High Dynamic Range Imaging

Introduction to Computational Photography:

EECS 395/495

Northwestern University

High Dynamic Range Imaging

How to extend the range of brightness values that can be represented by your digital camera

Topics:

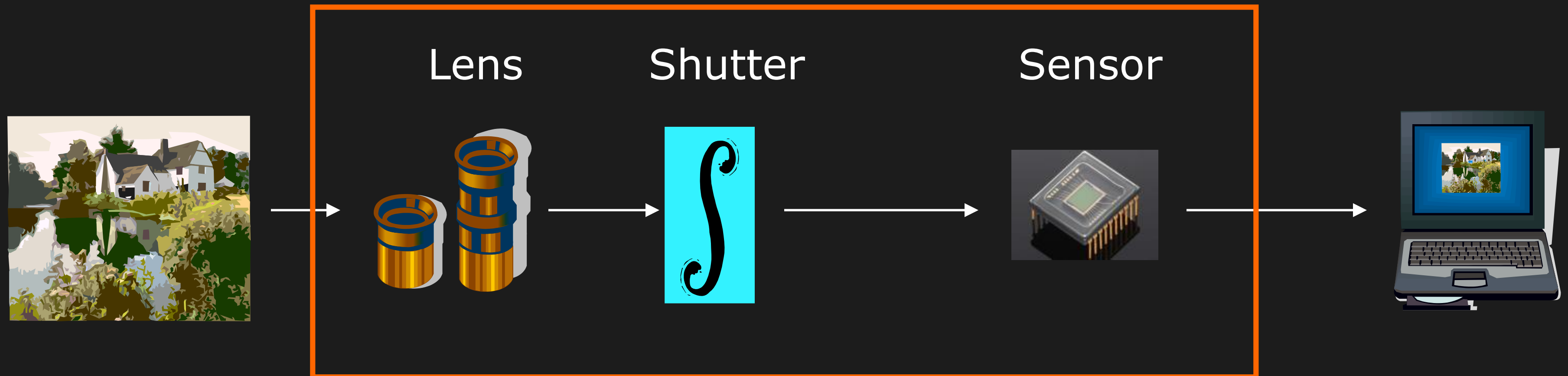
(1) Camera Response Function

(2) Recovering High Dynamic Range Images

(3) Tone-Mapping

(4) Assorted Pixel Camera

From Radiance to Intensity

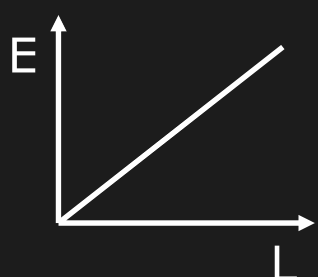


L: Scene Radiance watts/sr/m^2

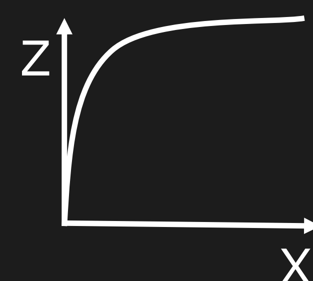
E: Image Irradiance watts/m^2

X: Exposure Value (EV) J/m^2

Z: Final Digital Image 8bit Intensity



$$\boxed{X = E \cdot \Delta t}$$

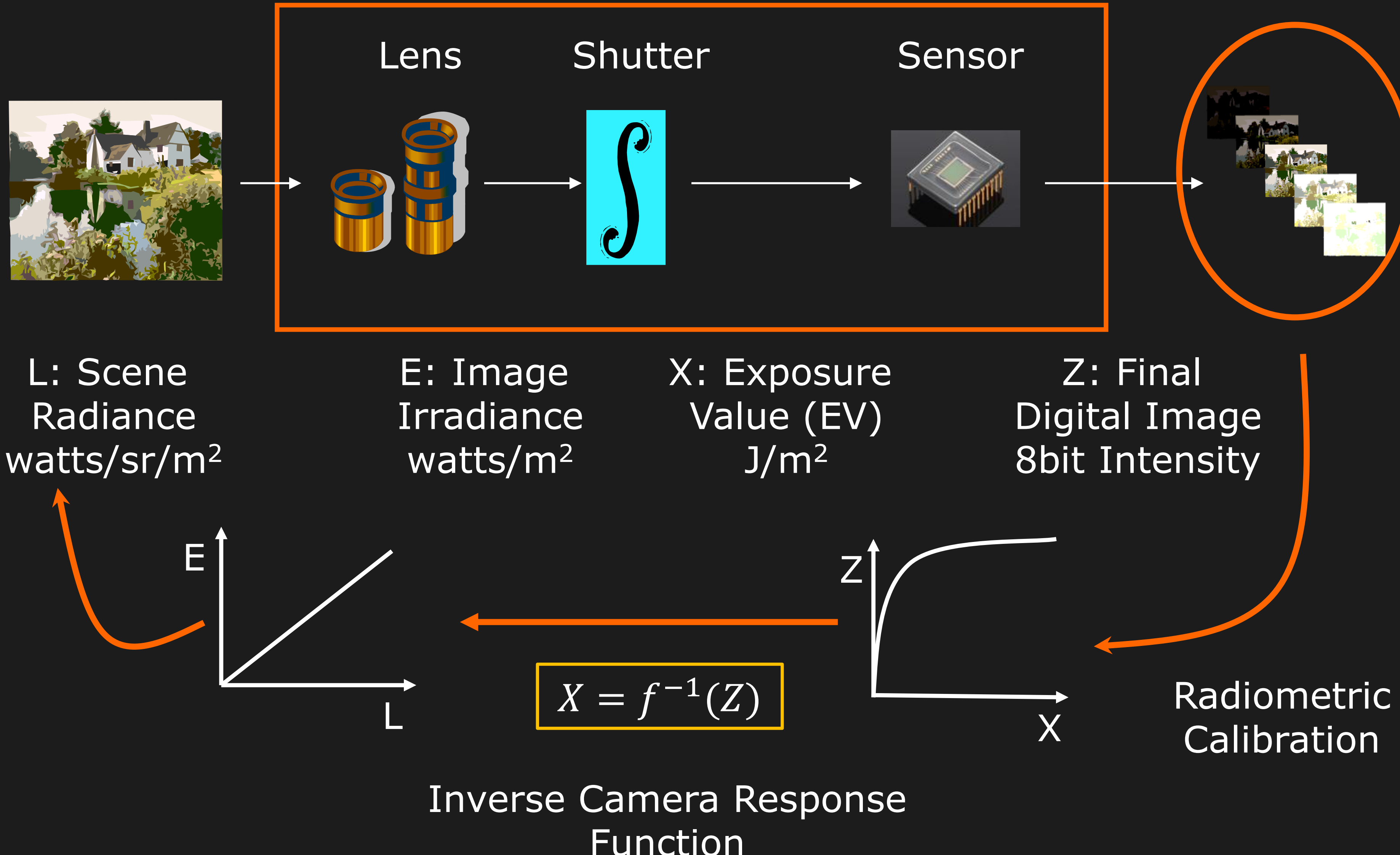


Camera Response Function

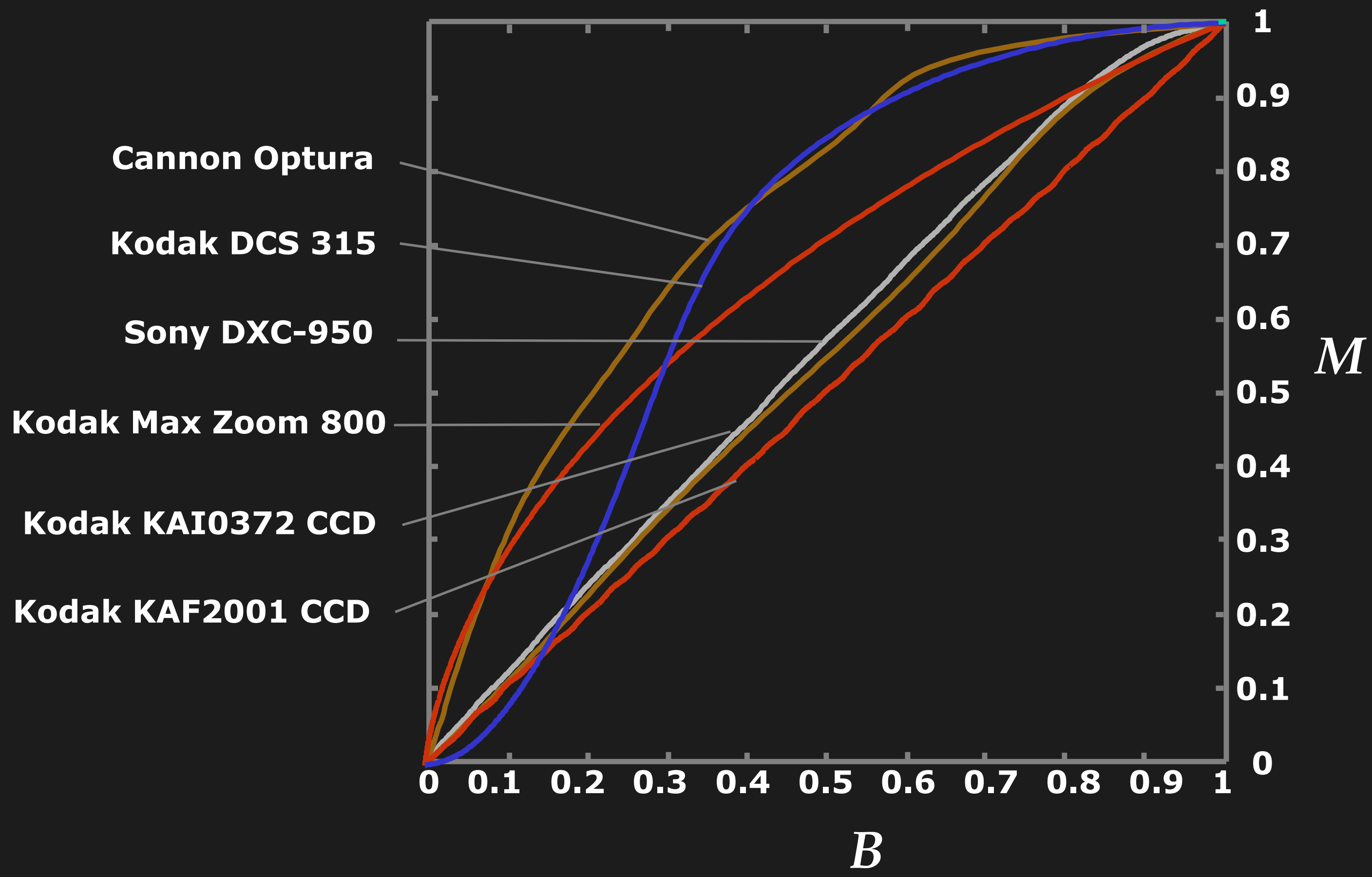
$$\boxed{E = L \cdot \frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4 \alpha}$$

$$\boxed{Z = f(X)}$$

From Intensity to Radiance



Camera Response Function $f(\cdot)$



"Gamma Curves"

