CSCE 448/748 - Computational Photography

HDR Imaging & Tonemapping

Nima Kalantari

Slides from Alexei Efros, Paul Debevec, Durand and Dorsey, and Mertens et al.

Problem: Dynamic Range

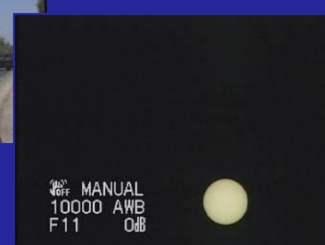
The real world is
high dynamic range.

1

1500

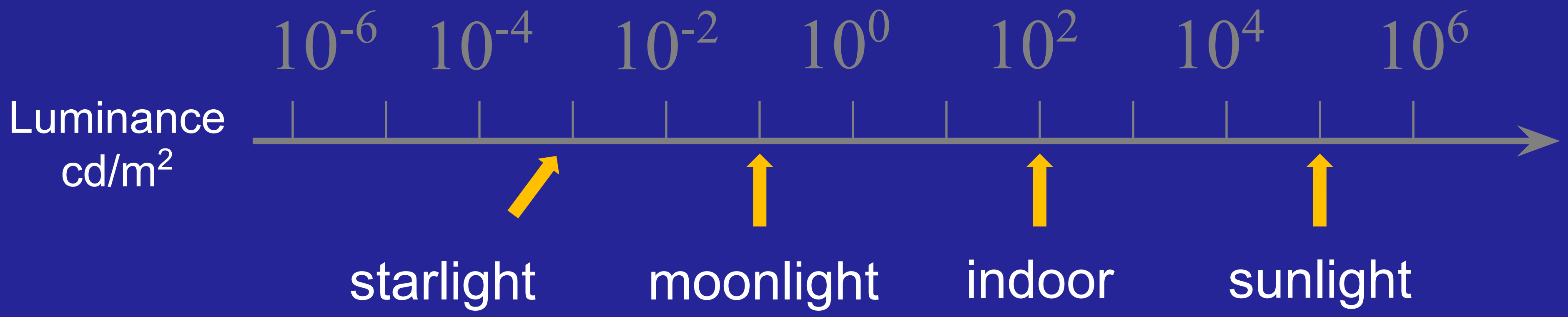
25,000

400,000



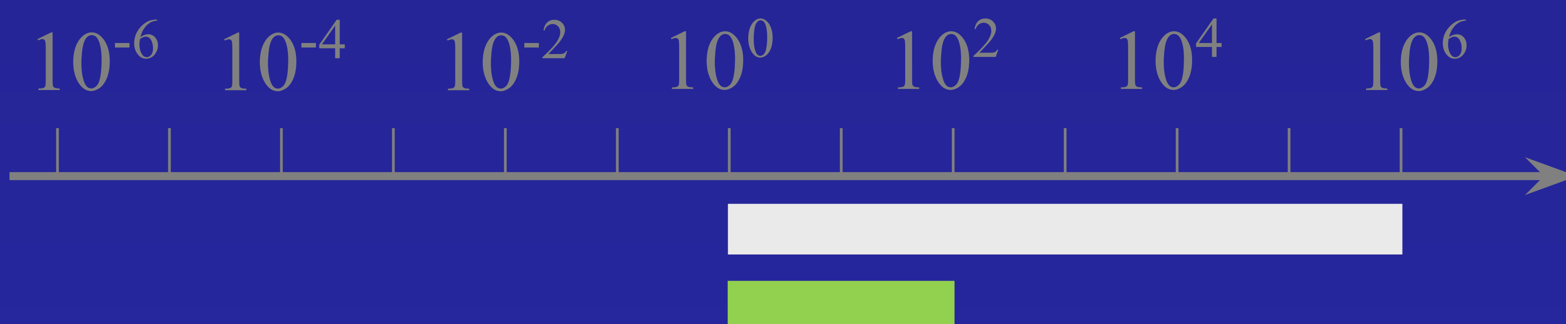
2,000,000,000

