High Dynamic Range & Tone-Mapping

Image Manipulation and Computational Photography
CS294-69 Fall 2011
Maneesh Agrawala

[Some slides from James Hays , Derek Hoiem, Alexei Efros and Fredo Durand]

A2 Lights Fields/HDR/Tone-Mapping Due Mon Oct 10

Implement Light Fields or HDR/Tone-Mapping

Adequate to implement, best solutions go beyond:

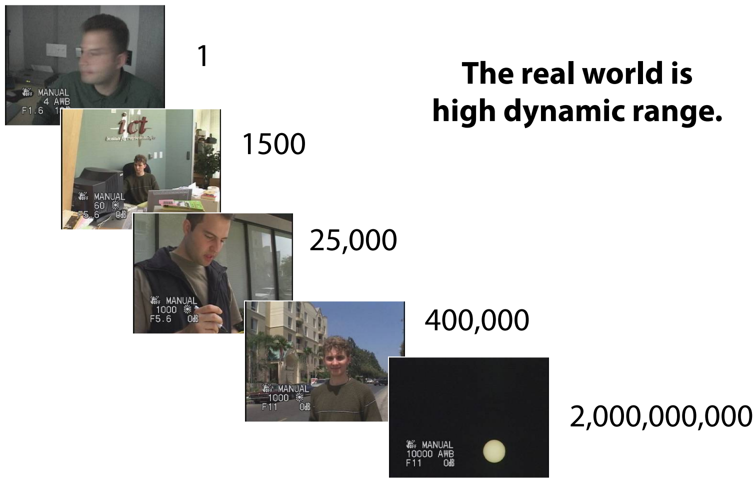
- Every technique has some limitations (well written papers usually describe some of them). Develop techniques to address one or more limitations?
- Sometimes different papers present different techniques for addressing the same problem Implement competing techniques and compare their strengths and weaknesses.
- It may be possible to combine ideas from multiple papers to produce a new hybrid technique that addresses a new problem. Develop a new way to combine the texture synthesis techniques your have read about to solve a new problem.

1 person = 1 paper,

2 people = 1 paper + issue from list above or 2 papers,

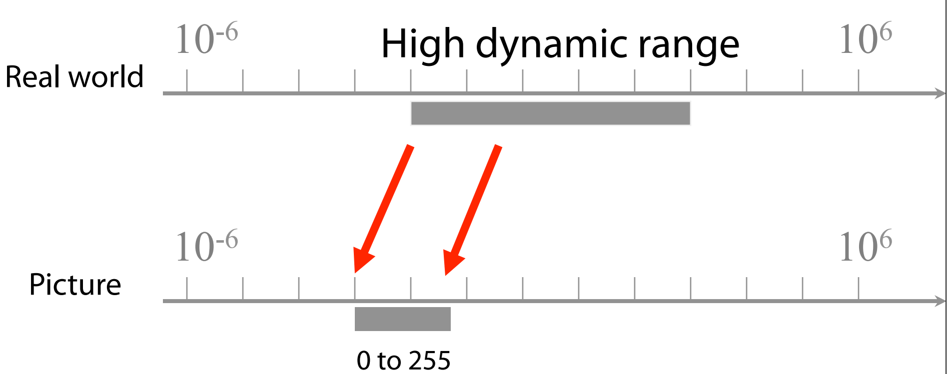
3 people = 2 papers + issue from list above

Problem: Dynamic Range

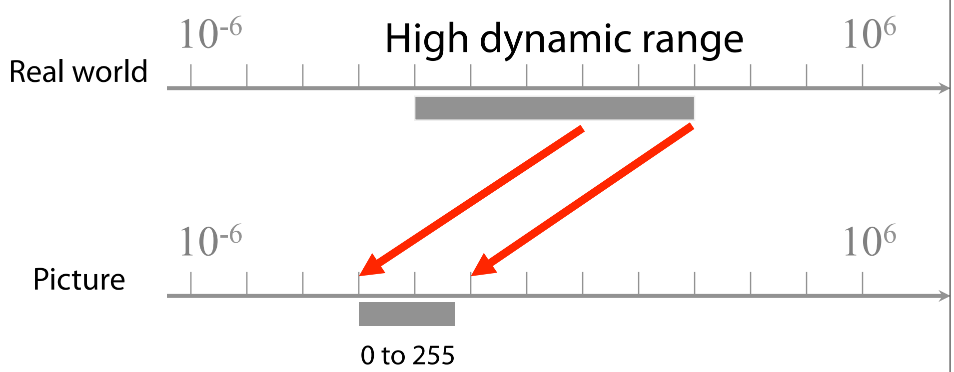


The real world is high dynamic range.

Long Exposure



Short Exposure



Image

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

42 photons?

Camera Calibration

Geometric

How pixel **coordinates** relate to **directions** in the world

Photometric

How pixel **values** relate to **radiance** amounts in the world

Camera Calibration

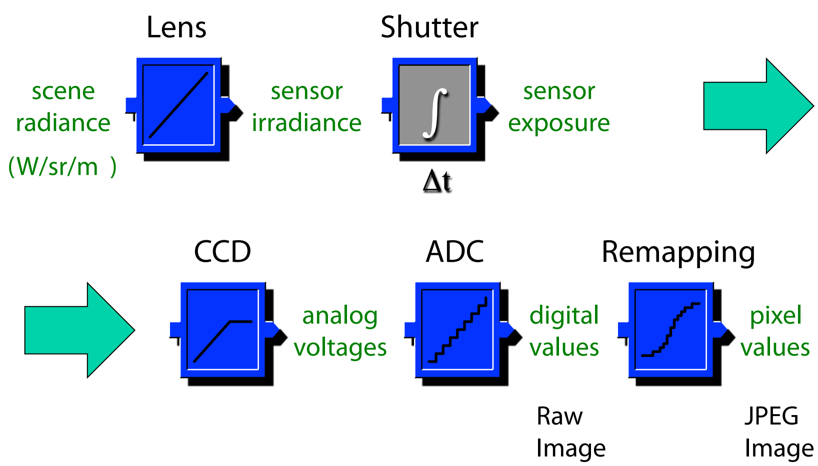
Geometric

How pixel **coordinates** relate to **directions** ~~in the world~~ in other images

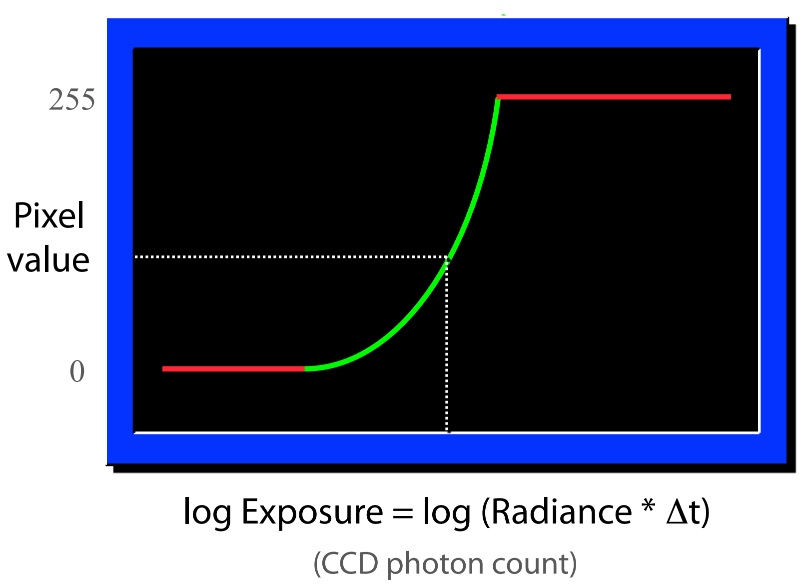
Photometric

How pixel **values** relate to **radiance** amounts ~~in the world~~ in other images

Image Acquisition Pipeline



Imaging System Response Function



Camera is Not a Photometer!