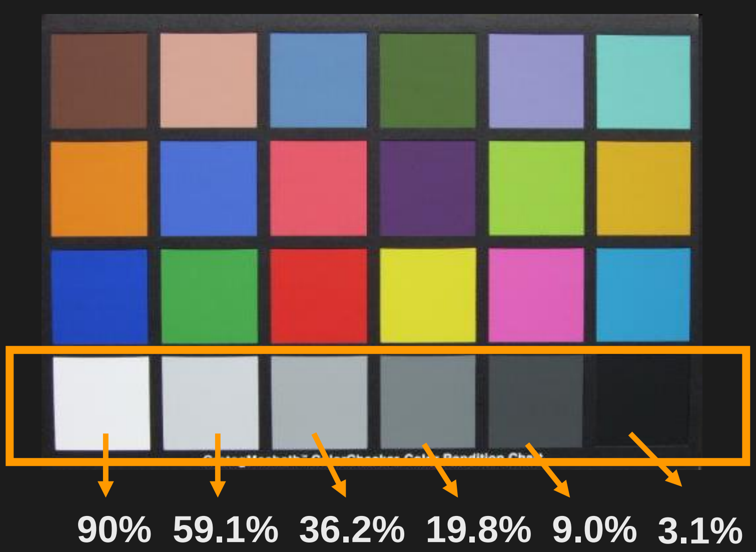
[Grossberg 2003]

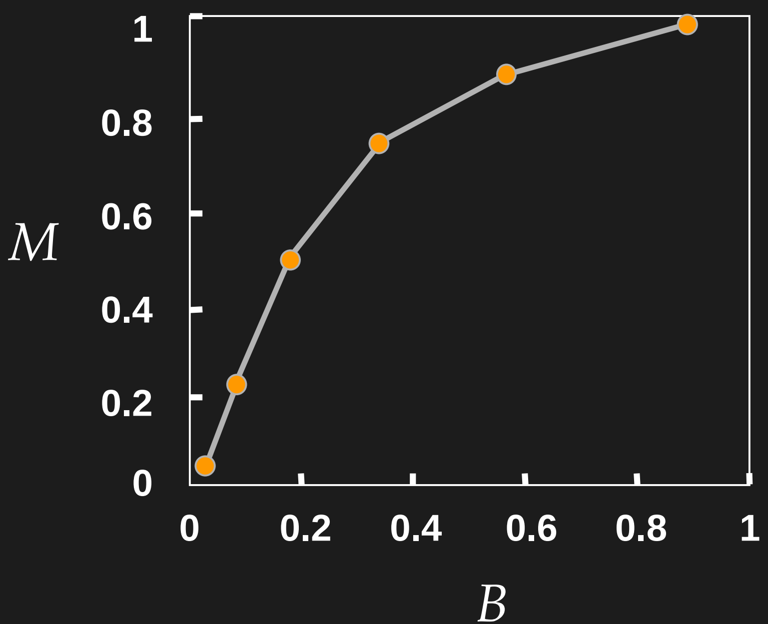
Radiometric Calibration: Finding $f(\cdot)$

Calibration using a chart:

1. Patches with known reflectance (when uniformly lit)
2. Fit linear segments or curve

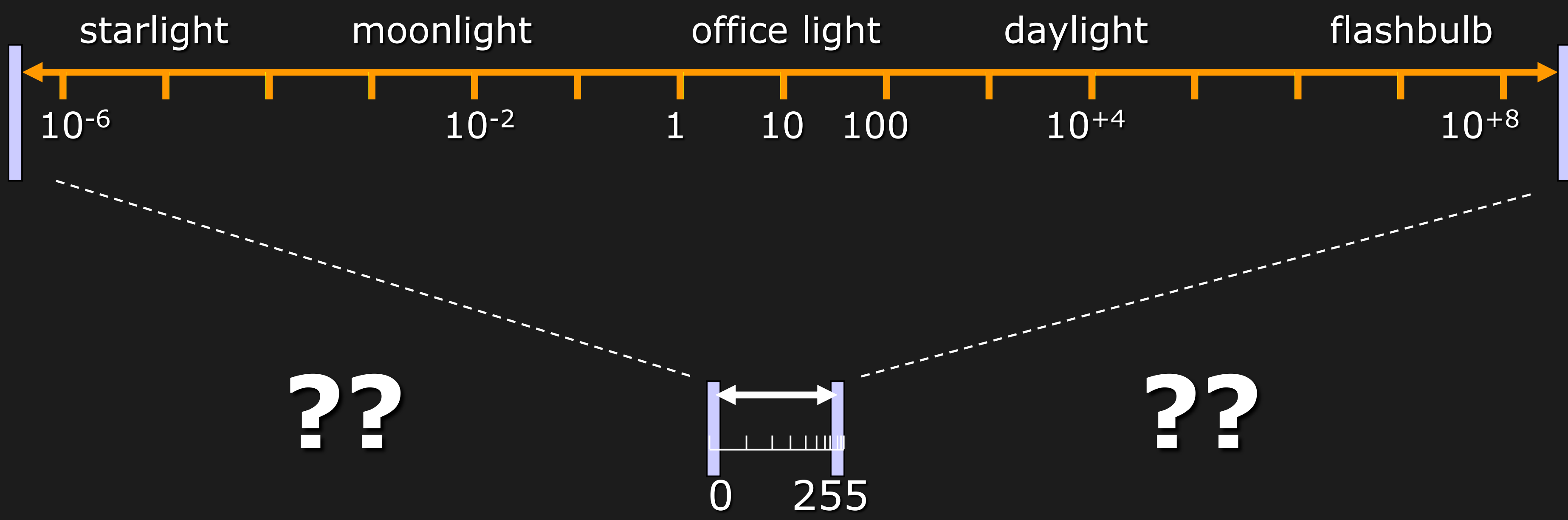
Macbeth Chart





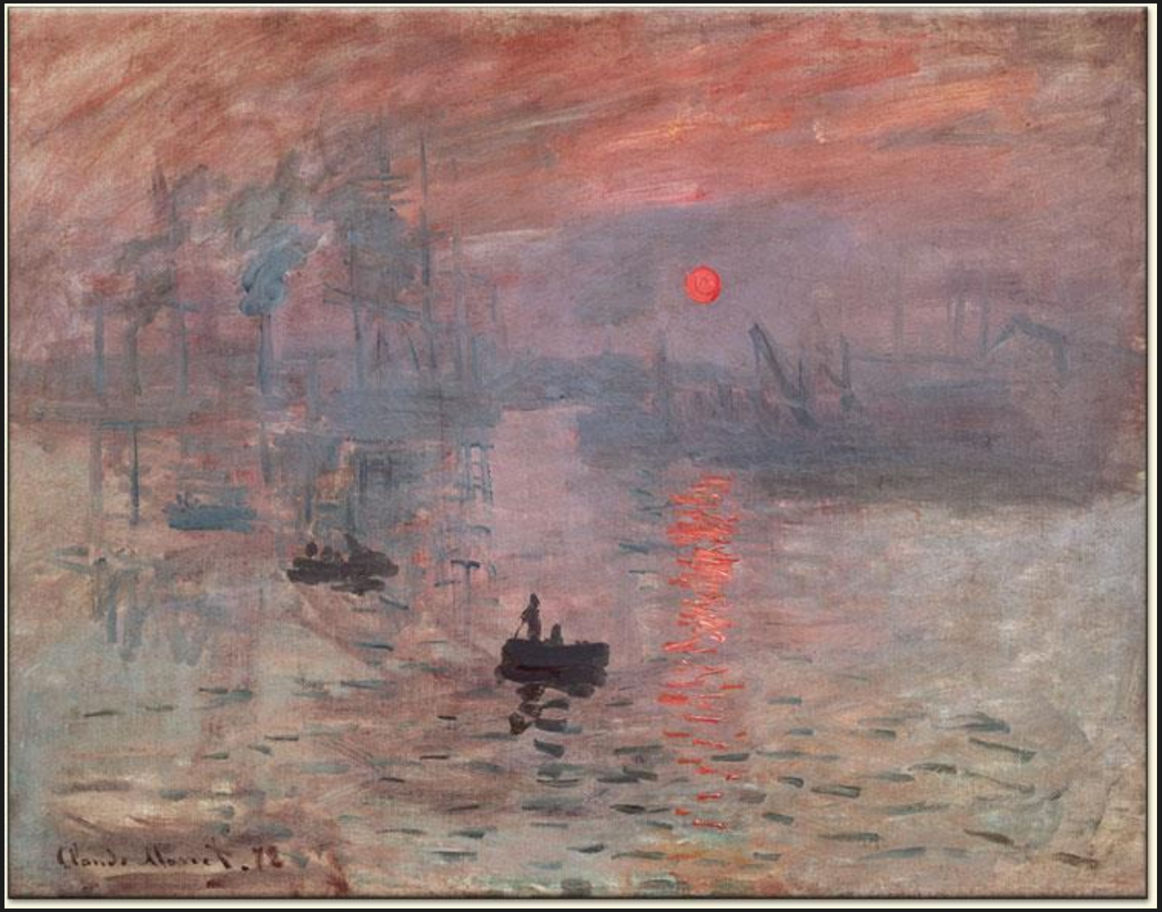
Dynamic Range of Imaging

Domain of Human Vision:
from ~10^{-6} to ~10^{+8} cd/m^2



Range of Typical Displays:
from ~1 to ~100 cd/m^2

Local Contrast and Perceived Higher Dynamic Range



Impression Sunrise, Claude Monet, 1873
http://webexhibits.org/colorart/monet.html