Dynamic range



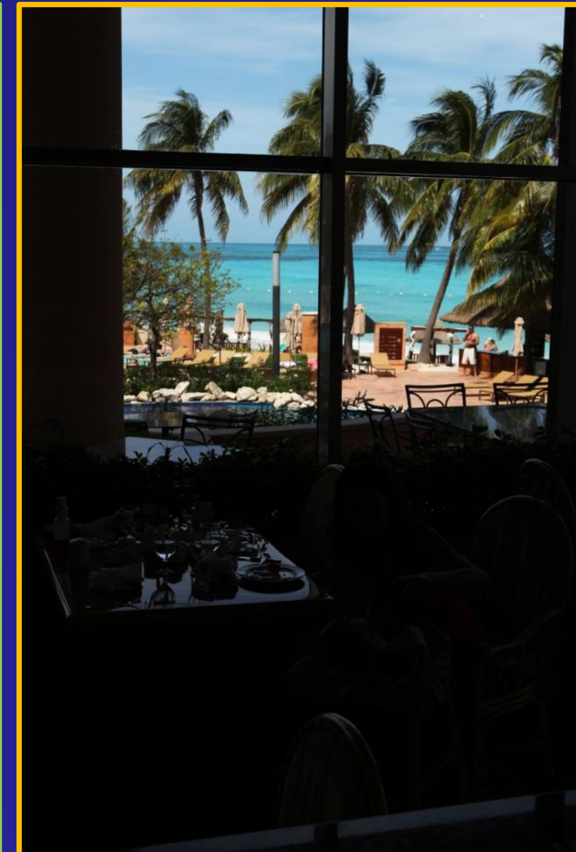
Real World Scene

High Exposure

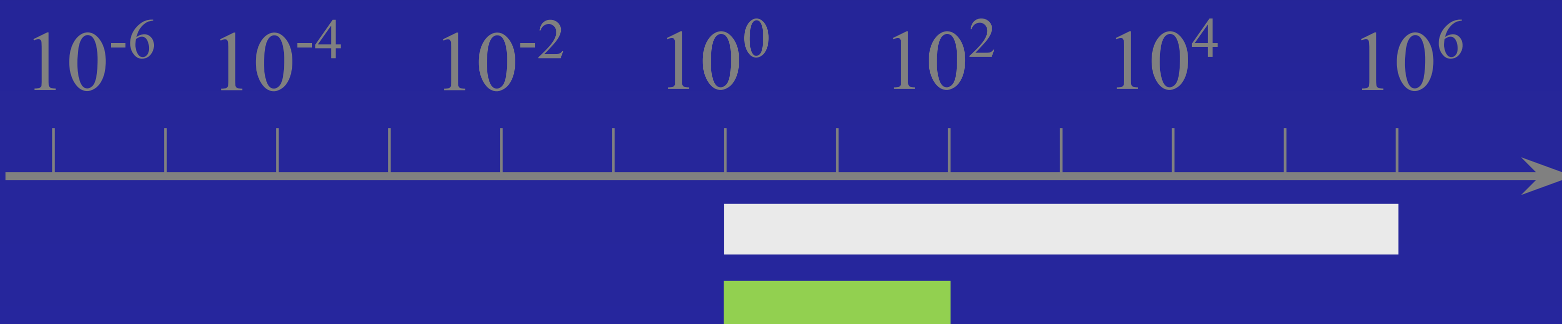


Real World Scene

High Exposure



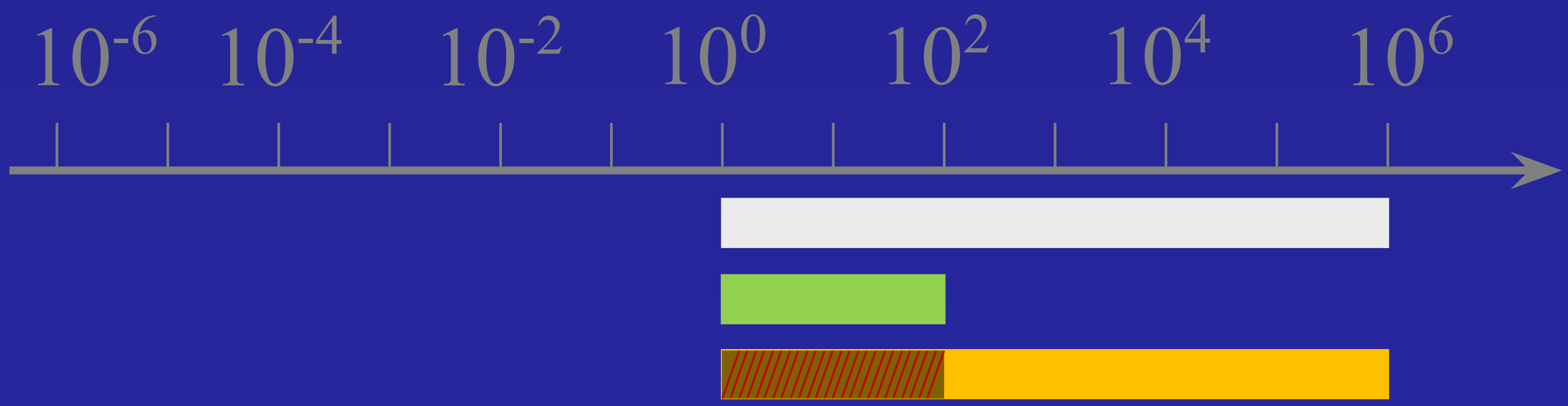
Low Exposure



Real World Scene

High Exposure

Low Exposure



Real World Scene

High Exposure

Low Exposure

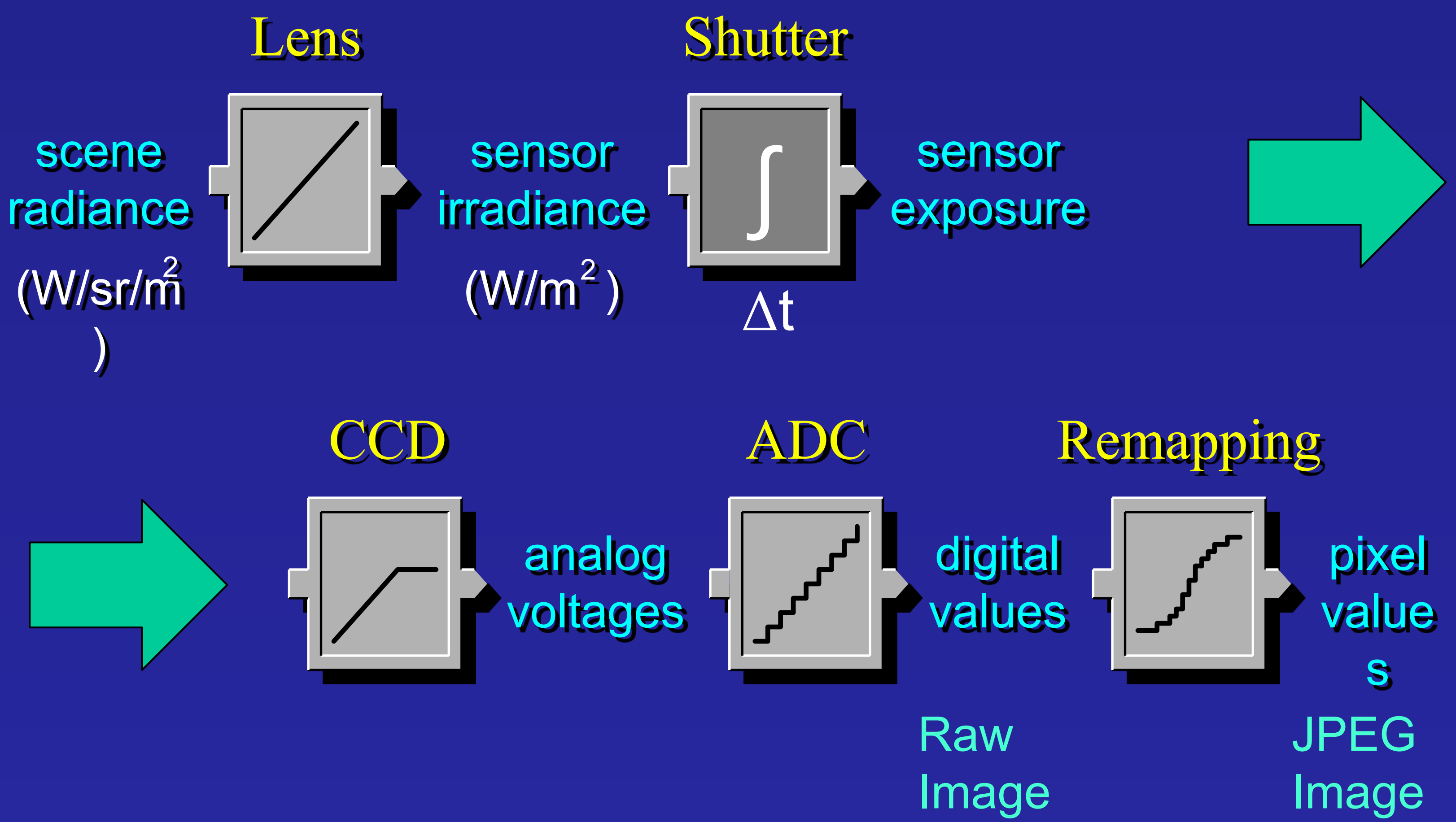
Goal: Recover the actual scene brightness