Limited dynamic range

Perhaps use multiple exposures?

Unknown, nonlinear response

Not possible to convert pixel values to radiance

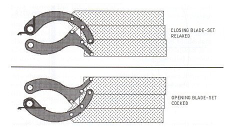
Solution:

Recover response curve from multiple exposures, then reconstruct the ***radiance map***

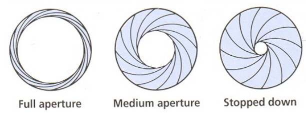
Varying Exposure

How to Vary Exposure?

Shutter speed



Aperture



ISO

Neutral density filter

Slide inspired by Siggraph 2005 course on HDR

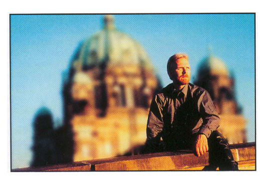
Tradeoffs

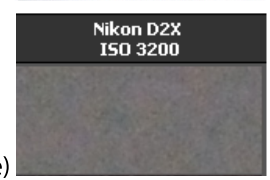
Shutter speed
- Range: ~30 sec to 1/4000sec (6 orders of magnitude)
- Pros: reliable, linear
- Cons: sometimes noise for long exposure

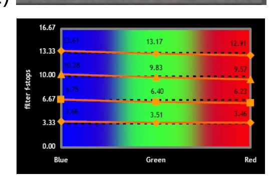
Aperture
- Range: ~f/1.4 to f/22 (2.5 orders of magnitude)
- Cons: changes depth of field
- Useful when desperate