High Dynamic Range: Multiple Exposures

Assume Camera Response $f(\cdot)$ is Linear

with t_0

t_1

t_2

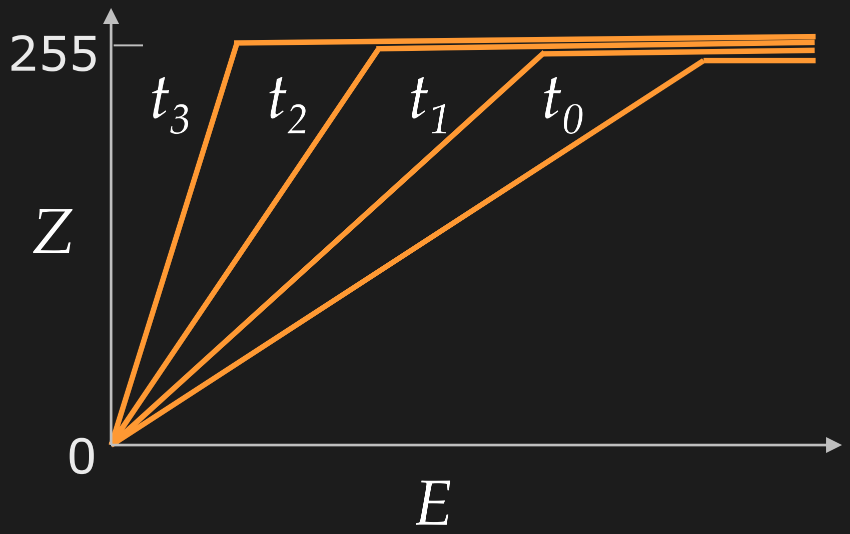
t_3

$$Z_0 = \min(t_0 \cdot \mathrm{E}, 255)$$

$$Z_1 = \min(t_1 \cdot \mathrm{E}, 255)$$

$$Z_2 = \min(t_2 \cdot \mathrm{E}, 255)$$

$$Z_3 = \min(t_3 \cdot \mathrm{E}, 255)$$

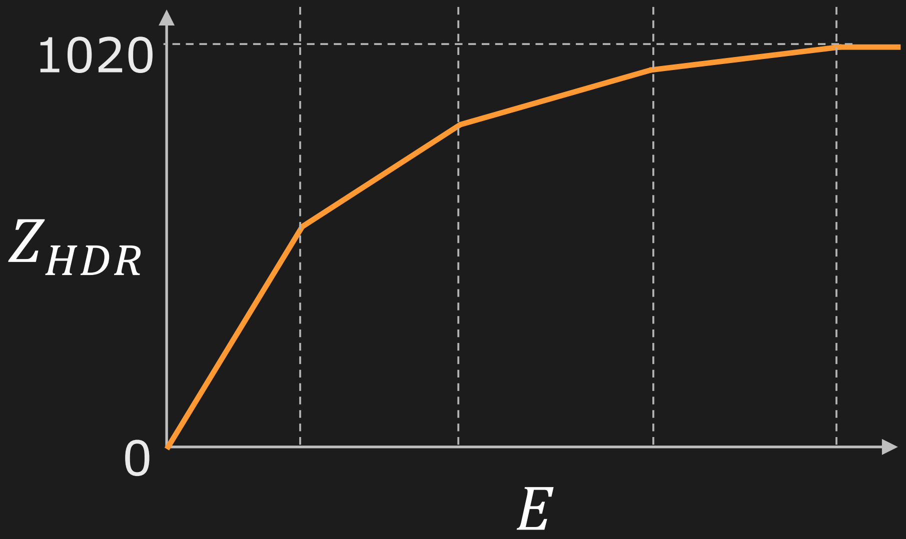


[Nayar 2000]

High Dynamic Range: Multiple Exposures

Aggregate Image: $Z_{HDR} = Z_0 + Z_1 + Z_2 + Z_3$

Camera Response $f(\cdot)$ for Aggregate Image:



[Nayar 2000]

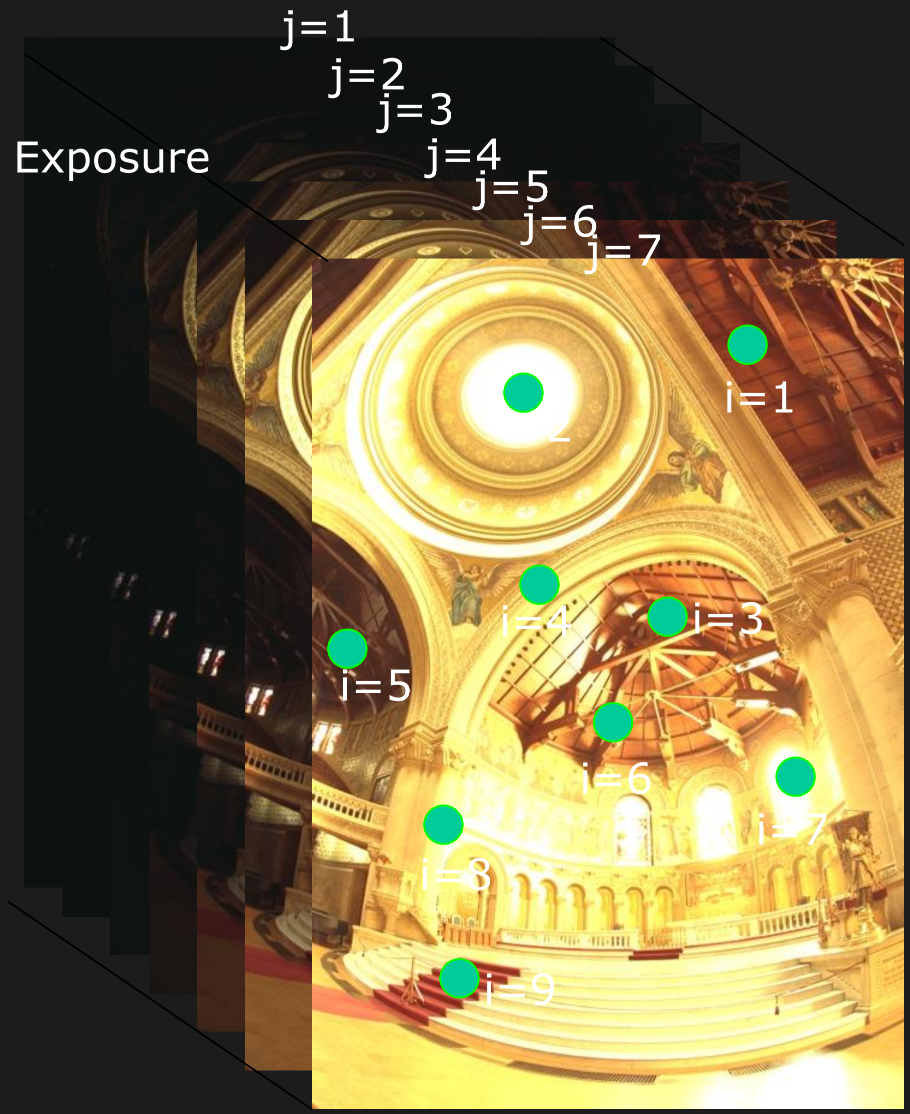
HDR Imaging without a ColorChecker

- Input: multiple images of the same scene with different exposures

- Assumptions:
  - Lighting changes can be ignored.
  - Images should be registered well.
  - Camera response is the same for all pixels.
  - Working in the middle range of the response curve, where *reciprocity* will not fail.
  - Smooth, monotonic camera response curve

[Debevec, et al., 1997]

Proposed Algorithm



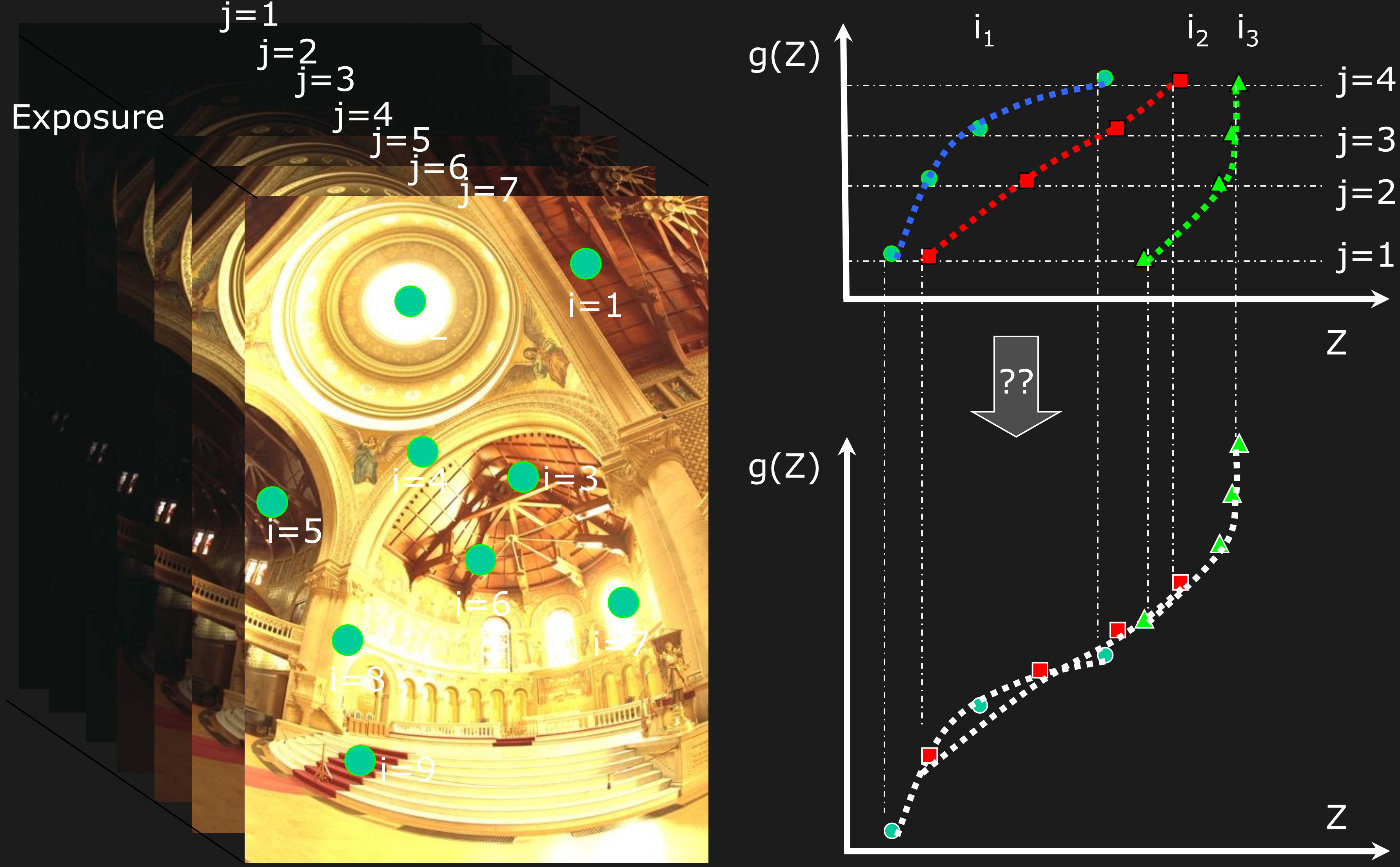
- Z_{ij}: image intenstiy of the i-th point in the j-th image.
- E_i: irradiance of the i-th point
- Δt_j: exposure time of the j-th image