Image

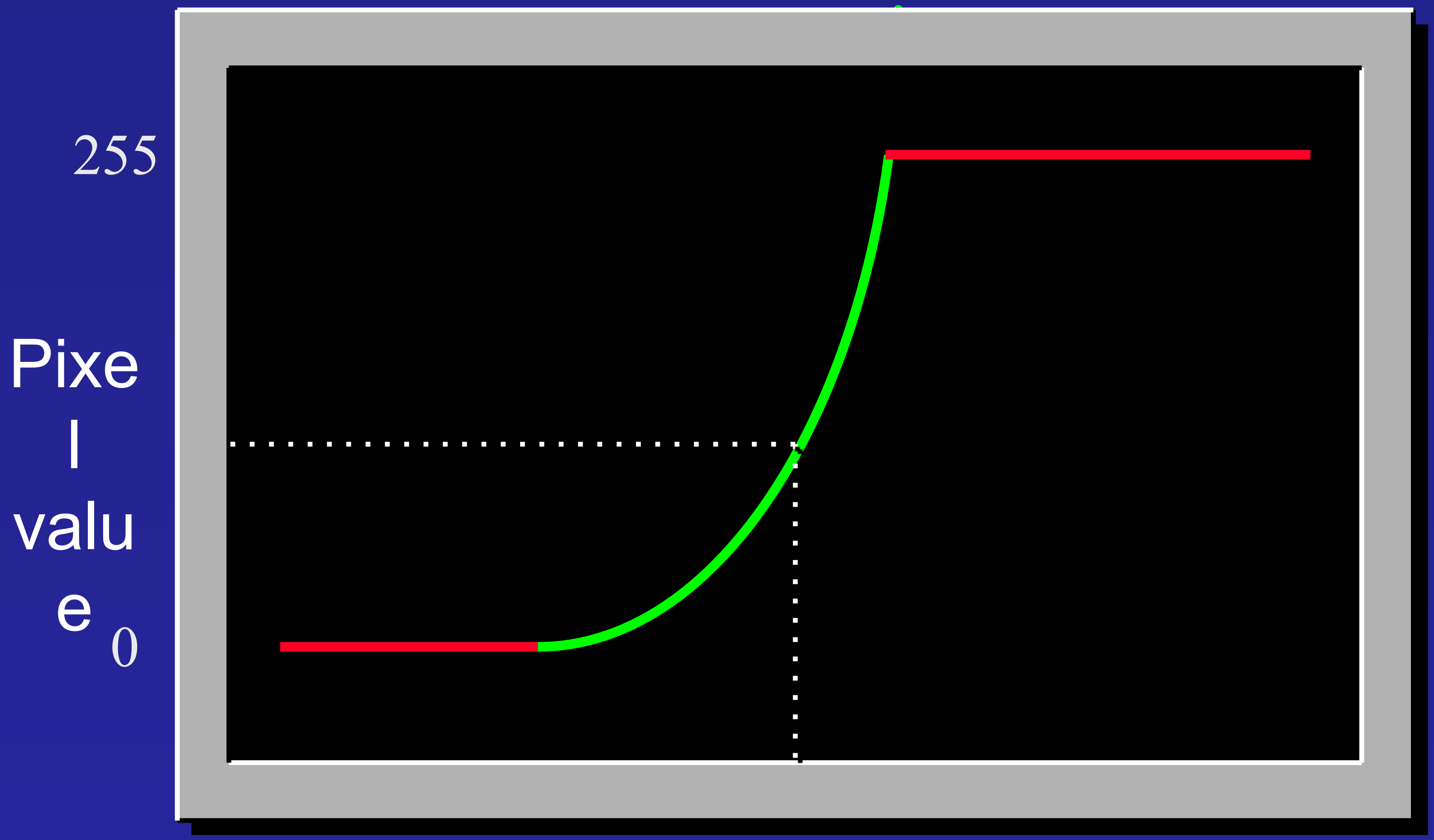
pixel (312, 284) = 42

42 photons?

The Image Acquisition Pipeline



Imaging system response function



log Exposure = log (Irradiance * Δt)

(CCD photon count)

Camera is not a photometer!

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

Recovering High Dynamic Range Radiance Maps from Photographs

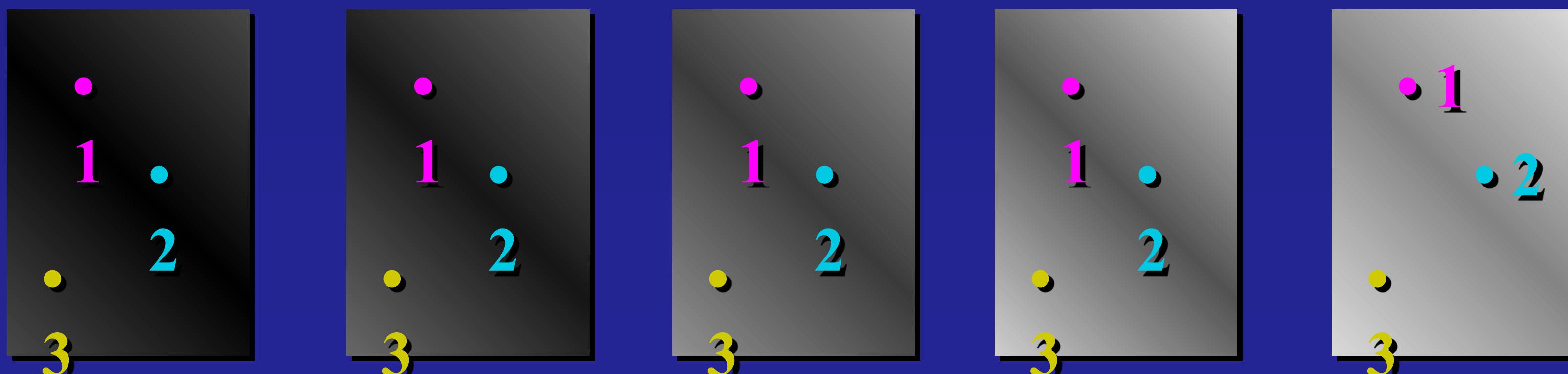
Paul Debevec
Jitendra Malik

Computer Science Division
University of California at Berkeley

August 1997

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

The Math

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

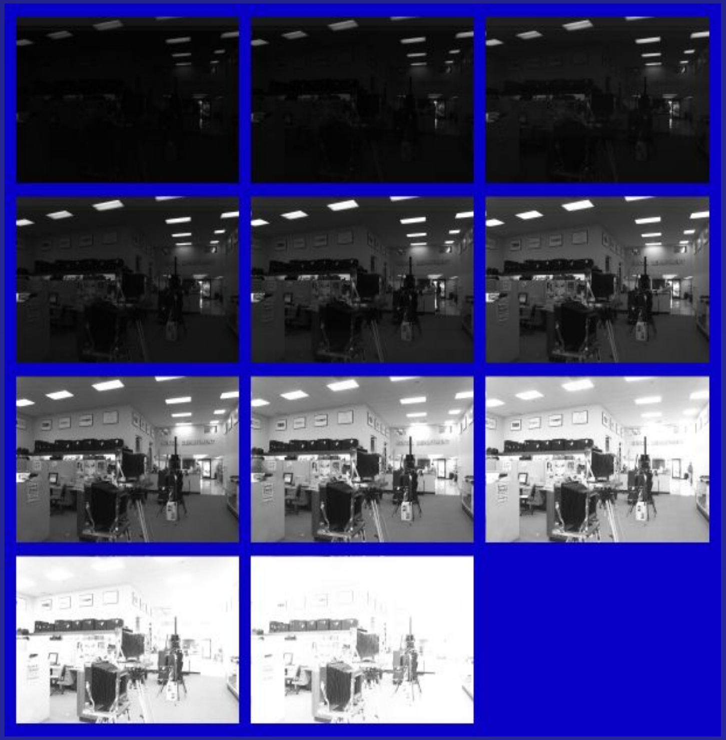
$$\ln Radiance_i + \ln \Delta t_j = g(Z_{ij})$$

- Solve the overdetermined linear system:

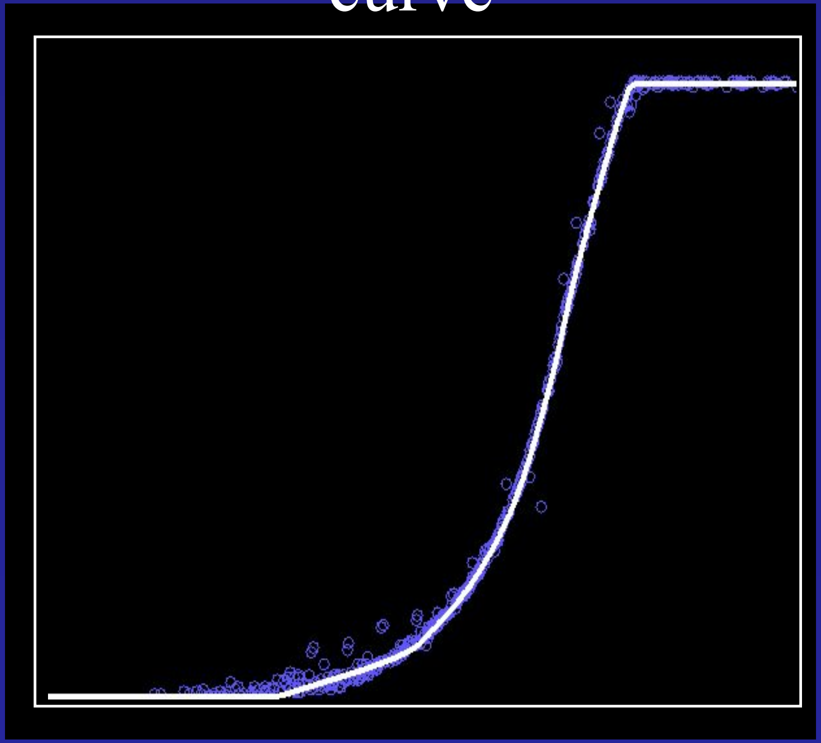
$$\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{P}\left[\ln Radiance_i + \ln \Delta t_j - g(Z_{ij})\right]^2}_{\text{fitting term}} + \underbrace{\lambda \sum_{z=Z_{min}}^{Z_{max}} g''(z)^2}_{\text{smoothness term}}$$