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



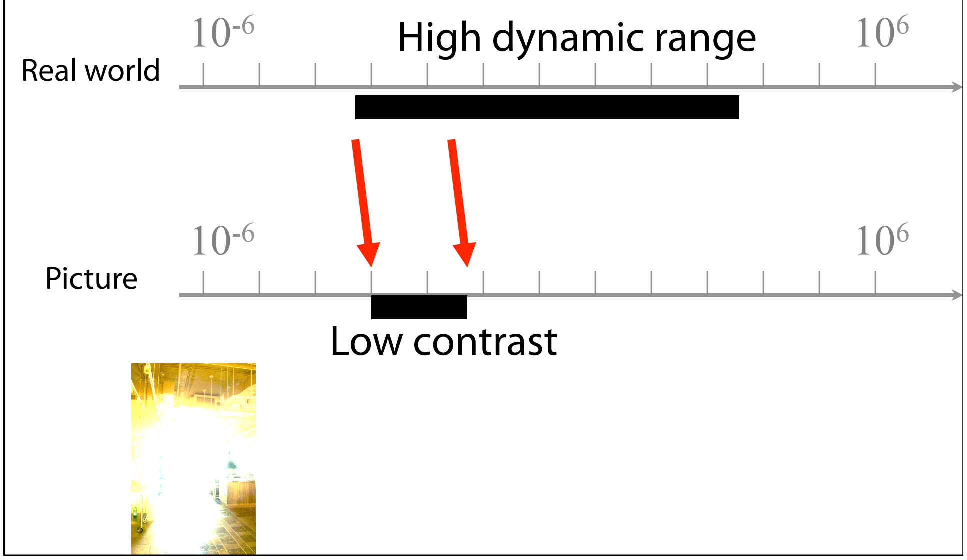




Slide after Siggraph 2005 course on HDR

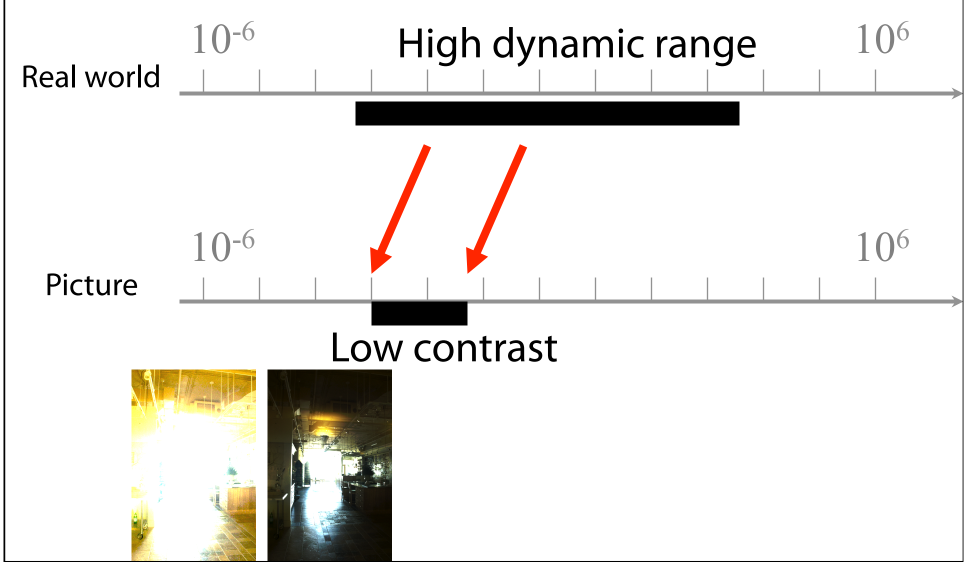
Multiple Exposure Photography

Sequentially measure all segments of the range



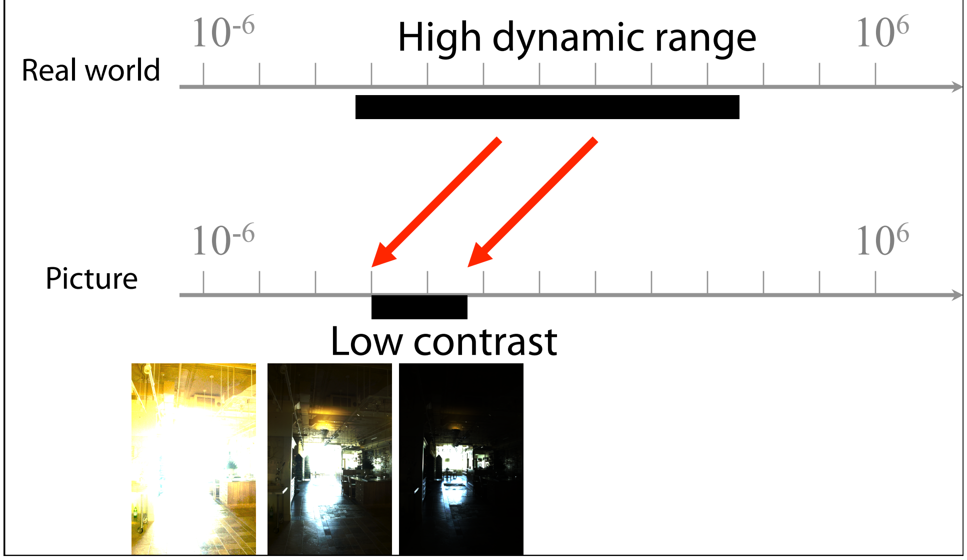
Multiple Exposure Photography

Sequentially measure all segments of the range



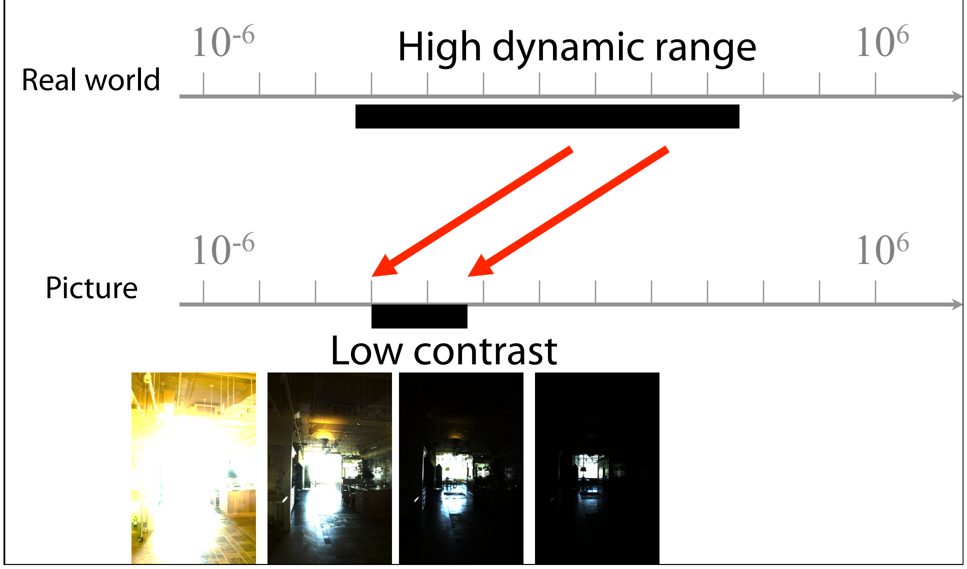
Multiple Exposure Photography

Sequentially measure all segments of the range



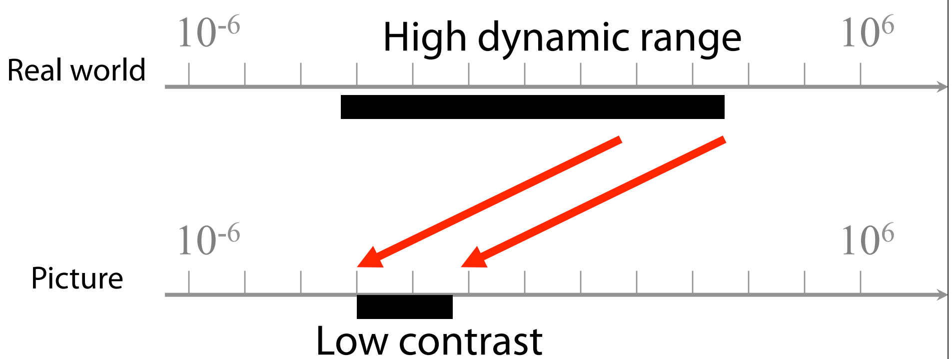
Multiple Exposure Photography

Sequentially measure all segments of the range



Multiple Exposure Photography

Sequentially measure all segments of the range



Problem:

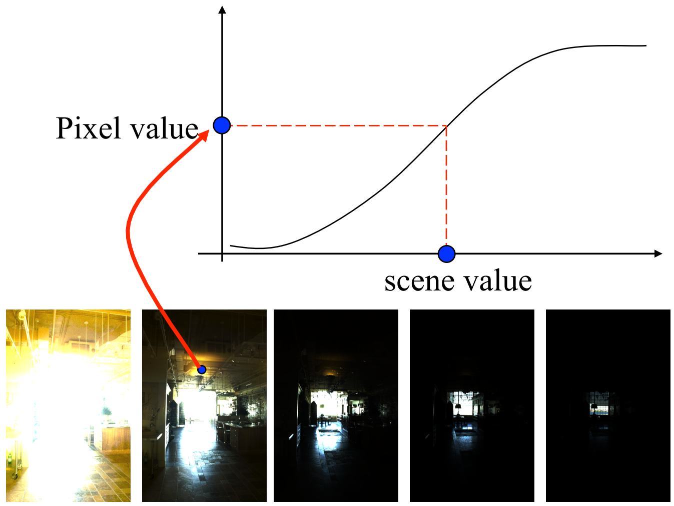
Given N photos at different exposure

Recover a HDR color for each pixel

If We Know Camera Response Curve

Just look up the inverse of the response curve

But how do we get the curve?



Calibrating Response Curve

Two basic solutions

- Vary scene luminance and see pixel values
  - Assumes we control and know scene luminance
- Vary exposure and see pixel value for one scene luminance
  - But note that we can usually not vary exposure more finely than by 1/3 stop

Best of both:

- Vary exposure
- Exploit the large number of pixels

Recovering High Dynamic Range Radiance Maps from Photographs

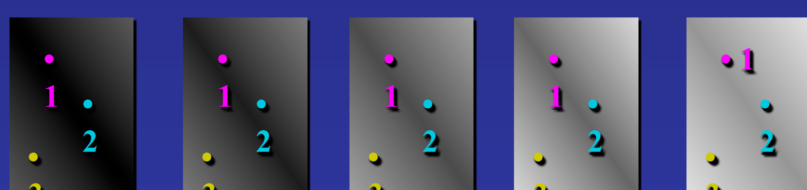
Paul Debevec
Jitendra Malik

Computer Science Division
University of California at Berkeley

August 1997

The Algorithm

Image series



Δt = 1/64 sec | **Δt = 1/16 sec** | **Δt = 1/4 sec** | **Δt = 1 sec** | **Δt = 4 sec**

Pixel Value Z = f(Exposure)

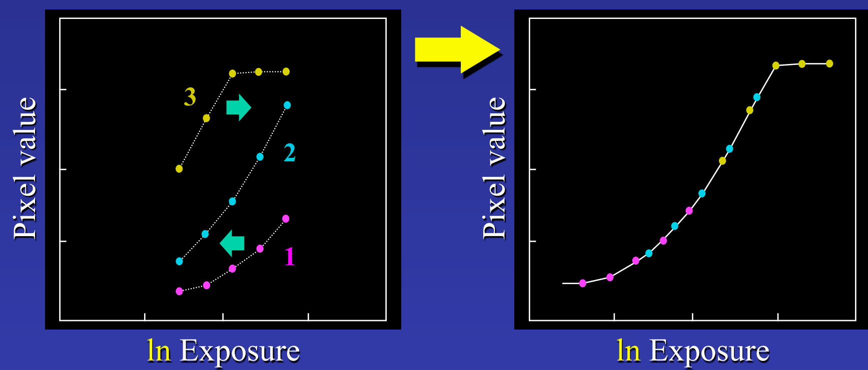
Exposure = Radiance * Δt

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:

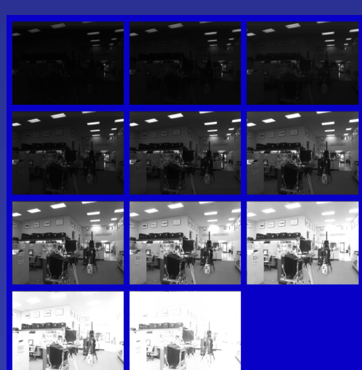
$$\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{P}\left[\ln Radiance_i + \ln \Delta t_j - g(Z_{ij})\right]^2}_{\text{fitting term}} + \underbrace{\lambda \sum_{z=Z_{min}}^{Z_{max}} g''(z)^2}_{\text{smoothness term}}$$

Matlab Code

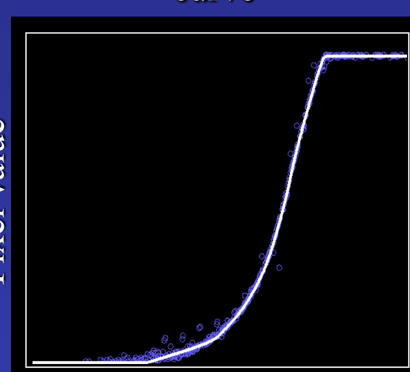
```
function [g,lE]=gsolve(Z,B,l,w)

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 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));
```

Results: Digital Camera

Kodak DCS460
1/30 to 30 sec



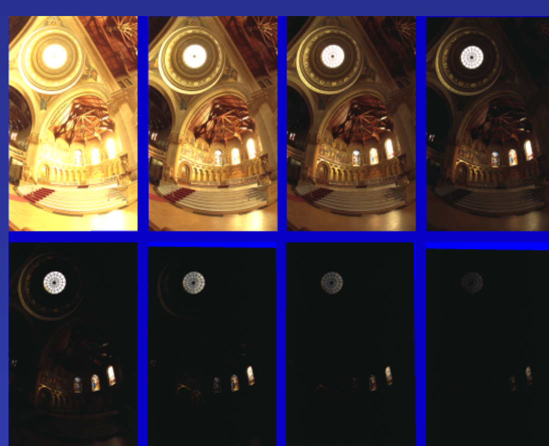
Recovered response curve



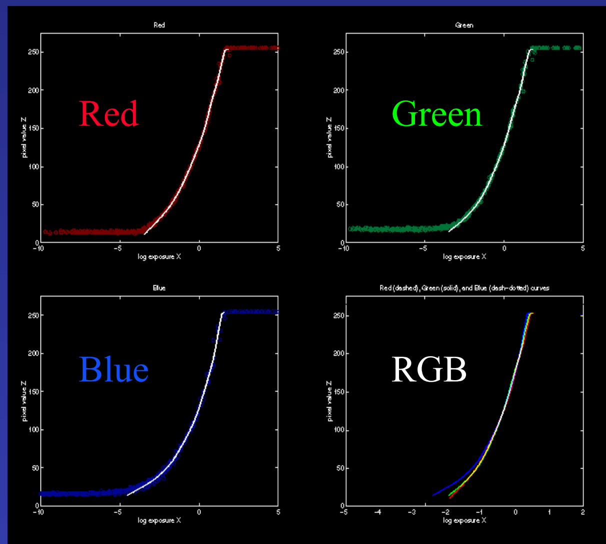
Reconstructed radiance map

Results: Color Film

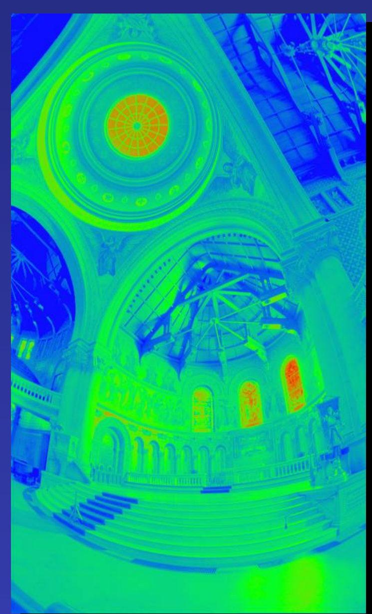
- Kodak Gold ASA 100, PhotoCD



Recovered Response Curves



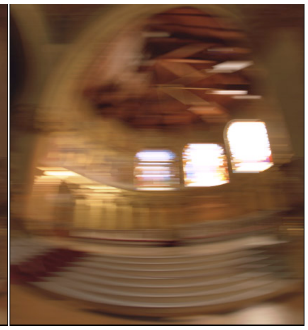
The Radiance Map



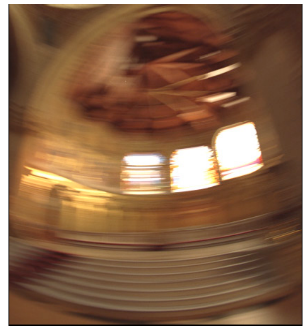
HDR Image Processing

Images from Debevec & Malik 1997

Motion blur applied to **low**-dynamic-range picture



Motion blur applied to **high**-dynamic-range picture



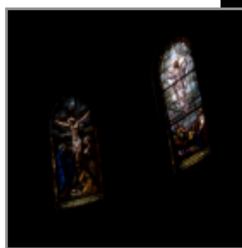
Real motion-blurred picture

Important also for depth of field post-process

Available in HDRShop

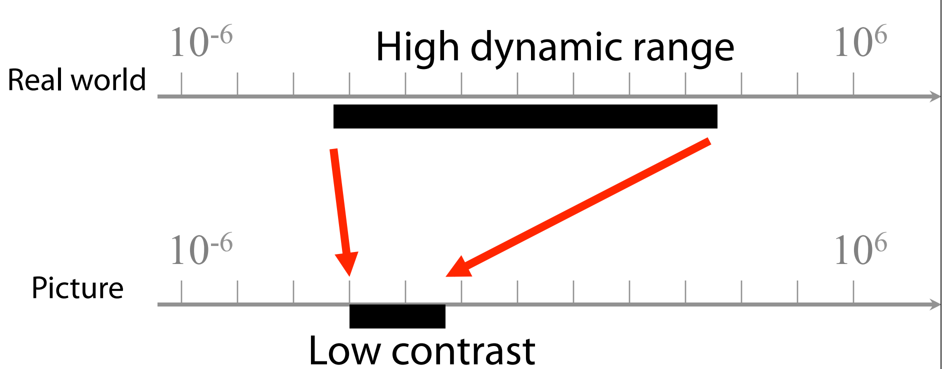
Slide from Siggraph 2005 course on HDR

The Radiance Map



Linearly scaled to display device

Tone-Mapping

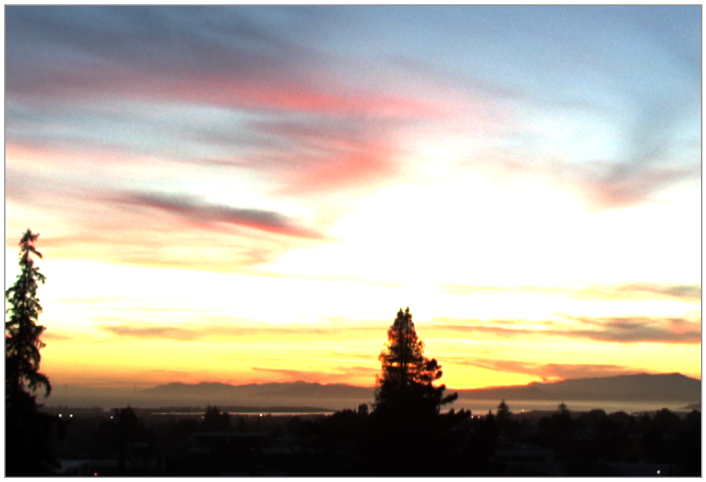


How can we do this remapping?

