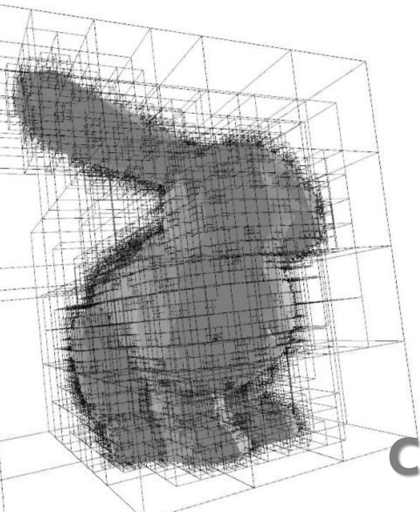
Image-Based Computer Graphics

Computational Photography

High Dynamic Range Imaging



CGRA 352

Dynamic range problem

- How to record and display as human eyes?

With High Dynamic Range Imaging

- Normal vs HDR (after Tone Mapping)

Dynamic range problem

- The "dynamic range" of a scene is the contrast ratio between its brightest and darkest parts

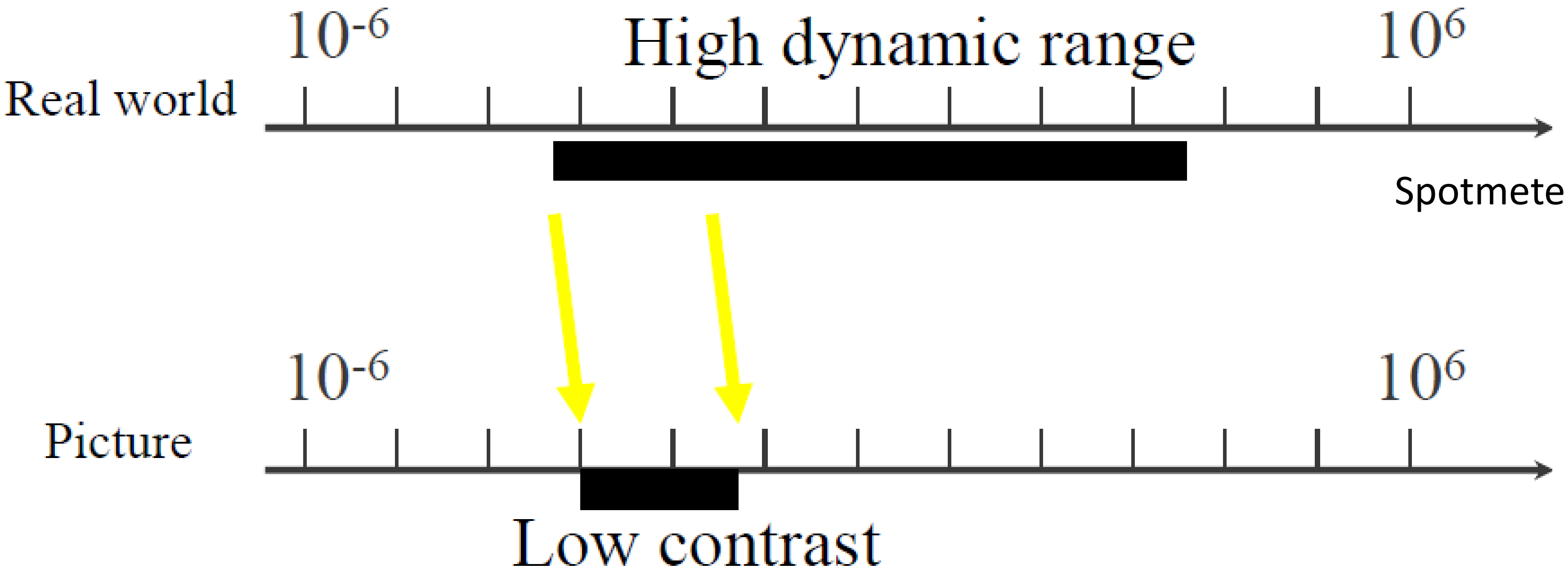
- Typical images displayed on screen are 24-bits
  - 8-bits per color component (RGB)
  - 256 different intensity levels
- Real-world dynamic range is far greater than 256 intensity levels!
  - Eye can adapt from ~ 10e-6 to 10e6 cd/m2
  - Often 1:100,000 in a scene

Dynamic range problem

- Dynamic range of different media (approximate and debatable)
  - 10:1 photographic print (higher for glossy paper)
  - 20:1 artist’s paints
  - 200:1 slide film
  - 500:1 negative film
  - 1000:1 LCD display
  - 2000:1 digital SLR (~11 bits)

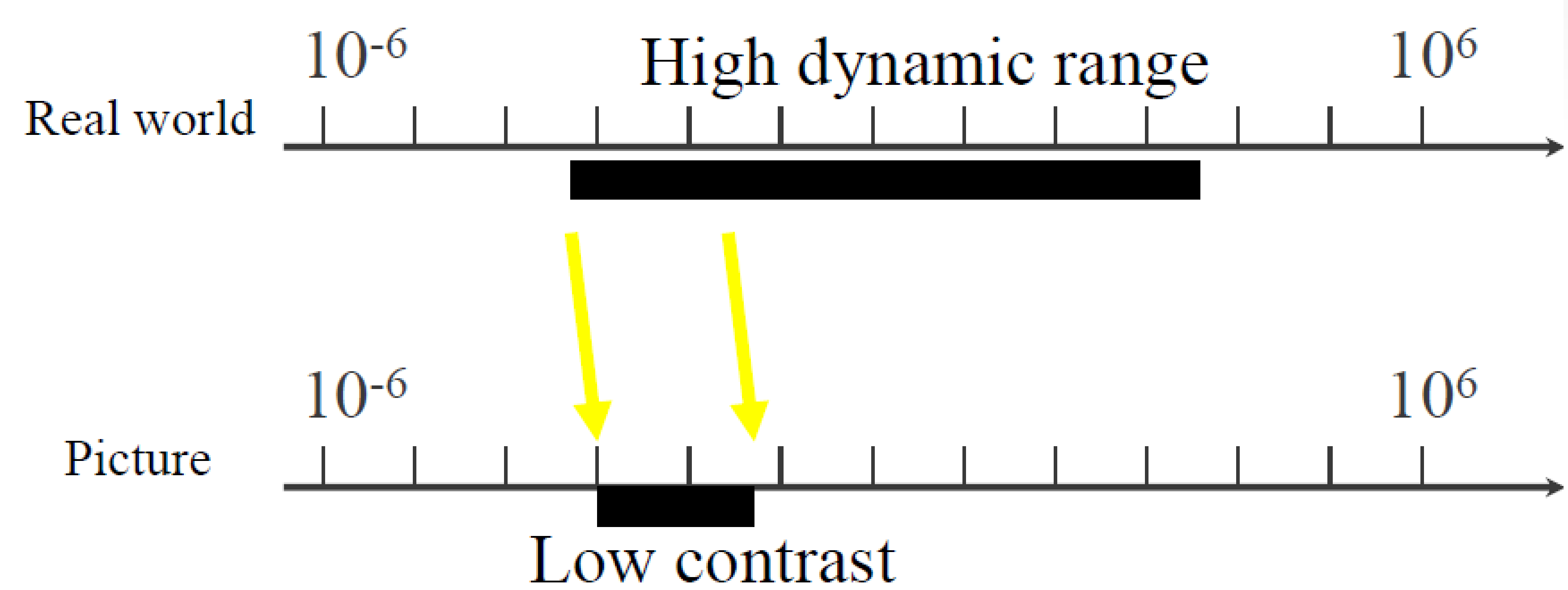
Problem 1: How to record it?

- The range of illumination levels that we encounter is 10 to 12 orders of magnitudes
- Negatives/sensors can record 2 to 3 orders of magnitude