Results: Digital Camera

Kodak DCS460
1/30 to 30 sec



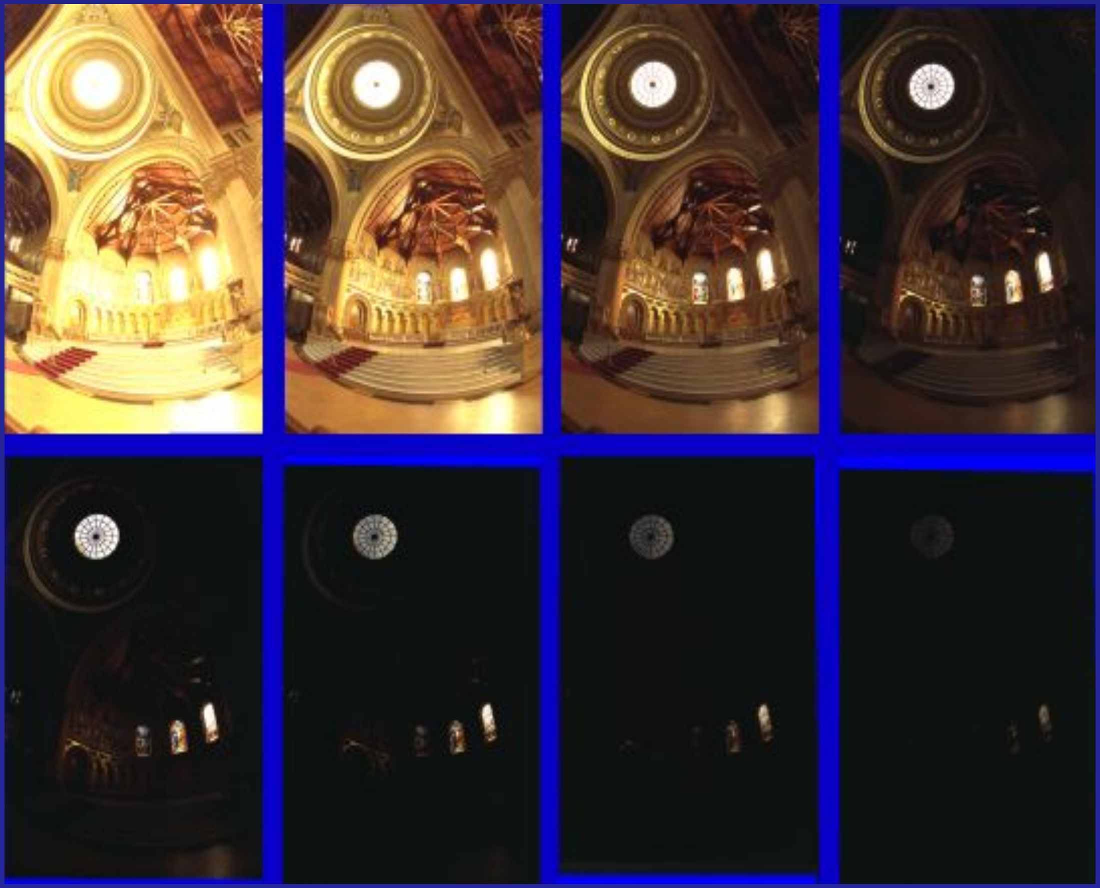
Recovered response curve



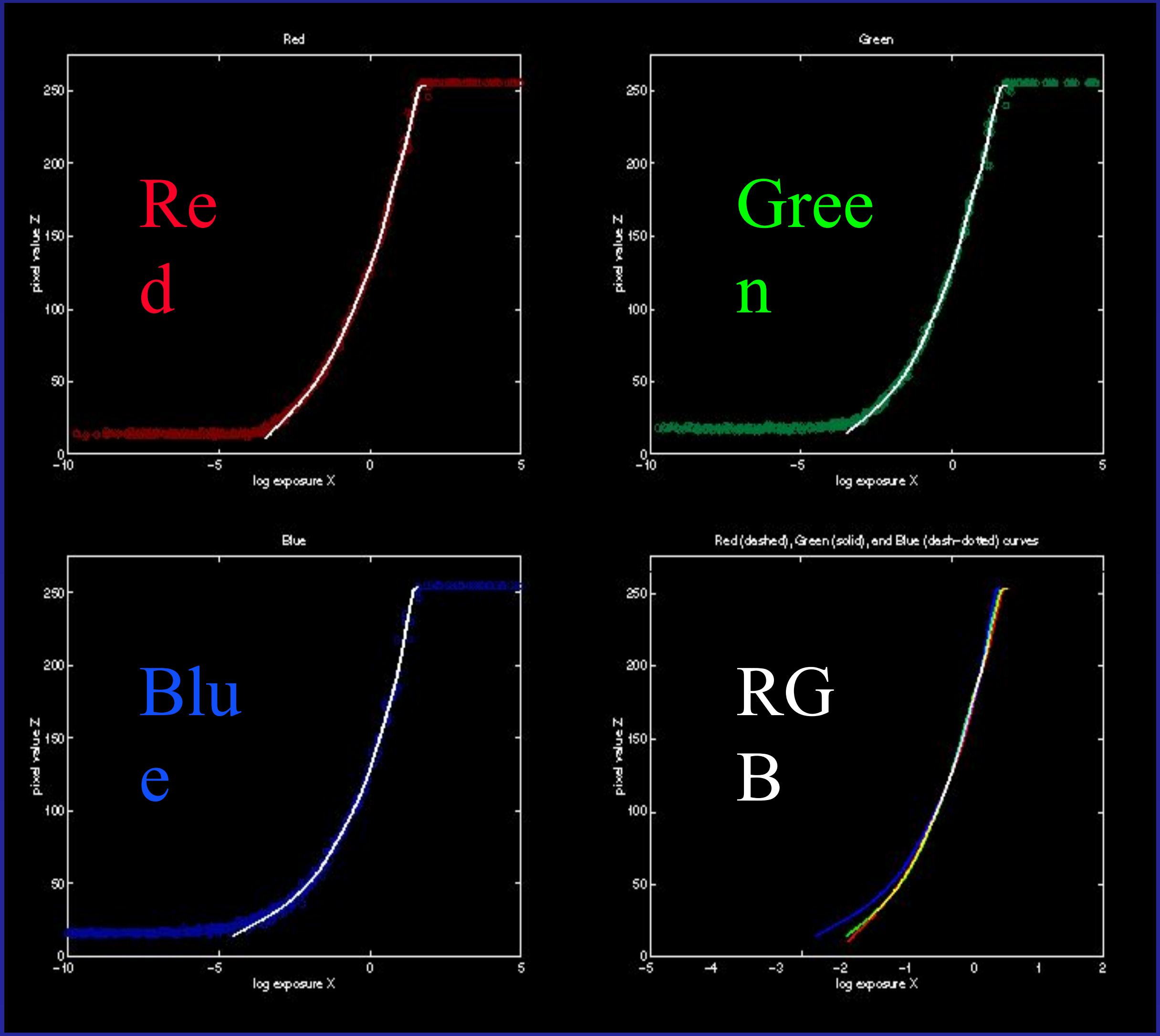
Reconstructed radiance map

Results: Color Film

