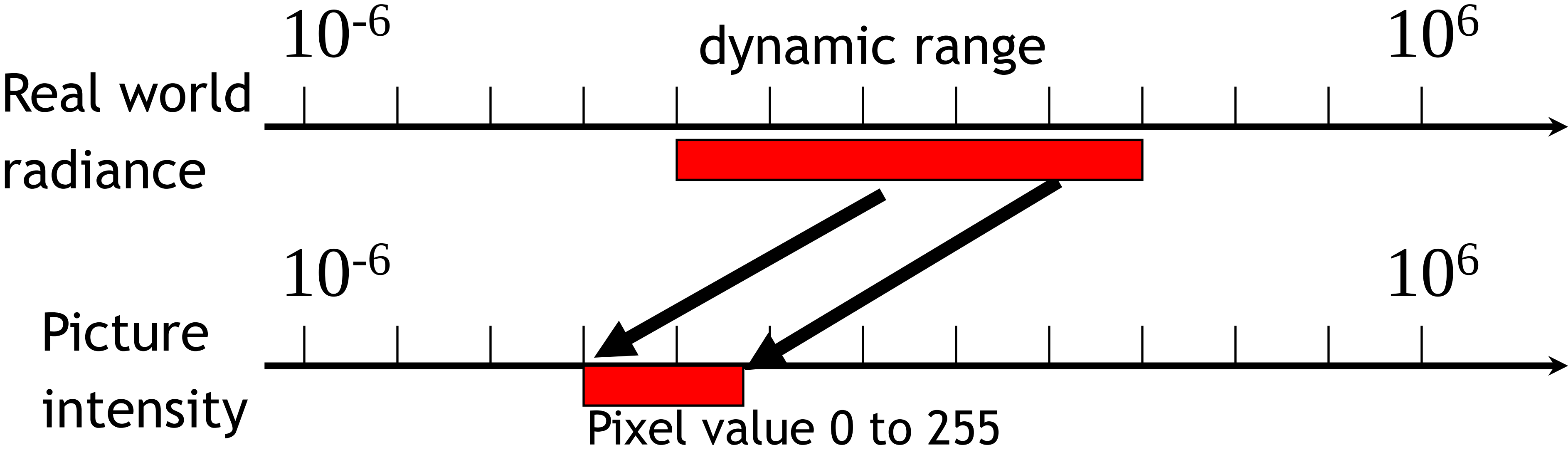
Real world dynamic range

- Eye can adapt from ~ 10^{-6} to 10^{6} cd/m^2
- Often 1 : 100,000 in a scene
- Typical 1:50, max 1:500 for pictures