Tone-Mapping Input

Input: high-dynamic-range image

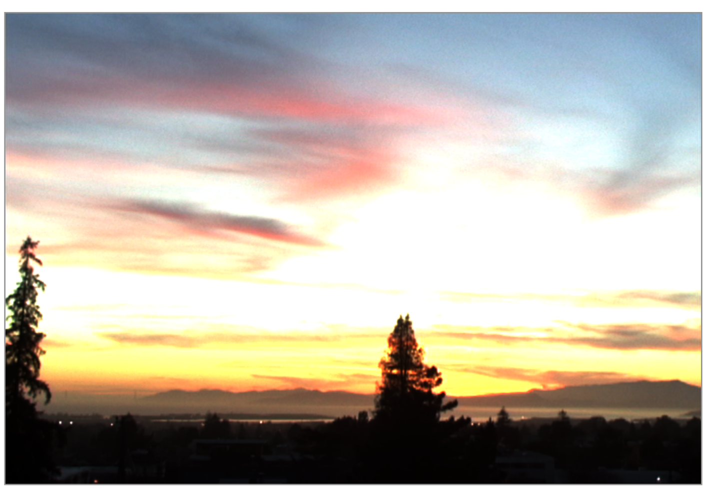
(floating point per pixel)



Naïve Technique

Scene has *1:10,000* contrast, display has *1:100*

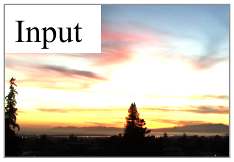
Simplest contrast reduction?

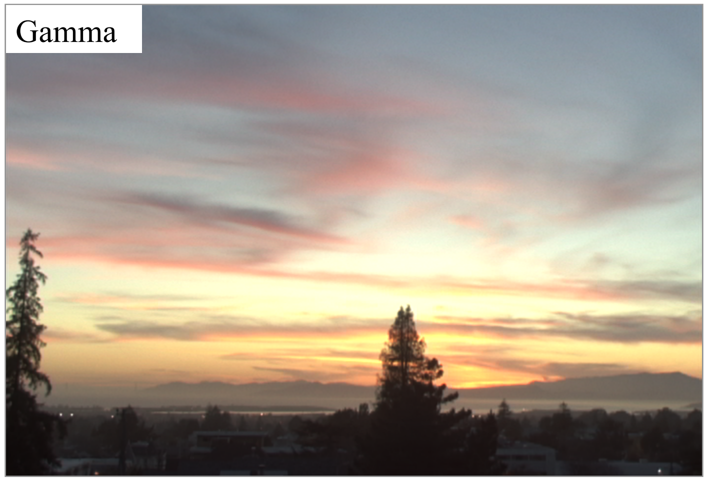


Naïve: Gamma Compression

X –>X$^{\gamma}$ (where γ=0.5 in our case)

But… colors are washed-out. Why?