Spotmeter

Multiple exposure photography



Real world

10^{-6}

High dynamic range

10^{6}

Picture

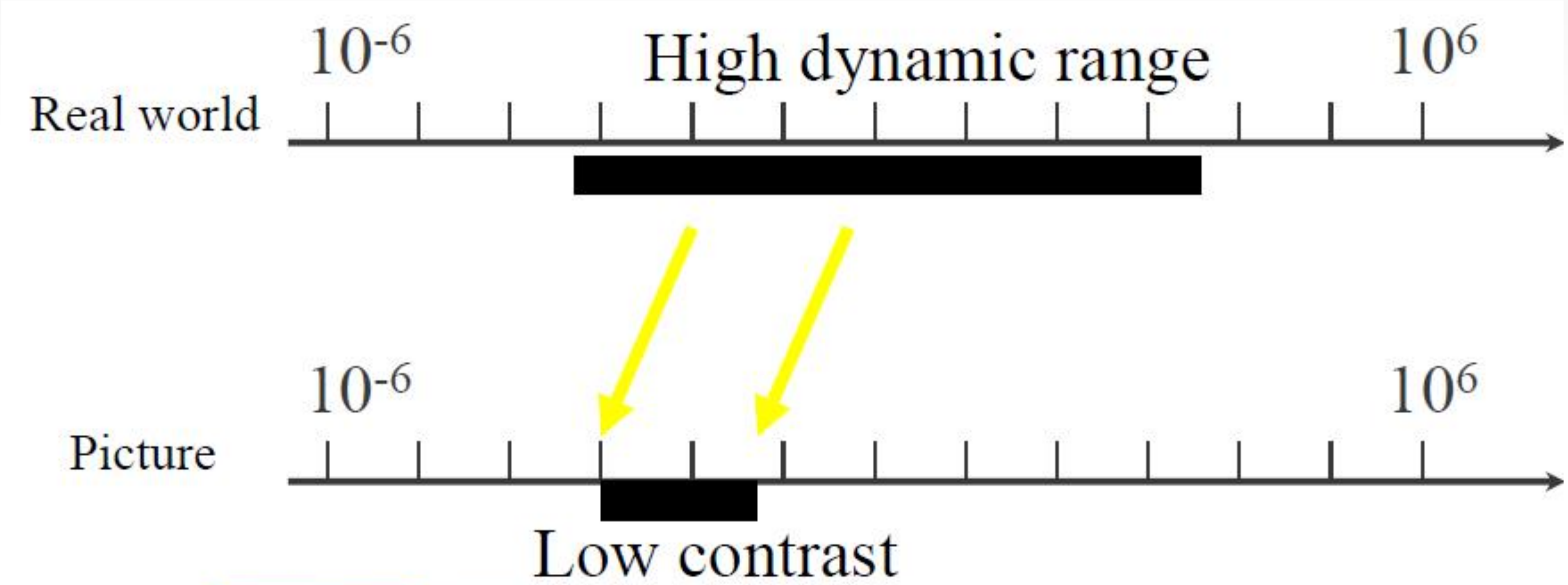
10^{-6}

10^{6}

Low contrast

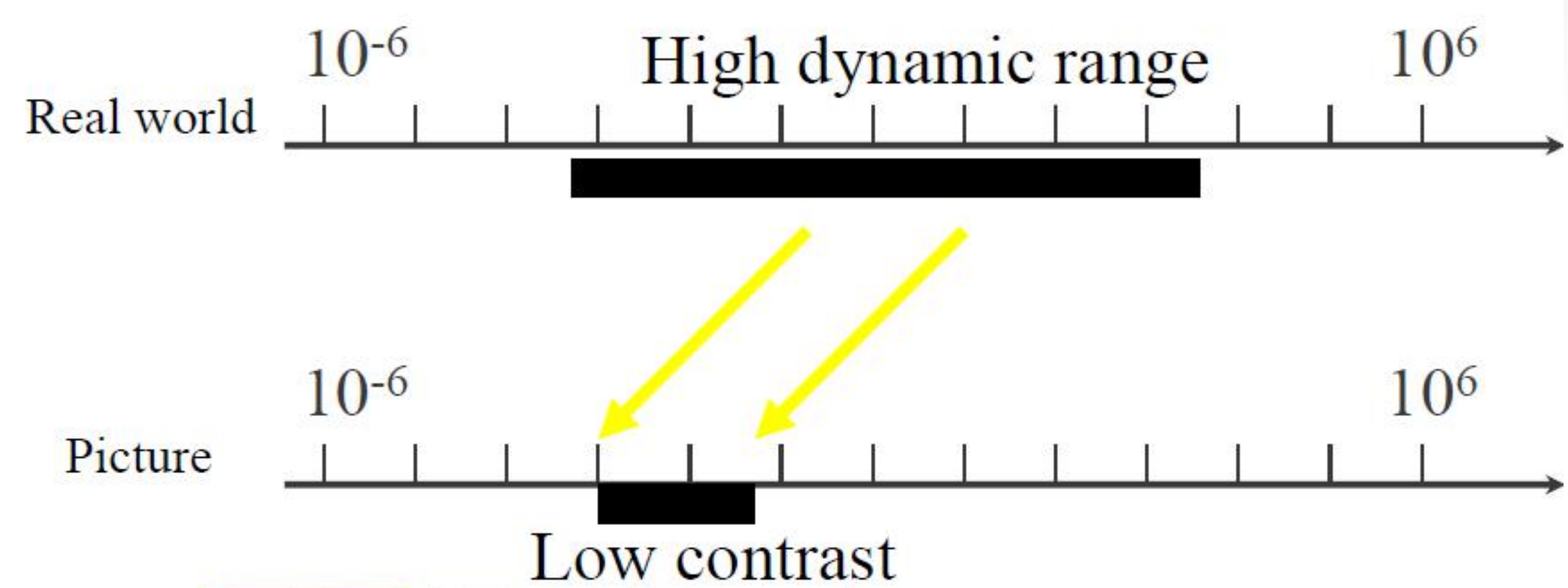
Shutter Speed: 10 sec

Multiple exposure photography



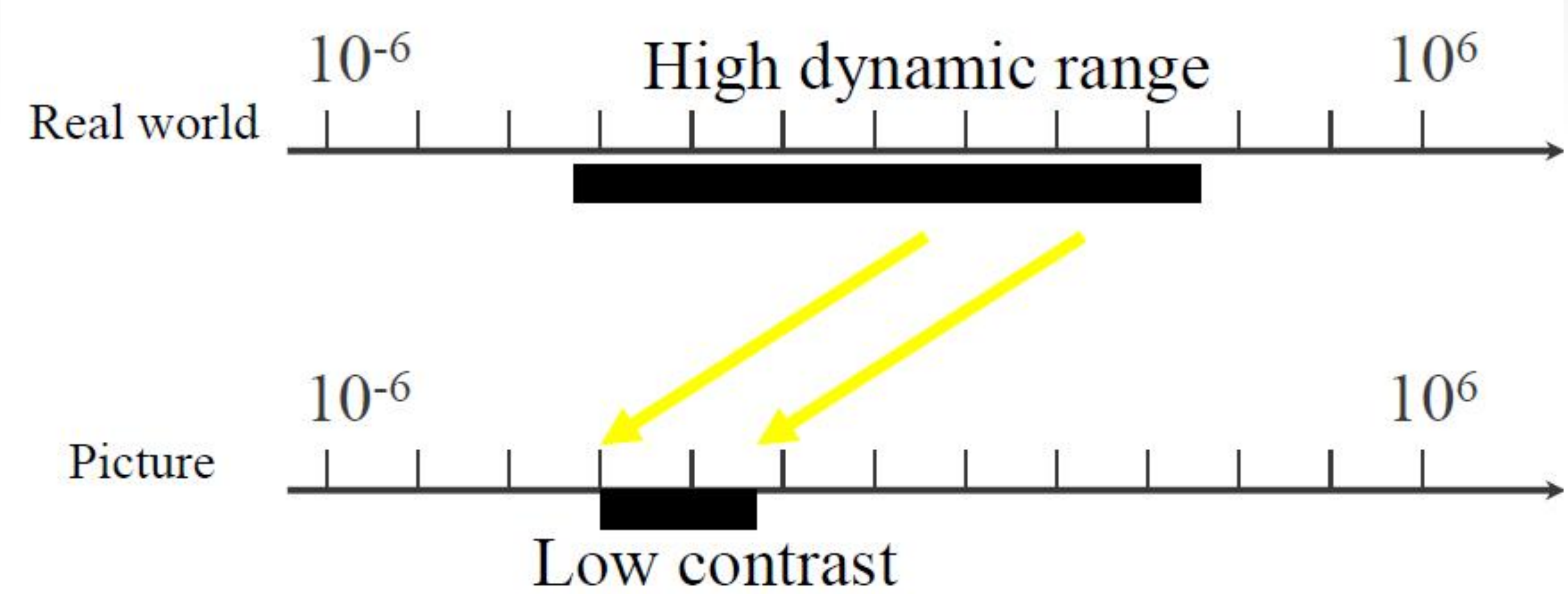
Shutter Speed: 1 sec

Multiple exposure photography



Shutter Speed: 0.1 sec

Multiple exposure photography



Real world

10^{-6} High dynamic range 10^{6}

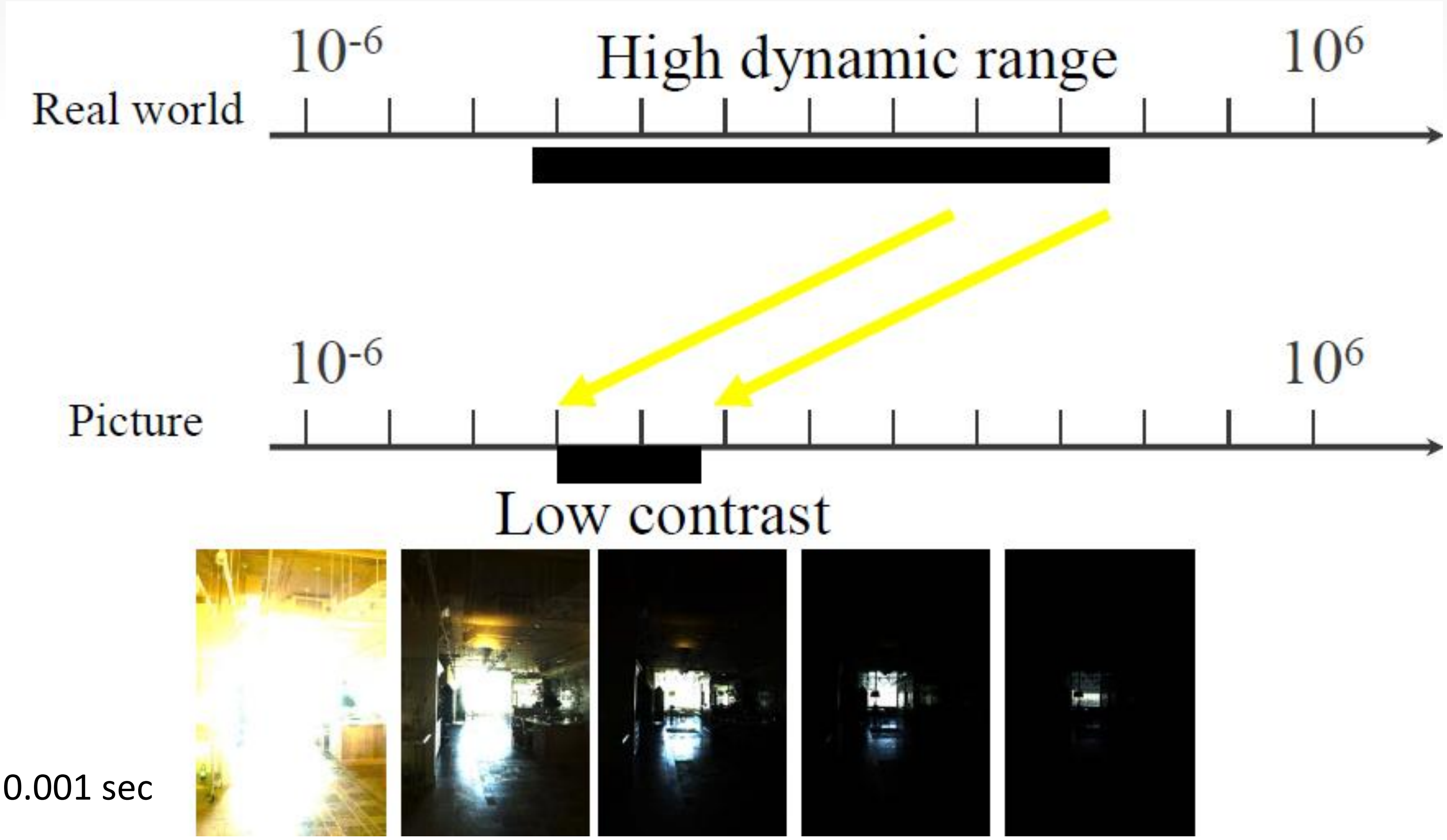
Picture

10^{-6} 10^{6}

Low contrast

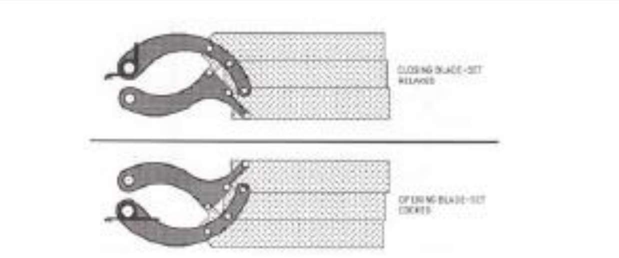
Shutter Speed: 0.01 sec

Multiple exposure photography



Shutter Speed: 0.001 sec

The ways to vary exposure



Shutter speed

- Range: ~30 sec to 1/4000sec (6 orders of magnitude)
- Pros: reliable, linear
- Cons: sometimes noise for long exposure

Aperture

- Range: ~f/1.4 to f/22 (2.5 orders of magnitude)
- Cons: changes depth of field
- Useful when desperate

ISO

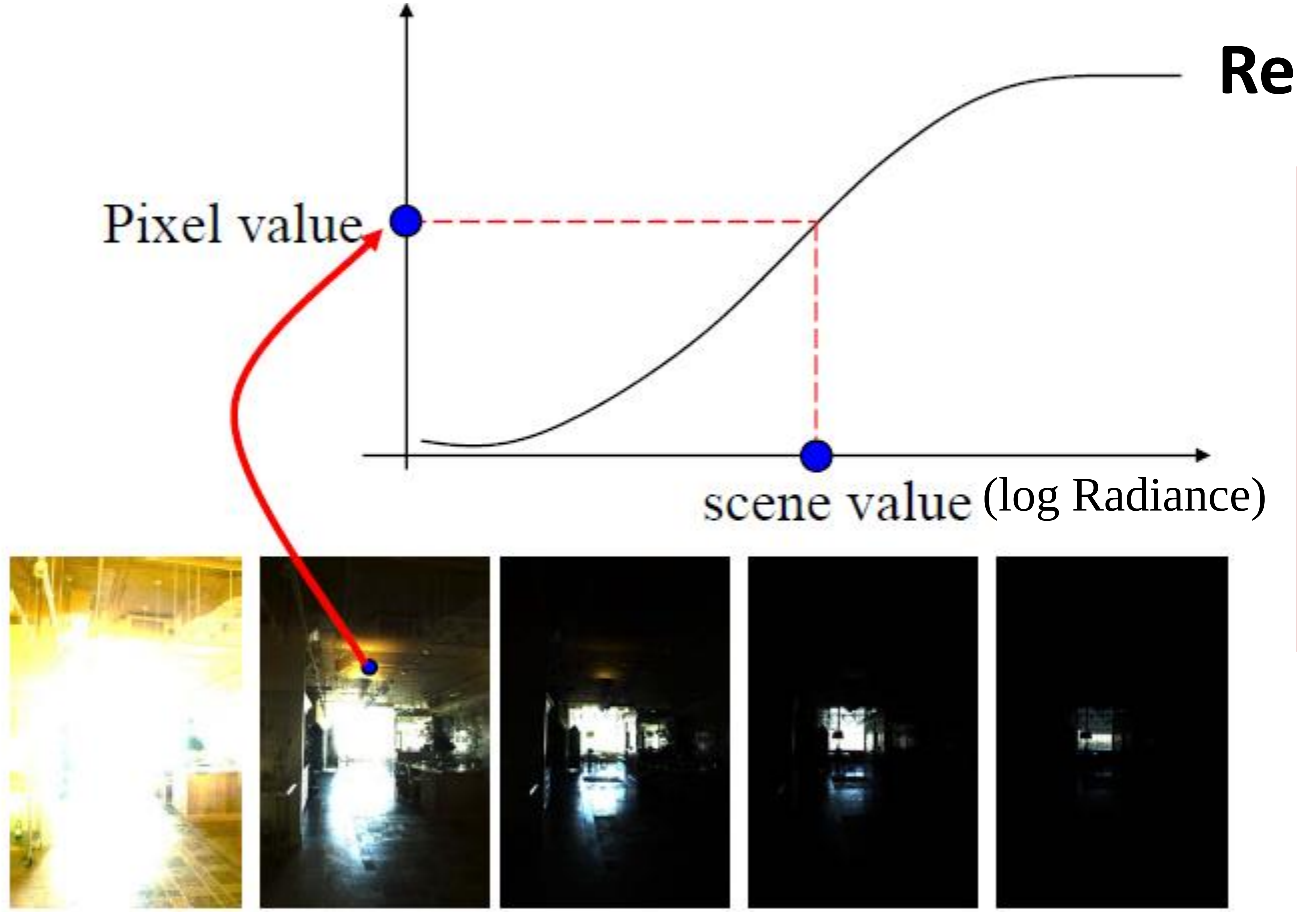
- Range: ~100 to 1600 (1.5 orders of magnitude)
- Cons: noise
- Useful when desperate