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

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

Proposed Algorithm

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



Proposed Algorithm

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

- g(Z) is unknown, but only discrete values are enough under smooth assumption.
- N*P equations, N+Zmax-Zmin+1 variables
- Thus, g(Z) and lnE can be solved by minimizing:

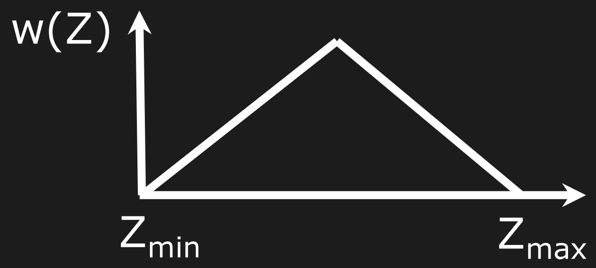
$$\sum_{i=1}^{N}\sum_{j=1}^{P}\left[g(Z_{ij}) - \ln E_i - \ln \Delta t_j\right]^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1}\left[g(z-1) - 2g(z) + g(z+1)\right]^2$$

Least Square Error — Smoothness of g(Z)

Linear Least Square Problem
which can be robustly solved by SVD

Proposed Algorithm

- g(Z) and E are solved up to a scale, but relative E are good enough for many applications.
- Weighting function is used to emphasize the central working range.



Optimization: Linear Least Square

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

1. Set partial derivatives zero
2.

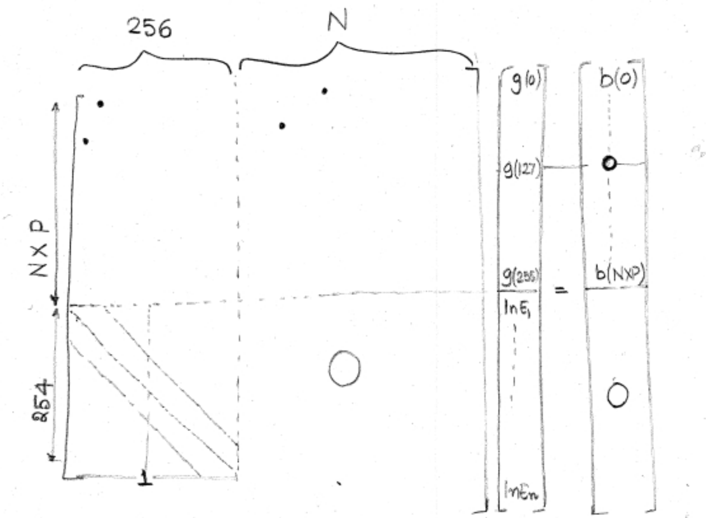
$$\min \sum_{i=1}^{M} (\mathbf{a_i x} - \mathbf{b_i})^2 \rightarrow \text{least-square solution of} \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \\ \vdots \\ \mathbf{a_N} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \vdots \\ \mathbf{b_N} \end{bmatrix}$$

Optimization: Linear Least Square

$$\sum_{i=1}^{N}\sum_{j=1}^{P}\left\{w(Z_{ij})\left[g(Z_{ij})-\ln E_i-\ln\Delta t_j\right]\right\}^2 + \lambda\sum_{Z=Z_{min+1}}^{Z_{max-1}}\left[w(Z)\,g''(Z)\right]^2 = 0$$

Least square Solver

$$[A]_{N\times P+255,\,256+N}\,\{x\}_{256+N,1} = \{b\}_{N\times P}$$



$A_{ij} = W(Z_{ij})\,g(Z_{ij}) - W(Z_{ij})\ln E_i \qquad i \leq N\times P$

$\quad = +\lambda W(Z_{i-1}) - 2W(Z_i)\lambda + \lambda W(Z_{i+1}) \qquad i > N\times P$

$b_{ij} = W(Z_{ij})\ln\Delta t_j$

http://pages.cs.wisc.edu/~csverma/CS766_09/HDRI/hdr.html

Proposed Algorithm

- Merging to HDR is straightforward once we know g(Z) :

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

- For color images, do it for R, G, B channels separately, and scale them by the color balance of the radiance map, either achromatic or proportional to color of the light sources.

Matlab Code

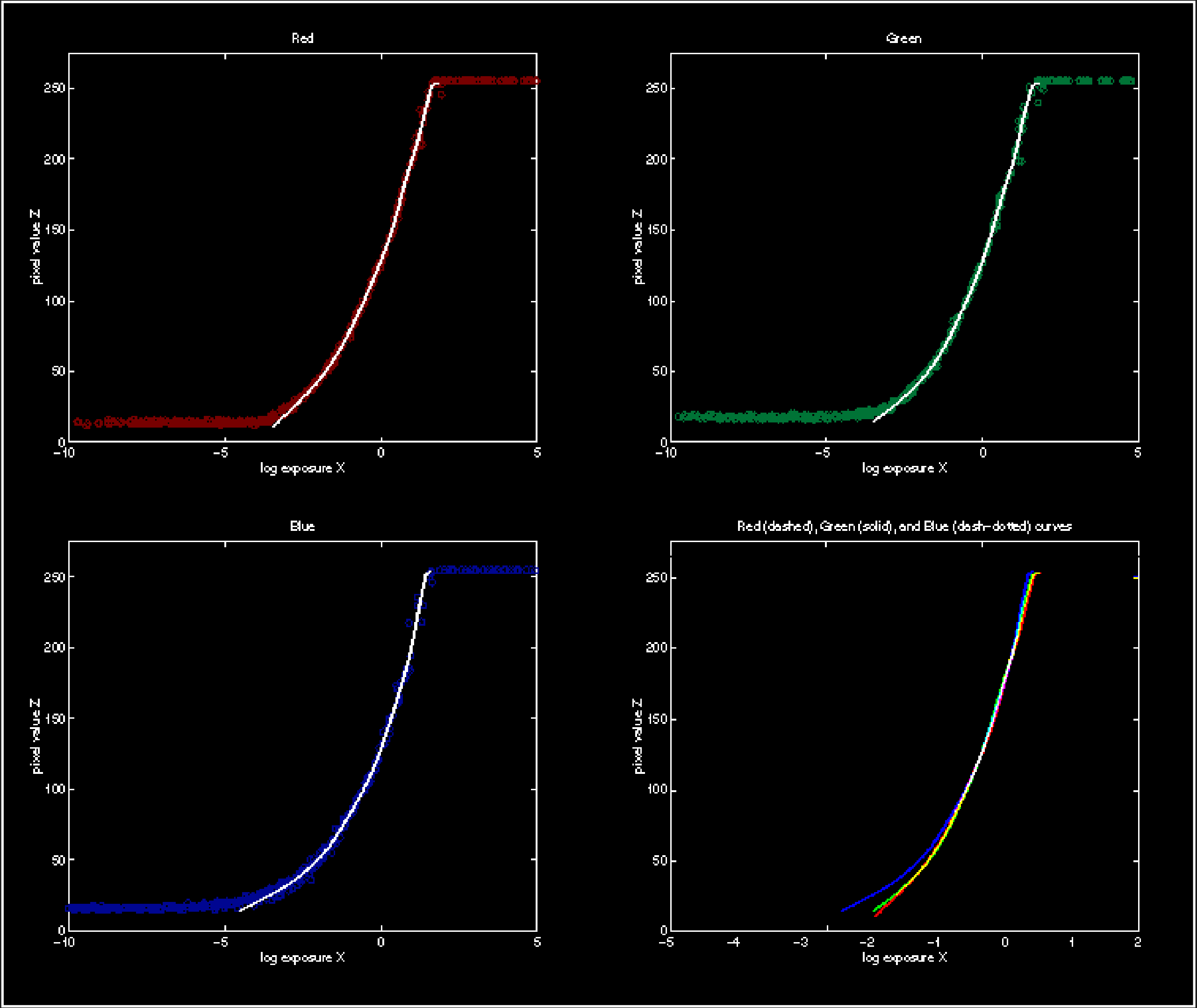
Matlab code

```
%
% 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

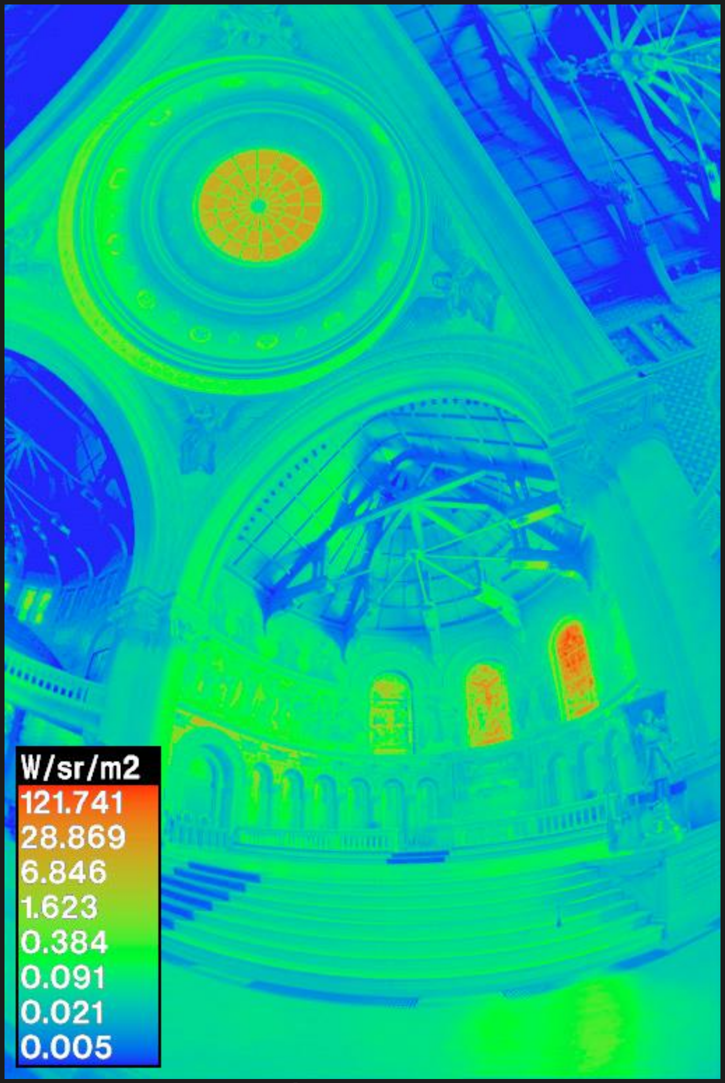
```
function [g,lE]=gsolve(Z,B,l,w)