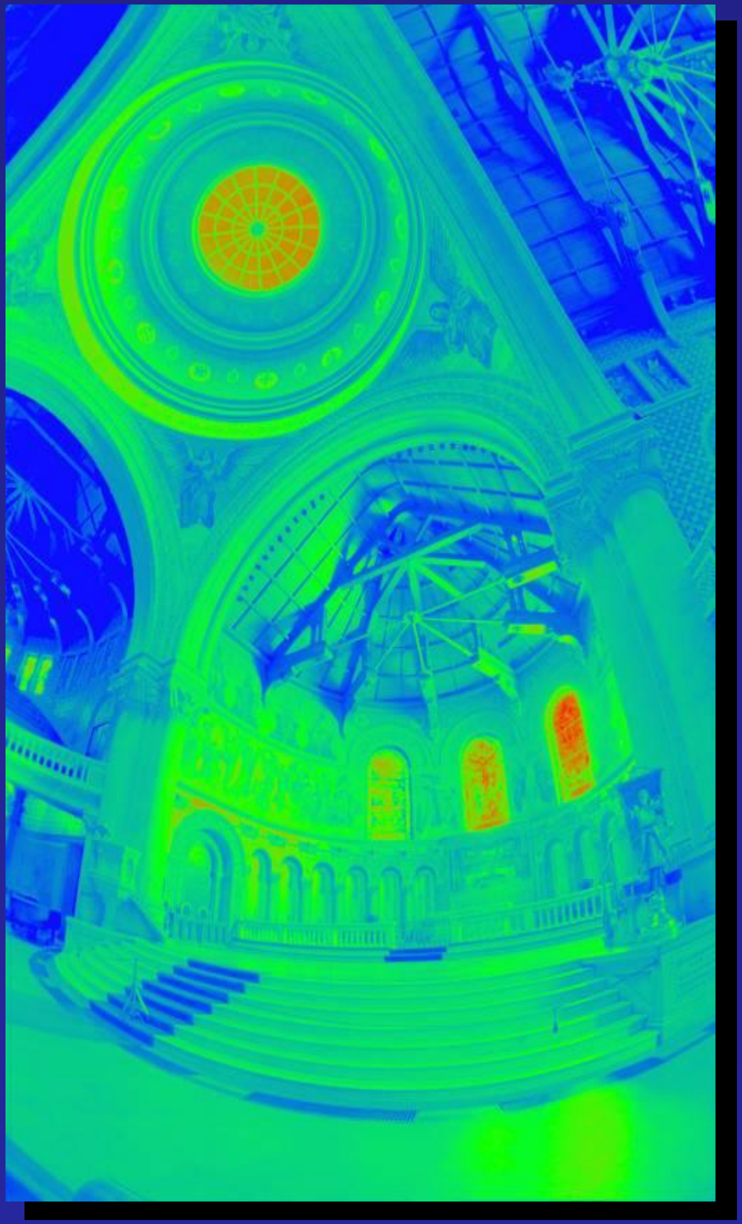
- Kodak Gold ASA 100, PhotoCD



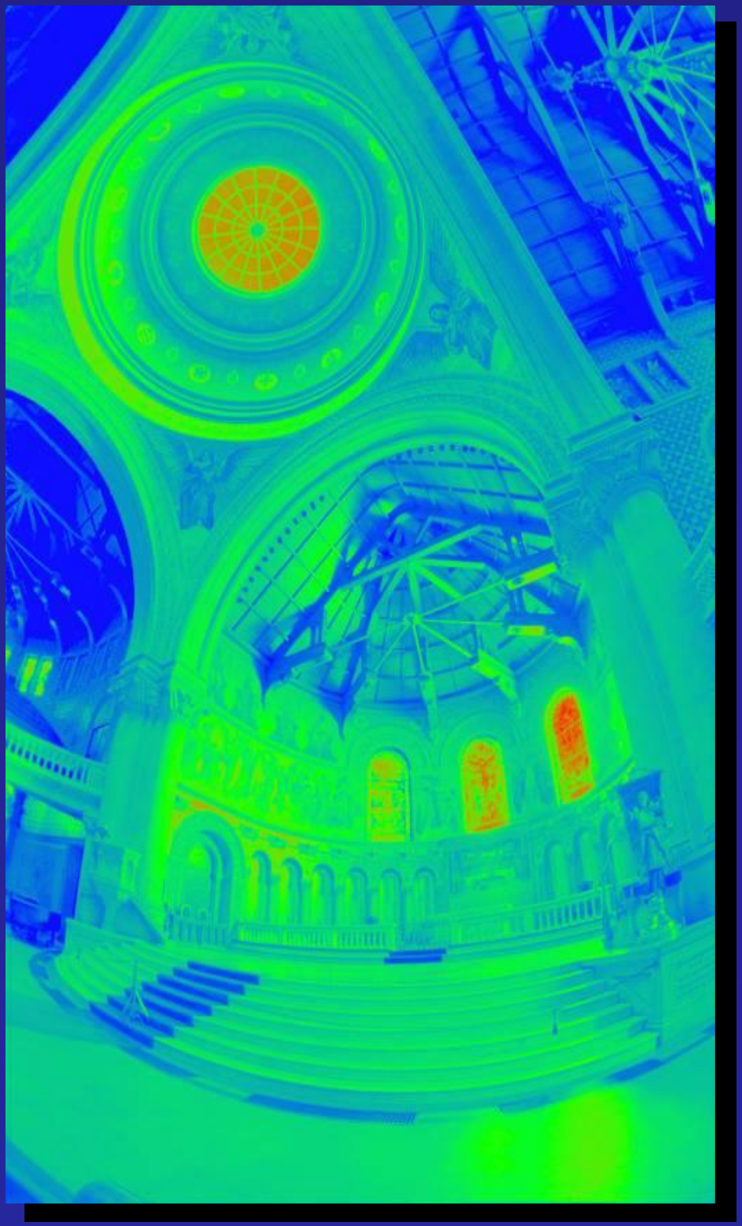
Recovered Response Curves



The Radiance Map

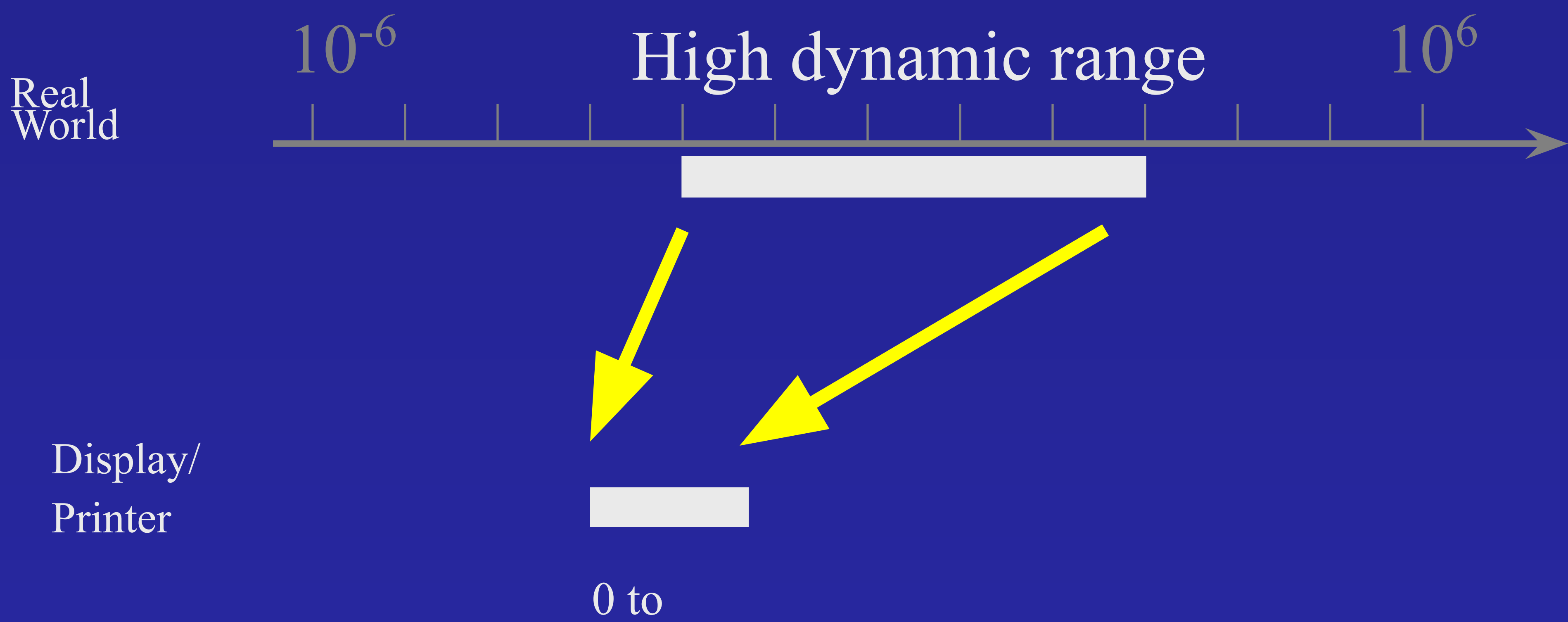


Now What?