Neutral density filter

- Range: up to 4 densities (4 orders of magnitude) & can be stacked
- Cons: not perfectly neutral (color shift), not very precise, need to touch camera (shake)
- Pros: works with strobe/flash, good complement when desperate

Response curve

- Given N photos at different exposure, recover a HDR color for each pixel, a response to its real Radiance (physical brightness)



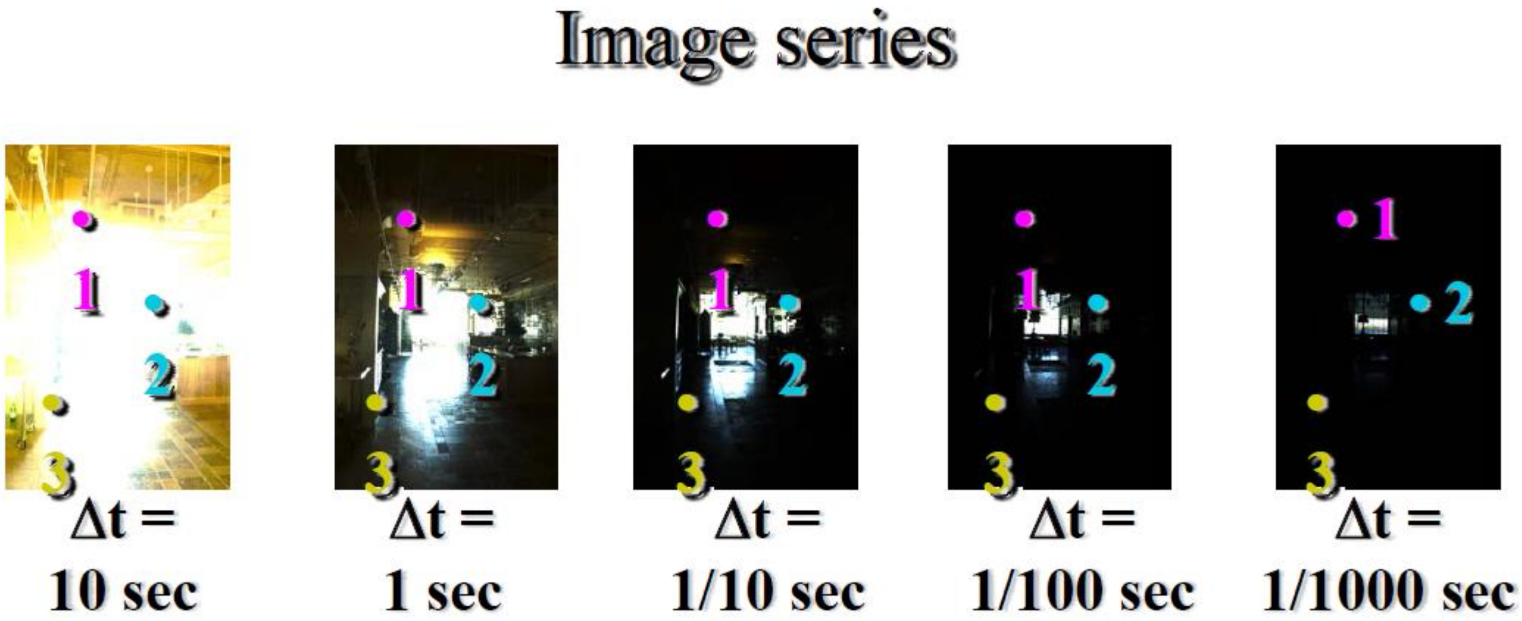
Response curve

Compress the range by replacing radiance value (scene value) with its logarithm.

Then the dynamic range will be not too large to be displayed on a screen

Response curve

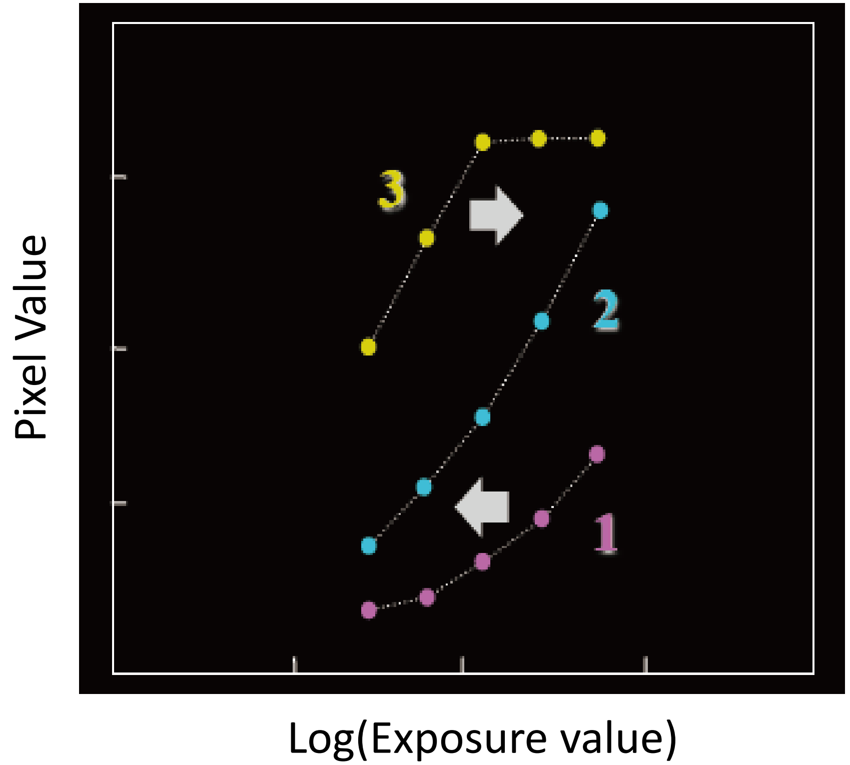
- Solutions
  - See pixel values as response one scene point's radiance with different exposure parameters (radiance * time)



*An alternative here : Vary scene luminance and see pixel values, assumes we control and know scene luminance

Response curve

- How to estimate a completed response curve?



For each pixel i and image j

Pixel Value $Z_{ij} = f(Exposure\ value)$

Exposure value = Radiance * Δt

Log (Exposure value) = log(Radiance) + log Δt

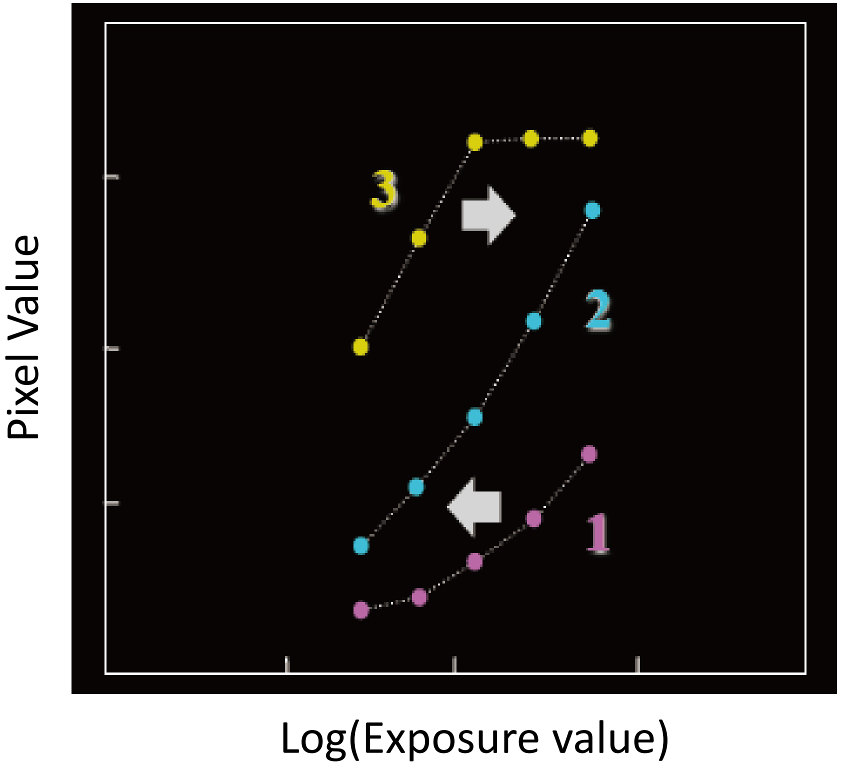
Response curve

- How to estimate a completed response curve?

Pixel Value $Z_{ij} = f(Exposure\ value)$

Exposure value = Radiance * Δt

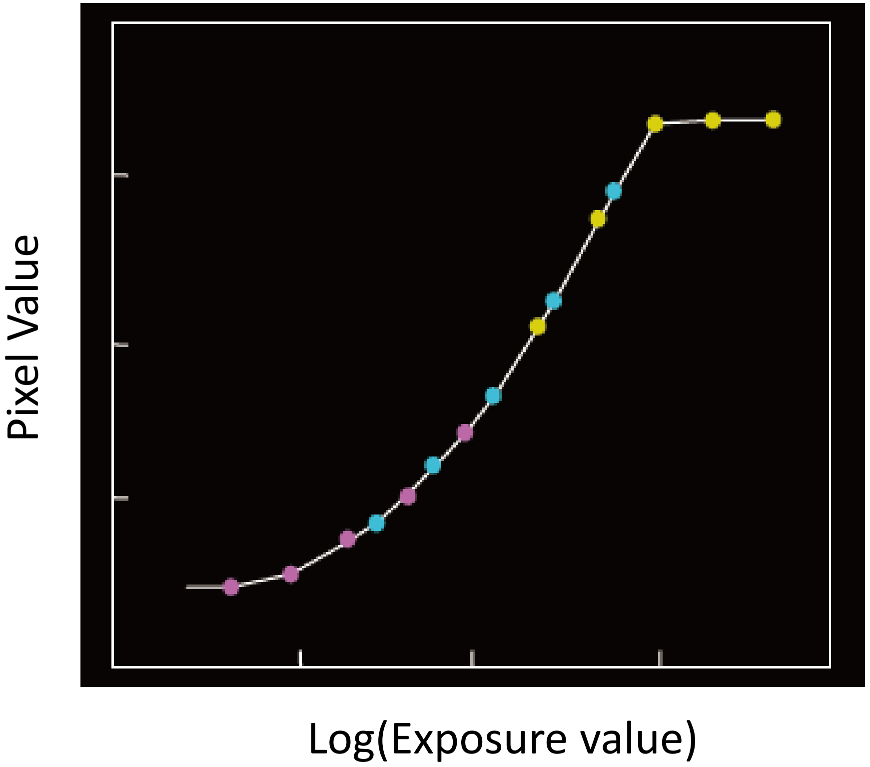
Log (Exposure value) = log(Radiance) + log Δt



The position of each curve for each scene point is **NOT** decided yet. We are aiming at forming a target curve like the right image.

We are actually seeking for a real radiance value for every scene point. Eventually, a radiance value of a point is a translation for each segment of the curve. Why?