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(i,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



Results

Original Photograph



Radiance Map

Radiance format (.pic, .hdr, .rad)

(145, 215, 87, 149) =

(145, 215, 87) * 2^(149-128) =

(1190000, 1760000, 713000)

(145, 215, 87, 103) =

(145, 215, 87) * 2^(103-128) =

(0.00000432, 0.00000641, 0.00000259)

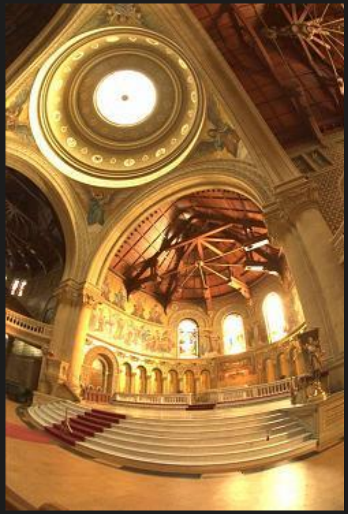
Ward, Greg. "Real Pixels," in Graphics Gems IV, edited by James Arvo, Academic Press, 1994

Rendering HDR Images: Tone Mapping

- Due to the limitation of current LDR 8-bit displays
- The goal is to compress the dynamic range of the input image and reproduce a realistic rendering based on human perception

Photograph

HDR



Photograph

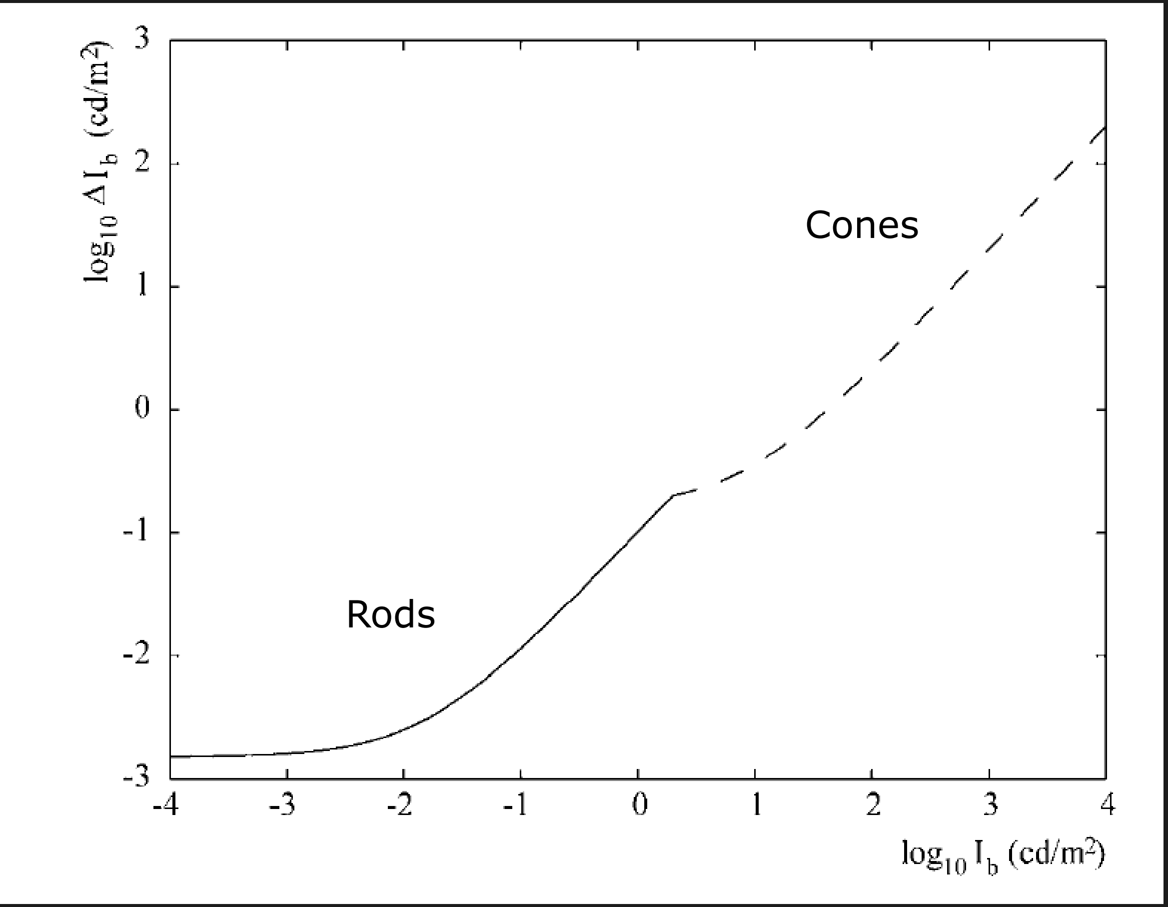
HDR

Human Visual Adaptation

FIGURE 6.4 *Although the headlights are on in both images, during daylight our eyes are less sensitive to car headlights than at night.*

[Reinhard, et al., 2006]

JND of Human Visual System

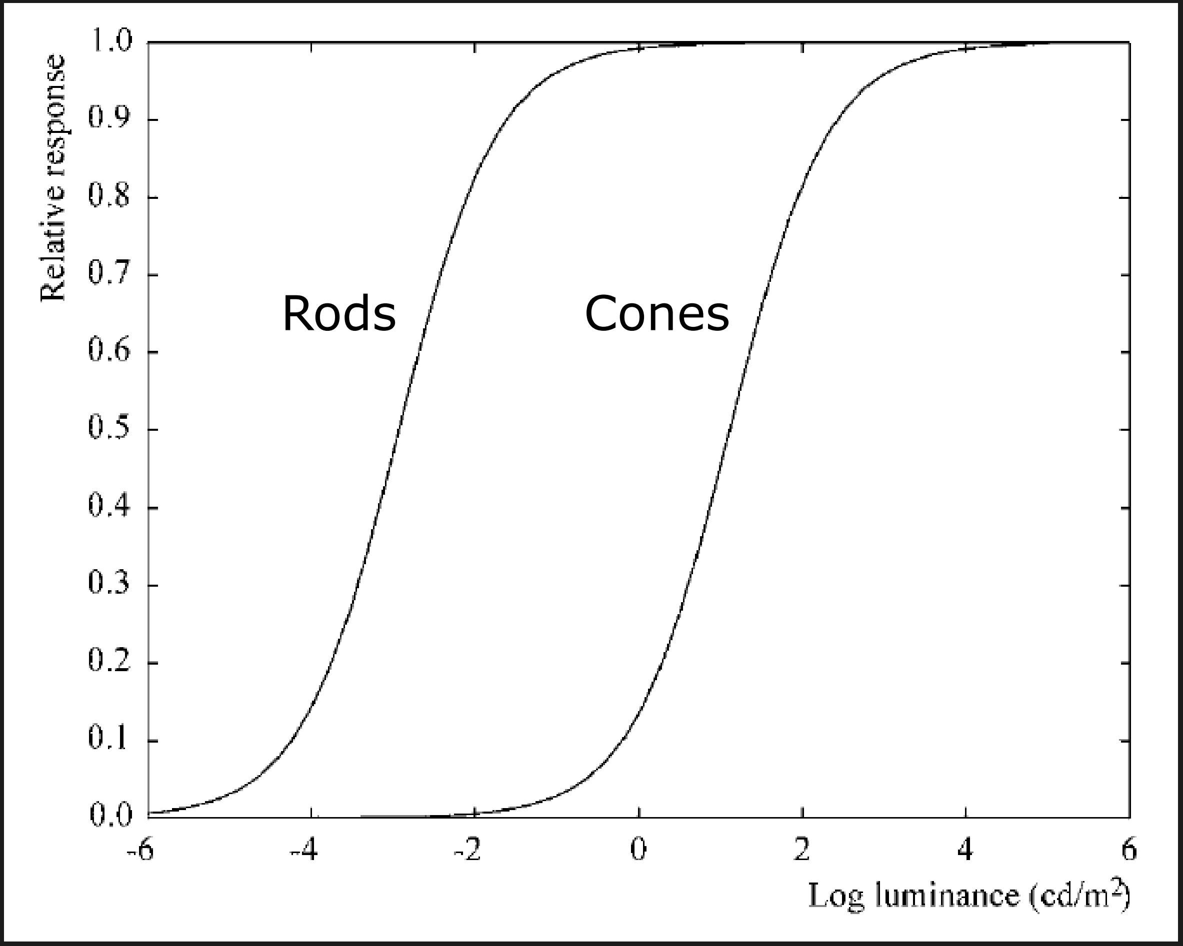


Weber's Law

$$\Delta I_b / I_b = const$$

[Reinhard, et al., 2006]