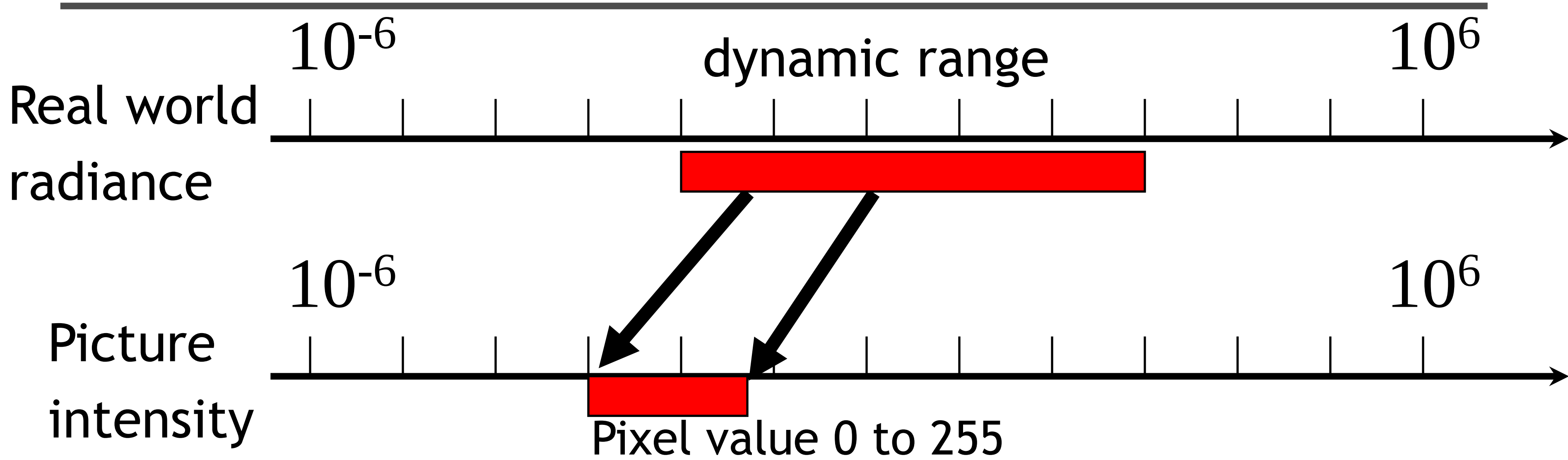


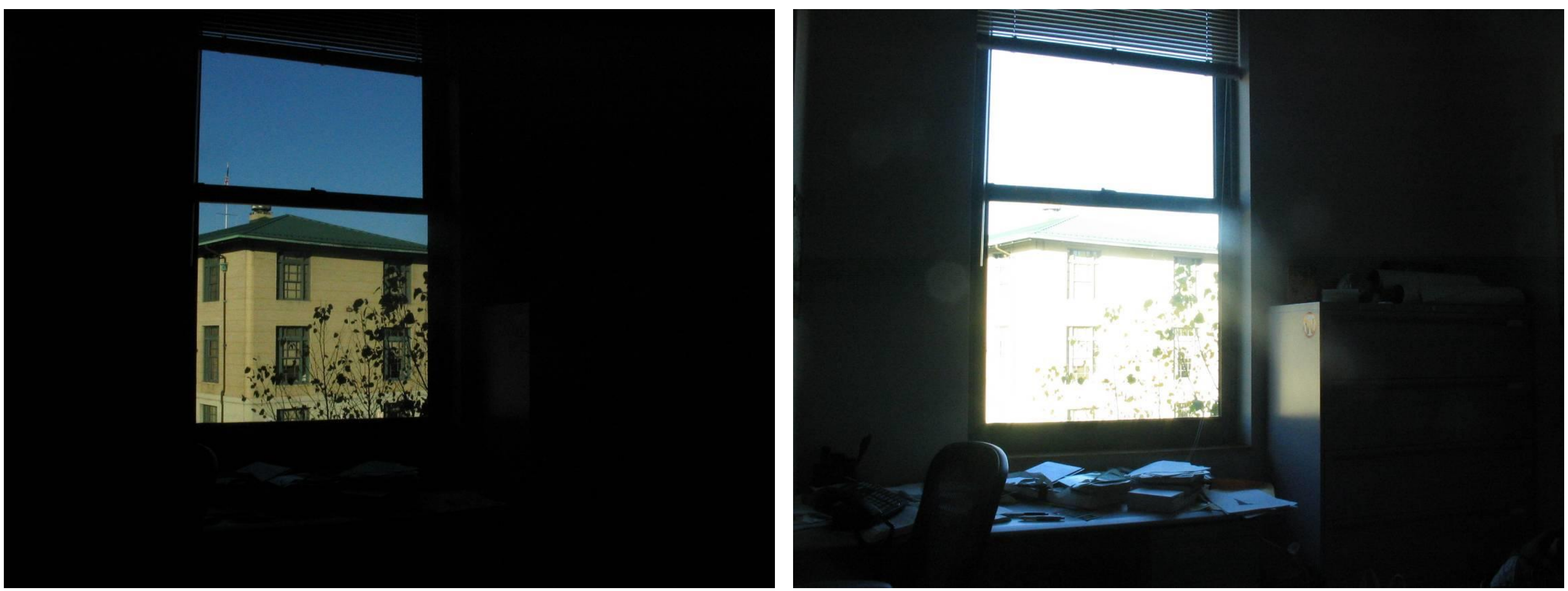
spotmeter

Short exposure



Long exposure





Camera is not a photometer

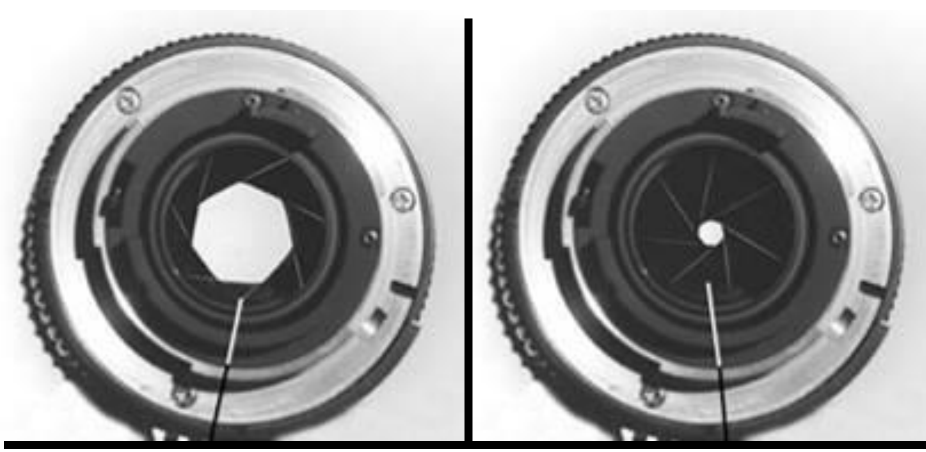
- Limited dynamic range

  $\Rightarrow$ Perhaps use multiple exposures?
- Unknown, nonlinear response

  $\Rightarrow$ Not possible to convert pixel values to radiance
- Solution:
  - Recover response curve from multiple exposures, then reconstruct the ***radiance map***

Varying exposure

- Ways to change exposure
  - Shutter speed
  - Aperture
  - Neutral density filters



Shutter speed

- Note: shutter times usually obey a power series – each "stop" is a factor of 2

¼, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Varying shutter speeds

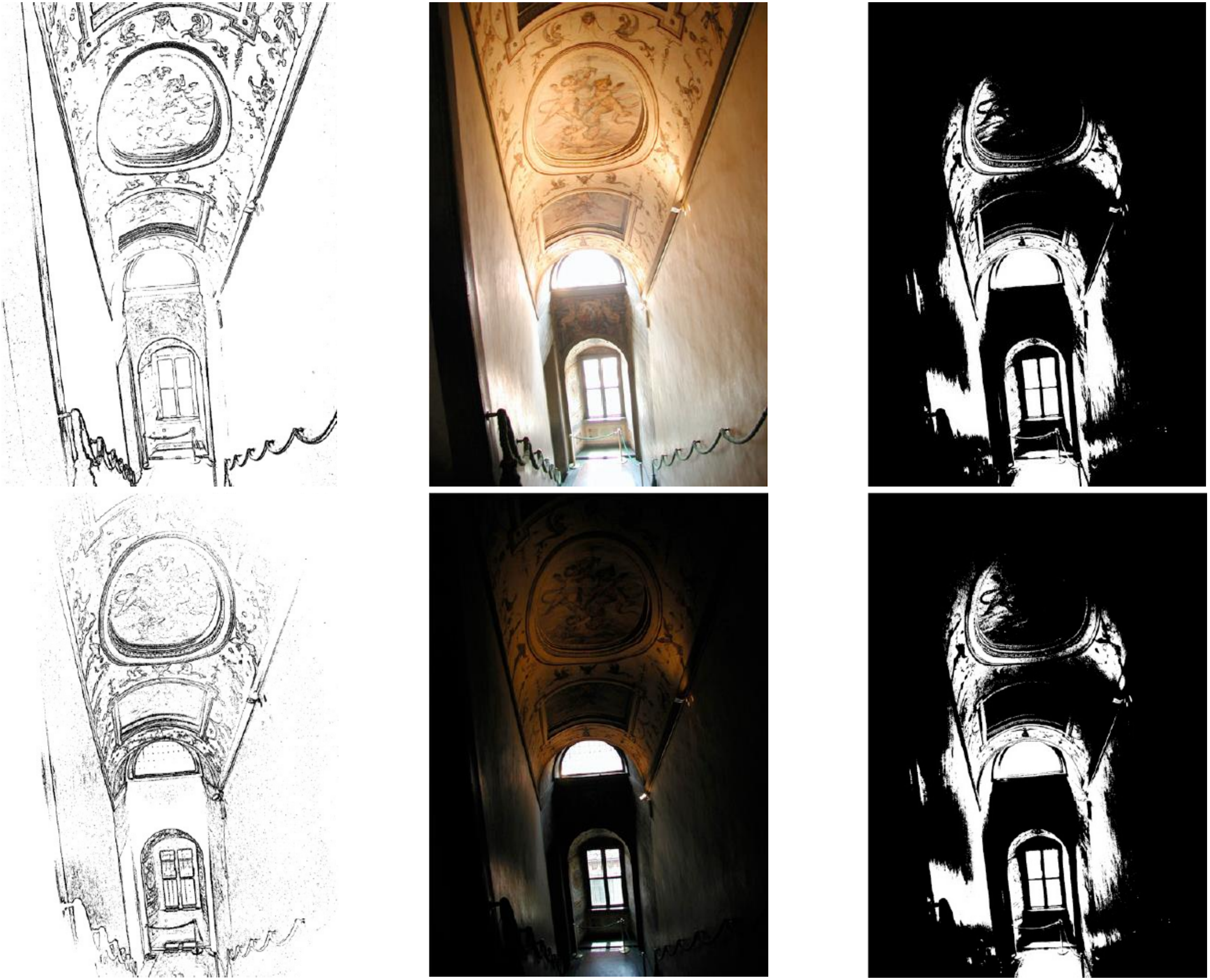
HDRI capturing from multiple exposures

- Capture images with multiple exposures
- Image alignment (even if you use tripod, it is suggested to run alignment)
- Response curve recovery
- Ghost/flare removal

Image alignment

- We will introduce a fast and easy-to-implement method for this task, called Median Threshold Bitmap (MTB) alignment technique.
- Consider only integral translations. It is enough empirically.
- The inputs are N grayscale images. (You can either use the green channel or convert into grayscale by Y=(54R+183G+19B)/256)
- MTB is a binary image formed by thresholding the input image using the median of intensities.

Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures .
Greg Ward. Journal of Graphics Tools



Edge map | Input | MTB

Why is MTB better than gradient?