After adjusting radiances to obtain a smooth response curve



Optimize the curve

Pixel Value $z_{ij} = f(Exposure\ value)$ -> Exposure value $= f^{-1}(z)$, denote f^{-1} as g

Exposure value = Radiance * Δt

Log (Exposure value) = log(Radiance) + log Δt

- We need to take it as an optimization problem to find the "response" to the exposure value (equivalent to find the relationship between the pixel value and a real radiance).
- **For each pixel site *i* in each image *j*, we want a function g of pixel value z :**

$$\log(\text{Radiance}_i) + \log \Delta t_j = g_z$$

Known: $\Delta t_j, z_{ij}$

Unknown: g_z (z is the response pixel value for the logarithm exposure), Radiance$_i$

Optimize the curve

$$\log(\mathrm{Radiance}_i) + \log \Delta t_j = g_z$$

- We want the value to satisfy the above equation. But we only have observed relationship between $\mathrm{Radiance}_i$ and g_z (a difference by Δt_j).
- More importantly, for every observed exposure value $\log(\mathrm{Radiance}_i) + \log \Delta t_j$, we actually have many pixels in j images with the same value, but may correspond to different pixel intensity level z.
- It is an over-determined system. We transfer it into an optimization problem, to minimize the difference between $\log(\mathrm{Radiance}_i) + \log \Delta t_j$ **and** g_z

Optimize the curve

$$\log(\text{Radiance}_i) + \log \Delta t_j = g_z$$

- If we have 0-M pixel intensity levels to represent the logarithm response curve, the actual unknown variables should be: $g_0, g_1, \dots, g_M$ and the radiance values for sampled scene points
- For every observed point in image j at pixel i, we can get a difference term: $\log(\text{Radiance}_i) + \log \Delta t_j - g_z$, where z means the observed pixel intensity level

Optimize the curve

$$\log(\text{Radiance}_i) + \log \Delta t_j = g_z$$

- Minimize this energy to make the curve fit the observed points.

$$\sum_{i=1}^{N}\sum_{j=1}^{P}[\log(Radiance_i) + \log \Delta t_j - g_z\,]^2$$

fitting term

- Add another smoothness term to get a smooth response curve

$$\sum_{i=1}^{N}\sum_{j=1}^{P}[\log(Radiance_i) + \log \Delta t_j - g_z\,]^2 + \lambda \sum_{z_{min}}^{z_{max}} (g''_z)^2$$

smoothness term

$$\boxed{g_{z-1} - 2g_z + g_{z+1}}$$

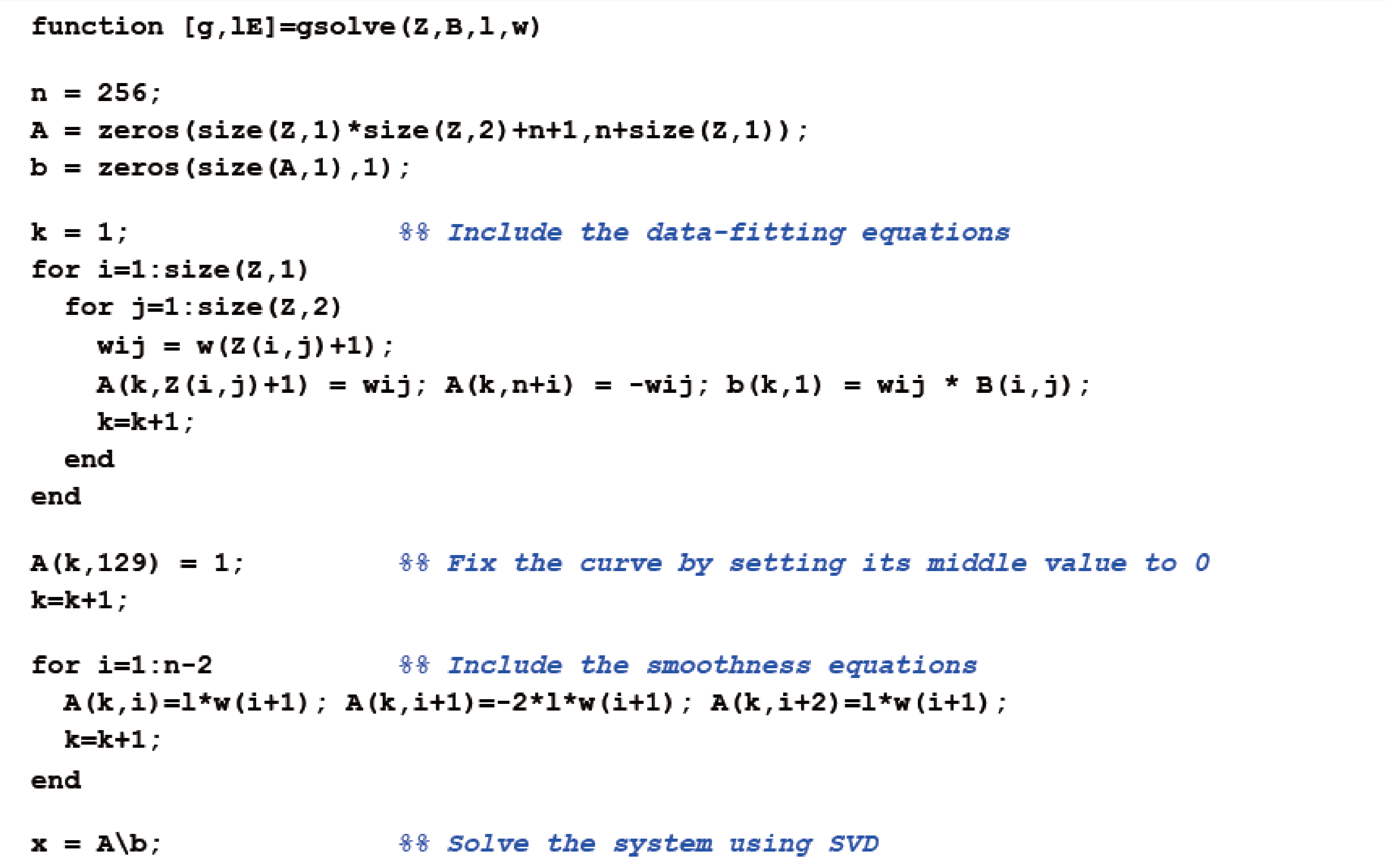
Optimize the curve

$$\sum_{i=1}^{N}\sum_{j=1}^{P}[\log(\text{Radiance}_i) + \log \Delta t_j - g_z]^2 + \lambda \sum_{z_{min}}^{z_{max}} (g''_z)^2$$

Solve an over-determined linear system to get the minimum of this energy, because the minimum value is where the derivative is zero.

Consider a vector x formed by all the unknowns
By setting the derivatives to zero, we will get a form of:
$Ax = b$ (A is a matrix, b is a know vector), then solve the linear system.

The code in the original paper by Debevec



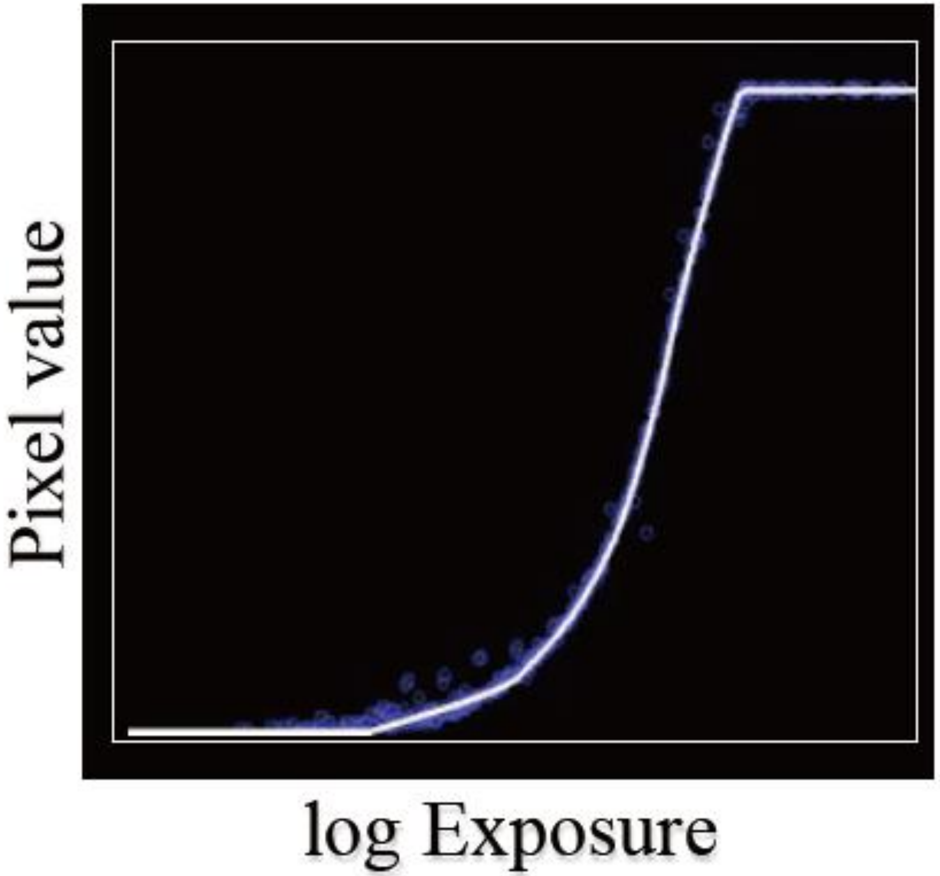
```
function [g,lE]=gsolve(Z,B,l,w)

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
```