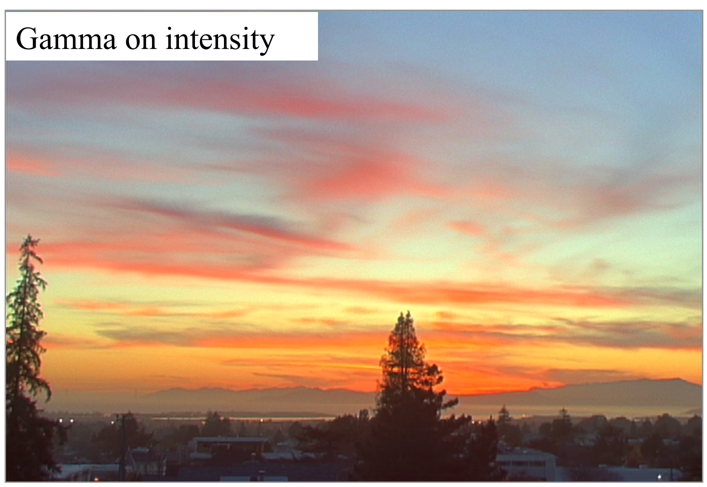

Input


Gamma

Gamma Compression on Intensity

Colors are OK, but details (intensity high-frequency) are blurred


Intensity


Gamma on intensity


Color

Oppenheim 1968, Chiu et al. 1993

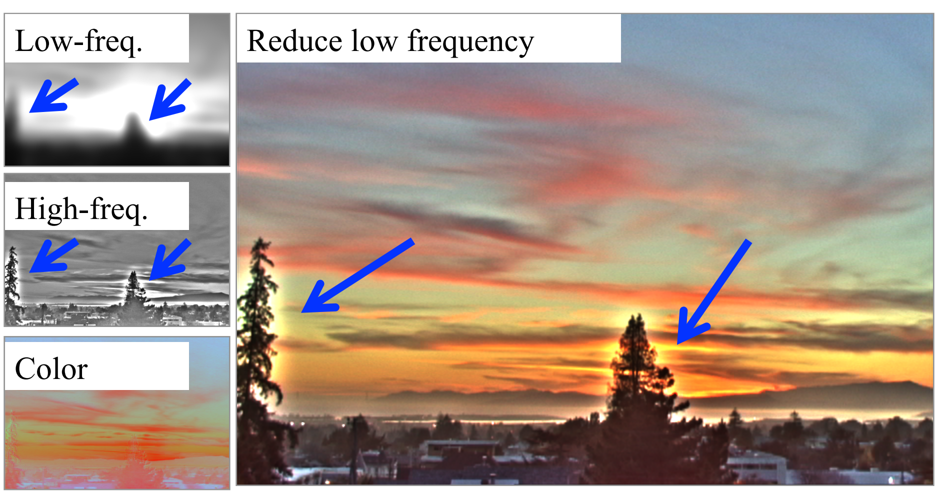
Reduce contrast of low-frequencies

Keep high frequencies

The Halo Nightmare

For strong edges

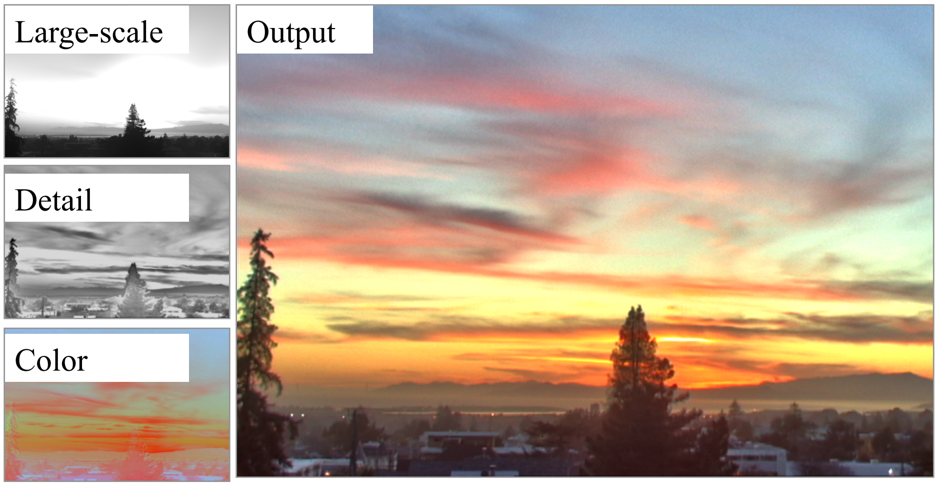
Because they contain high frequency



Durand & Dorsey 02

Do not blur across edges

Non-linear filtering



Bilateral Filter

Tomasi and Manduci 1998

http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf

Related to

SUSAN filter [Smith and Brady 95]
http://citeseer.ist.psu.edu/smith95susan.html

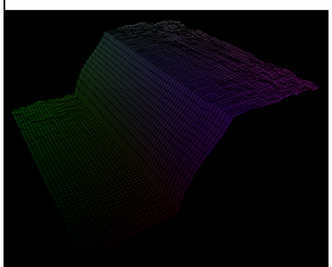
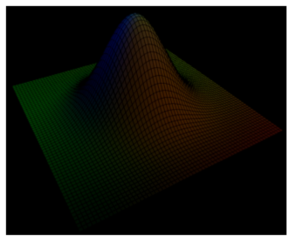
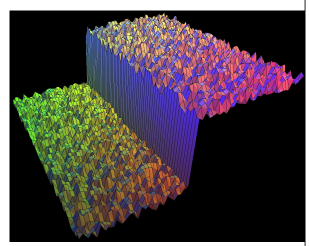
Digital-TV [Chan, Osher and Chen 2001]
http://citeseer.ist.psu.edu/chan01digital.html

sigma filter http://www.geogr.ku.dk/CHIPS/Manual/f187.htm

Start with Gaussian Filter

Here, input is a step function + noise

$$J = f \otimes I$$

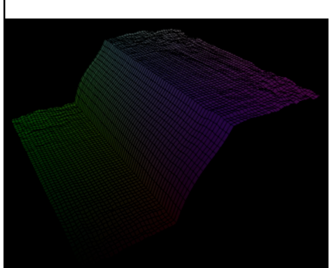
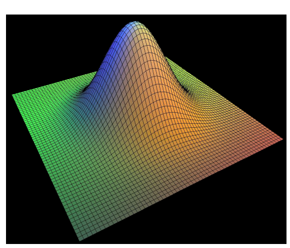
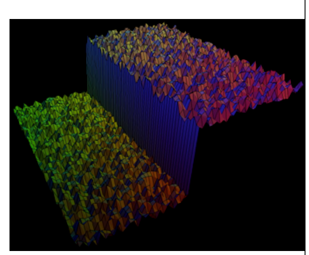
  

output ⟵ input

Start with Gaussian Filter

Spatial Gaussian f

$$J = f \otimes I$$

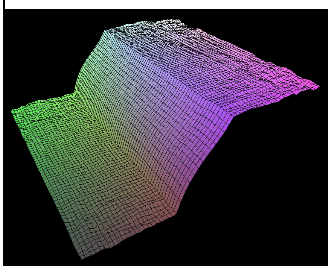
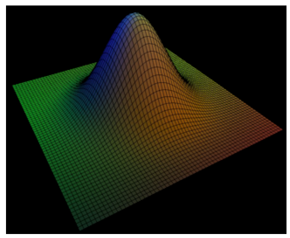
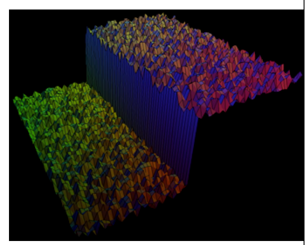
  

output ⟵ input

Start with Gaussian Filter

Output is blurred

$$J = f \otimes I$$

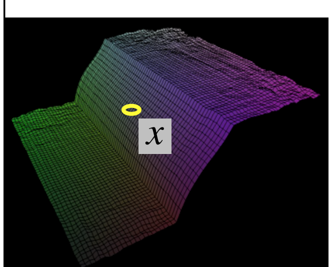
  

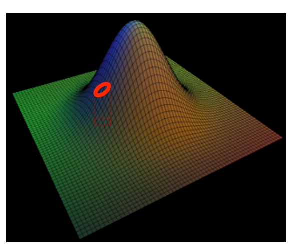
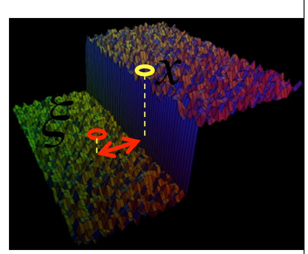
output ⟵ input

Gaussian Filter as Weighted Average

Weight of ξ depends on distance to x

$$J(x) = \sum_{\xi} f(x,\xi) \; I(\xi)$$



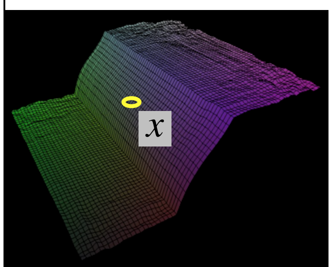
output ⟵ input

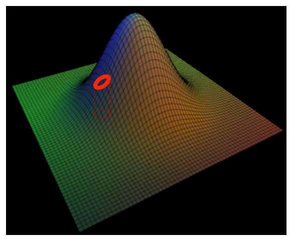
The Problem of Edges

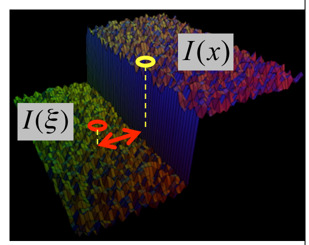
Here, $I(\xi)$"pollutes" our estimate J(x)

It is too different

$$J(x) = \sum_{\xi} f(x,\xi) \quad I(\xi)$$







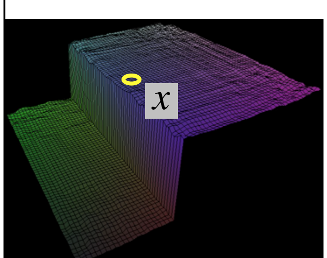
output ⟵ input

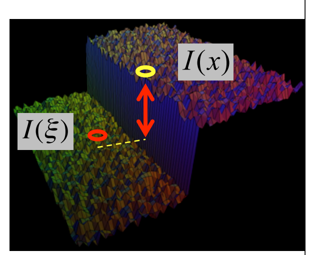
Principle of Bilateral Filter

[Tomasi and Manduchi 1998]

Penalty g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$





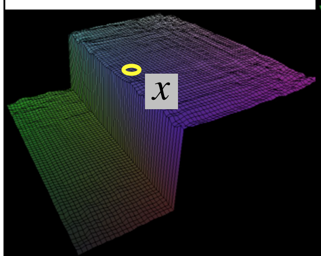
output ⟵ input

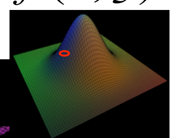
Bilateral Filter

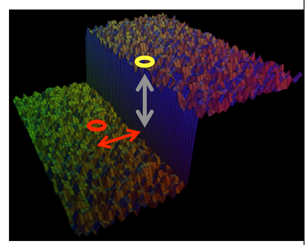
[Tomasi and Manduchi 1998]

