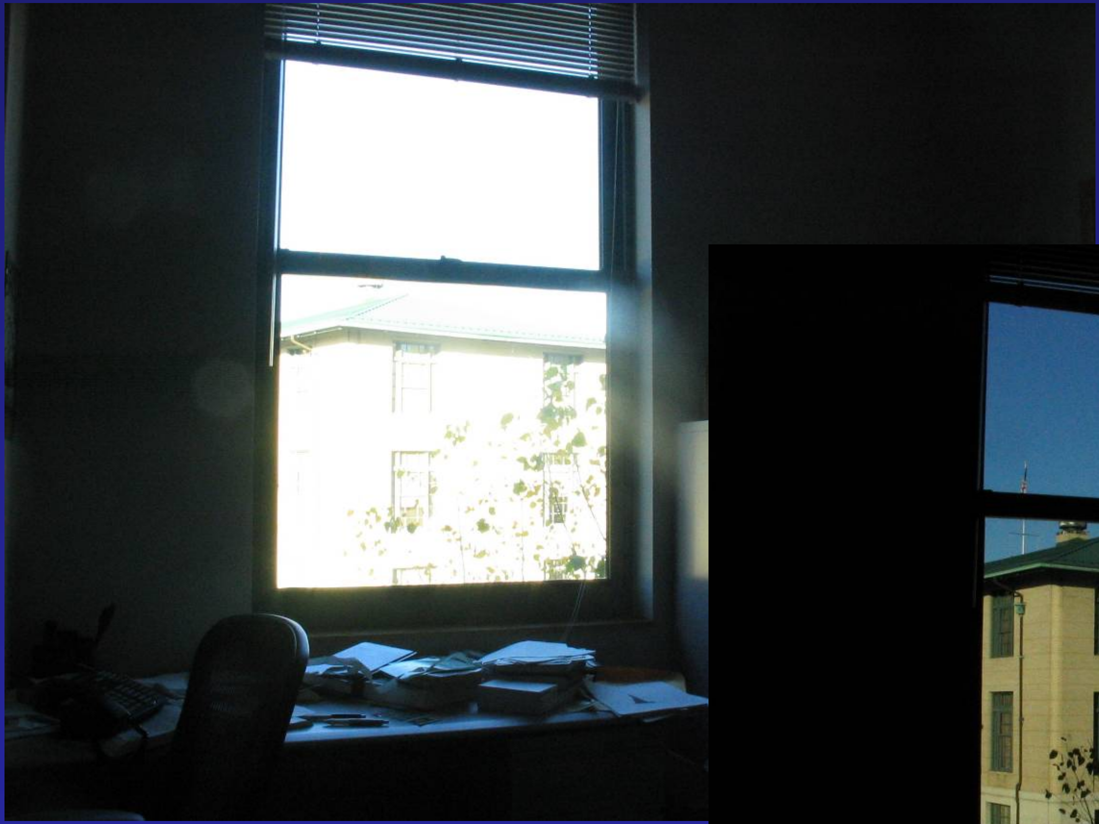
High Dynamic Range Images

© Alyosha Efros

15-463: Computational Photography
Alexei Efros, CMU, Fall 2005

…with a lot of slides stolen from Paul Debevec and Yuanzhen Li,

The Grandma Problem



Problem: Dynamic Range

The real world is high dynamic range.

1

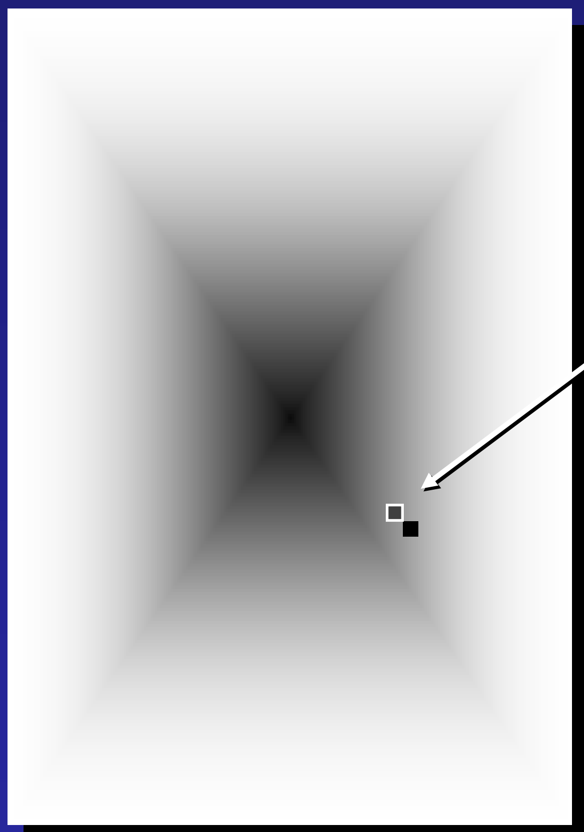
1500

25,000

400,000

2,000,000,000

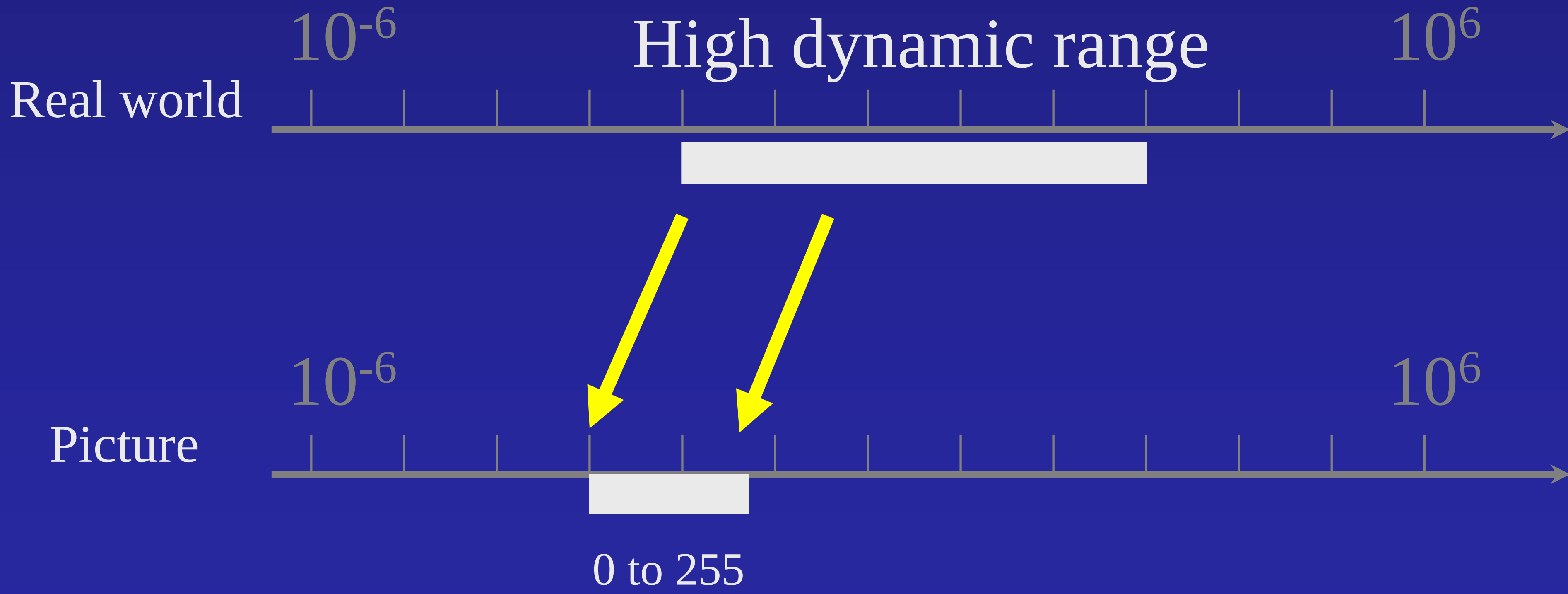
Image



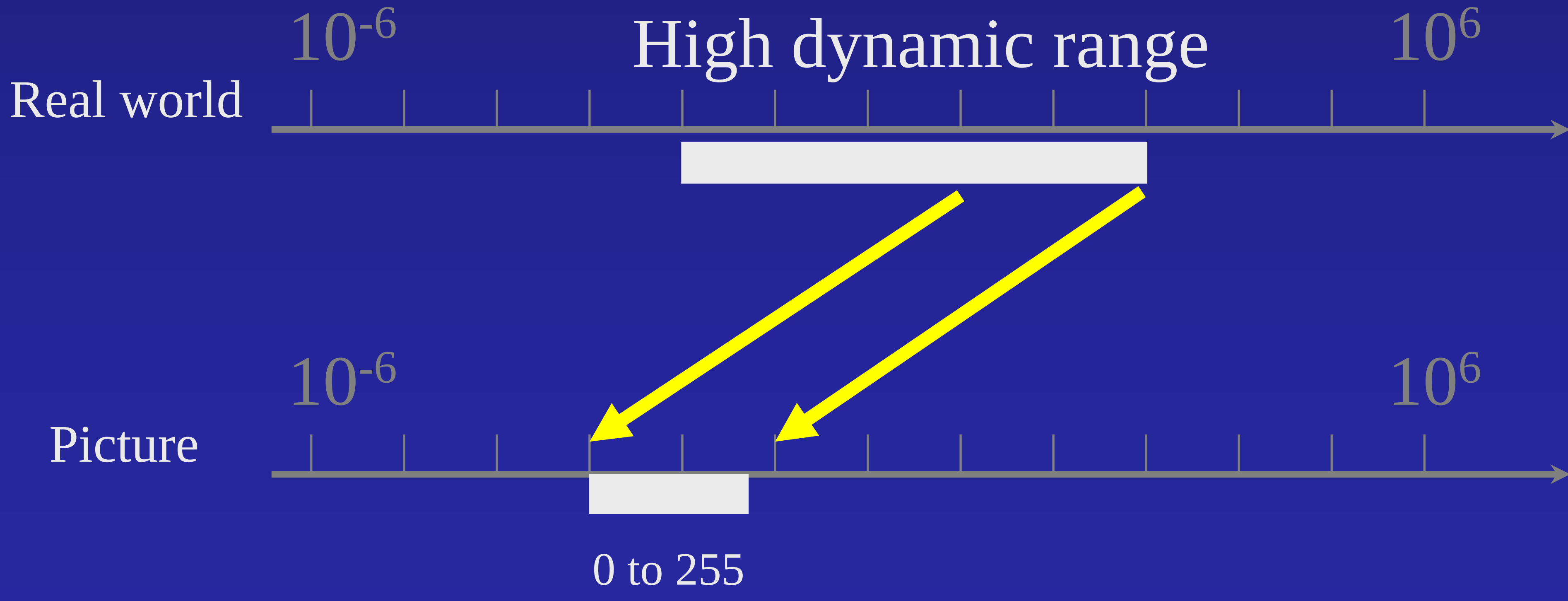
pixel (312, 284) = **42**

42 photos?