Slide stolen from Alyosha Efros who stole it from Paul Debevec

Response Curve

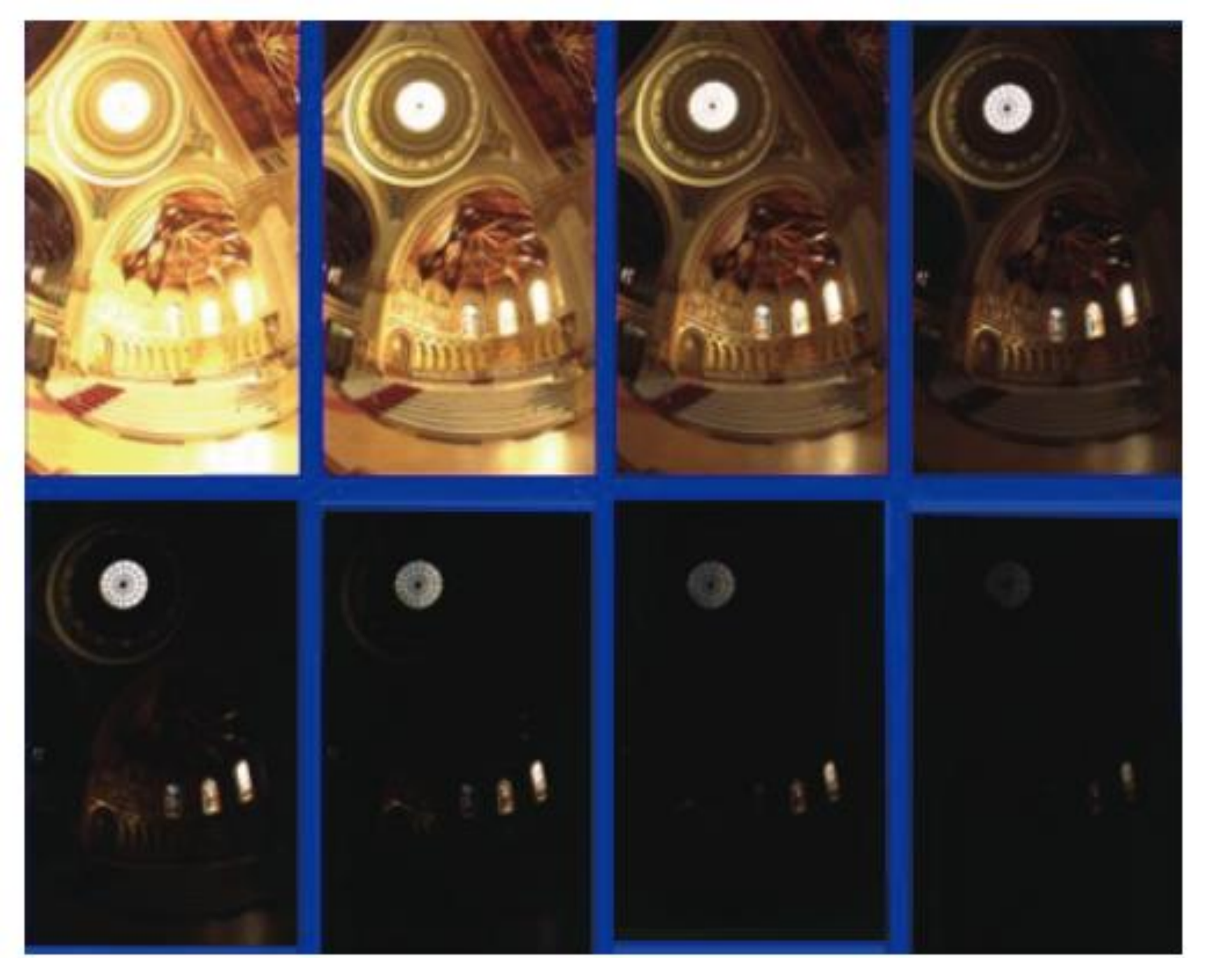
Kodak DCS460
1/30 to 30 sec

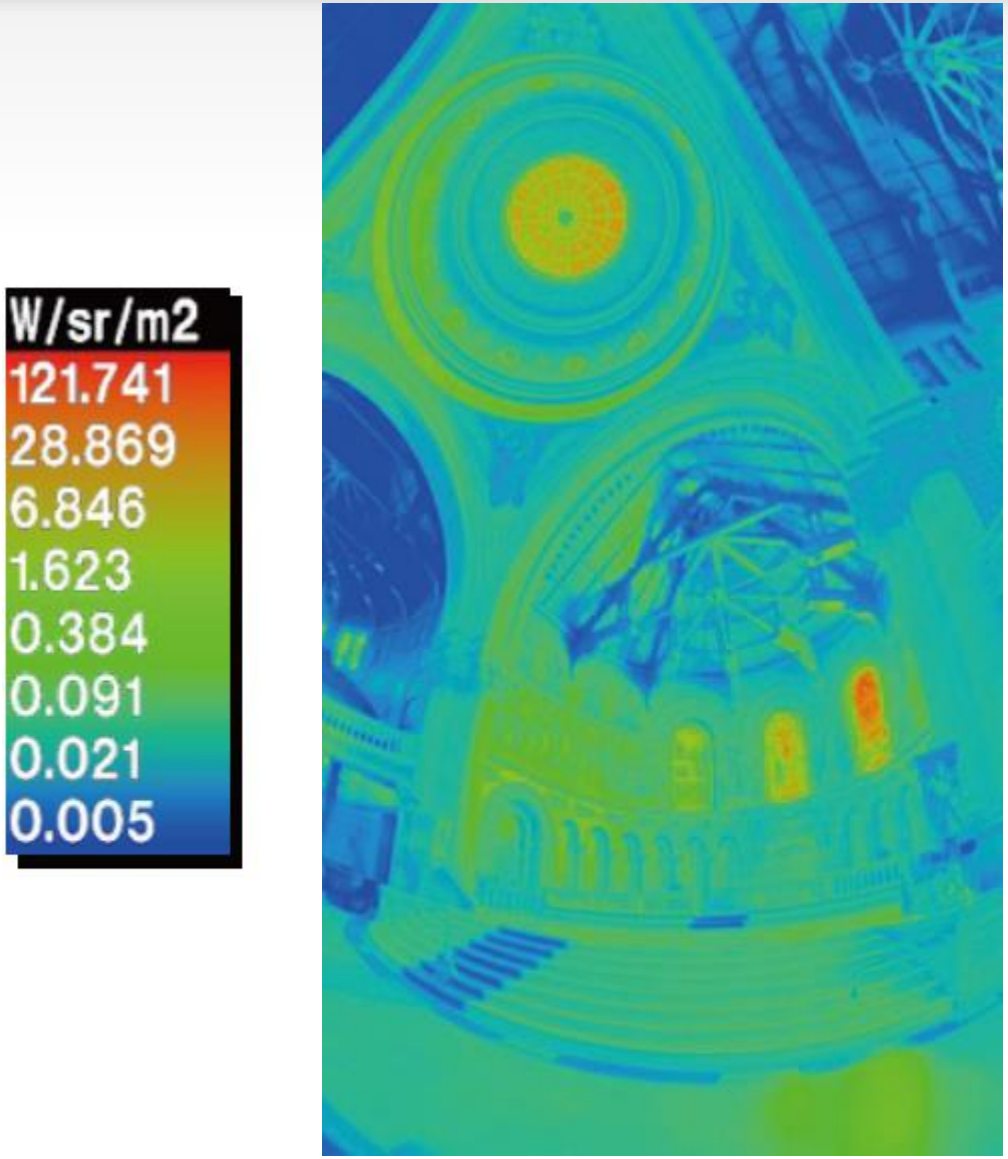
Recovered response curve



Reconstructed radiance map

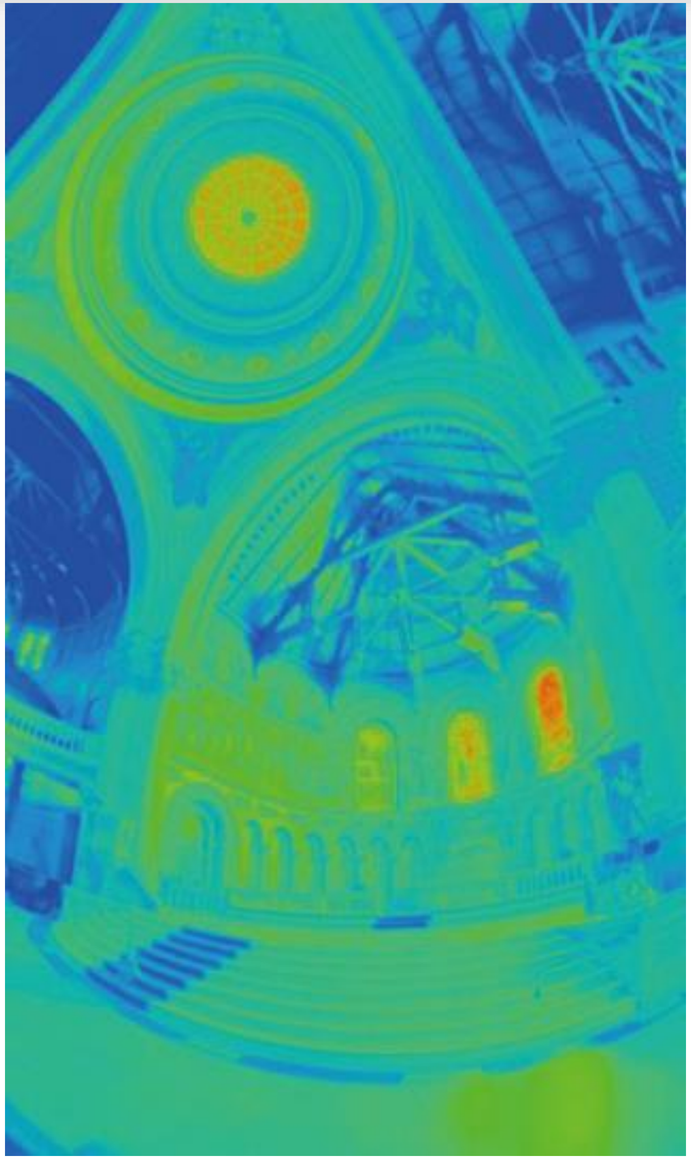
Reconstructed radiance map





Problem 2: How to display?

Linearly scaled
to display device



How to display?

- Input: high-dynamic-range image (Radiance map, floating point per pixel)
- Scene has *1:10,000* contrast, display has *1:100, so* simplest contrast reduction?

Naïve: Gamma compression

- $Z = r^{\gamma}$ In this case, $\gamma = 0.5$ can map the radiance into an acceptable range.
- Colors are washed-out.

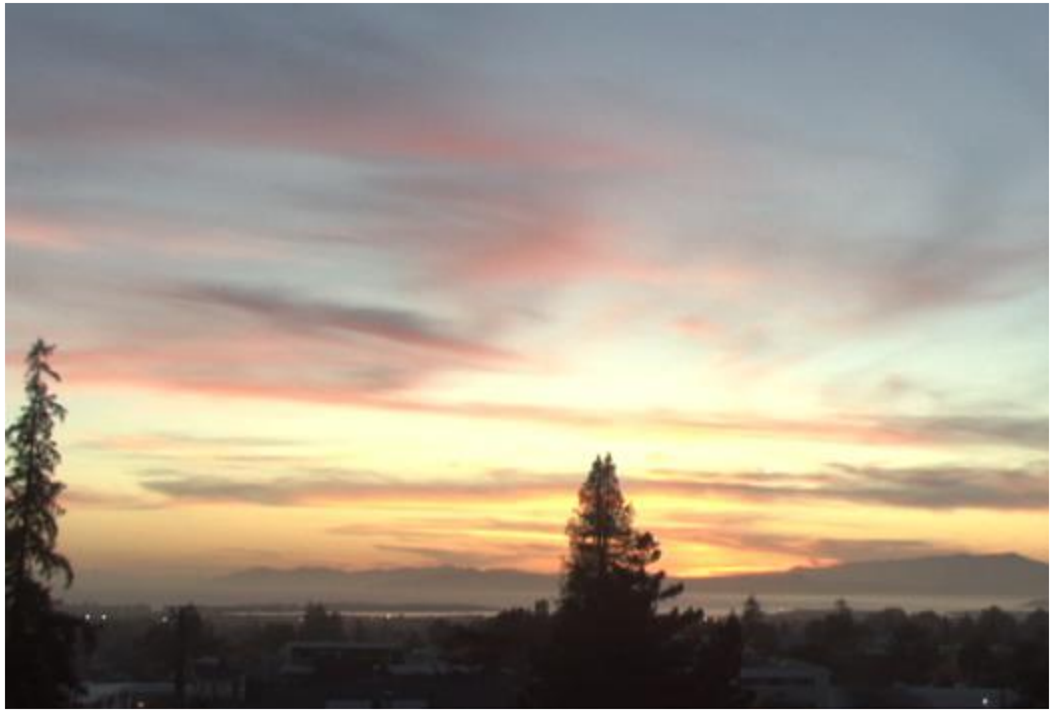
  -Why ?

  -Larger RGB difference means larger color saturation

RGB: 80,93,120 🟦
Saturation: 51

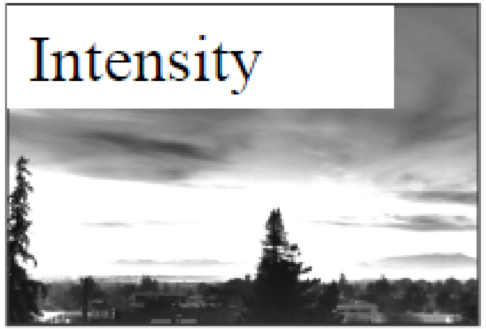
41,93,159 🟦
Saturation: 150

It is applied independently on R, G & B. The linear relationship between 3 channels are not preserved by Gamma compression

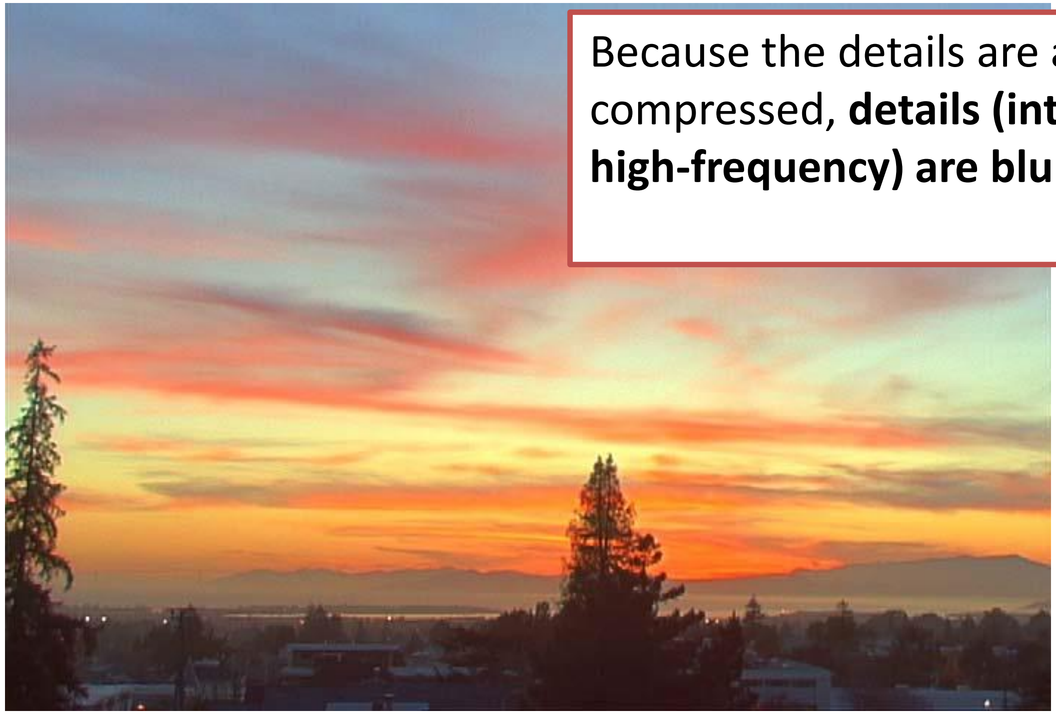


Naïve: Gamma compression

- Keep the color, just compress the intensity.
- We can do it by Lab color space, or just extract the relative value of RGB channels.