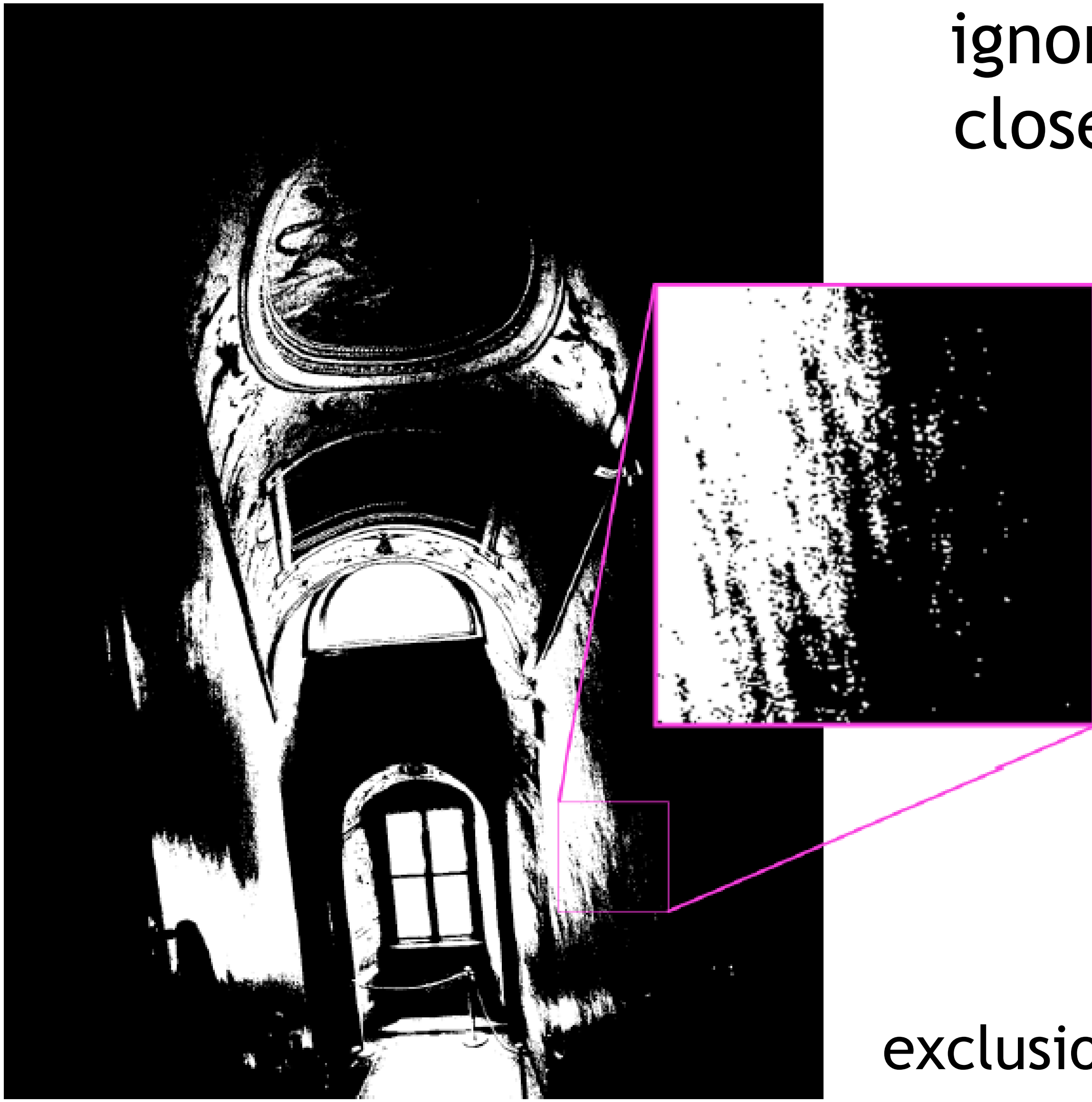
- Edge-detection filters are dependent on image exposures
- Taking the difference of two edge bitmaps would not give a good indication of where the edges are misaligned.

Search for the optimal offset

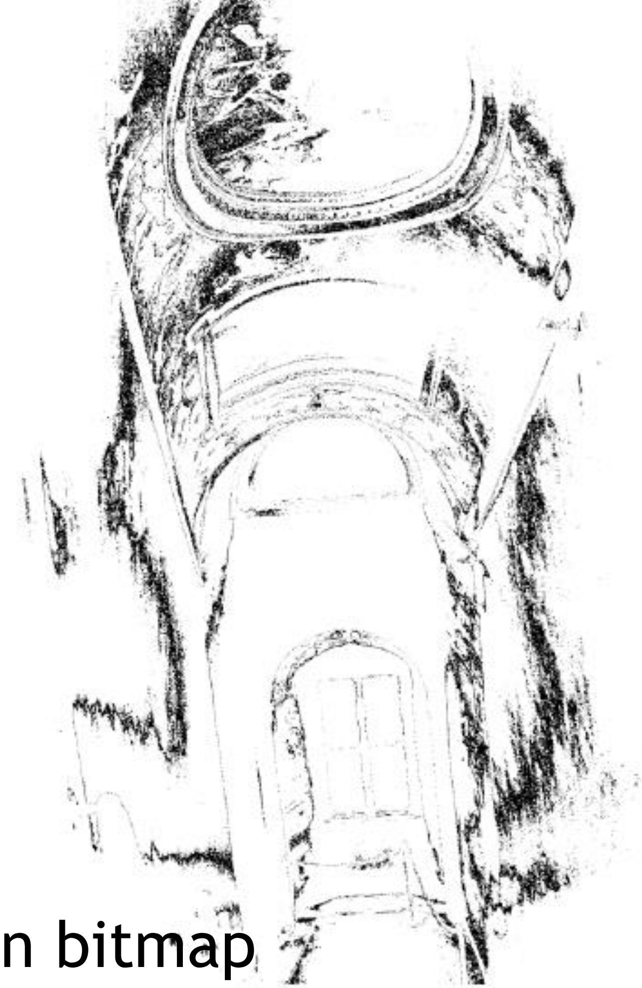
- Try all possible offsets.
- Gradient descent
- Multiscale technique

- log(max_offset) levels
- Try 9 possibilities for the top level
- Scale by 2 when passing down; try its 9 neighbors