Long Exposure

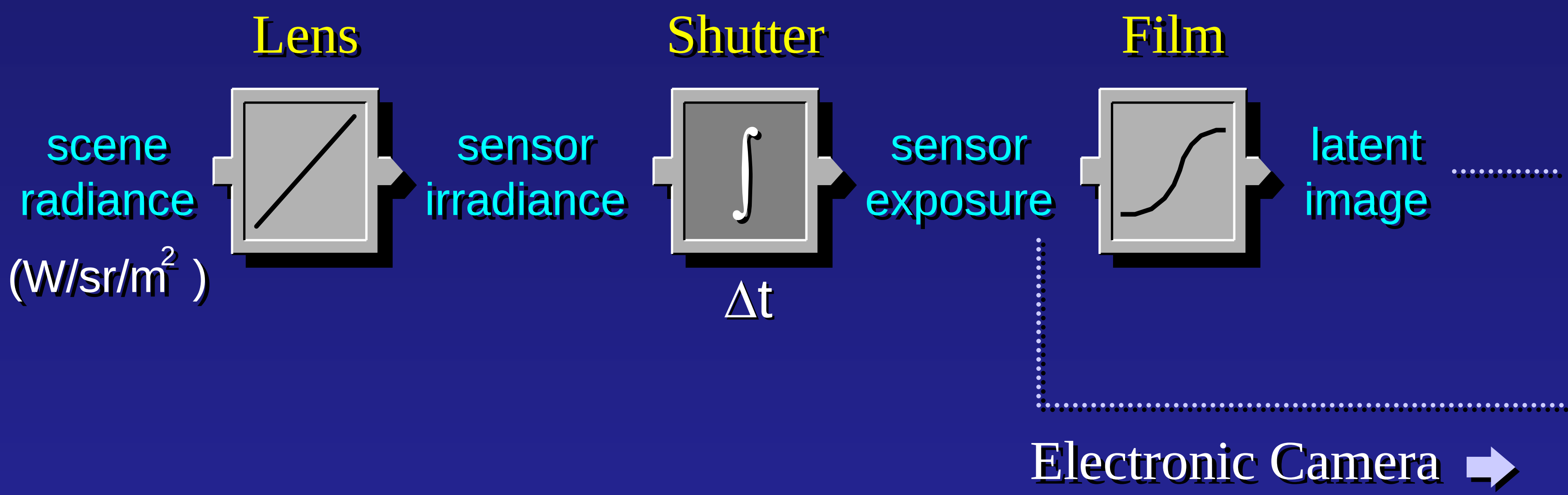


Short Exposure

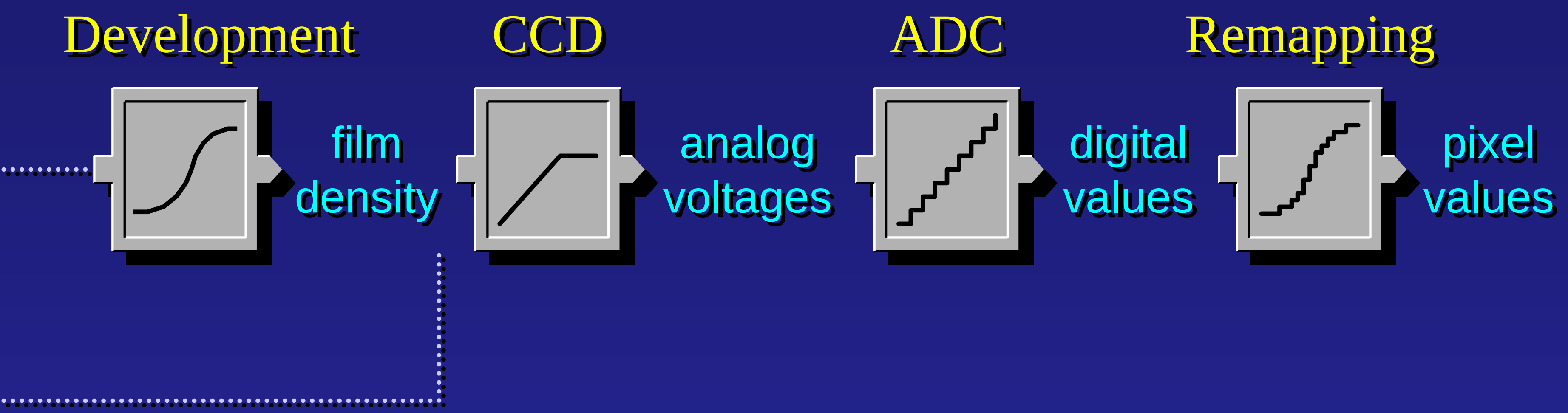


Camera Calibration

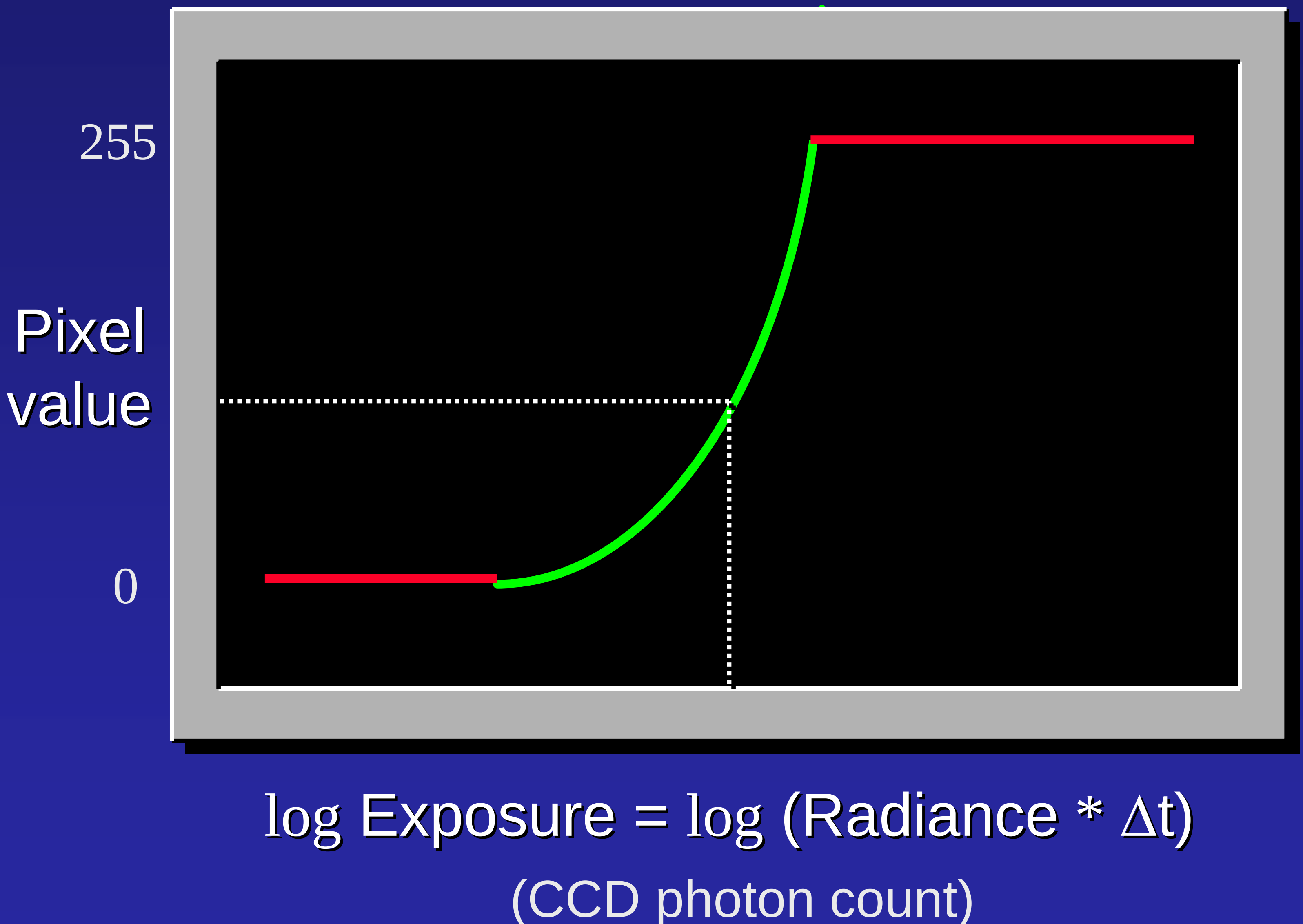
- Geometric
  - How pixel **coordinates** relate to **directions** in the world
- Photometric
  - How pixel **values** relate to **radiance** amounts in the world



The Image Acquisition Pipeline


Development
CCD
ADC
Remapping
film density
analog voltages
digital values
pixel values

Imaging system response function



Varying Exposure

Camera is not a photometer!

- Limited dynamic range
  - ⇒ Perhaps use multiple exposures?
- Unknown, nonlinear response
  - ⇒ Not possible to convert pixel values to radiance
- Solution:
  - Recover response curve from multiple exposures, then reconstruct the ***radiance map***

Recovering High Dynamic Range Radiance Maps from Photographs

Paul Debevec
Jitendra Malik

Computer Science Division
University of California at Berkeley

August 1997

Ways to vary exposure

- Shutter Speed (*)
- F/stop (aperture, iris)

- Neutral Density (ND) Filters

Shutter Speed

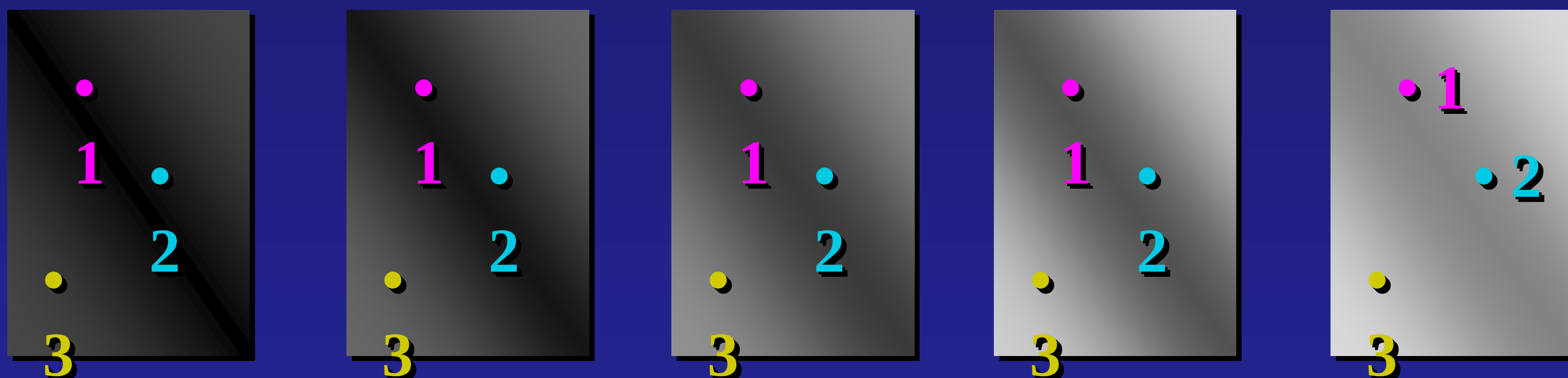
- Ranges: Canon D30: 30 to 1/4,000 sec.
- Sony VX2000: ¼ to 1/10,000 sec.
- Pros:
- Directly varies the exposure
- Usually accurate and repeatable
- Issues:
- Noise in long exposures

Shutter Speed

- **Note: shutter times usually obey a power series – each “stop” is a factor of 2**
- **¼, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec**
- **Usually really is:**
- **¼, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec**

The Algorithm

Image series



$\Delta t =$ 1/64 sec | $\Delta t =$ 1/16 sec | $\Delta t =$ 1/4 sec | $\Delta t =$ 1 sec | $\Delta t =$ 4 sec