Tone Mapping

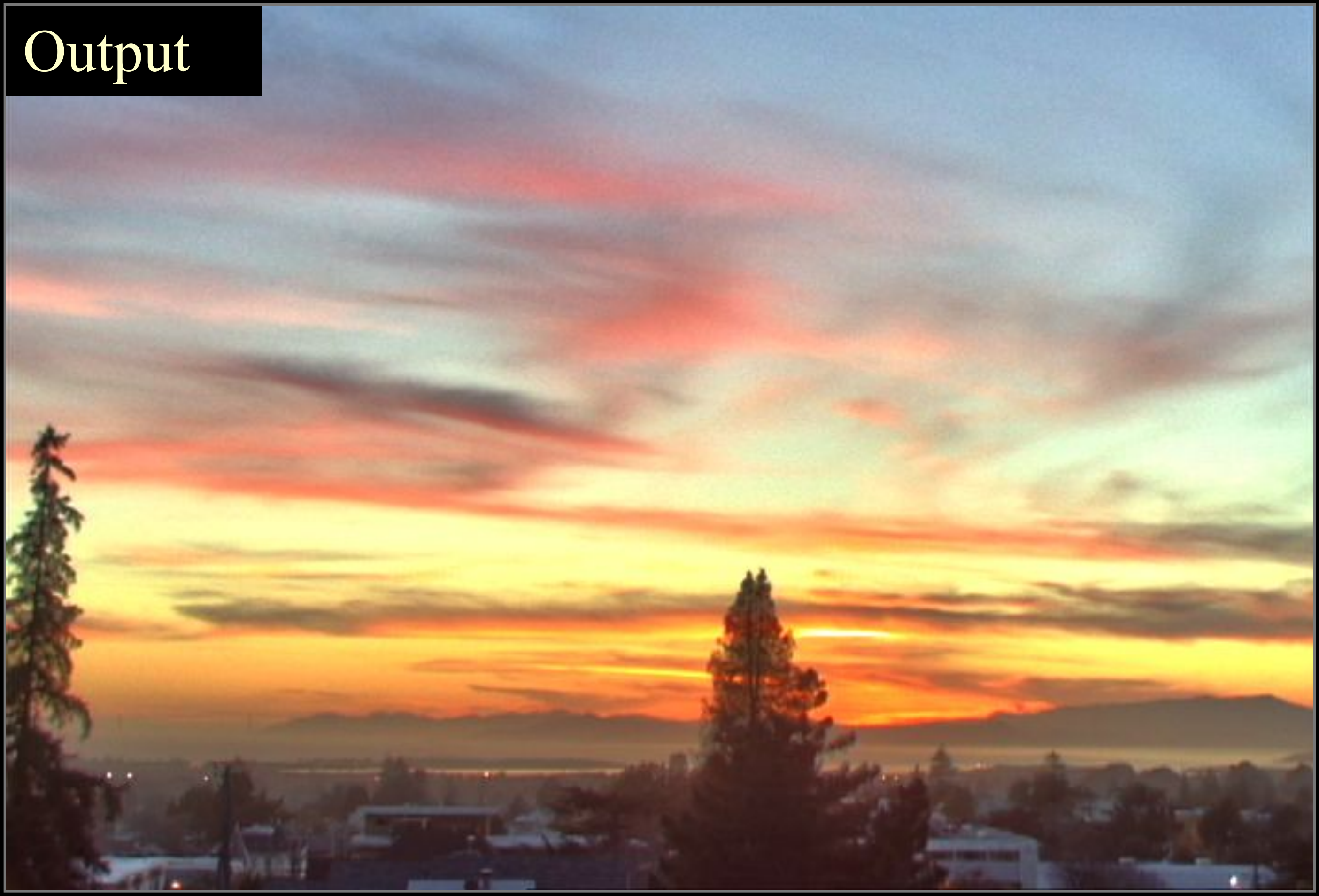
- How can we do this?



Fast Bilateral Filtering for the Display of High-Dynamic-Range Images

Frédo Durand & Julie Dorsey
Laboratory for Computer Science
Massachusetts Institute of Technology
SIGGRAPH 2002

Our approach



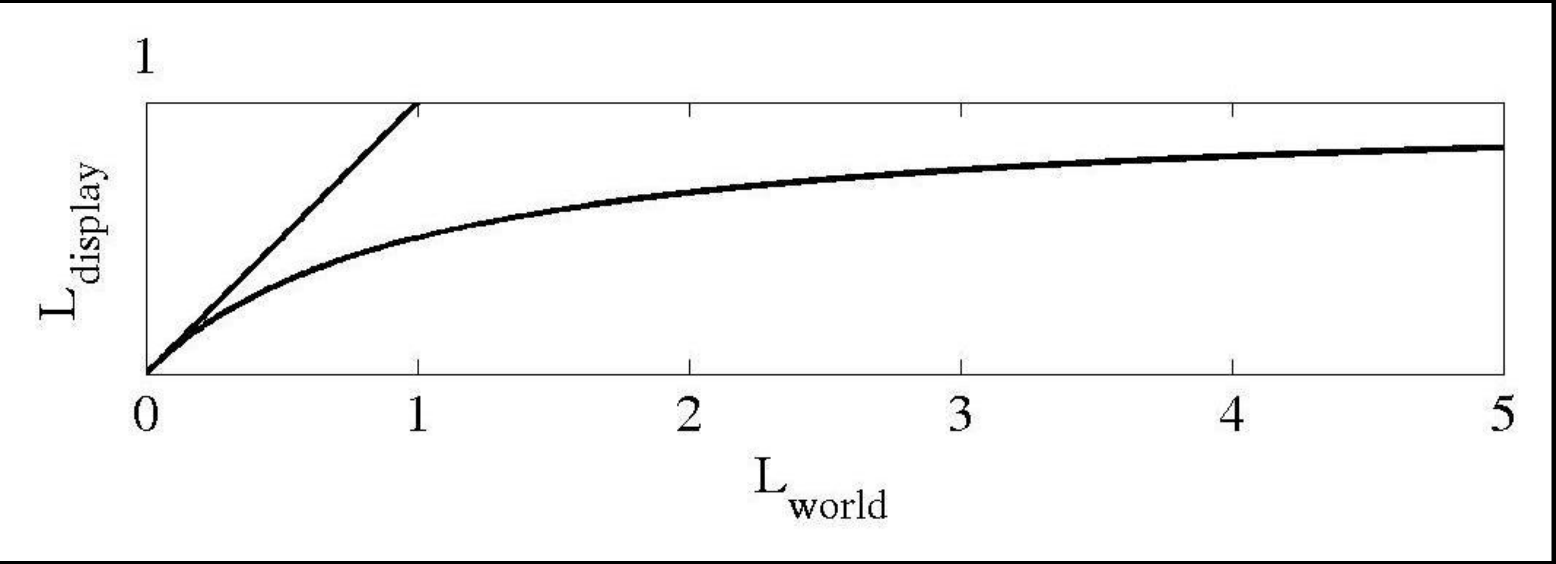
Clipping

- Sun is overexposed
- Foreground is underexposed

Scaling

Global Operator (Reinhart et al)