R’=R/intensity
G’=G/intensity
B’=B/intensity
important to use ratios
(makes it luminance invariant)



Because the details are also compressed, **details (intensity high-frequency) are blurred**

Tone Mapping by Bilateral Filtering

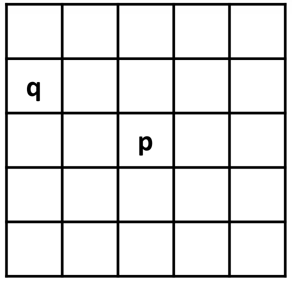
- **Do not blur details** by **non-linear filtering (Bilateral filtering)**

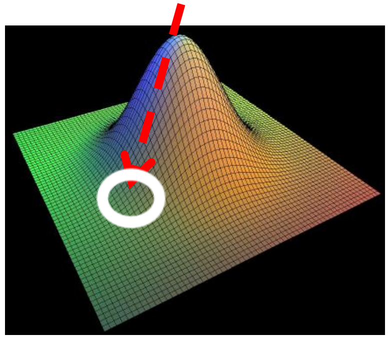
Recap Bilateral Filtering

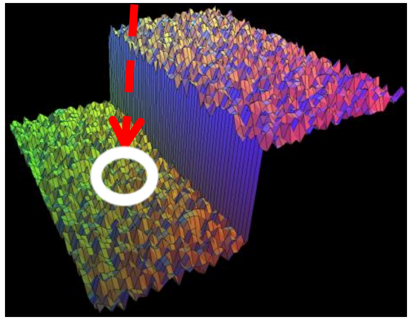
- Weighted average of **SIMILAR** pixels

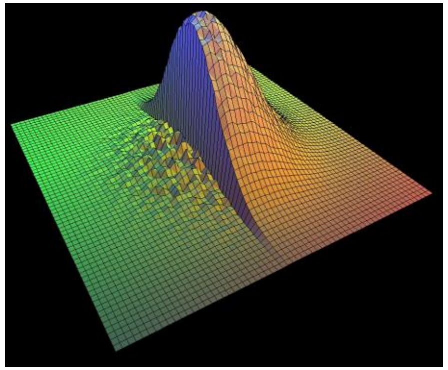
$$B(\boldsymbol{p}) = 1/W_{\boldsymbol{p}} \sum_{\boldsymbol{q}\in N} G_s(||\boldsymbol{p}-\boldsymbol{q}||)\,G_r\left(||I_p - I_q||\right) I_{\boldsymbol{q}}$$

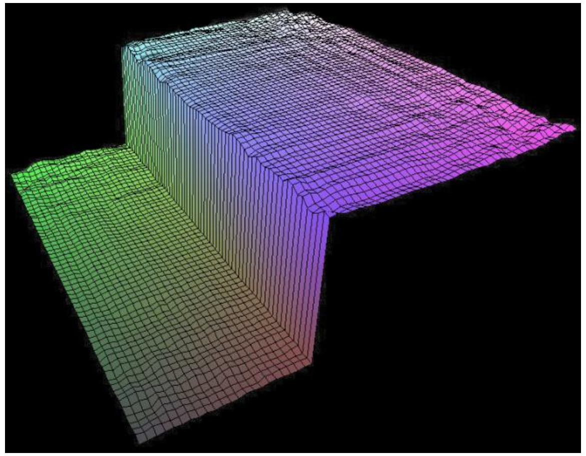
$$\boxed{W_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in N} G_s(||\boldsymbol{p}-\boldsymbol{q}||)\,G_r\left(||I_p - I_q||\right)}$$

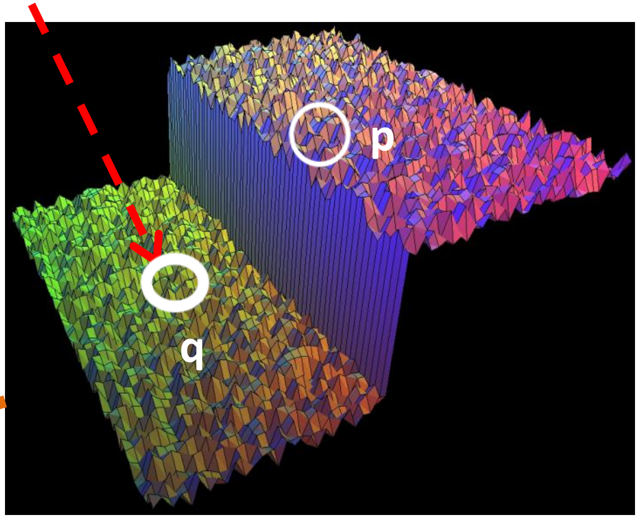



Spatial distance weight


Color difference weight



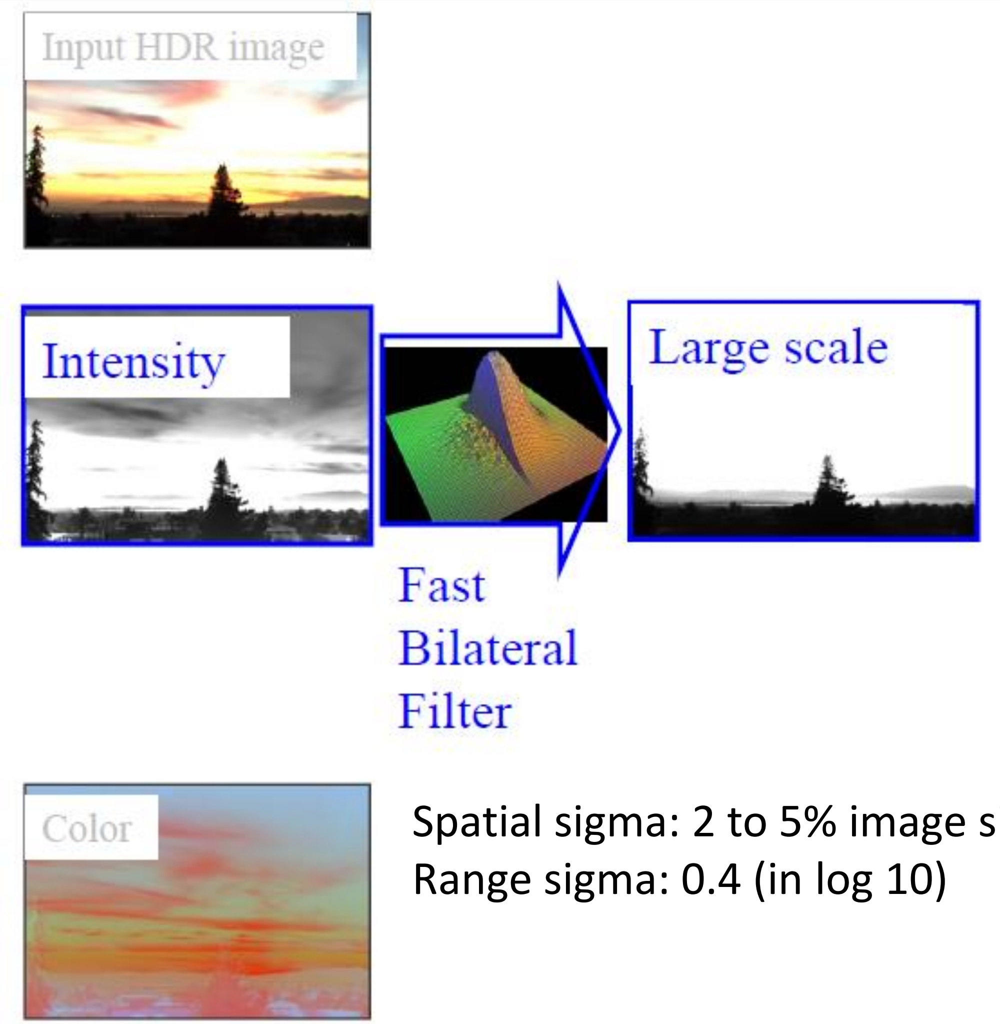

Output



Input

Tone Mapping by Bilateral Filtering

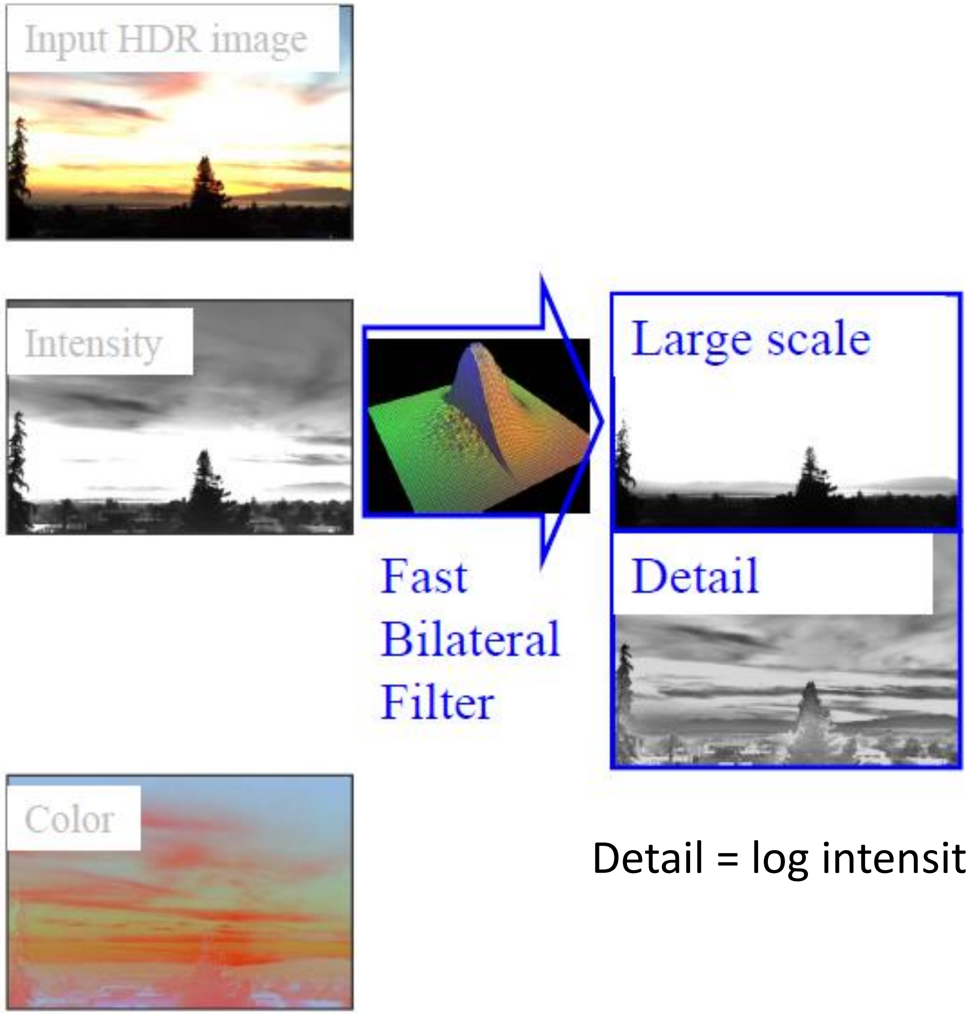
- **Do not blur details** by **non-linear filtering (Bilateral filtering)**



Spatial sigma: 2 to 5% image size
Range sigma: 0.4 (in log 10)