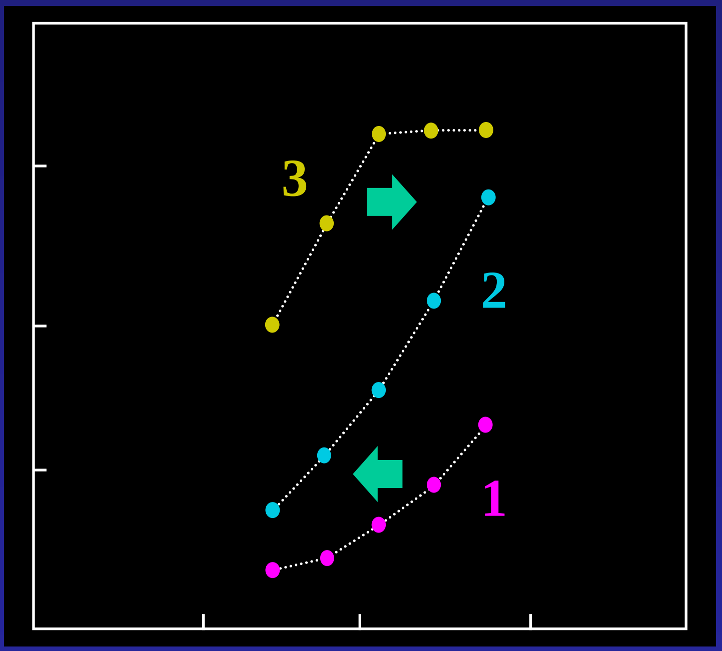
Pixel Value Z = f(Exposure)

Exposure = Radiance $\times$ Δt

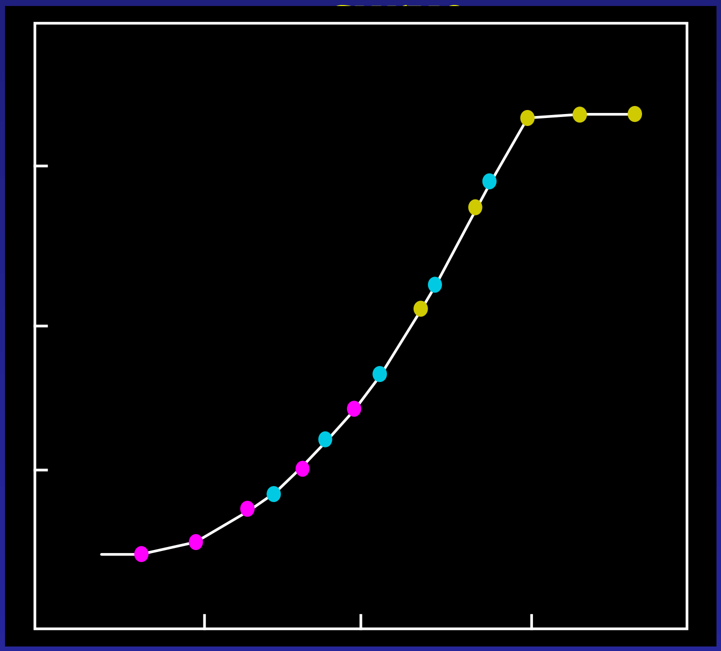
log Exposure = log Radiance + log Δt

Response Curve

Assuming unit radiance for each pixel



After adjusting radiances to obtain a smooth response



The Math

- Let *g(z)* be the *discrete* inverse response function
- For each pixel site *i* in each image *j*, want:

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

- Solve the overdetermined linear system:

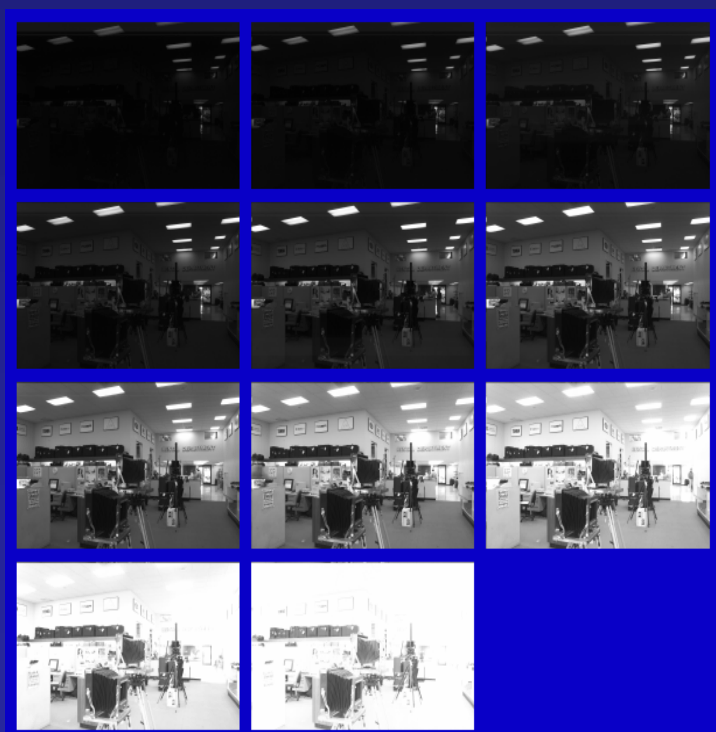
$$\sum_{i=1}^{N}\sum_{j=1}^{P}\underbrace{\left[\ln Radiance_i + \ln \Delta t_j - g(Z_{ij})\right]^2}_{\text{fitting term}} + \underbrace{\lambda \sum_{z=Z_{min}}^{Z_{max}} g''(z)^2}_{\text{smoothness term}}$$

Matlab Code

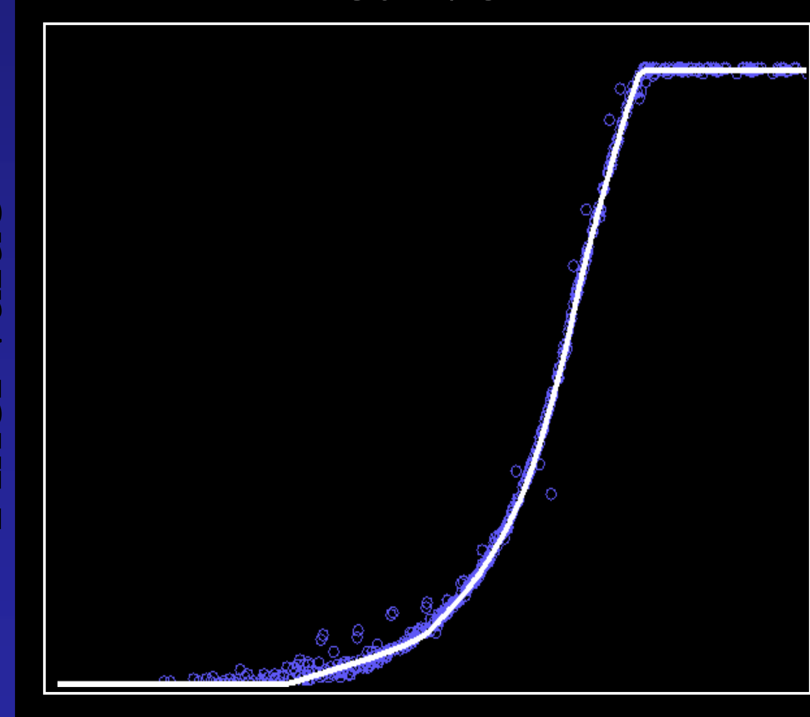
```
function [g,lE]=gsolve(Z,B,l,w)

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
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));
```