Response Curves of Rods and Cones



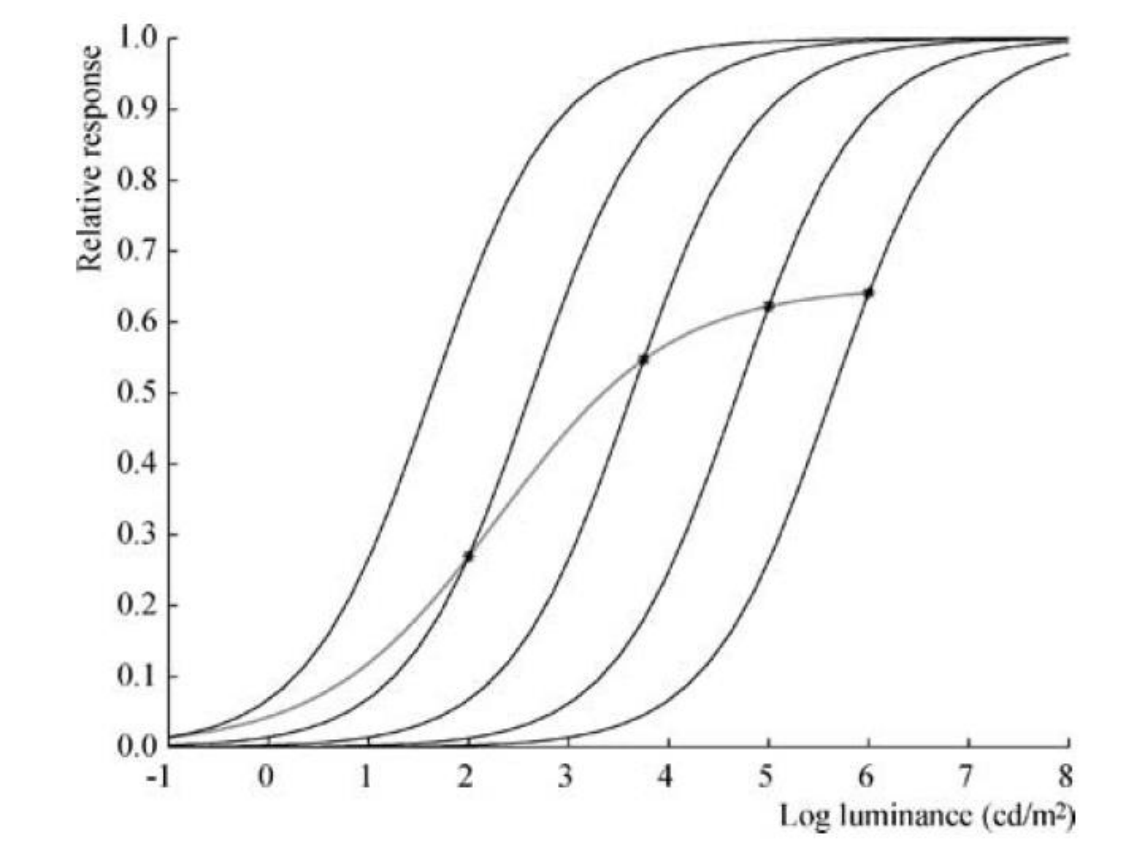
Michaelis-Menten Equation
(or, Naka-Rushton Equation)

$$\frac{R}{R_{\max}} = \frac{I^n}{I^n + \sigma^n}$$

n is between 0.7 and 1.

[Reinhard, et al., 2006]

Photoreceptor Adaption

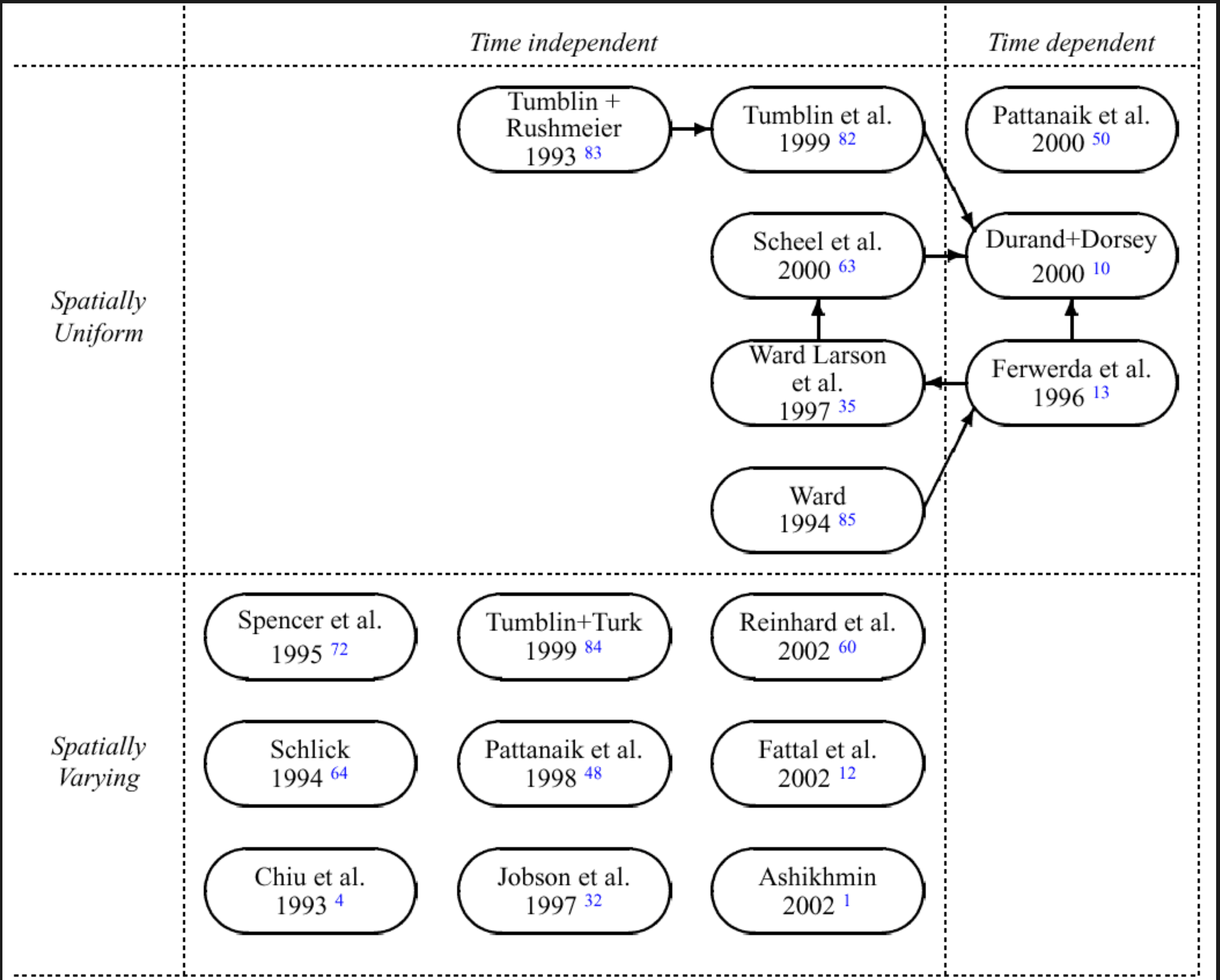


$$\frac{R}{R_{\max}} = \frac{I^n}{I^n + \sigma_b^n}$$

σ_b varies with background light intensity.

[Reinhard, et al., 2006]

Tone Mapping: Many Operators



Taxonomy of Tone Reproduction Methods

Devlin, Chalmers, Wilkie, Purgathofer / Tone Reproduction and Physically Based Spectral Rendering

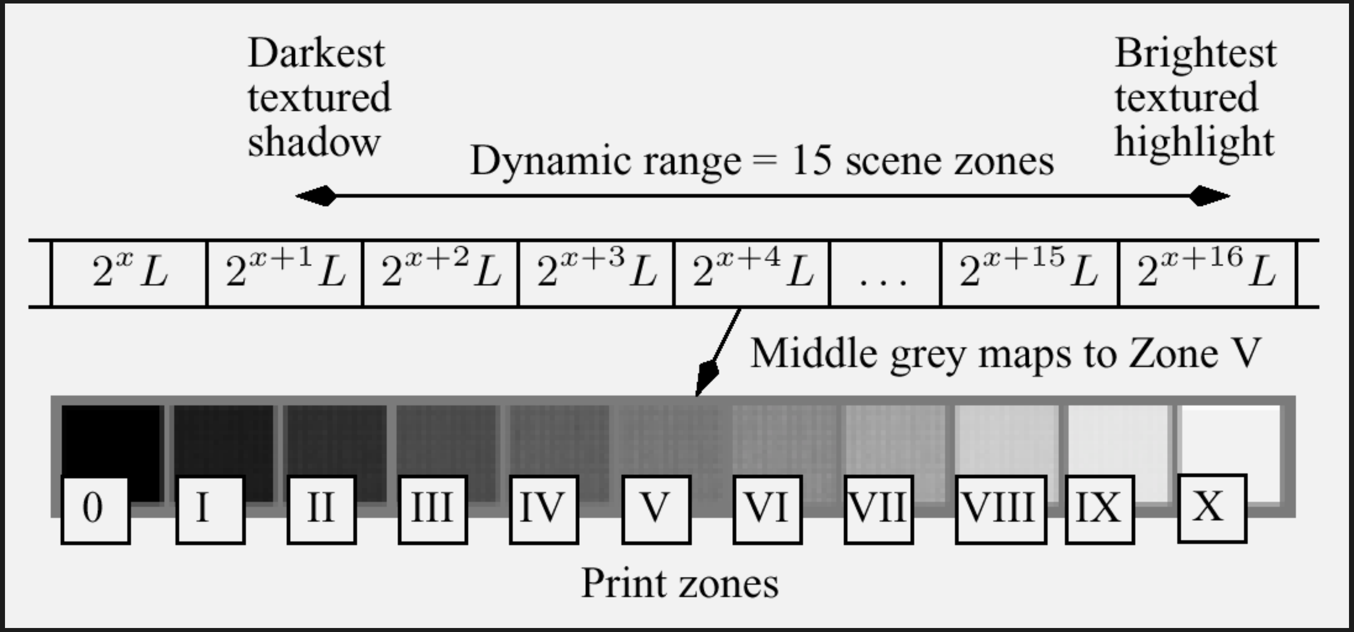
Spatially Uniform Operators

- "Gamma Curve" operator
- Focused on preserving overall brightness
- Subjective brightness, B
- k = constant
- L_0 = minimum luminance visible
- alpha = [.333, 0.49]
- Not valid for complex scenes; chosen for computational simplicity

$$B = k(L - L_0)^{\alpha}$$

Spatially Varying Operators

- Reinhard et al. 2002 "Photographic Tone Reproduction for Digital Images
- Technique is based on famous photographer Ansel Adams studies on tone reproduction using the Zone System (his invention); still widely used and practical



Spatially Varying Operators

Algorithm:

- Use the log-average luminance to find the "key" of a scene
- Automatic “dodging” or “burning” (as in photography): all portions of the print receive difference exposure time.