Spatial Gaussian f

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$







output input

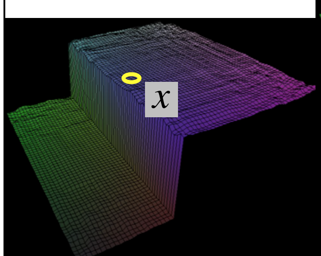
Bilateral filtering

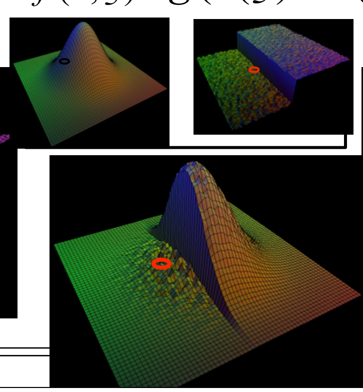
[Tomasi and Manduchi 1998]

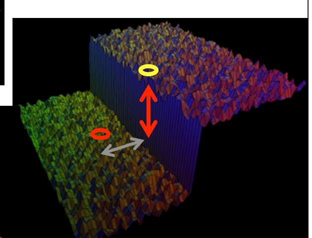
Spatial Gaussian f

Gaussian g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$







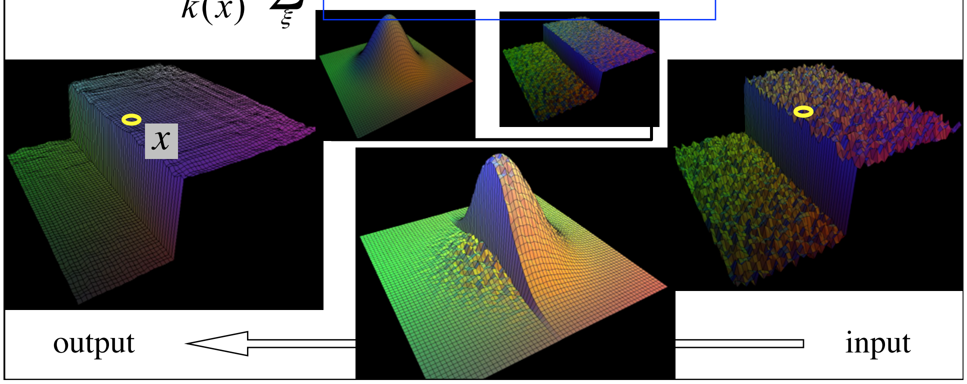
output input

Normalization Factor

[Tomasi and Manduchi 1998]

k(x)= $\sum_{\xi} \boxed{f(x,\xi) \quad g(I(\xi)-I(x))}$

$$J(x) = \frac{1}{k(x)} \sum_{\xi} \boxed{f(x,\xi) \quad g(I(\xi)-I(x))} \quad I(\xi)$$

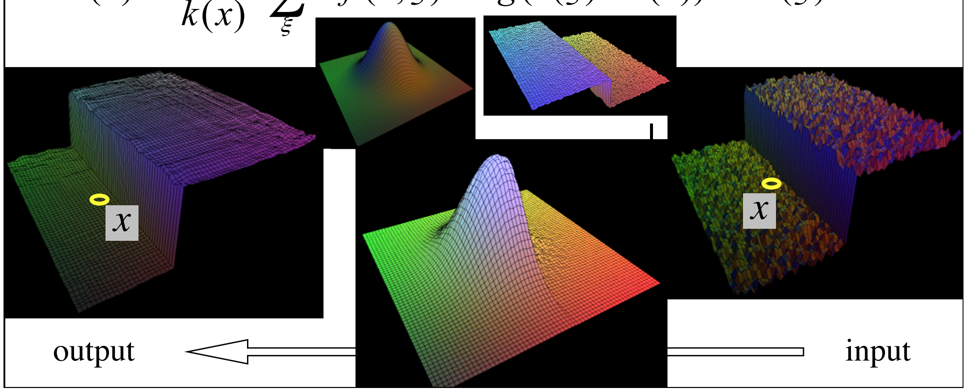


Bilateral Filter is Non-Linear

[Tomasi and Manduchi 1998]

The weights are different for each output pixel

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$

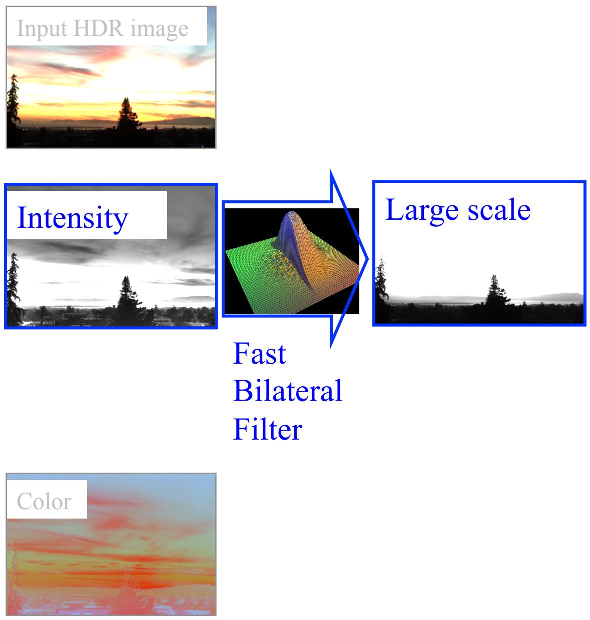


Contrast Reduction

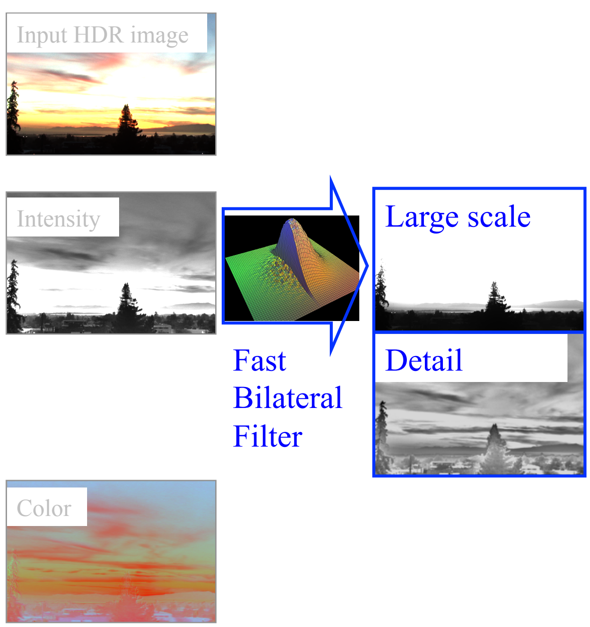
Contrast
too high!