Results: Digital Camera

Kodak DCS460
1/30 to 30 sec



Recovered response curve



Pixel value

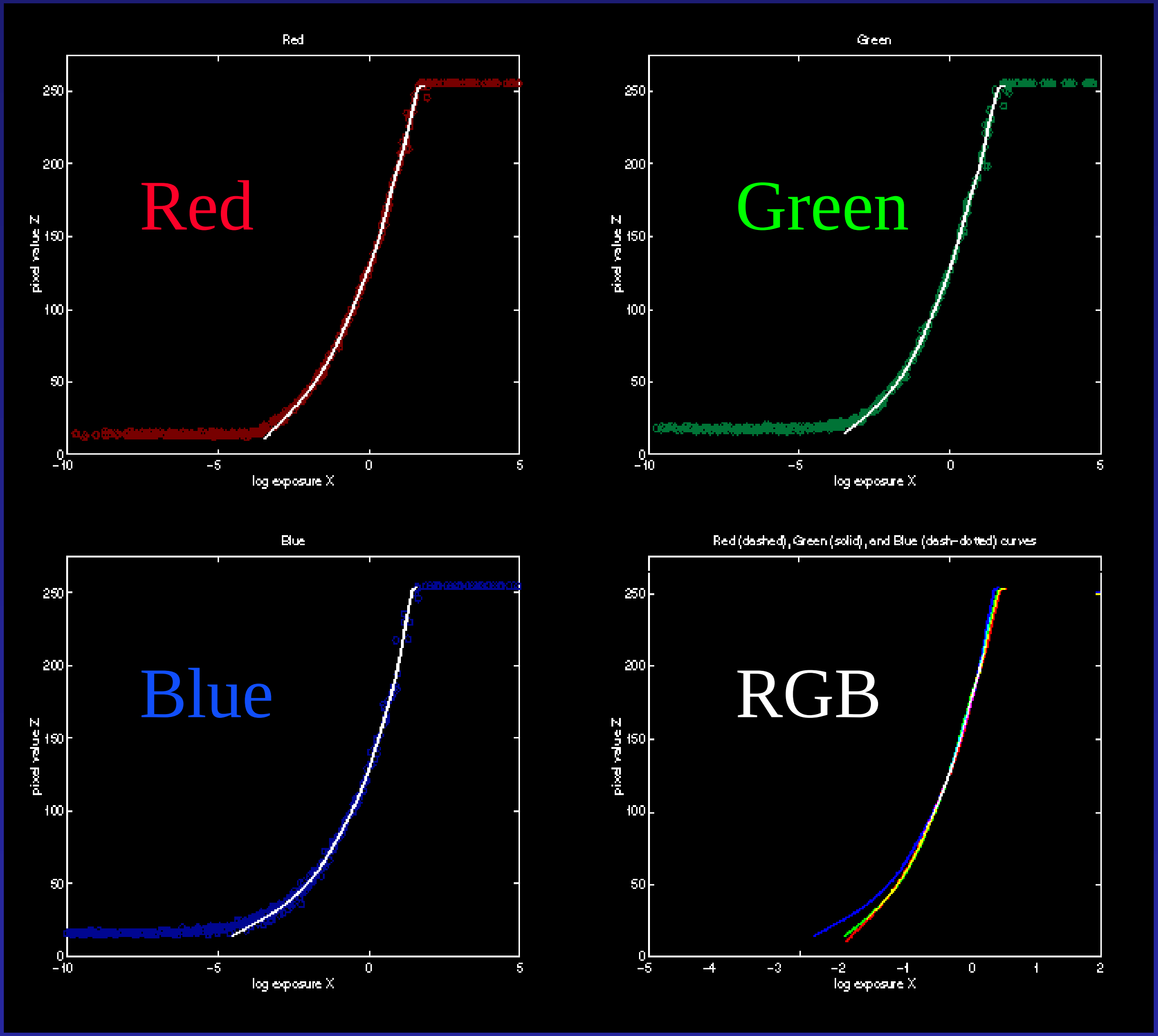
log Exposure

Reconstructed radiance map

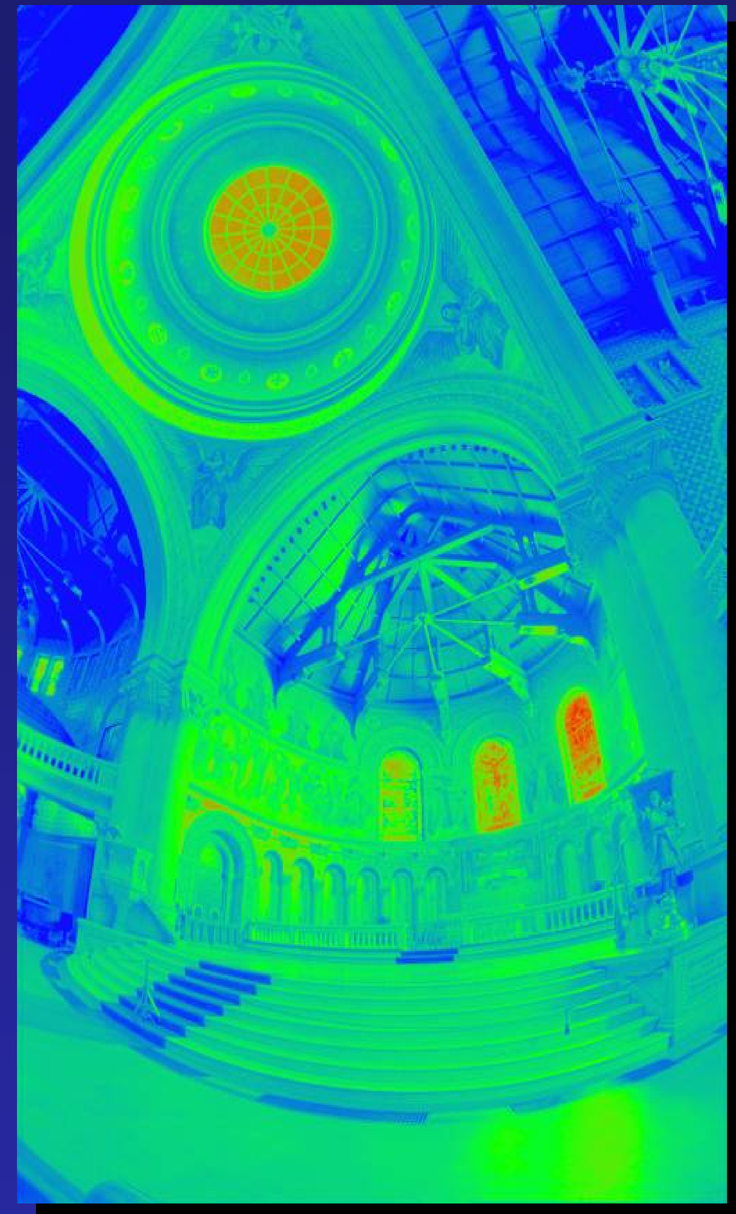
Results: Color Film

- Kodak Gold ASA 100, PhotoCD

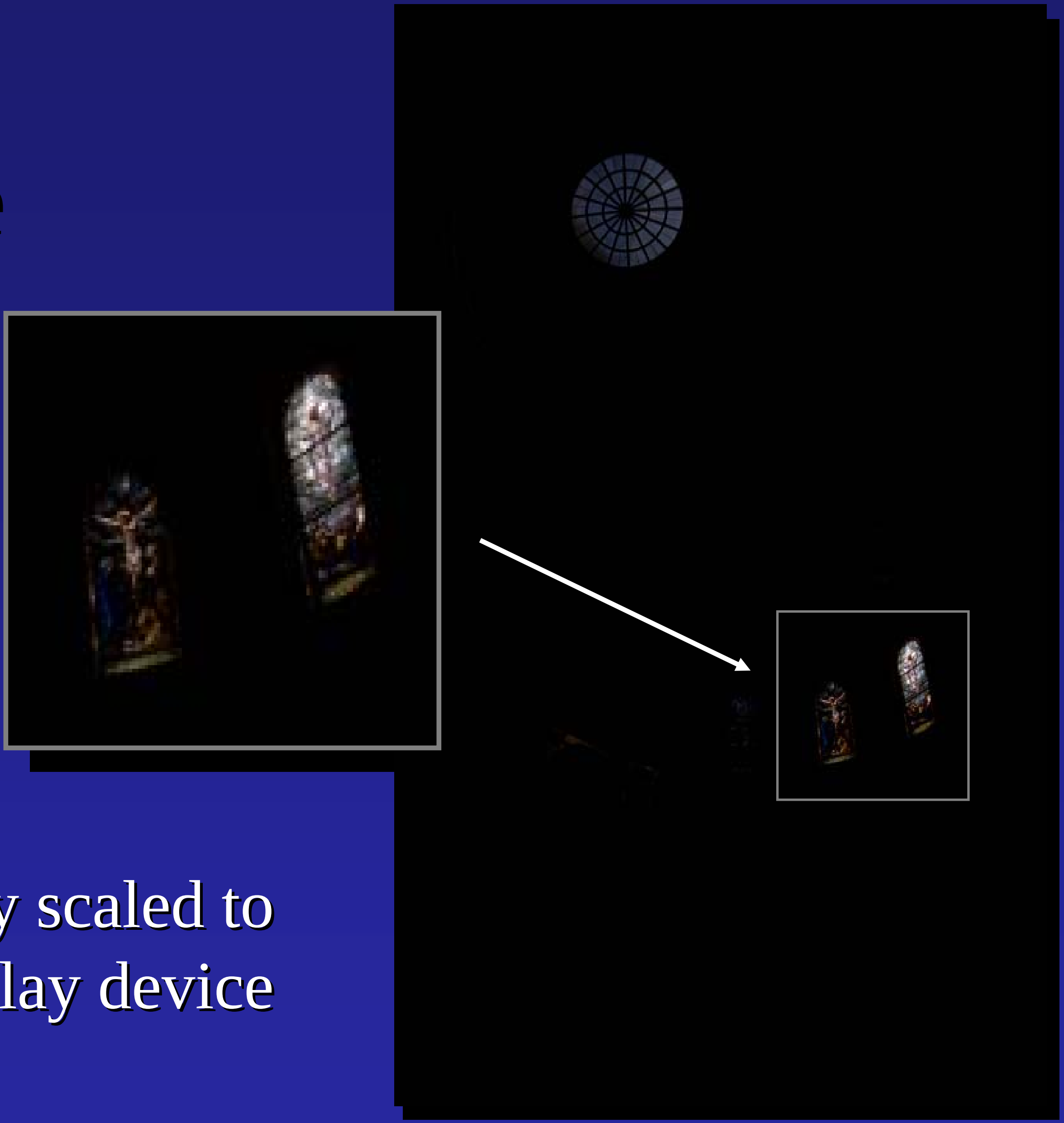
Recovered Response Curves



The Radiance Map



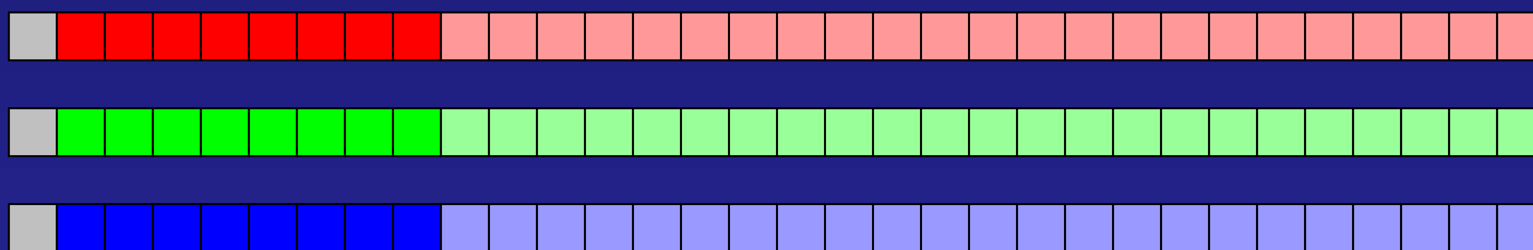
The Radiance Map



Linearly scaled to display device

Portable FloatMap (.pfm)

- 12 bytes per pixel, 4 for each channel



sign exponent mantissa

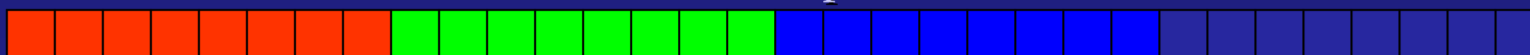
Text header similar to Jeff Poskanzer’s .ppm image format:

```
PF
768 512
1
<binary image data>
```

Floating Point TIFF similar

Radiance Format (.pic, .hdr)

32 bits / pixel



Red Green Blue Exponent

(145, 215, 87, 149) =

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

(1190000, 1760000, 713000)

(145, 215, 87, 103) =

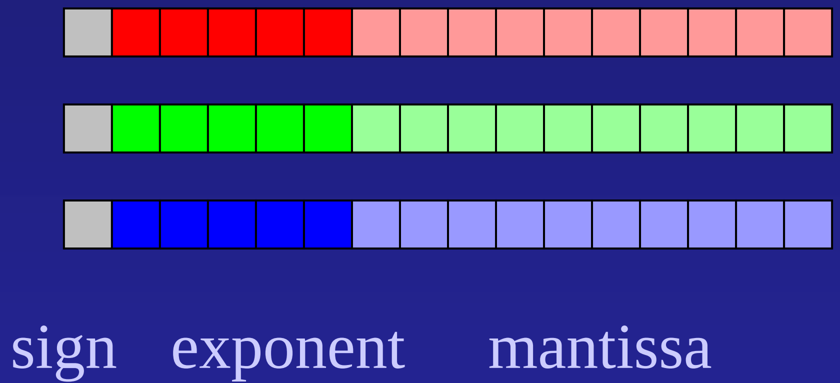
(145, 215, 87) * 2^(103-128) =

(0.00000432, 0.00000641, 0.00000259)

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

ILM's OpenEXR (.exr)