Spatially Varying Operators

- Step 1. Log Average:

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_{x,y}\log\left(\delta + L_w(x,y)\right)\right)$$

- Step 2. Scale Luminance to a key:

$$L(x,y) = \frac{a}{\bar{L}_w} L_w(x,y)$$

a is called the “key value”

Spatially Varying Operators

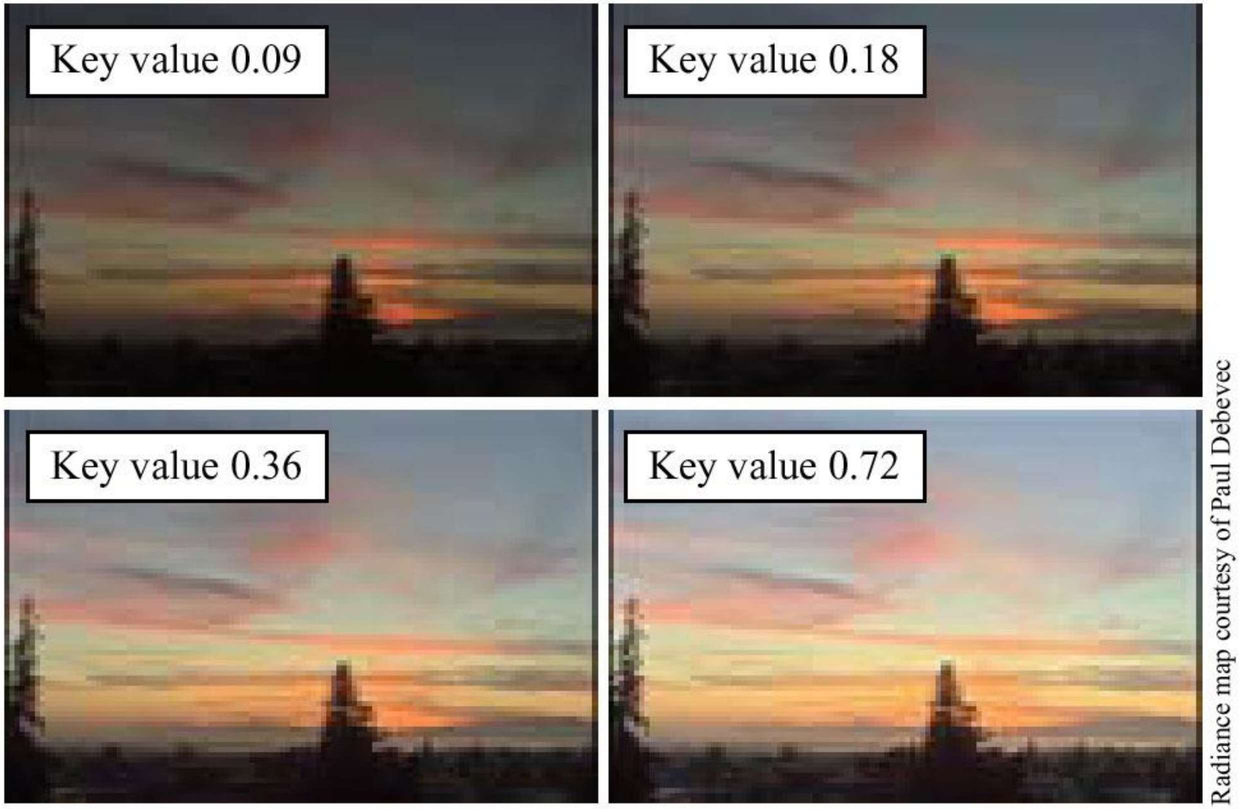


Figure 5: *The linear scaling applied to the input luminance allows the user to steer the final appearance of the tone-mapped image. The dynamic range of the image is 7 zones.*

Spatially Varying Operators

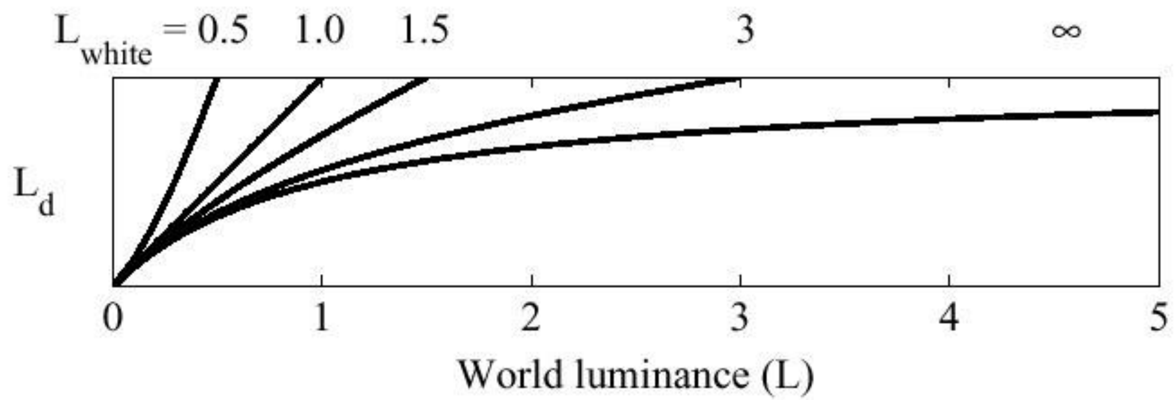
- Step 3. Compress the high luminance:

$$L_d(x,y) = \frac{L(x,y)}{1 + L(x,y)}$$

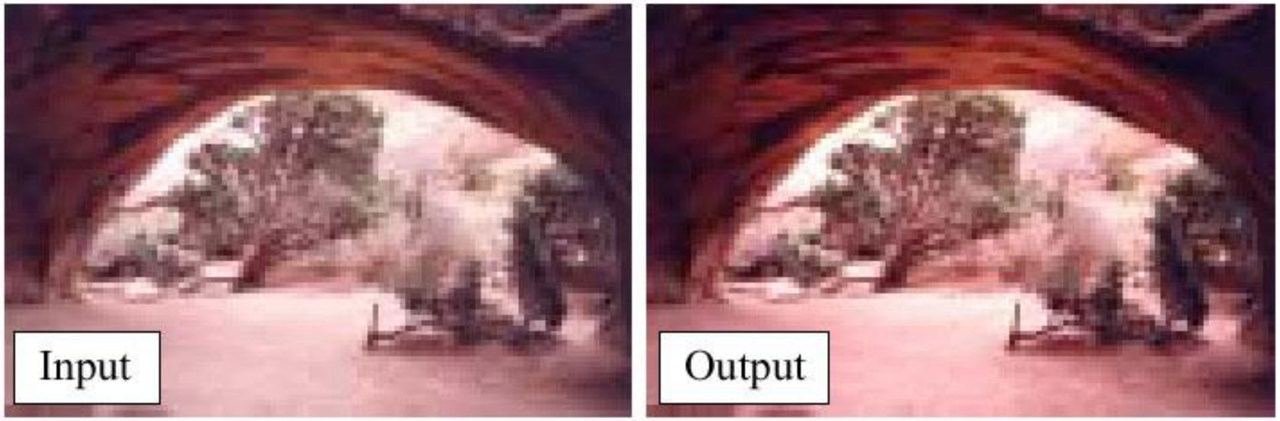
- Or, Step 3. Burning high luminance in a controlled fashion:

$$L_d(x,y) = \frac{L(x,y)\left(1 + \frac{L(x,y)}{L^2_{\text{white}}}\right)}{1 + L(x,y)}$$

Spatially Varying Operators



Display luminance as function of world luminance for a family of values for L_{white}.