Threshold noise



ignore pixels that are close to the threshold



exclusion bitmap

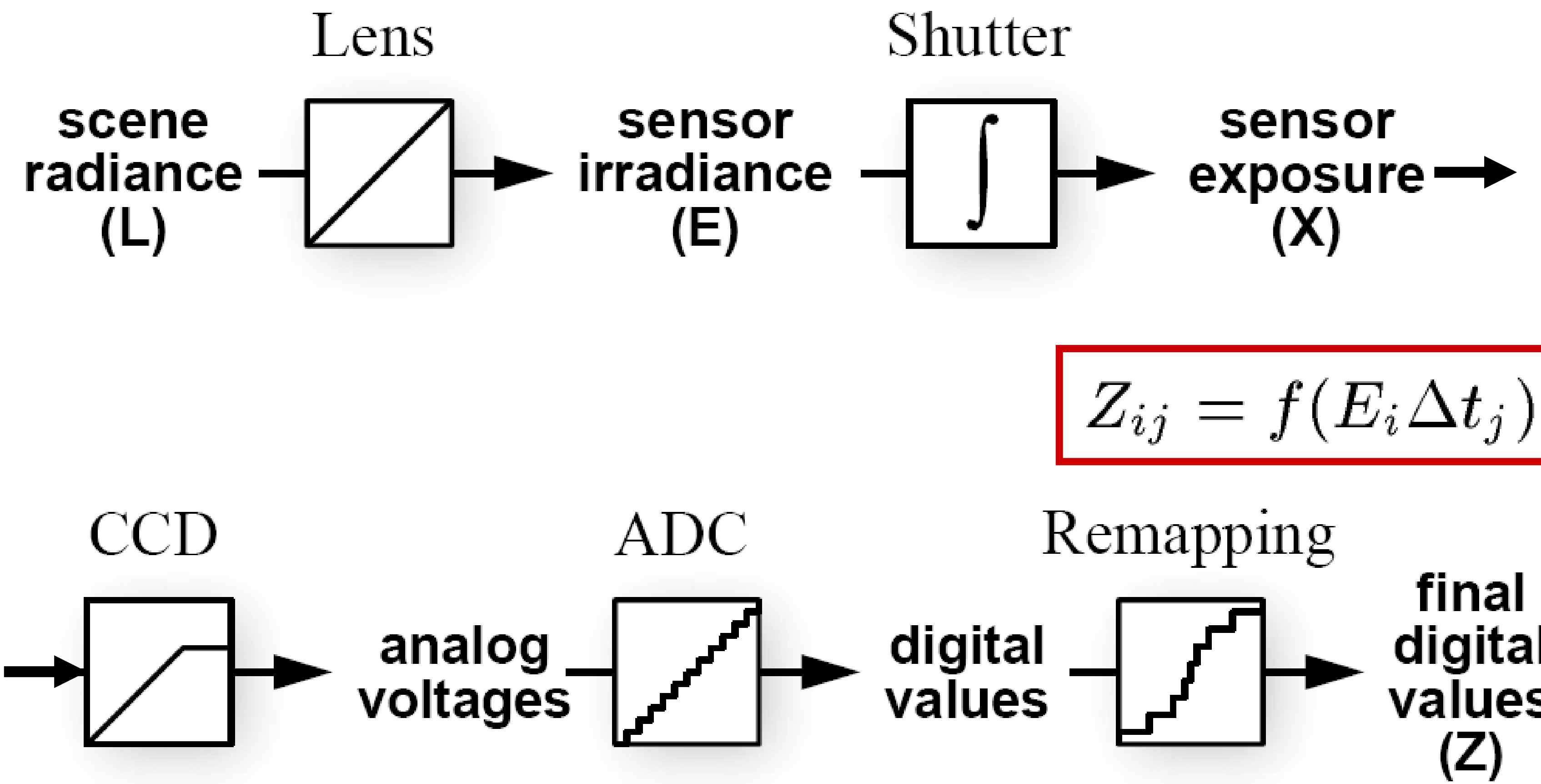
Results

Success rate = 84%. 10% failure due to rotation. 3% for excessive motion and 3% for too much high-frequency content.