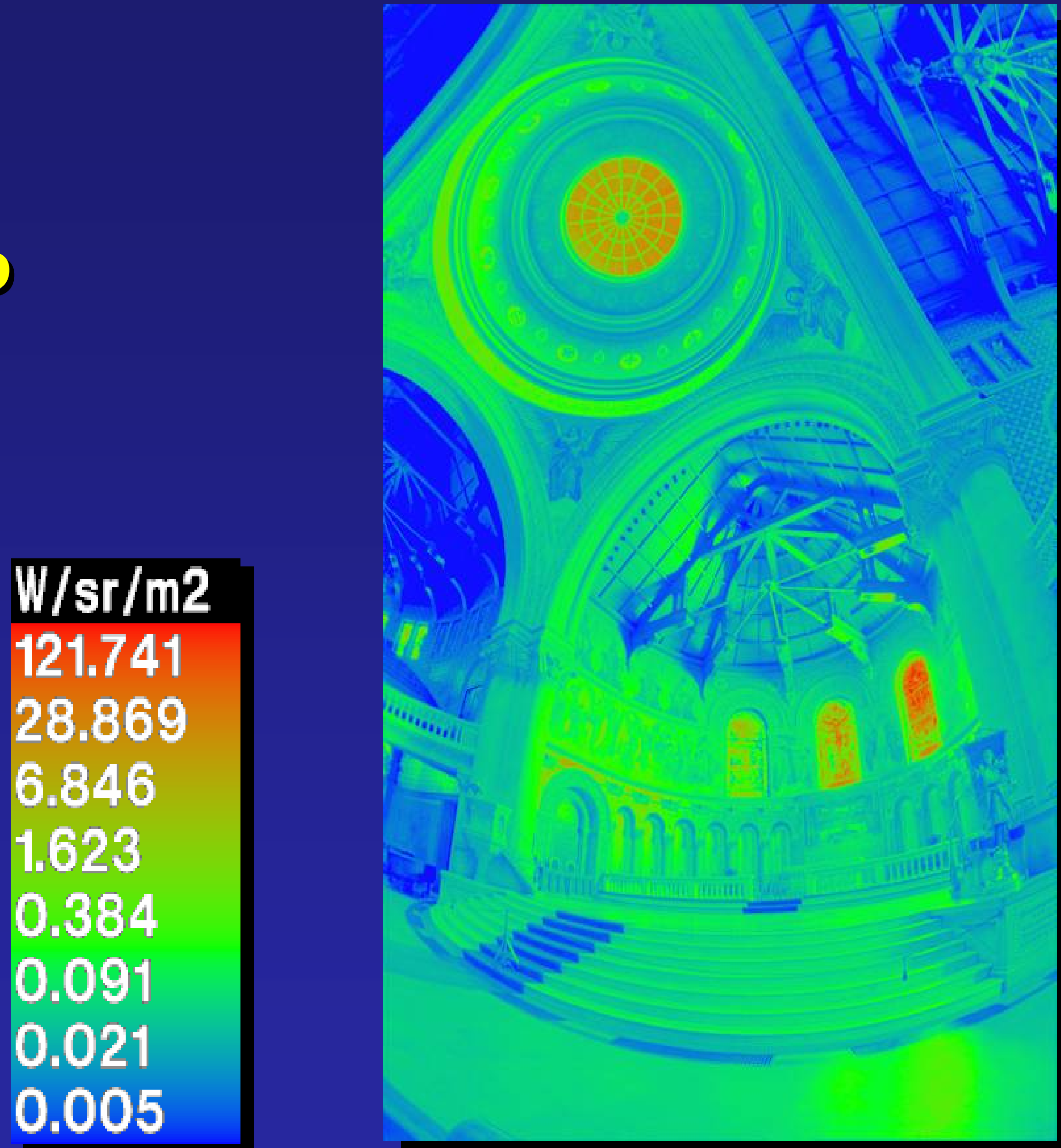
- 6 bytes per pixel, 2 for each channel, compressed



- Several lossless compression options, 2:1 typical
- Compatible with the "half" datatype in NVidia's Cg
- Supported natively on GeForce FX and Quadro FX

- Available at http://www.openexr.net/

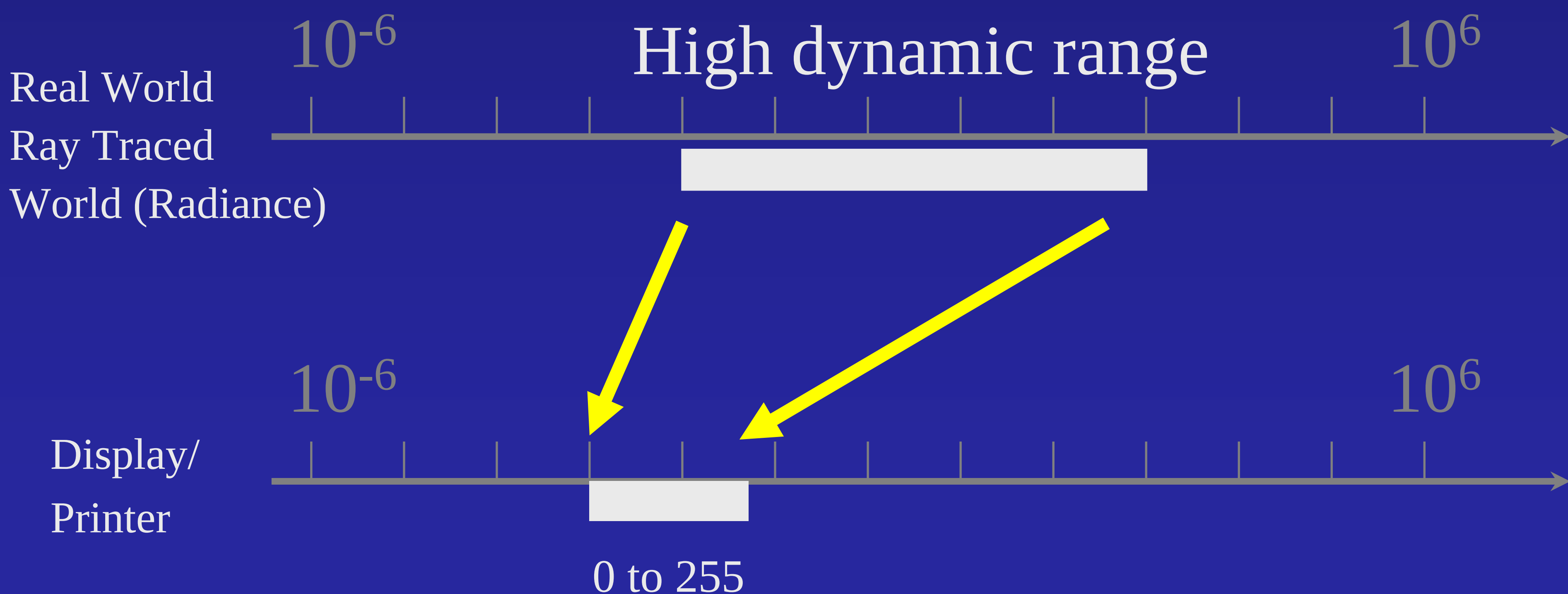
Now What?



Tone Mapping

- How can we do this?

  Linear scaling?, thresholding? Suggestions?

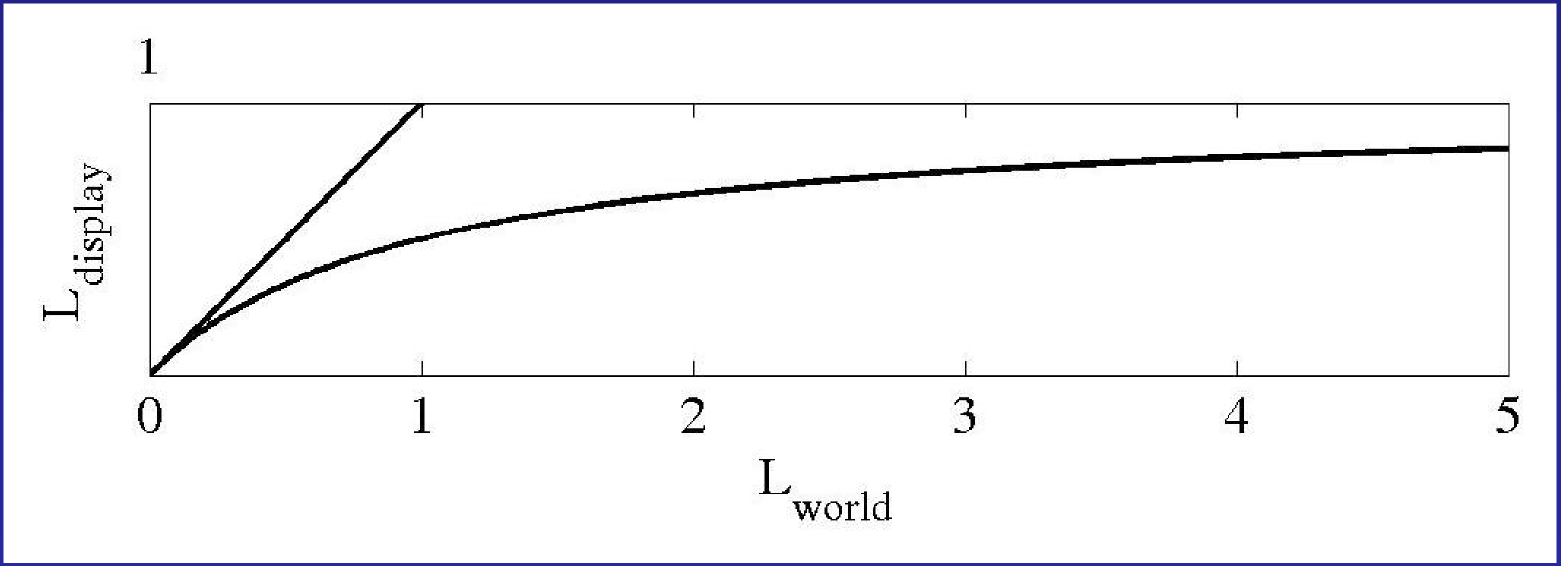


Simple Global Operator

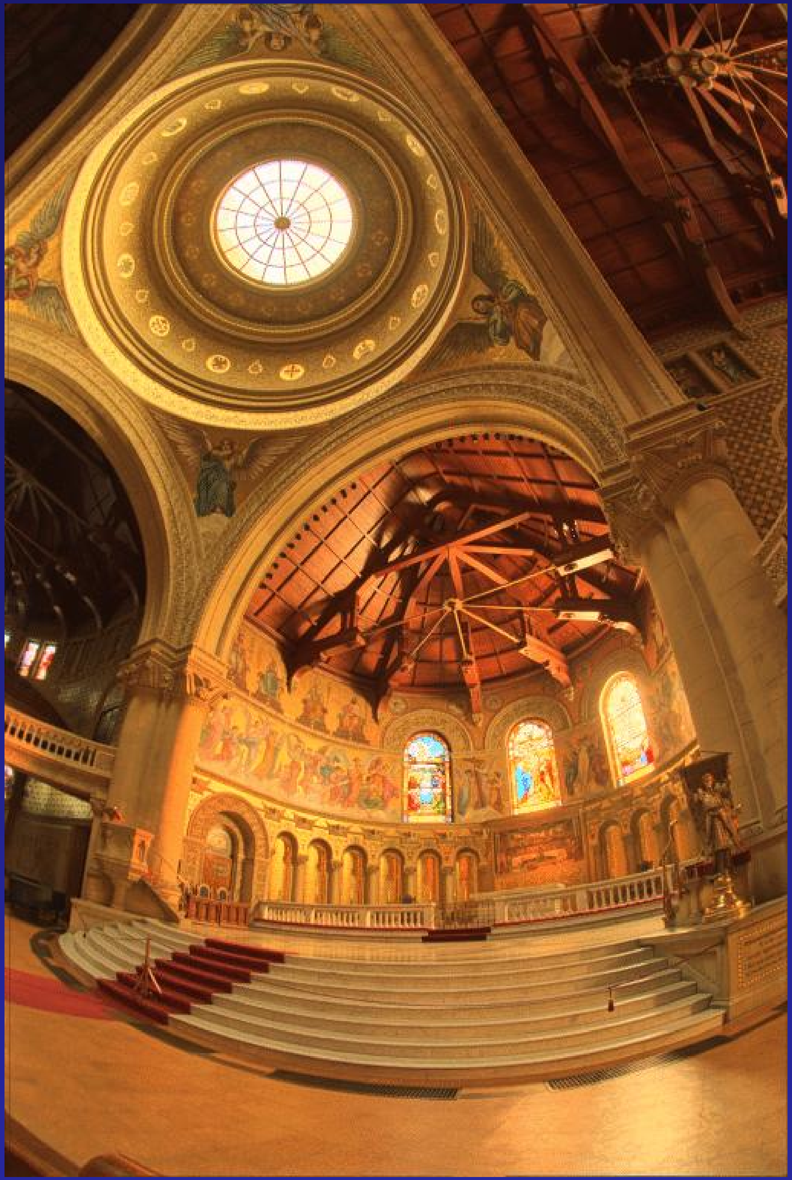
- Compression curve needs to
  - Bring everything within range
  - Leave dark areas alone
- In other words
  - Asymptote at 255
  - Derivative of 1 at 0