Tone Mapping by Bilateral Filtering

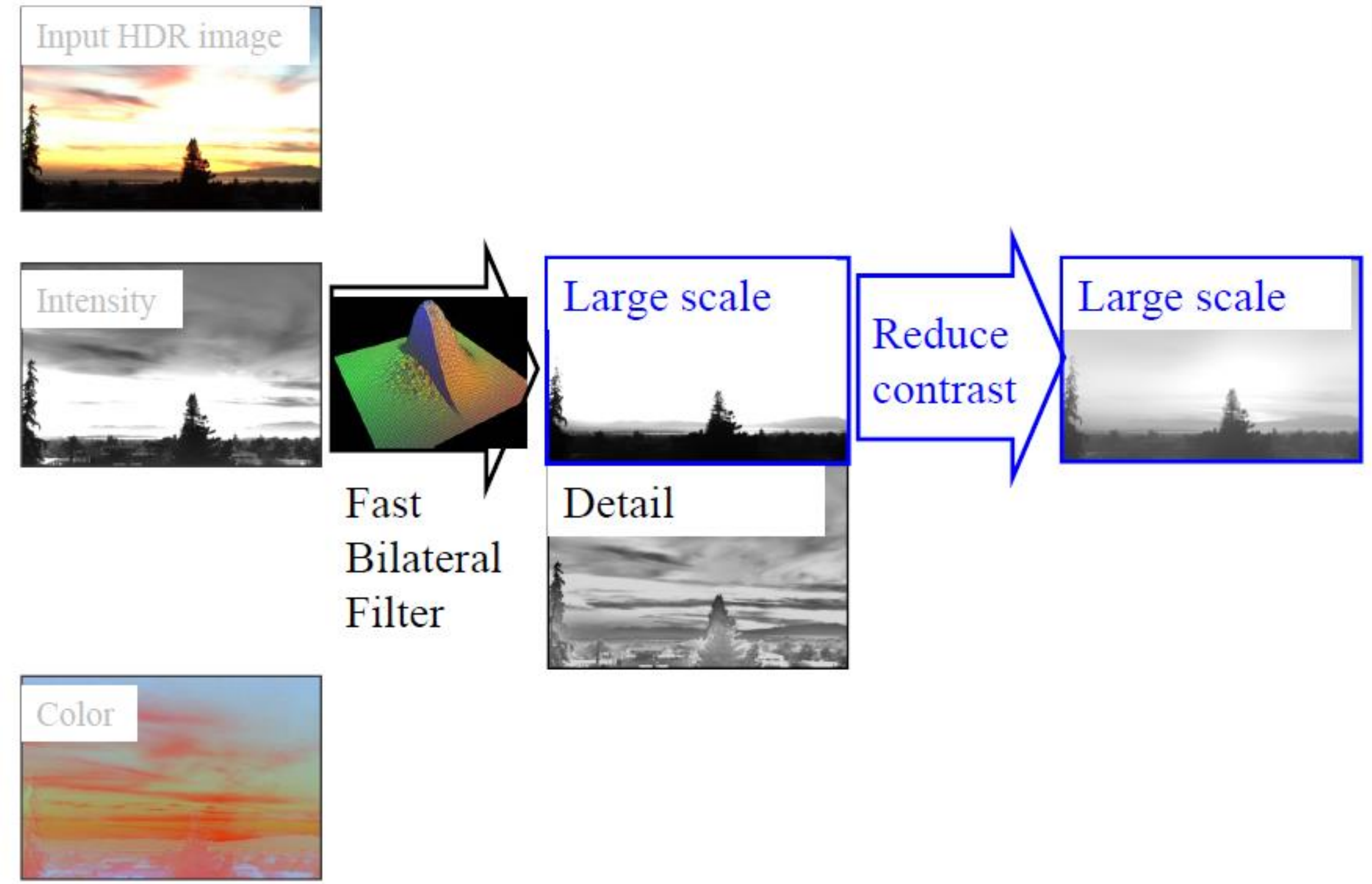
- **Do not blur details** by **non-linear filtering (Bilateral filtering)**



Detail = log intensity - large scale (residual)