Unaligned HDR

Aligned HDR

Recovering response curve



$$Z_{ij} = f(E_i \Delta t_j)$$

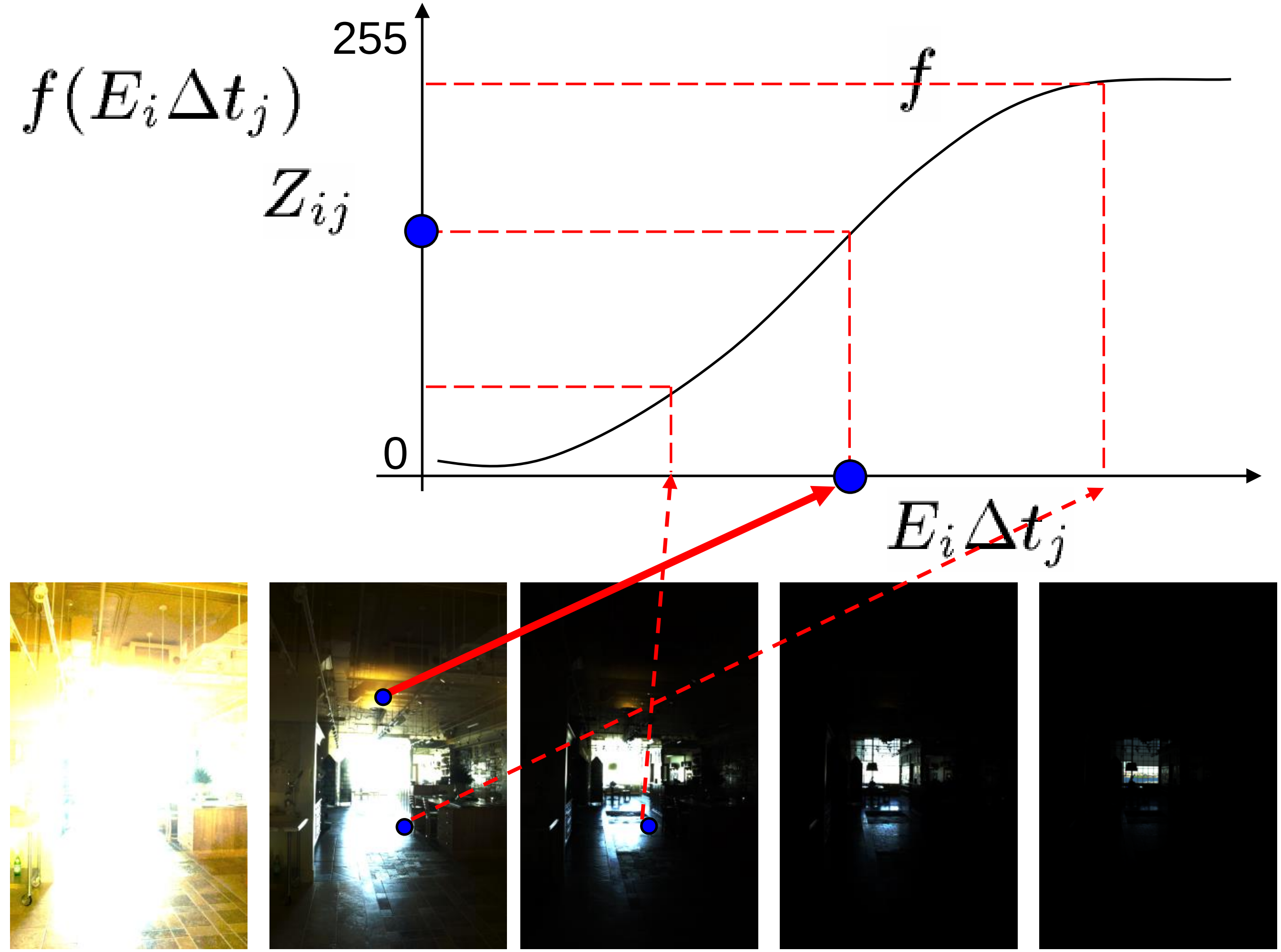
12 bits

8 bits

Recovering response curve

- We want to obtain the inverse of the response curve

$$Z_{ij} = f(E_i \Delta t_j)$$

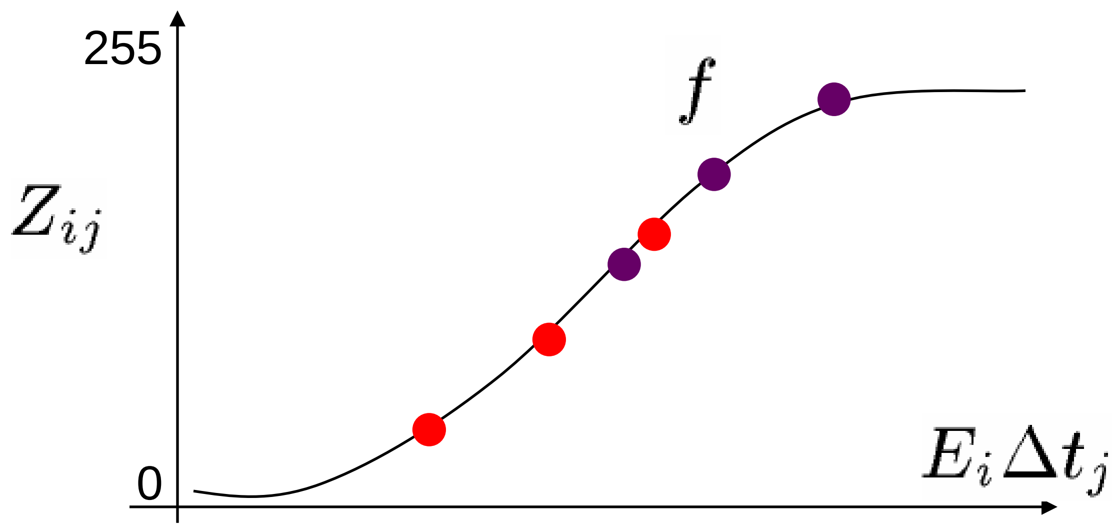


Recovering response curve

Image series

Δt = 2 sec	Δt = 1 sec	Δt = 1/2 sec	Δt = 1/4 sec	Δt = 1/8 sec

$$Z_{ij} = f(E_i \Delta t_j)$$



Recovering response curve

Image series

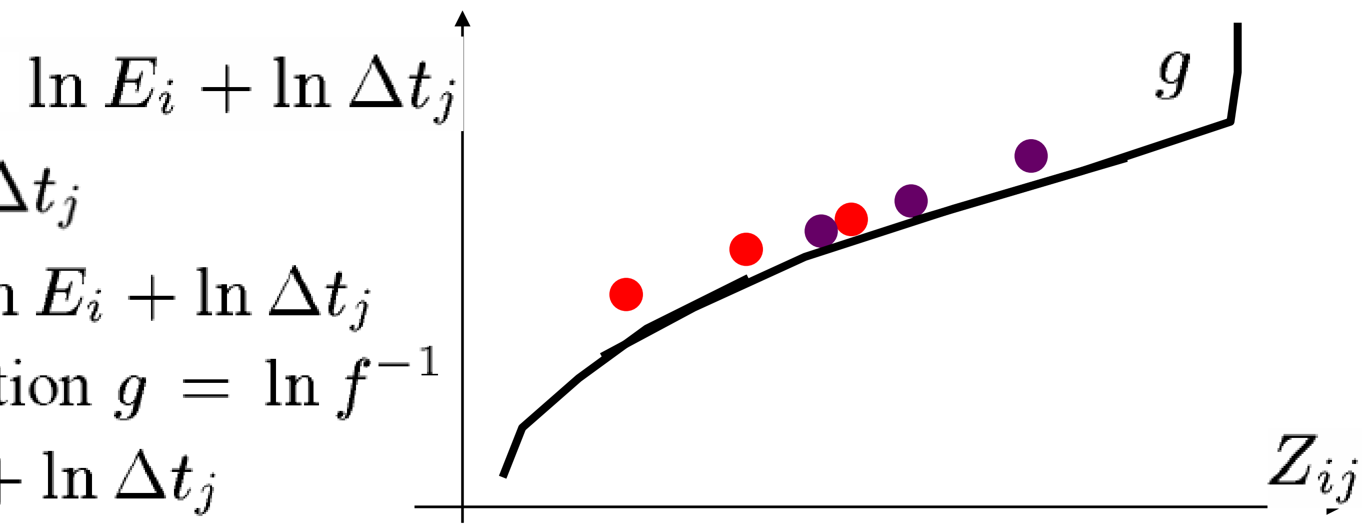
Δt = 2 sec	**Δt = 1 sec**	**Δt = 1/2 sec**	**Δt = 1/4 sec**	**Δt = 1/8 sec**

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

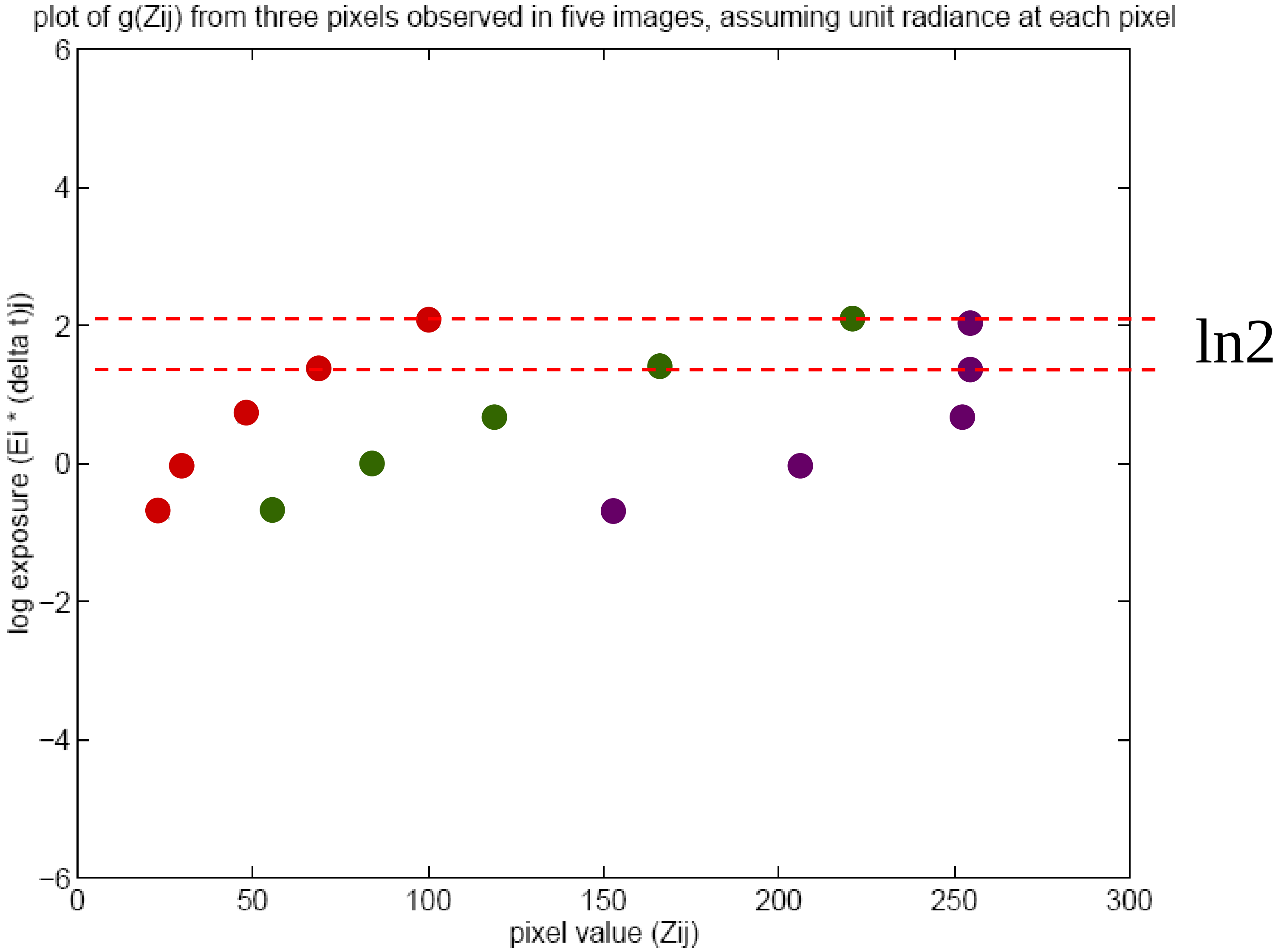
$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