Global Operator (Reinhart et al)

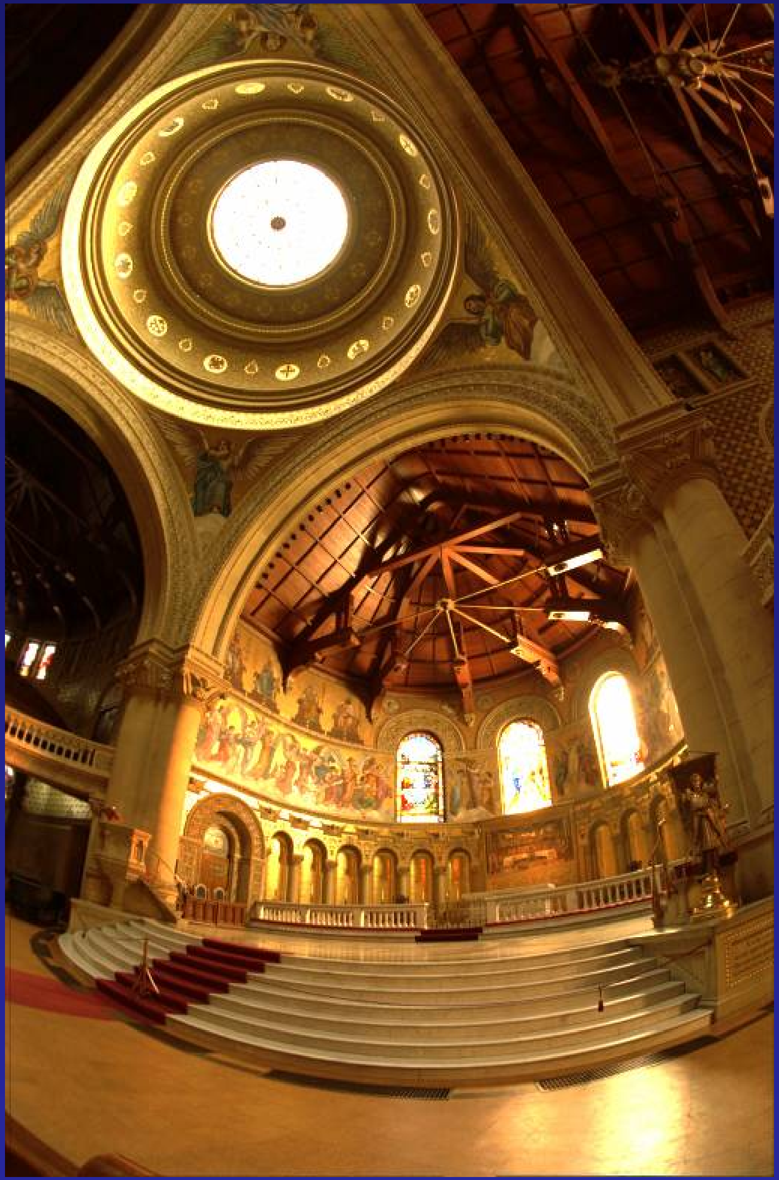
$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$



Global Operator Results

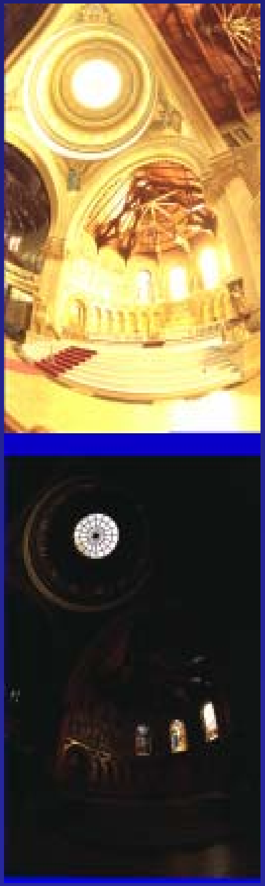


Reinhart Operator

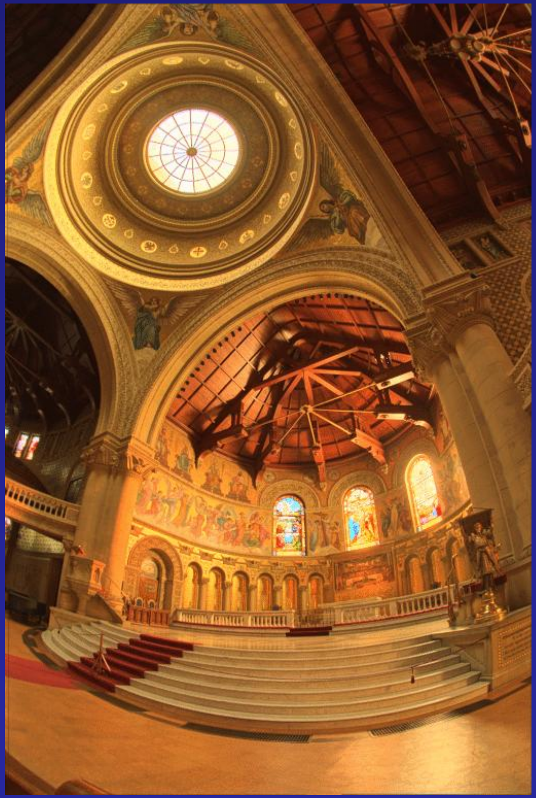


Darkest 0.1% scaled to display device

What do *we* see?



Vs.



What does the eye sees?

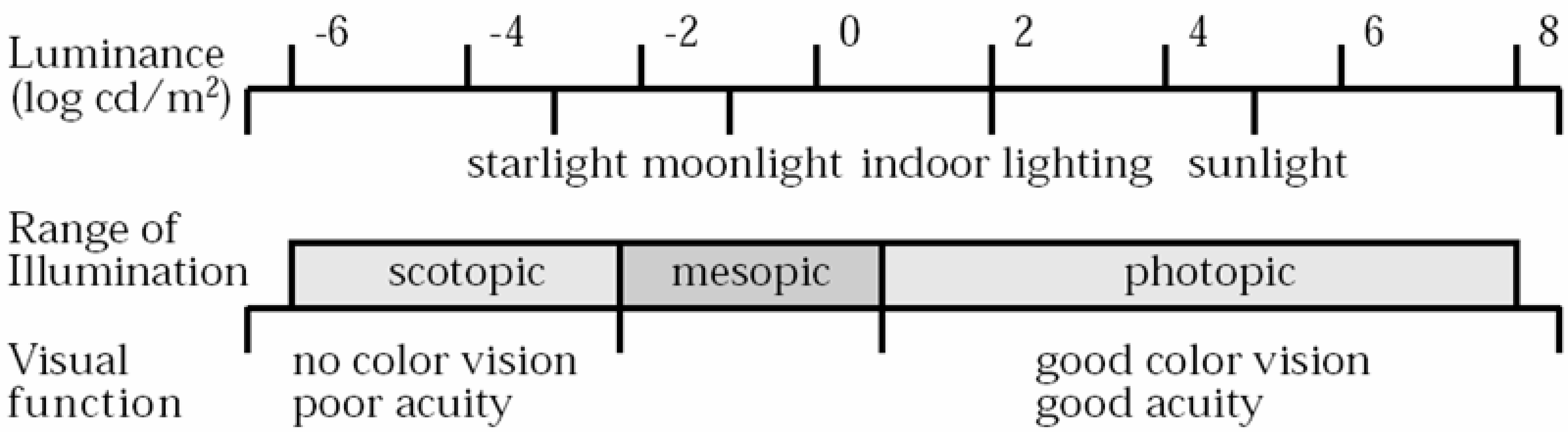
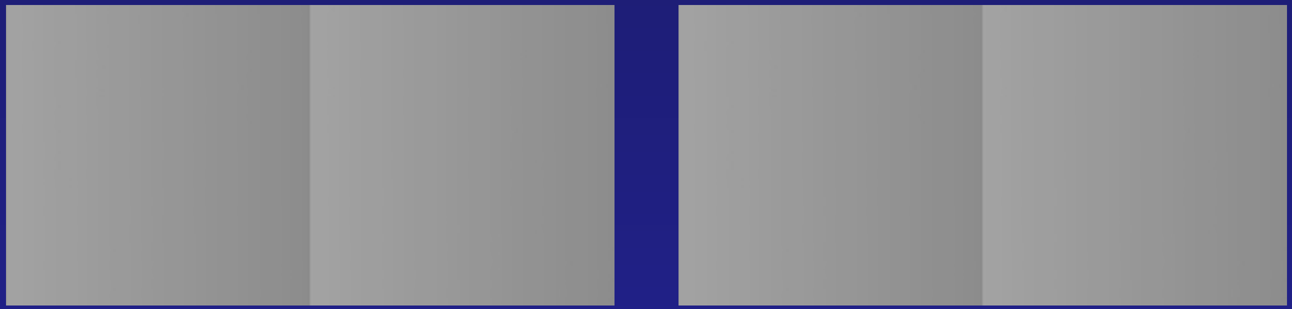


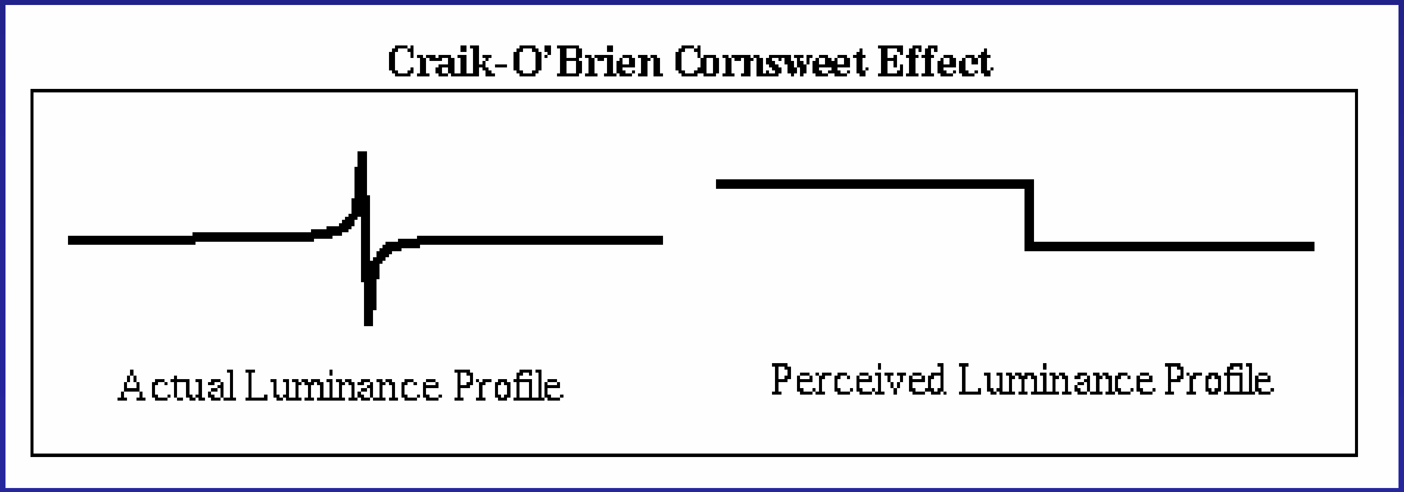
Figure 1: The range of luminances in the natural environment and associated visual parameters. After Hood (1986).

The eye has a huge dynamic range

Do we see a true radiance map?