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



Global vs local

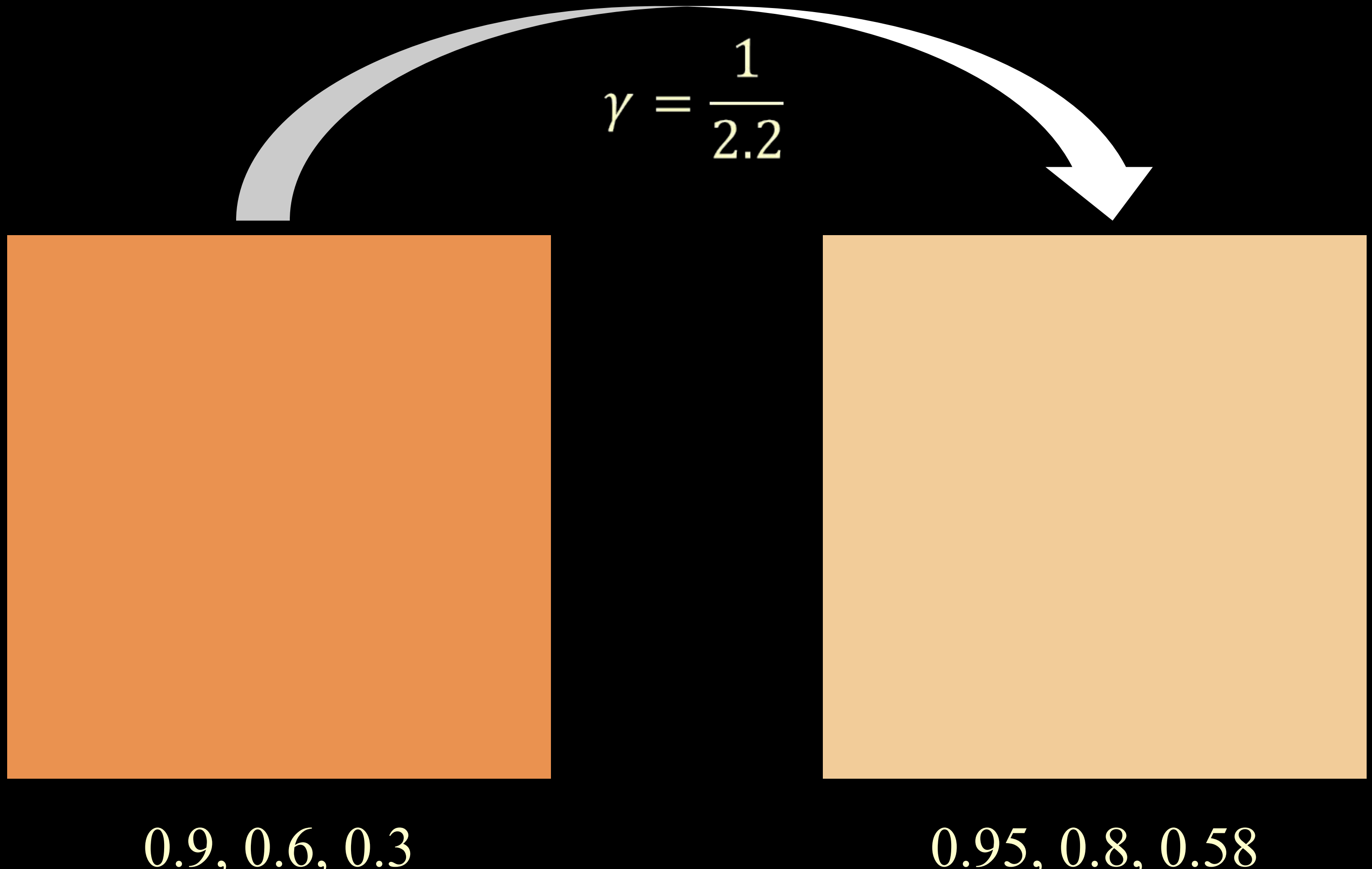
Photographic
[Reinhard et al.]

Bilateral
[Durand et al.]

Gamma compression

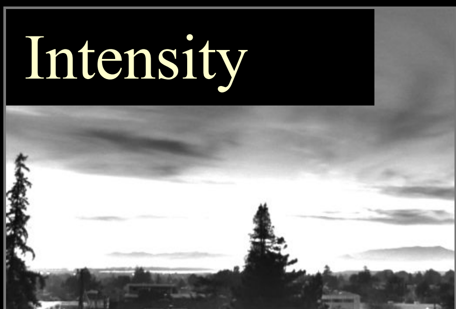
- $X \rightarrow X^{\gamma}$
- Colors are washed-out