let us define function $g = \ln f^{-1}$

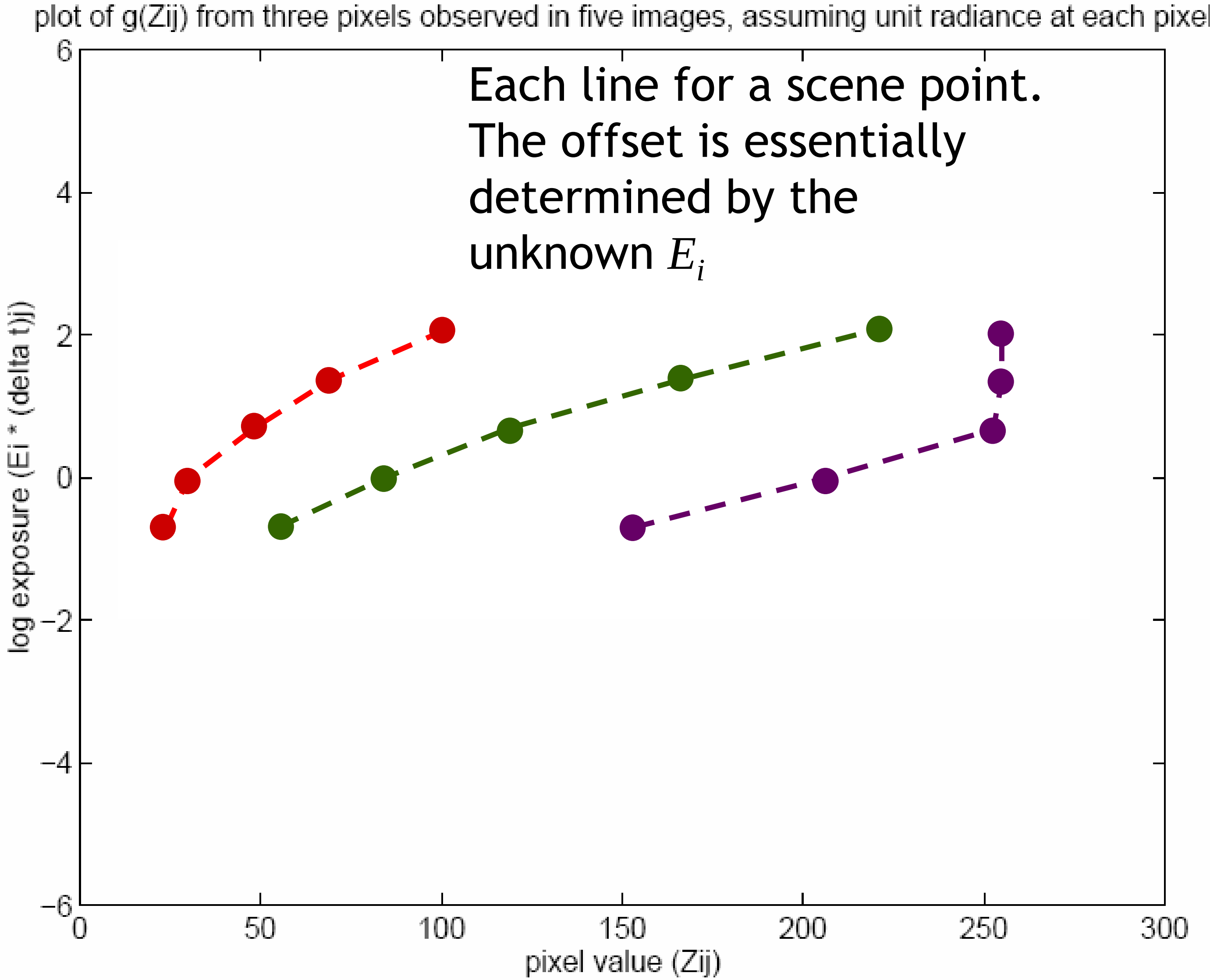
$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$