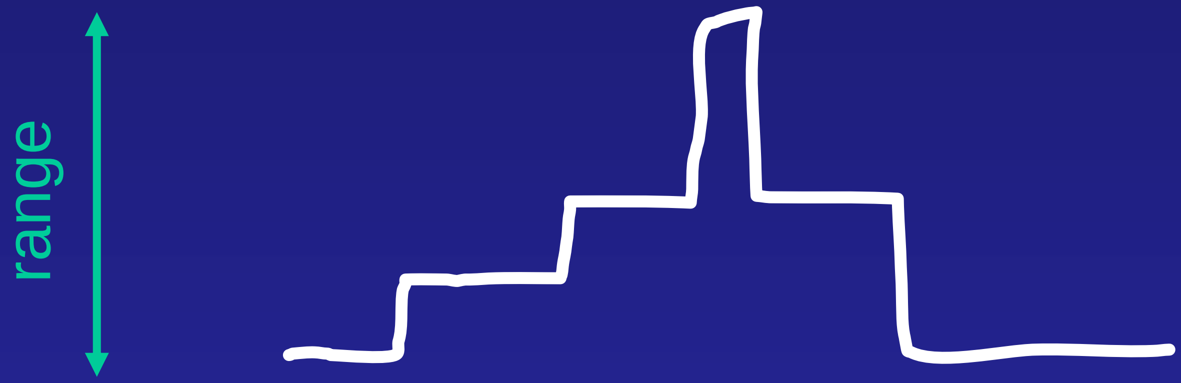
Metamores





Can we use this for range compression?

Compressing Dynamic Range



SIGGRAPH2005

Compressing and Companding High Dynamic Range Images with Subband Architectures

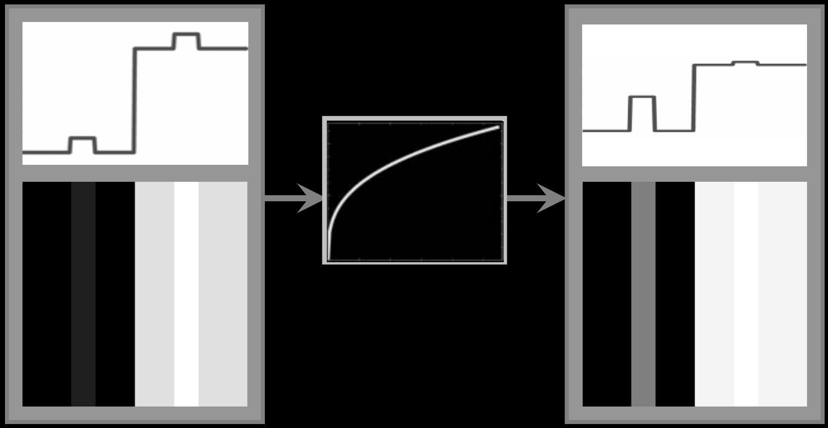
Yuanzhen Li, Lavanya Sharan,
Edward Adelson
Massachusetts Institute of Technology

Dynamic Range Problem

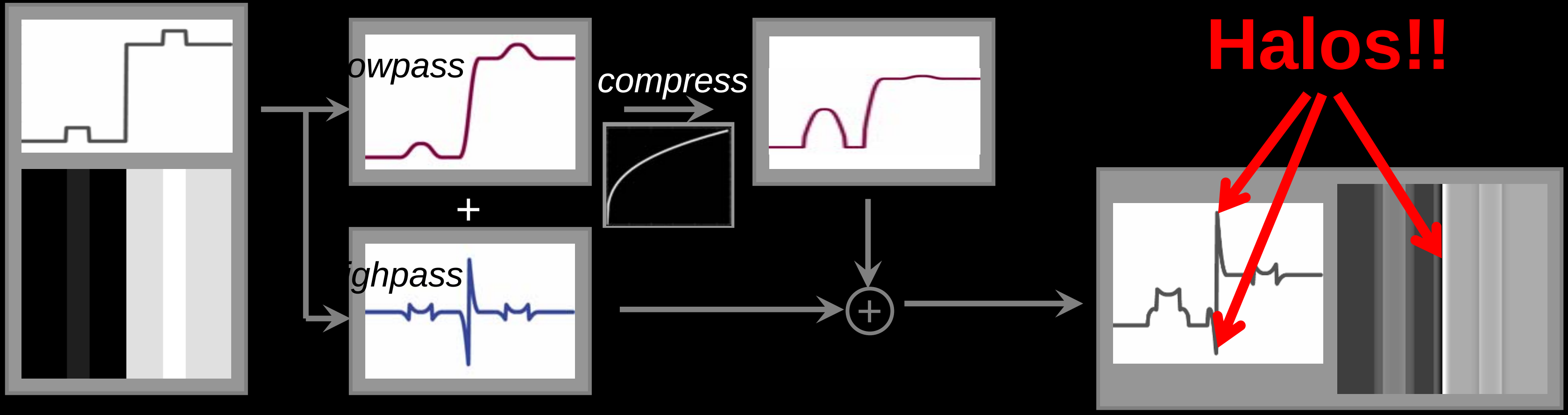
Source: Shree Nayar

Range Compression

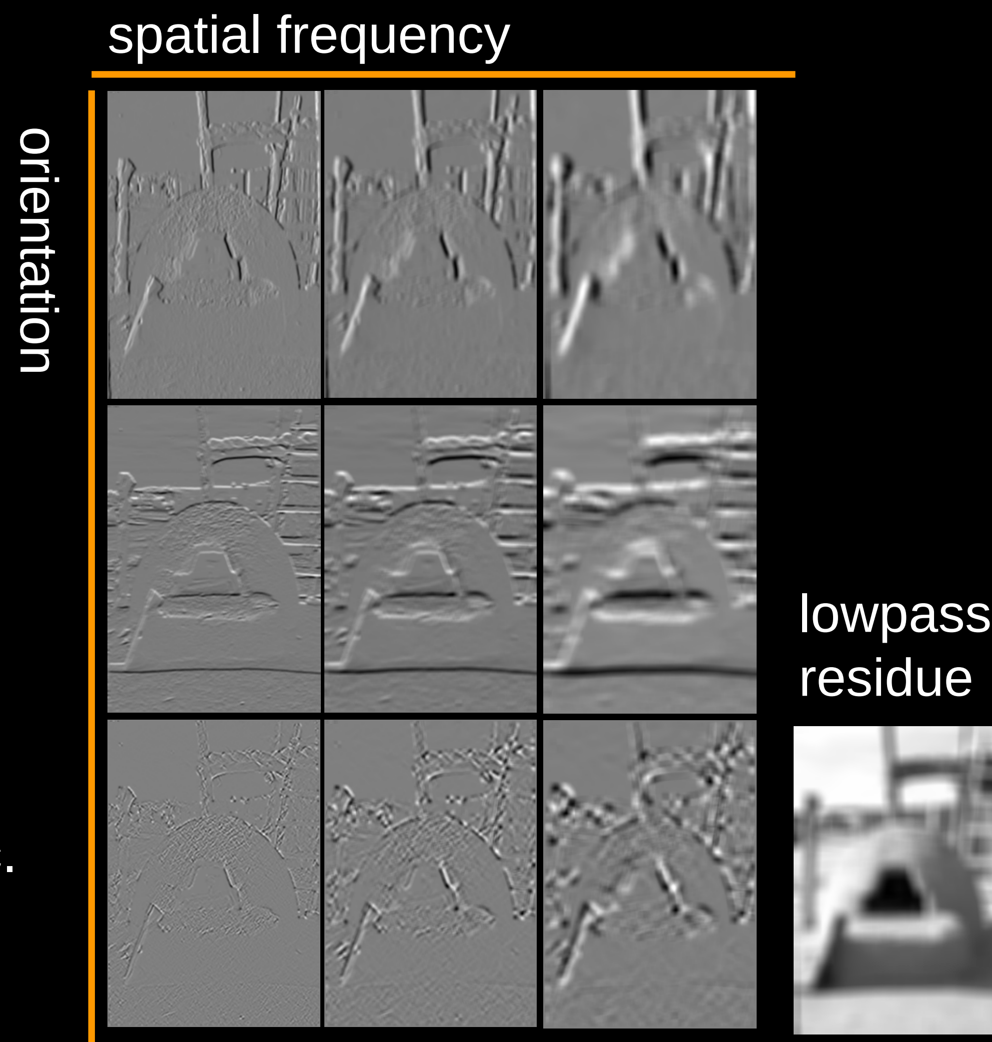
Method: Gamma or log on intensities.
Problem: loss of detail.



Solution: filtering.
Problem: halos.



Multiscale Subband Decomposition



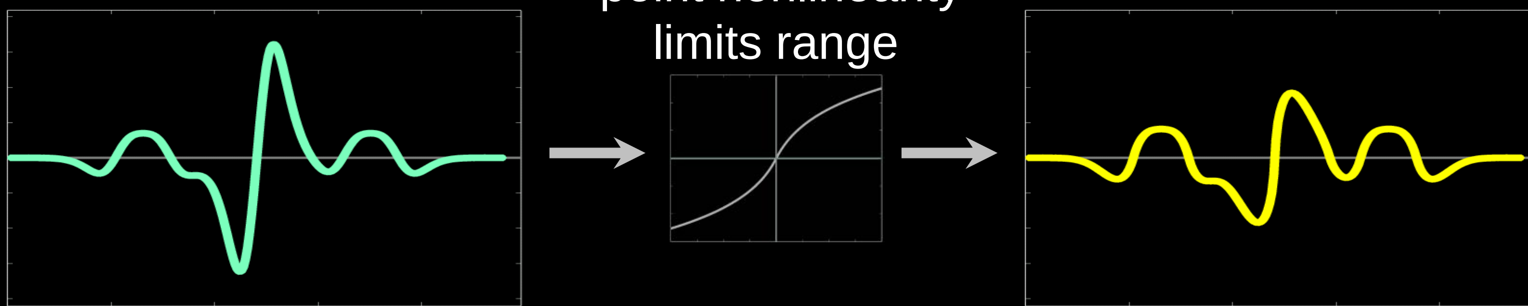
Choice of filters:
Wavelets, QMFs, Laplacian, etc.
They all worked.