Contrast Reduction

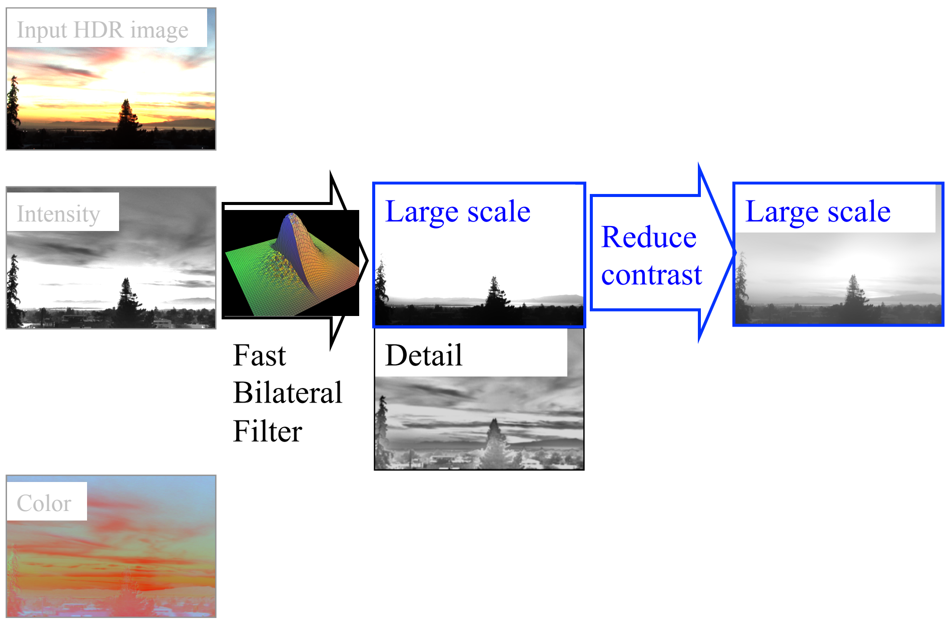
Contrast Reduction



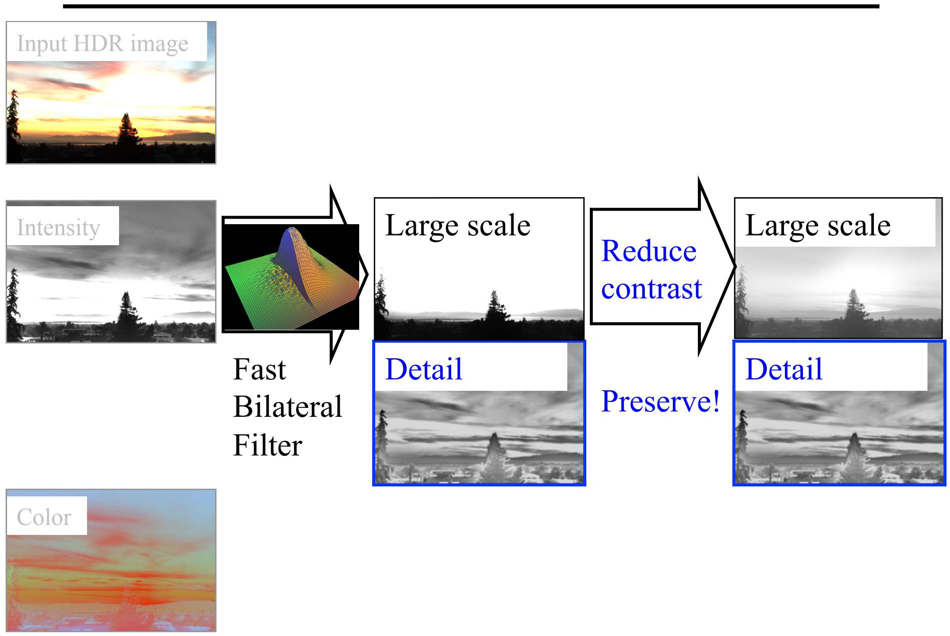
Contrast Reduction



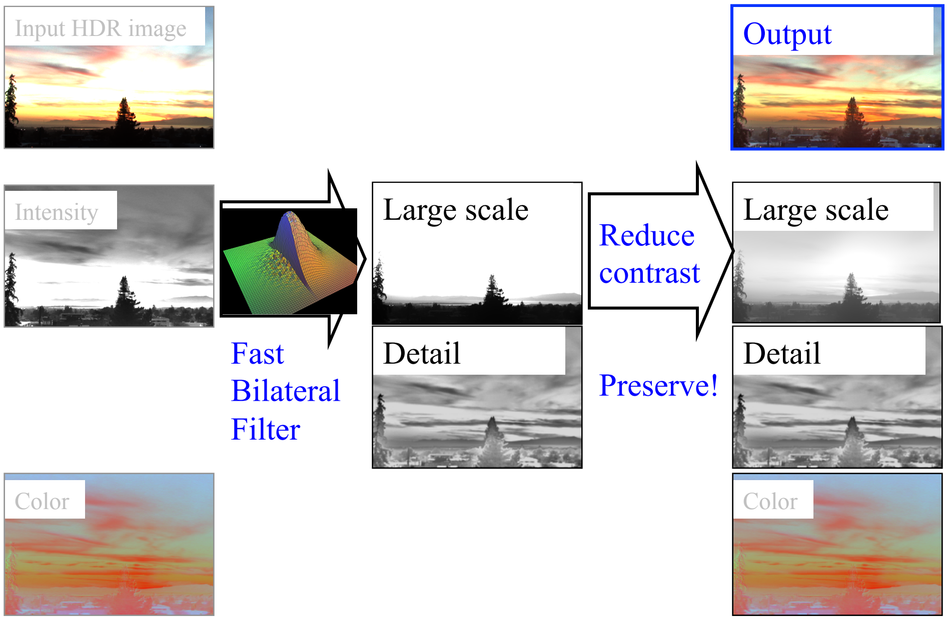
Contrast Reduction



Contrast Reduction



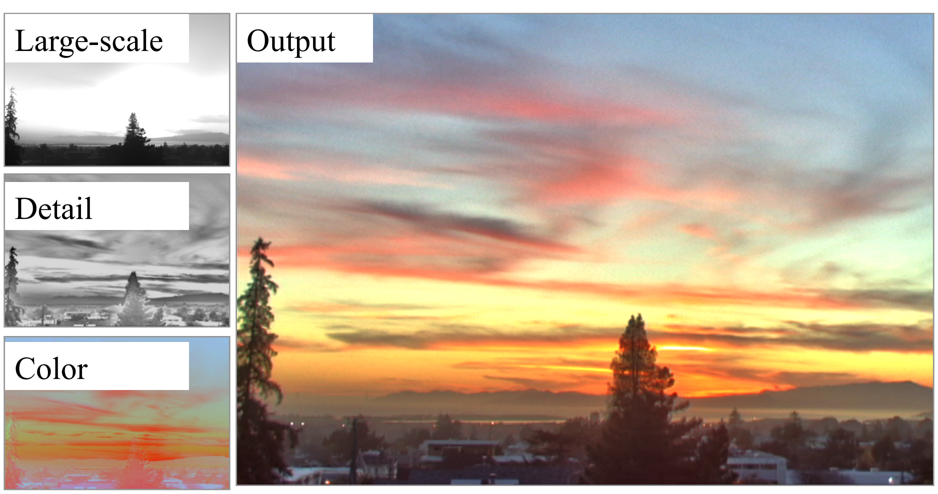
Contrast Reduction



Durand & Dorsey 02

Do not blur across edges

Non-linear filtering



Informal Comparison

Gradient-space [Fattal et al.] | Bilateral [Durand et al.] | Photographic [Reinhard et al.]

Informal Comparison

Gradient-space [Fattal et al.] | Bilateral [Durand et al.] | Photographic [Reinhard et al.]