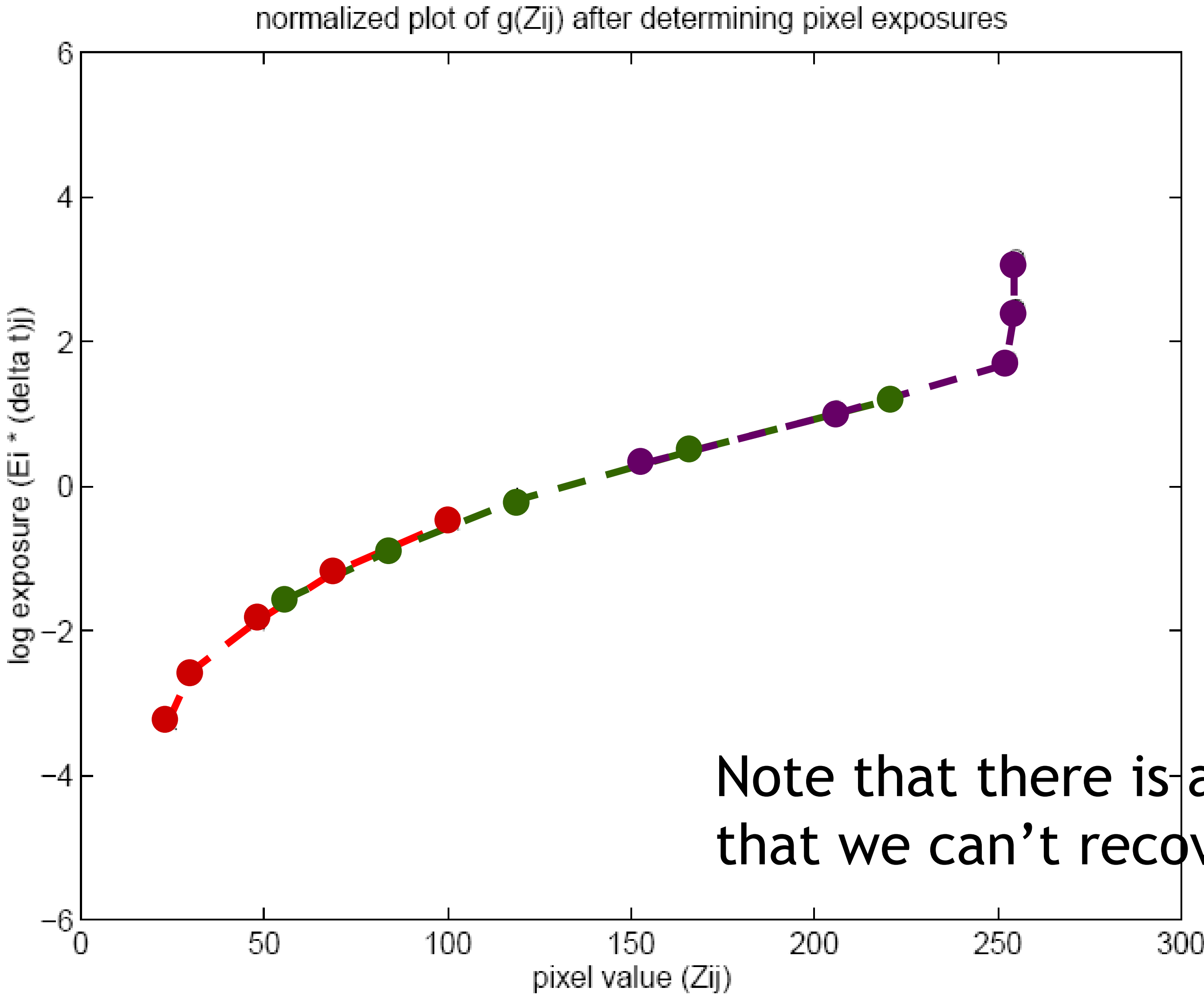
Idea behind the math



Idea behind the math



Idea behind the math



Note that there is a shift that we can't recover

Basic idea

- Design an objective function
- Optimize it

Math for recovering response curve

DigiVFX

$$Z_{ij} = f(E_i \Delta t_j)$$

f is monotonic, it is invertible

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

let us define function $g = \ln f^{-1}$

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

minimize the following

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2$$

$$g''(z) = g(z-1) - 2g(z) + g(z+1)$$

Recovering response curve

DigiVFX

- The solution can be only up to a scale, add a constraint

$$g(Z_{mid}) = 0, \text{ where } Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max})$$

- Add a hat weighting function

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij}) [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 +$$

$$\lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z) g''(z)]^2$$

Recovering response curve

- We want $N(P-1) > (Z_{max} - Z_{min})$

  If P=11, N~25 (typically 50 is used)
- We prefer that selected pixels are well distributed and sampled from constant regions. They picked points by hand.
- It is an overdetermined system of linear equations and can be solved using SVD

How to optimize?

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \left\{ w(Z_{ij}) \left[g(Z_{ij}) - \ln E_i - \ln \Delta t_j \right] \right\}^2 +$$

$$\lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} \left[w(z) g''(z) \right]^2$$

1. Set partial derivatives zero
2.

$$\min \sum_{i=1}^{N} (\mathbf{a_i x} - \mathbf{b_i})^2 \rightarrow \text{least-square solution of } \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \\ \vdots \\ \mathbf{a_N} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \vdots \\ \mathbf{b_N} \end{bmatrix}$$

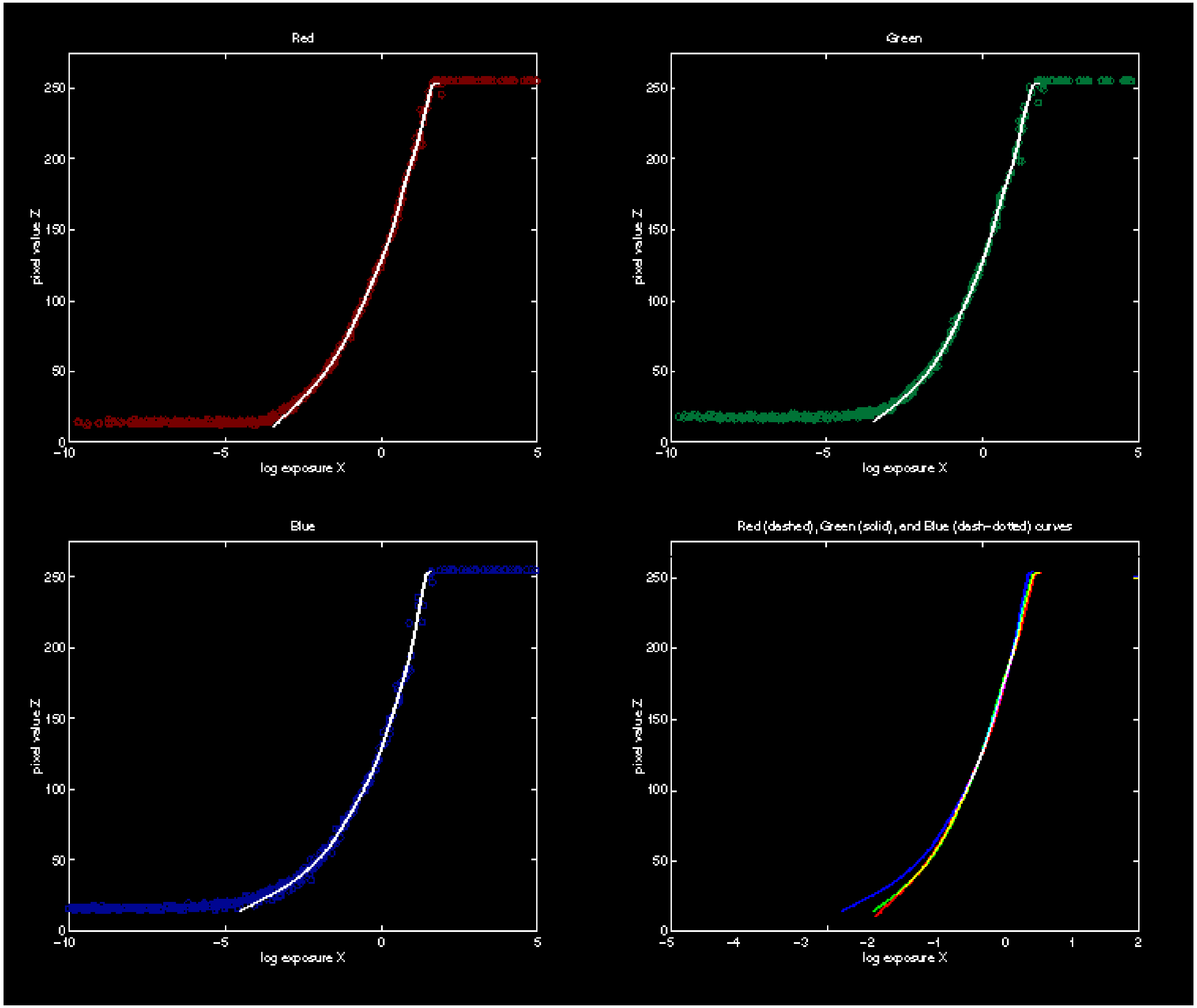
Matlab code

```
%
% gsolve.m - Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system's response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
%
%  Zmin = 0
%  Zmax = 255
%
% Arguments:
%
%  Z(i,j) is the pixel values of pixel location number i in image j
%  B(j)   is the log delta t, or log shutter speed, for image j
%  l      is lamdba, the constant that determines the amount of smoothness
%  w(z)   is the weighting function value for pixel value z
%
% Returns:
%
%  g(z)   is the log exposure corresponding to pixel value z
%  lE(i)  is the log film irradiance at pixel location i
%
```