Spatially Varying Operators

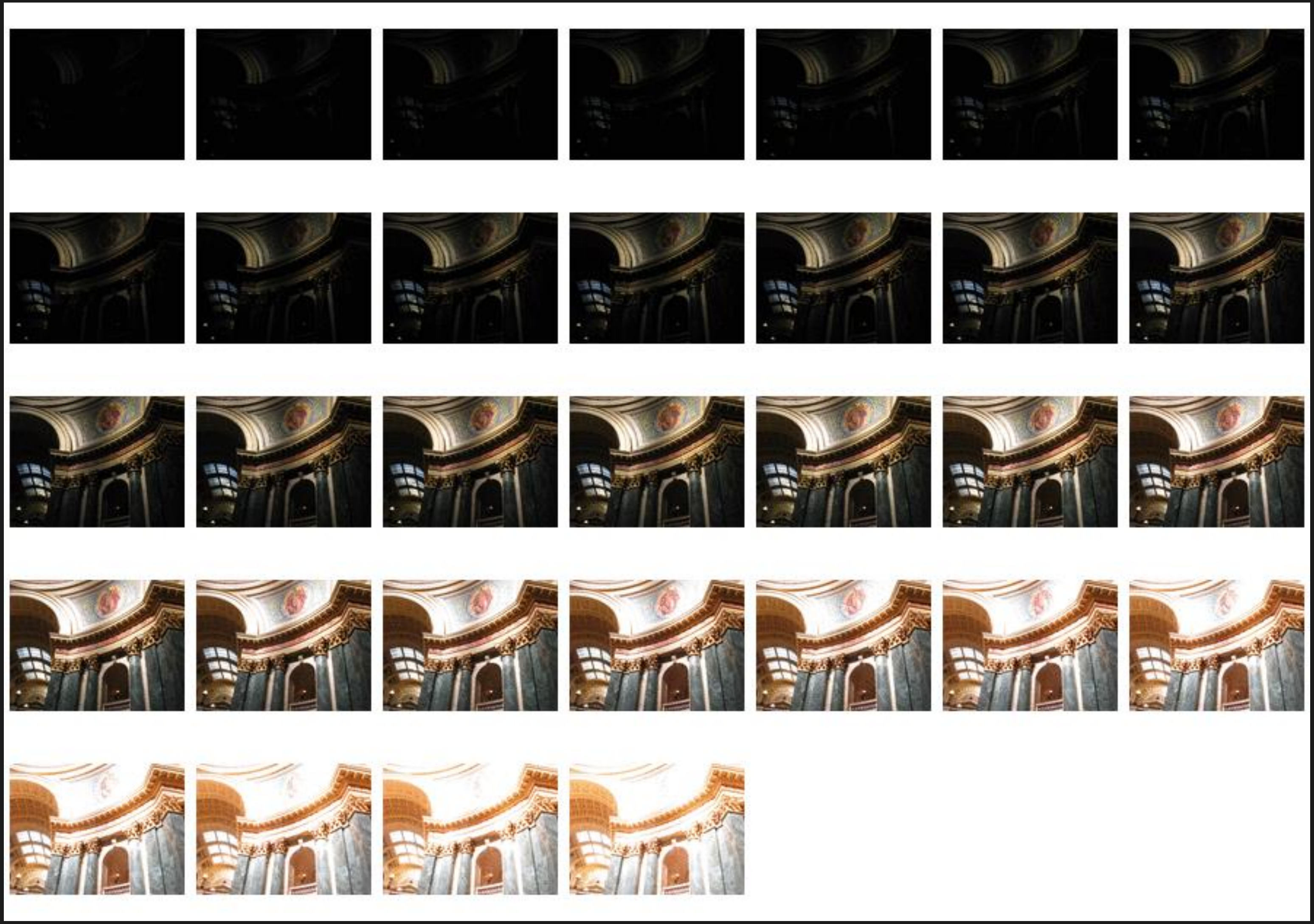
L_w=0.15

L_w=0.25

L_w=0.35

L_w=0.45

Spatially Varying Operators



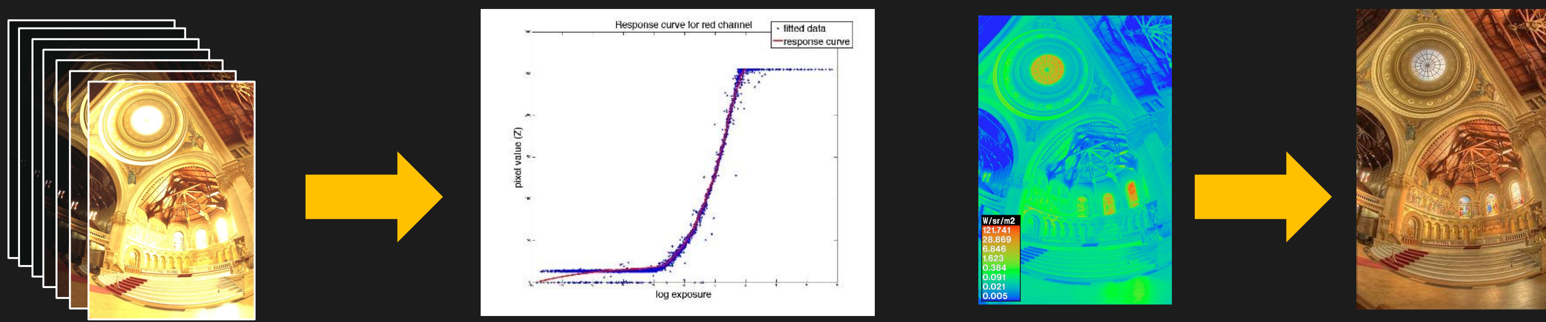
http://pages.cs.wisc.edu/~csverma/CS766_09/HDRI/hdr.html

Spatially Varying Operators

Reinhard ‘02 Tone Mapping

http://pages.cs.wisc.edu/~csverma/CS766_09/HDRI/hdr.html

Homework 3: HDR Imaging



Exposure Sequence | Camera Response | Radiance Map | Tone-mapped Image

<u>Fcam Programming</u>

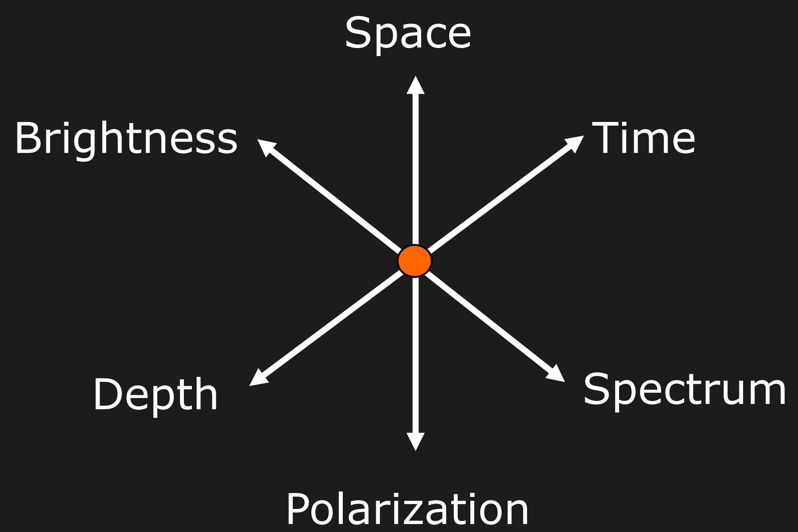
1. Write an exposure bracketing routine for your N900 phone

<u>MATLAB Programming</u>

1. Estimate the camera response curve and the radiance map of the scene
2. Apply a tone mapping operator to radiance map

Due before class on Thursday 2/7

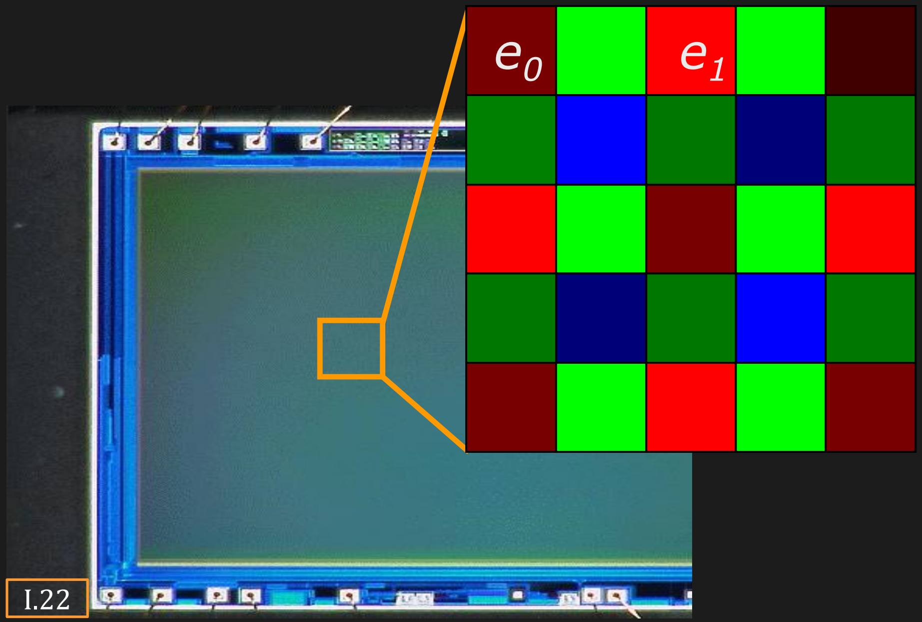
Dimensions of Imaging



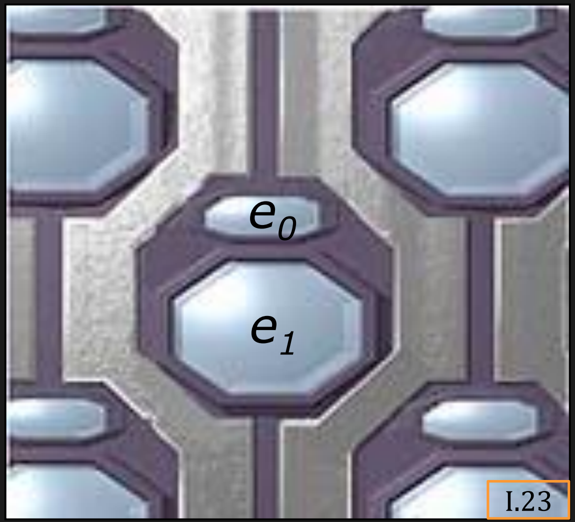
How can we capture information in multiple dimensions using only a two-dimensional array of pixels ?

[Nayar, et al., 2002]

High Dynamic Range: Single Shot

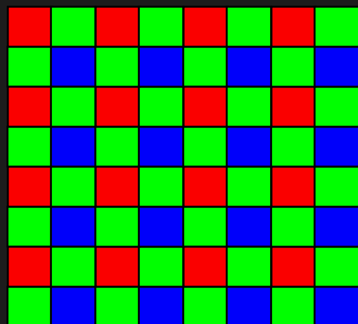


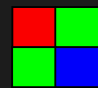
Assorted Pixels:
Spatially Varying Color & Exposure



SuperCCD SR, FujiFilm:
Pixels with Subpixels

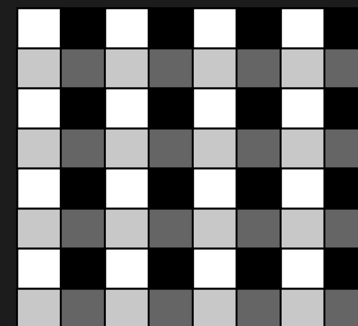
Assorted Pixels: Multi-Sampled Images





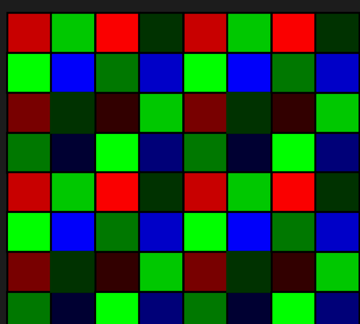
Base Pattern

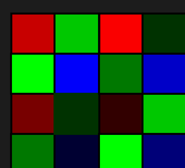
SVC : Spatially Varying Color
(Bayer Pattern)



Base Pattern

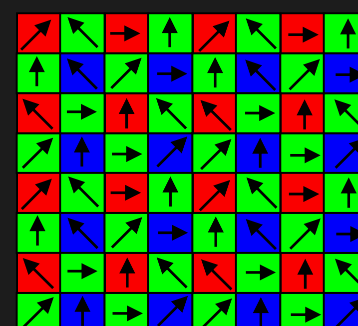
SVE : Spatially Varying Exposure
(Nayar et al., 00)

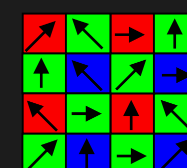




Base Pattern

SVEC : Spatially Varying Color and Exposure





Base Pattern

SVCP : Spatially Varying Color and Polarization

Simultaneous Sampling along Multiple Dimensions

Assorted Pixel Sensor

Assorted Pixel Camera: HDR Example

References: Papers

[Grossberg 2003] M. D. Grossberg and S. K. Nayar. "What is the Space of Camera Response Function?". *CVPR,* 2003.

[Mitsunaga 1999] T. Mitsunaga and S. K. Nayar. "Radiometric Self Calibration". *CVPR*, 1999.

[Nakamura 2006] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras.* CRC Press, 2006.

[Nayar 2000] S. K. Nayar and T. Mitsunaga. "High Dynamic Range Imaging: Spatially Varying Pixel Exposures". *CVPR,* 2000.

[Nayar 2002] S. K. Nayar and S. G. Narasimhan. "Assorted Pixels: Multi-Sampled Imaging with Structured Models". *ECCV,* 2002.

Image Credits

I.20 S. K. Nayar and T. Mitsunaga. “High Dynamic Range Imaging: Spatially Varying Pixel Exposures”. CVPR, 2000

I.21 S. K. Nayar and T. Mitsunaga. “High Dynamic Range Imaging: Spatially Varying Pixel Exposures”. CVPR, 2000.

I.22 http://www.dpreview.com/news/0301/03012202fujisuperccdsr.asp

I.23 http://www.dpreview.com/news/0301/03012202fujisuperccdsr.asp