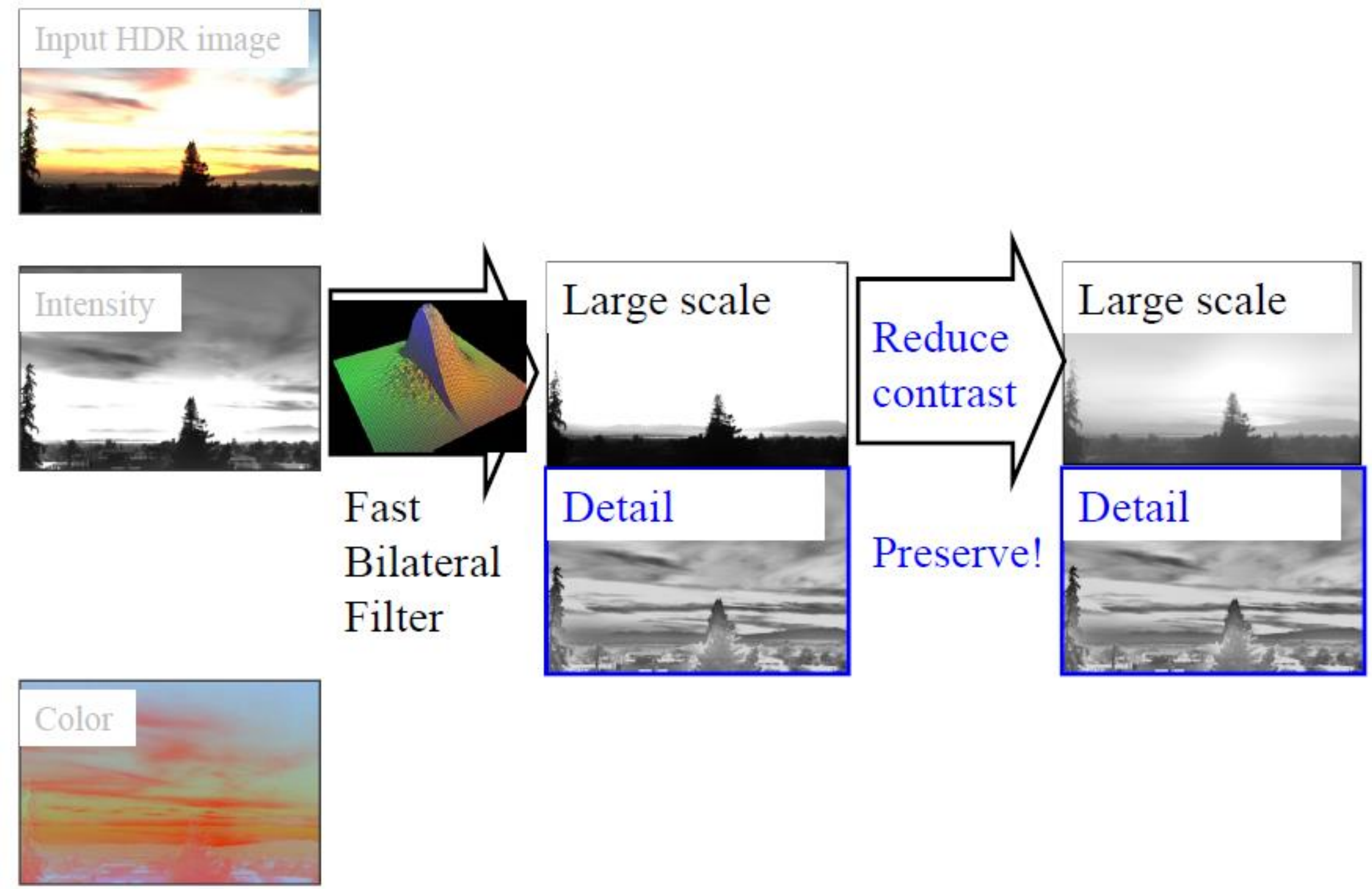
Tone Mapping by Bilateral Filtering

- **Do not blur details** by **non-linear filtering (Bilateral filtering)**



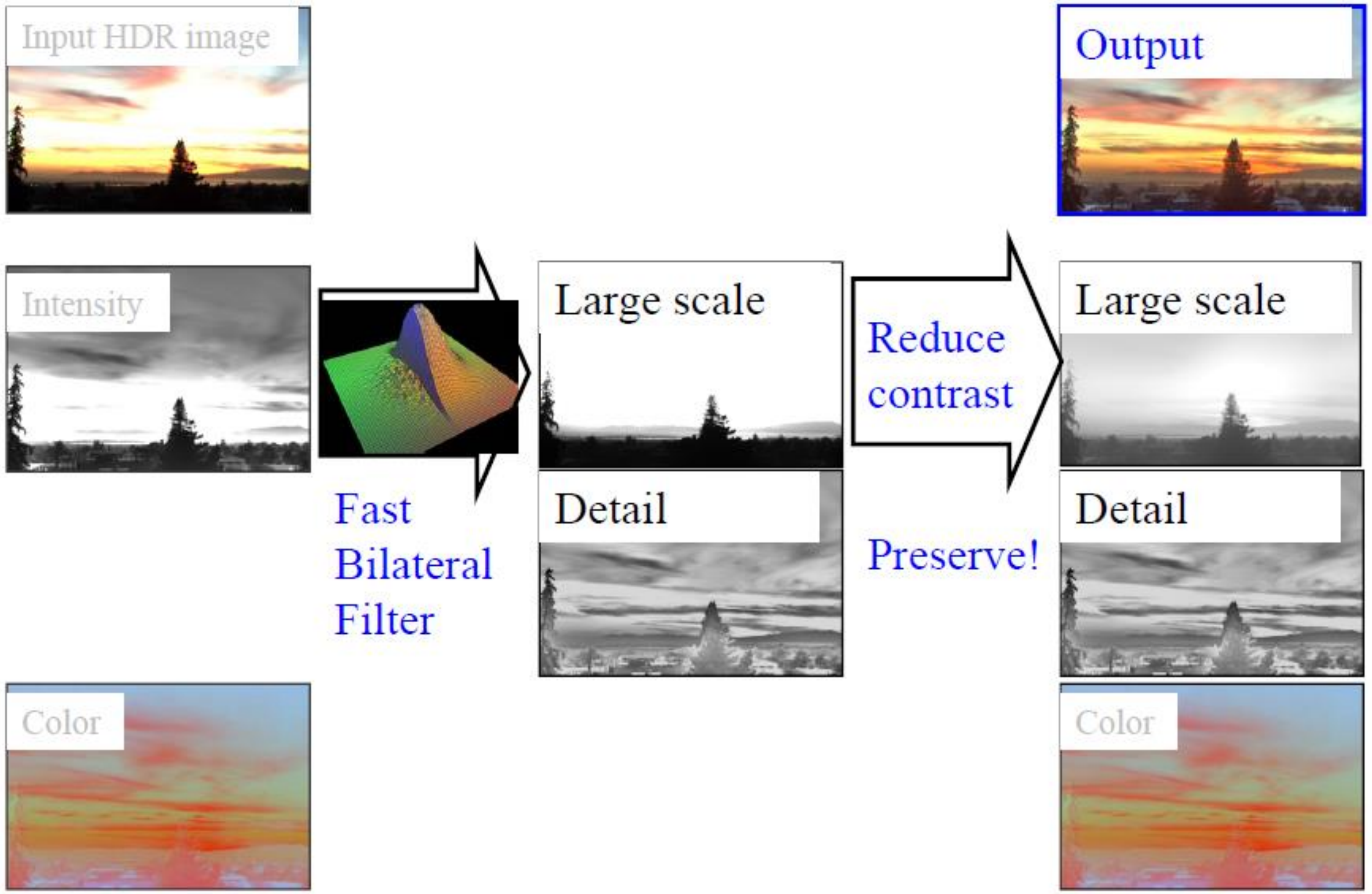
Tone Mapping by Bilateral Filtering

- **Do not blur details** by **non-linear filtering (Bilateral filtering)**



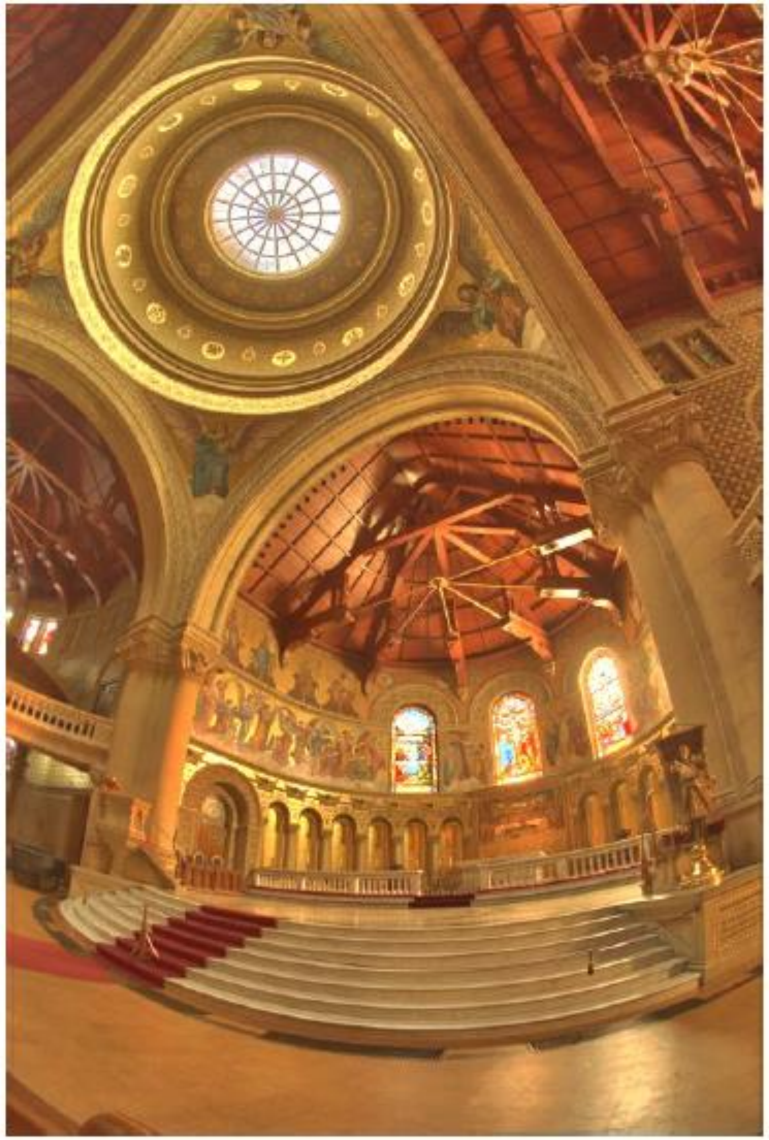
Tone Mapping by Bilateral Filtering

- **Do not blur details** by **non-linear filtering (Bilateral filtering)**

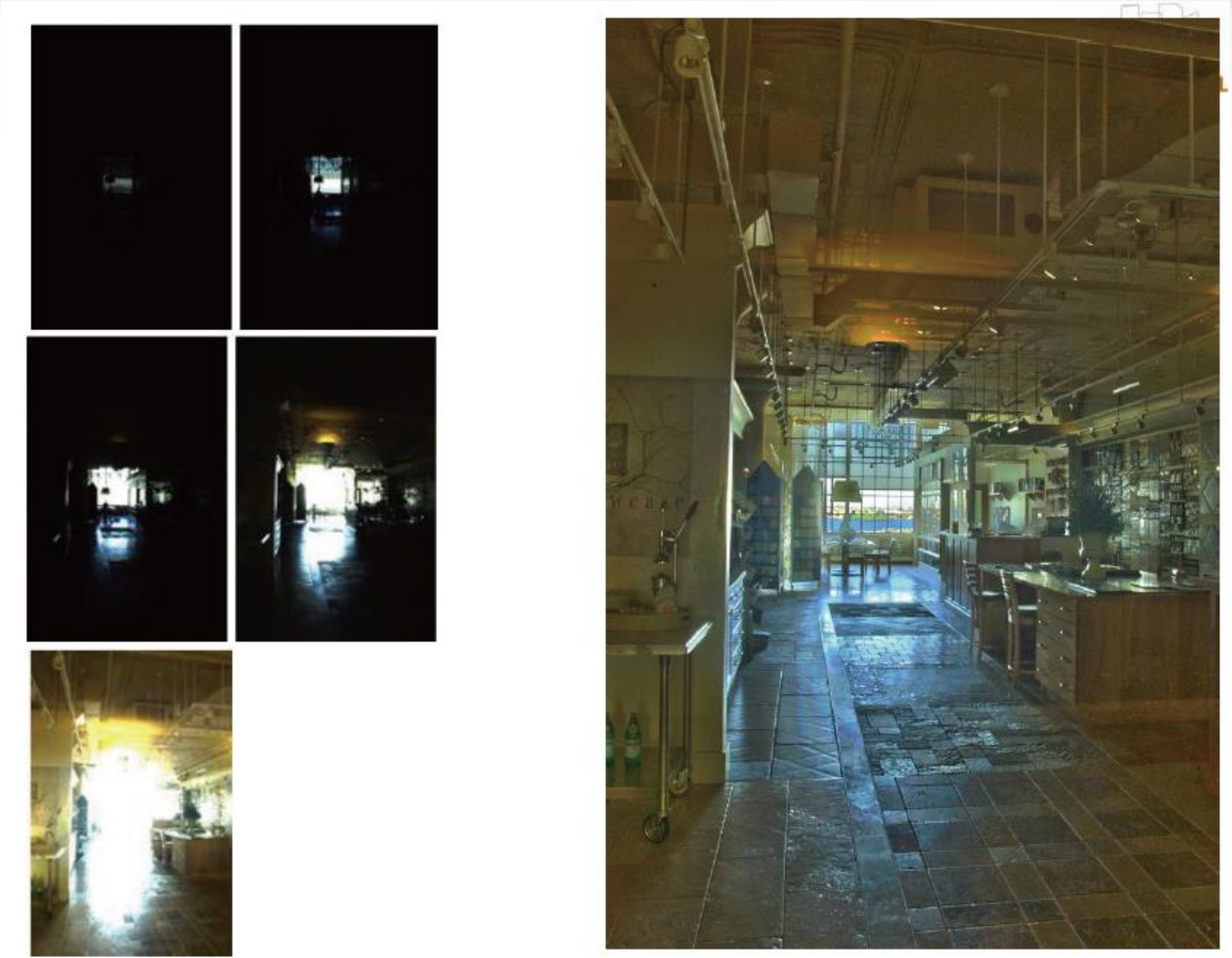


HDR Imaging results

- Tone mapped result of the four input images



HDR Imaging results



Accelaration

- Bilateral filtering is slow

- Fast algorithm: bilateral grid

- http://groups.csail.mit.edu/graphics/bilagrid/
- http://people.csail.mit.edu/sparis/publi/2009/ijcv/Paris_09_Fast_Approximation.pdf
- http://graphics.stanford.edu/papers/gkdtrees/

HDR Storage

RGBE Format

- Denoted by “.pic” or “.hdr” extensions
  (Developed by G. Ward for Radiance system)
- It is based on a 4 bytes representation
  - 8-bit mantissas (R, G, B)
  - Append to each pixel 8-bits for the common exponent
- Absolute accuracy of about 1%, covering a range of over 76 orders of magnitude
- Total 32-bit per pixel => 3 times smaller than RAW

HDR Storage

Other Formats

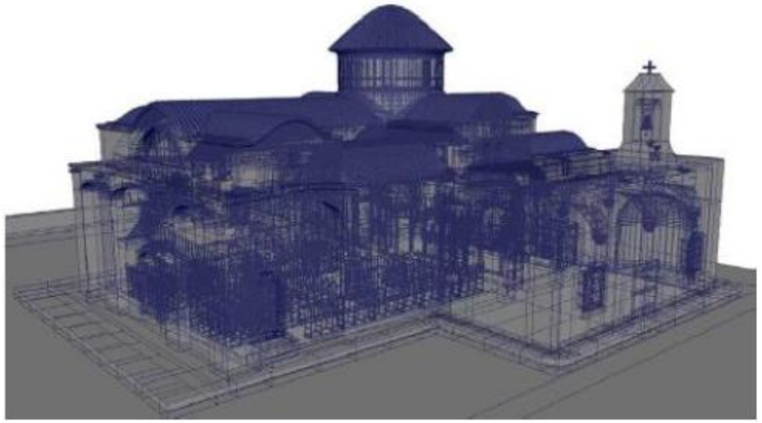
- LogLUV
- OpenEXR (.exr)
- Tiff (extension)
- Compressed formats

HDR Applications - IBL

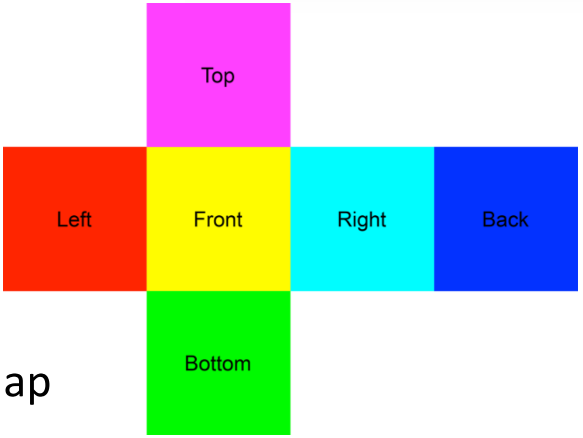
Image Based Lighting (IBL)

- HDR images contain physical values of light
- Use HDRI to light virtual object
  - Commonly called augmented reality
- Used
  - Entertainment
  - Advertising
  - Design

IBL-Rendering with natural lighting



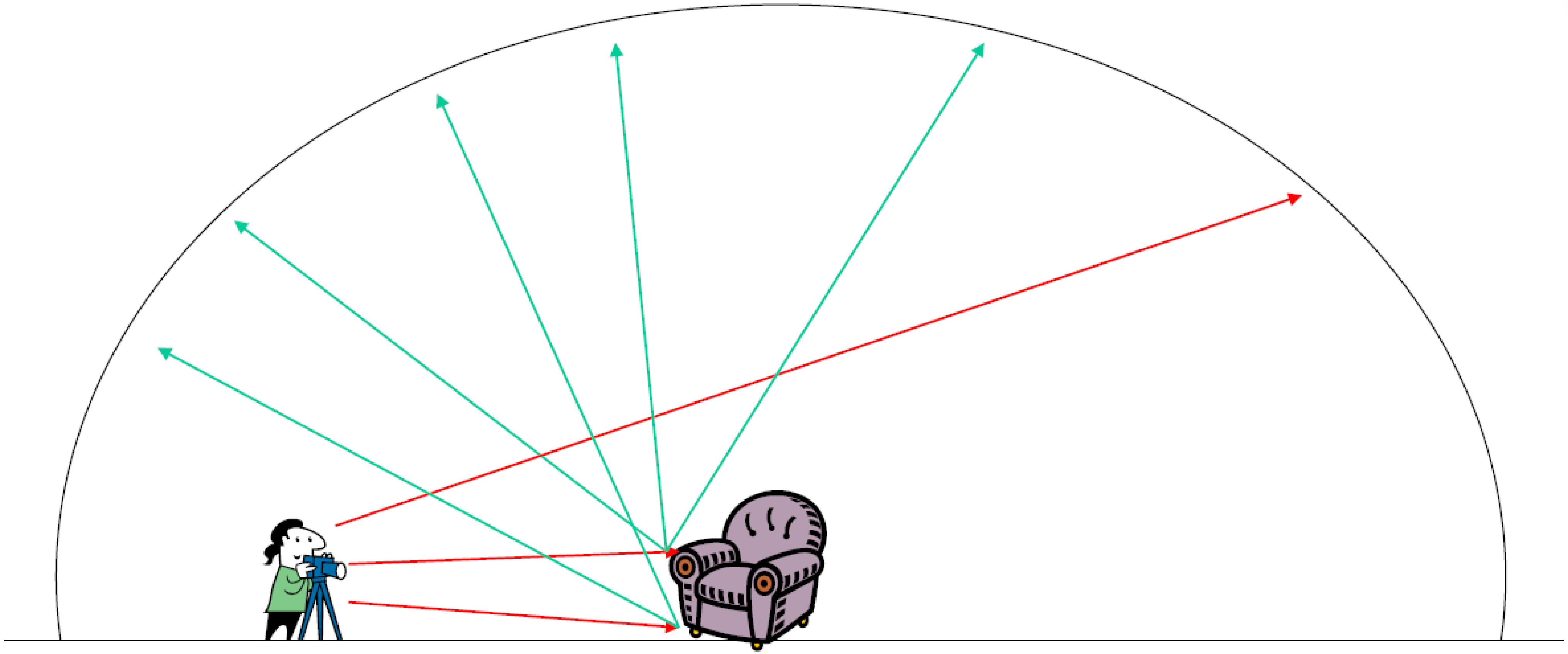
360 lighting map



equirectangular

Cube map

IBL



Indoor Scene by IBL