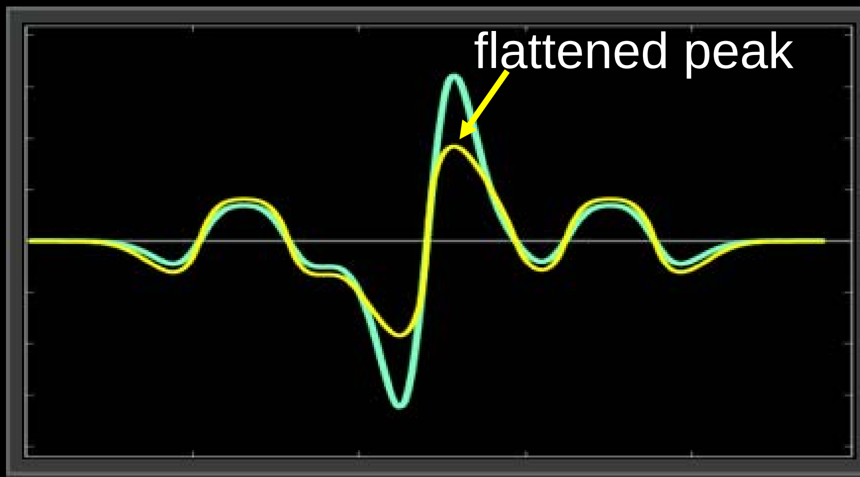
Point Nonlinearity on Subbands

Original subband

point nonlinearity limits range

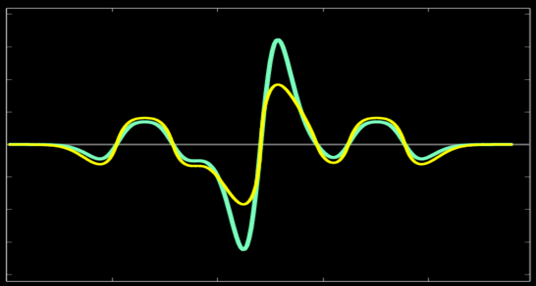
Modified subband

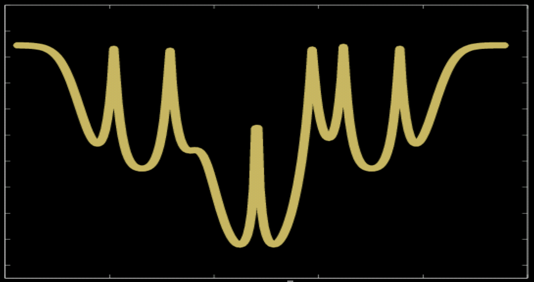




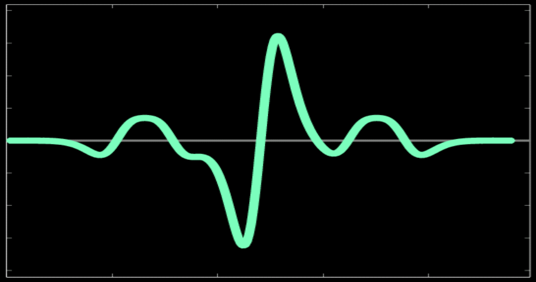
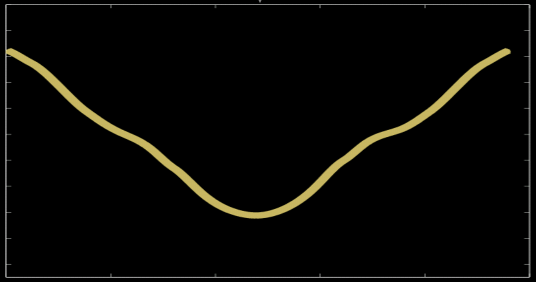
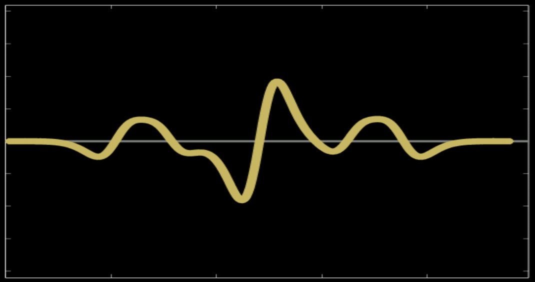
Problem: Nonlinear distortion.

Smooth Gain Control



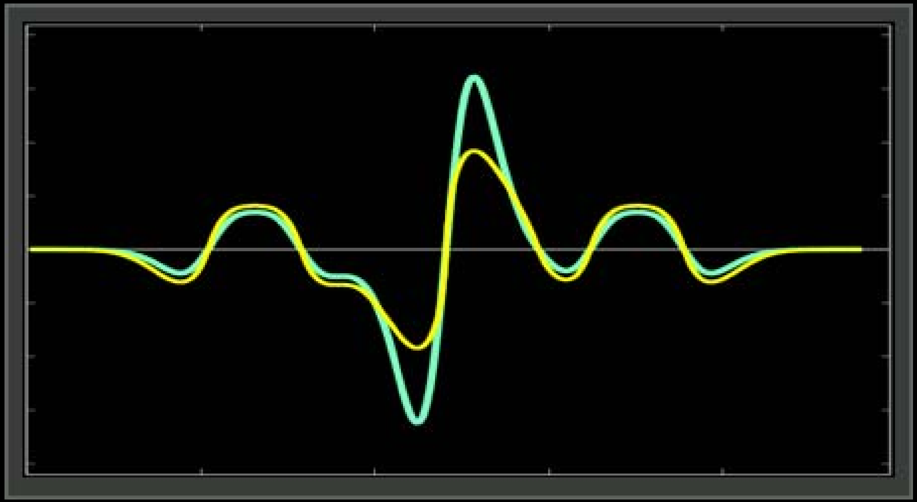
$gain(x) = b'(x) \,/\, b(x) =$  /  $=$ 

smooth

 x  = 

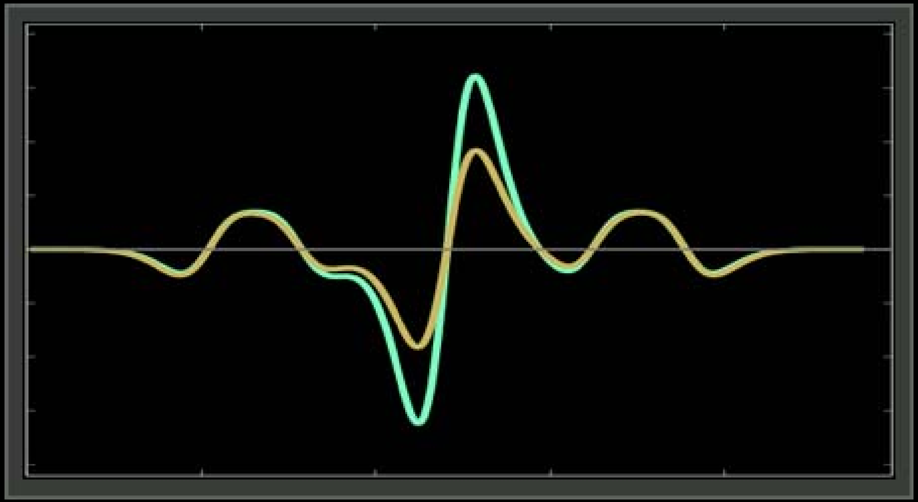
Smooth Gain Control Reduces Distortion

Point nonlinearity



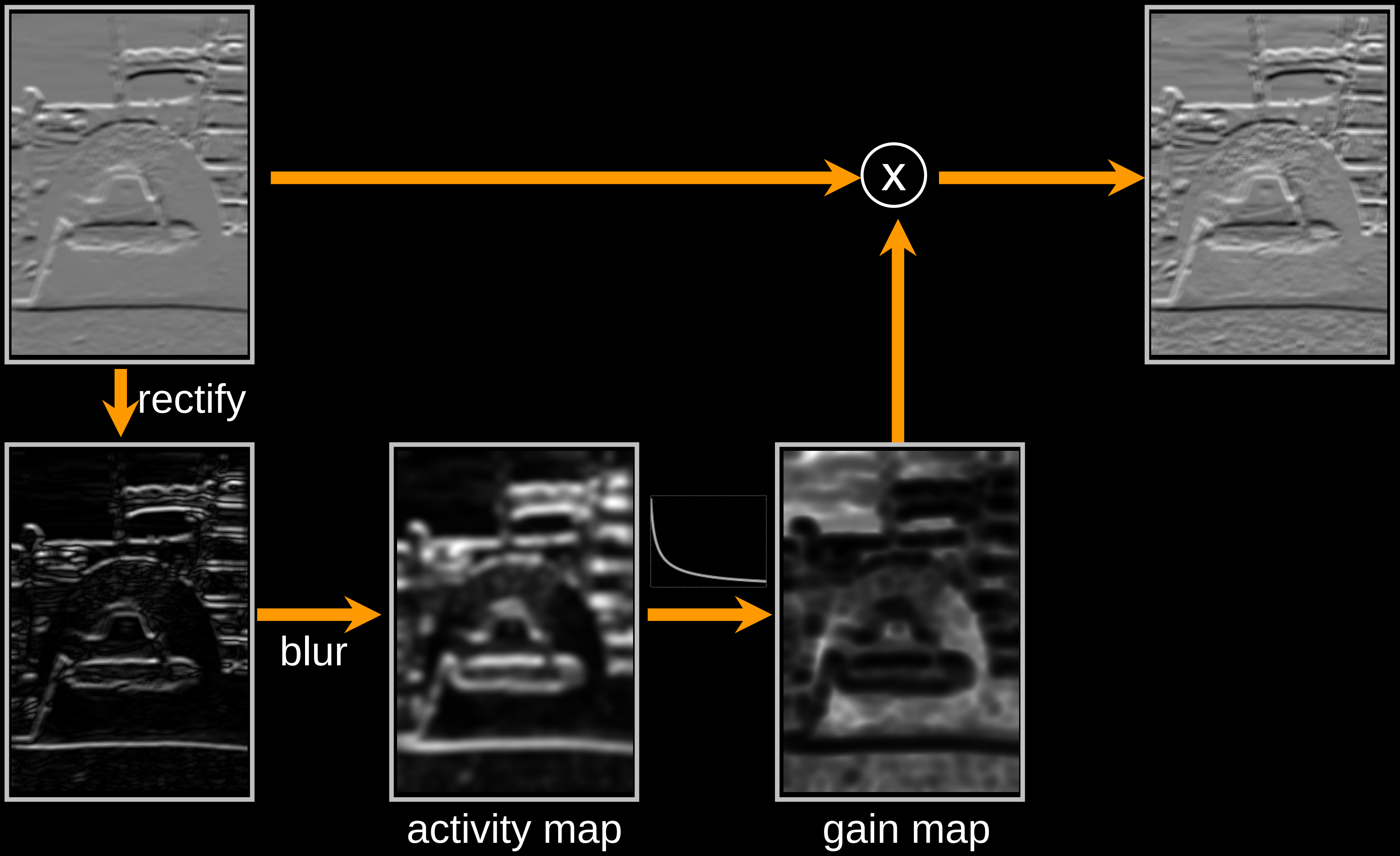
Distorted.

Smooth gain control



Distortion reduced.

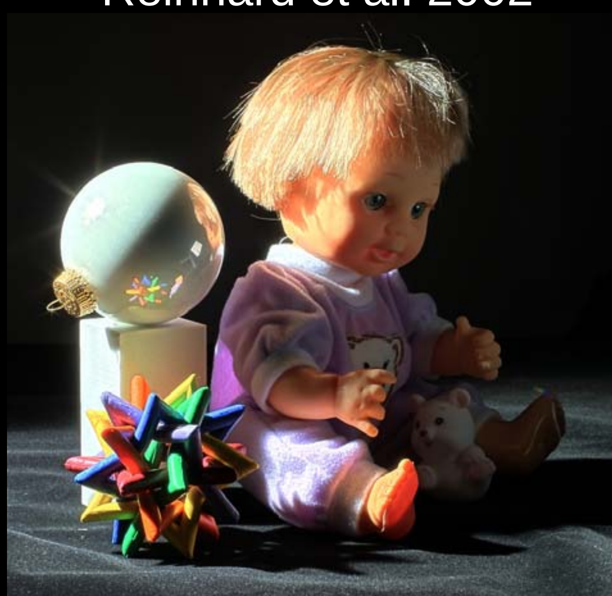
Smooth Gain Control on Subbands



Ours

Fattal et al. 2002

Reinhard et al. 2002



Durand & Dorsey 2002

Reinhard et al. 2002

Ours

Fattal et al. 2002

Reinhard et al. 2002

Durand & Dorsey 2002

Ours

Fattal et al. 2002

Reinhard et al. 2002

Durand & Dorsey 2002

Ours

Fattal et al. 2002

Reinhard et al. 2002

Durand & Dorsey 2002