Matlab code

```
function [g,lE]=gsolve(Z,B,l,w)

n = 256;
A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

k = 1;                  %% Include the data-fitting equations
for i=1:size(Z,1)
  for j=1:size(Z,2)
    wij = w(Z(i,j)+1);
    A(k,Z(i,j)+1) = wij; A(k,n+i) = -wij; b(k,1) = wij * B(  j);
    k=k+1;
  end
end

A(k,129) = 1;           %% Fix the curve by setting its middle value to 0
k=k+1;

for i=1:n-2             %% Include the smoothness equations
  A(k,i)=l*w(i+1); A(k,i+1)=-2*l*w(i+1); A(k,i+2)=l*w(i+1);
  k=k+1;
end

x = A\b;                %% Solve the system using SVD

g = x(1:n);
lE = x(n+1:size(x,1));
```

Recovered response function



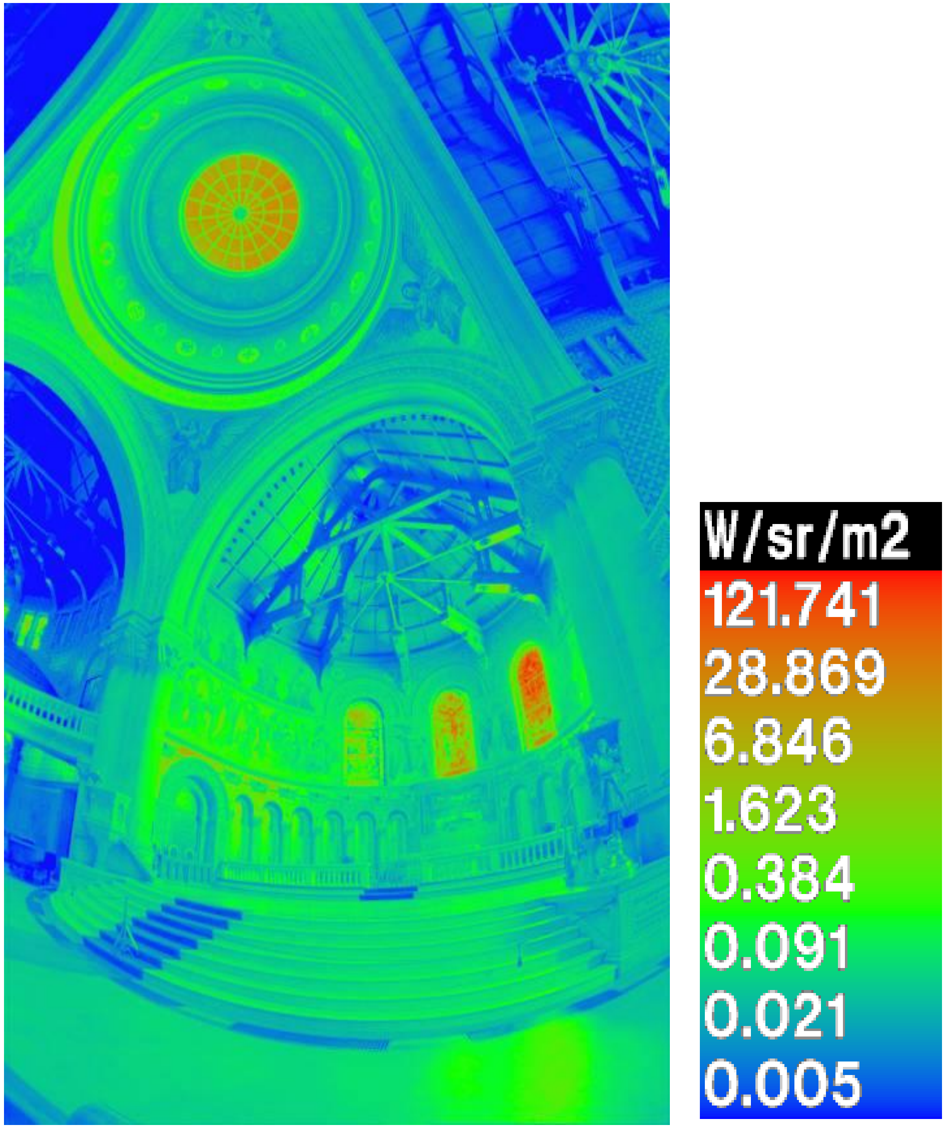
Constructing HDR radiance map

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

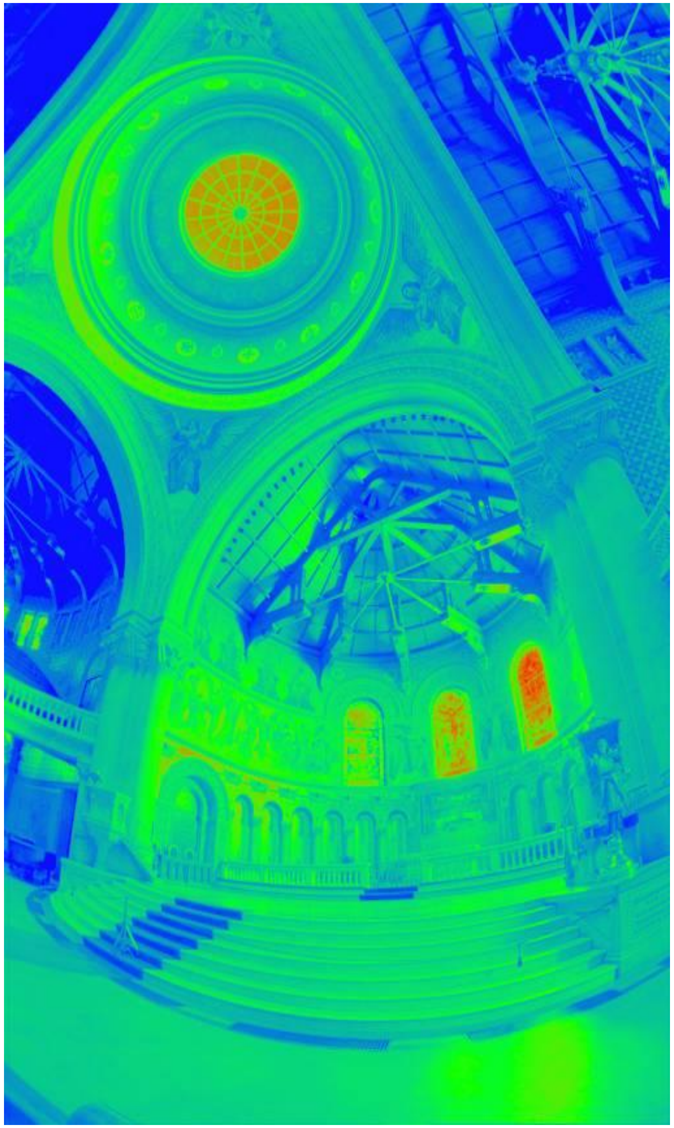
combine pixels to reduce noise and obtain a more reliable estimation

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}$$

Reconstructed radiance map



What is this for?



- Human perception
- Vision/graphics applications

Tone Mapping

Reprint from [Debevec and Malik 97]

Student paper presentation

Robust color-to-gray via nonlinear global mapping

Yongjin Kim, Cheolhun Jang

Julien Demouth, and Seungyong Lee

SIGGRAPH ASIA 2009

Presenter: Chen, Ray

Next Time

- Panorama
- Student paper presentations
  - 04/21: Nguyen, Henry
    - Colorization Using Optimization
      A. Levin, D. Lischinski, and Y. Weiss
      SIGGRAPH 2004
  - 04/21: Gerendasy, Daniel R.
    - Color harmonization
      D. Cohen-Or, O. Sorkine, R. Gal, T. Leyv, and, Y. Xu
      ACM SIGGRAPH 2006