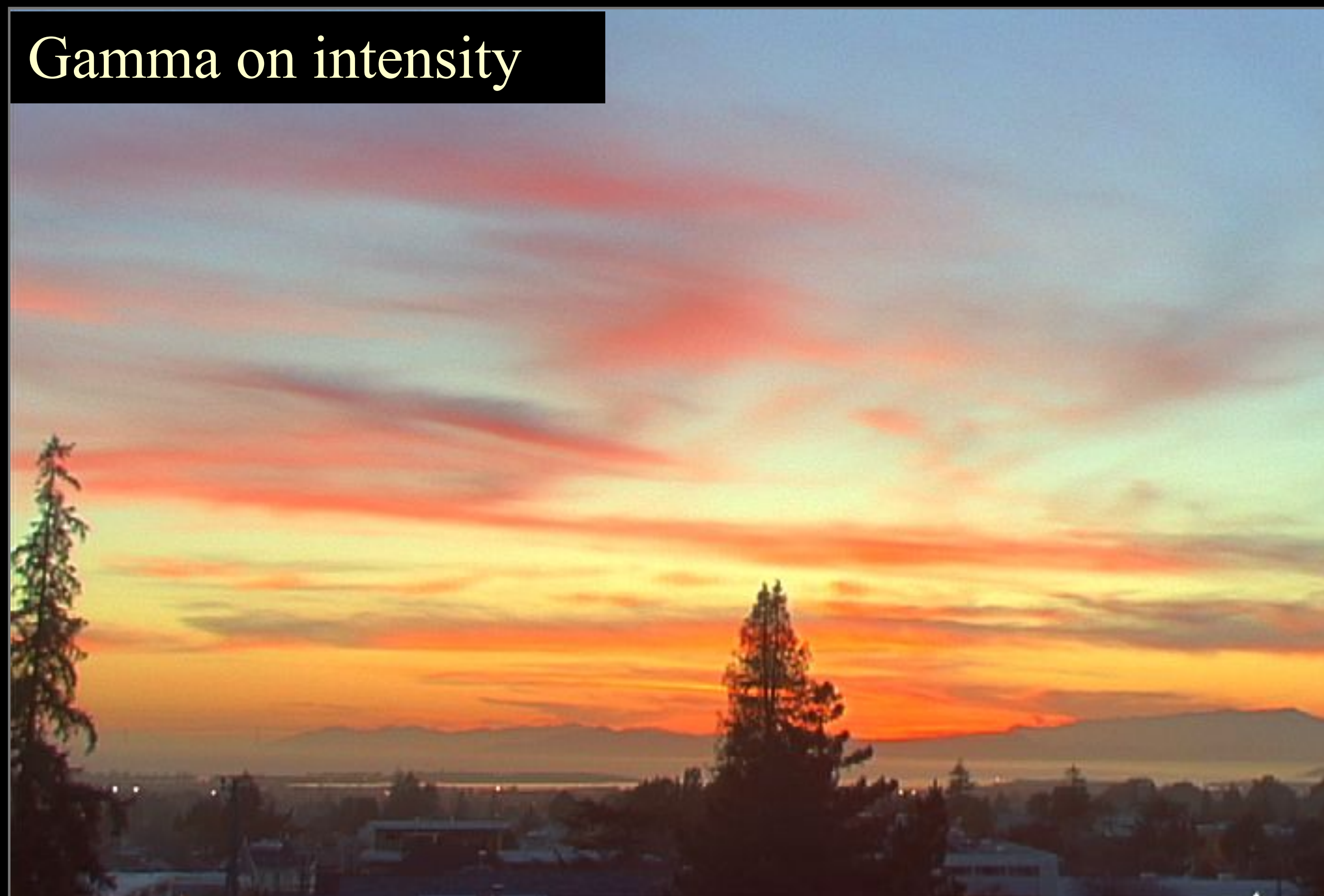
Effect of gamma on color



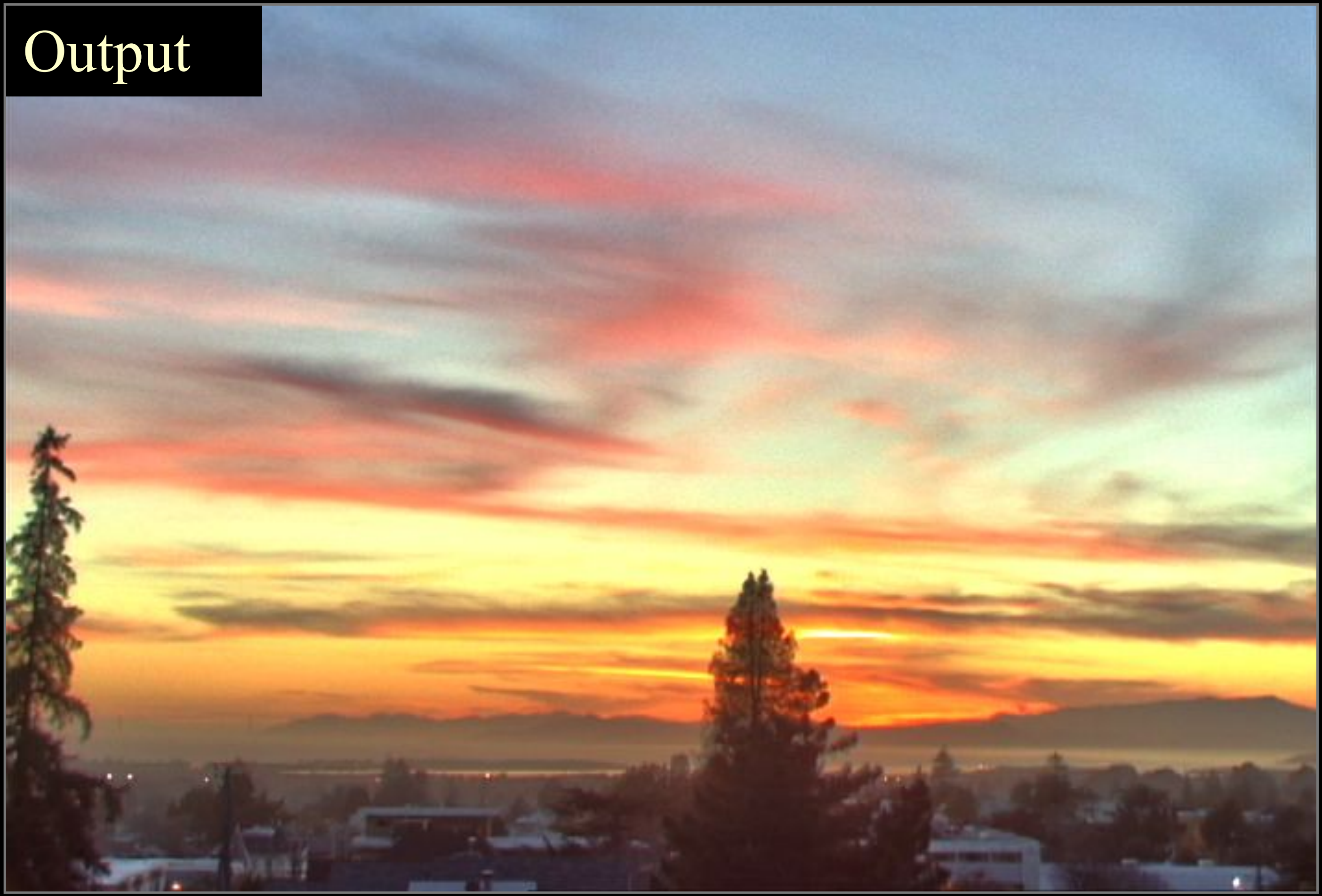
Gamma compression on intensity

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





Our approach



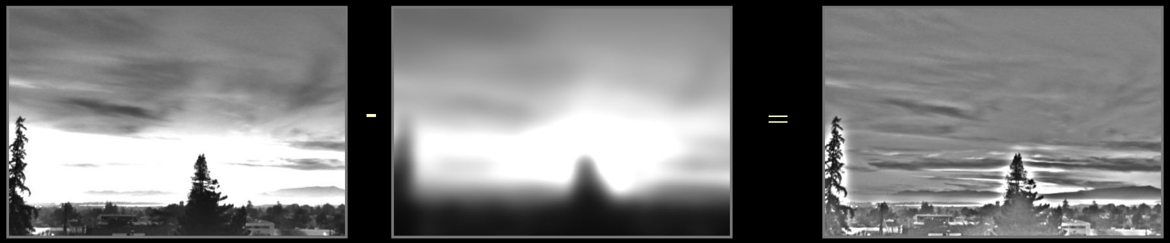
Chiu et al. 1993

- Reduce contrast of low-frequencies
- Keep high frequencies

Quick low/high frequency computation

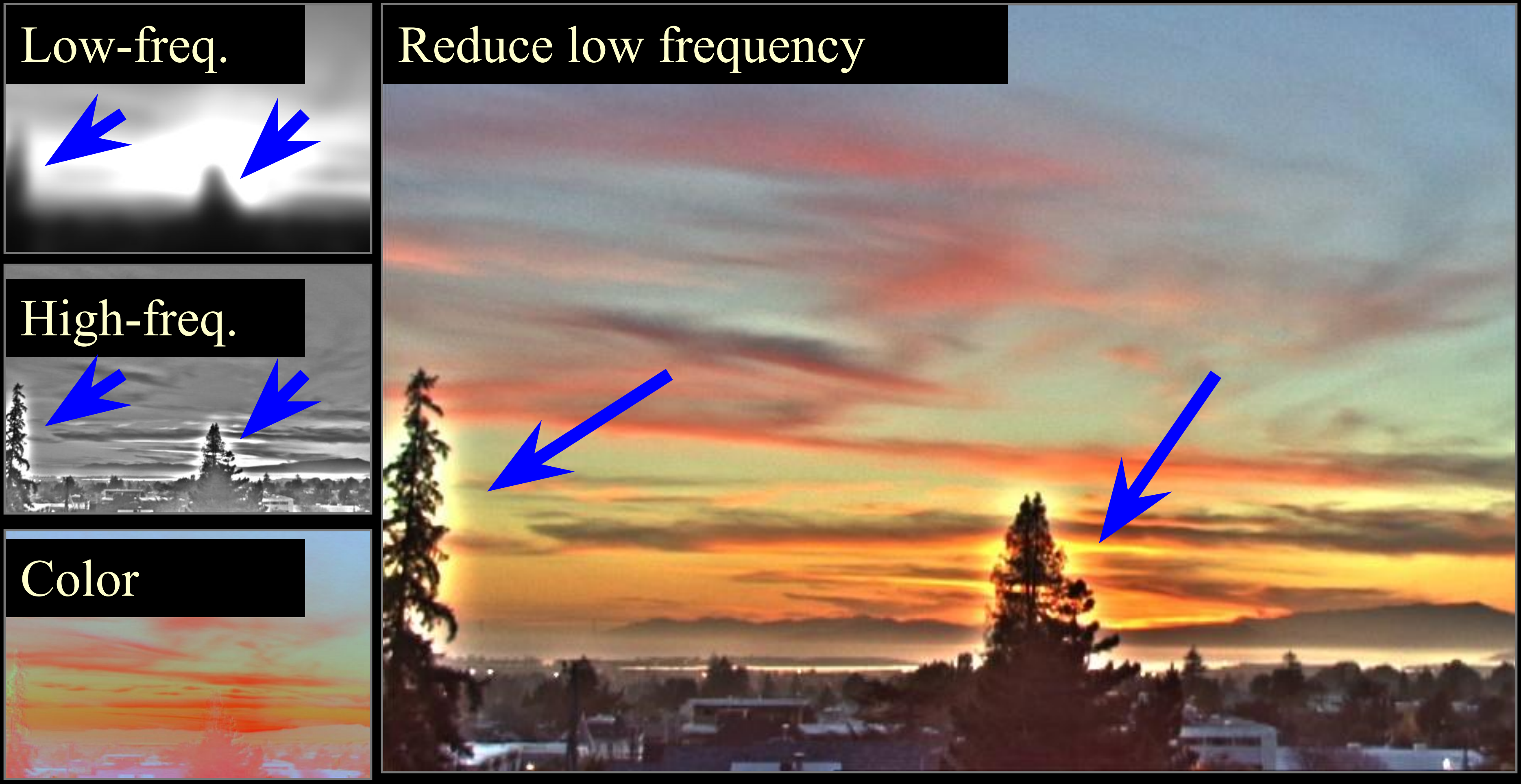
- Blur intensity to get low frequencies

- Subtract original from low frequency



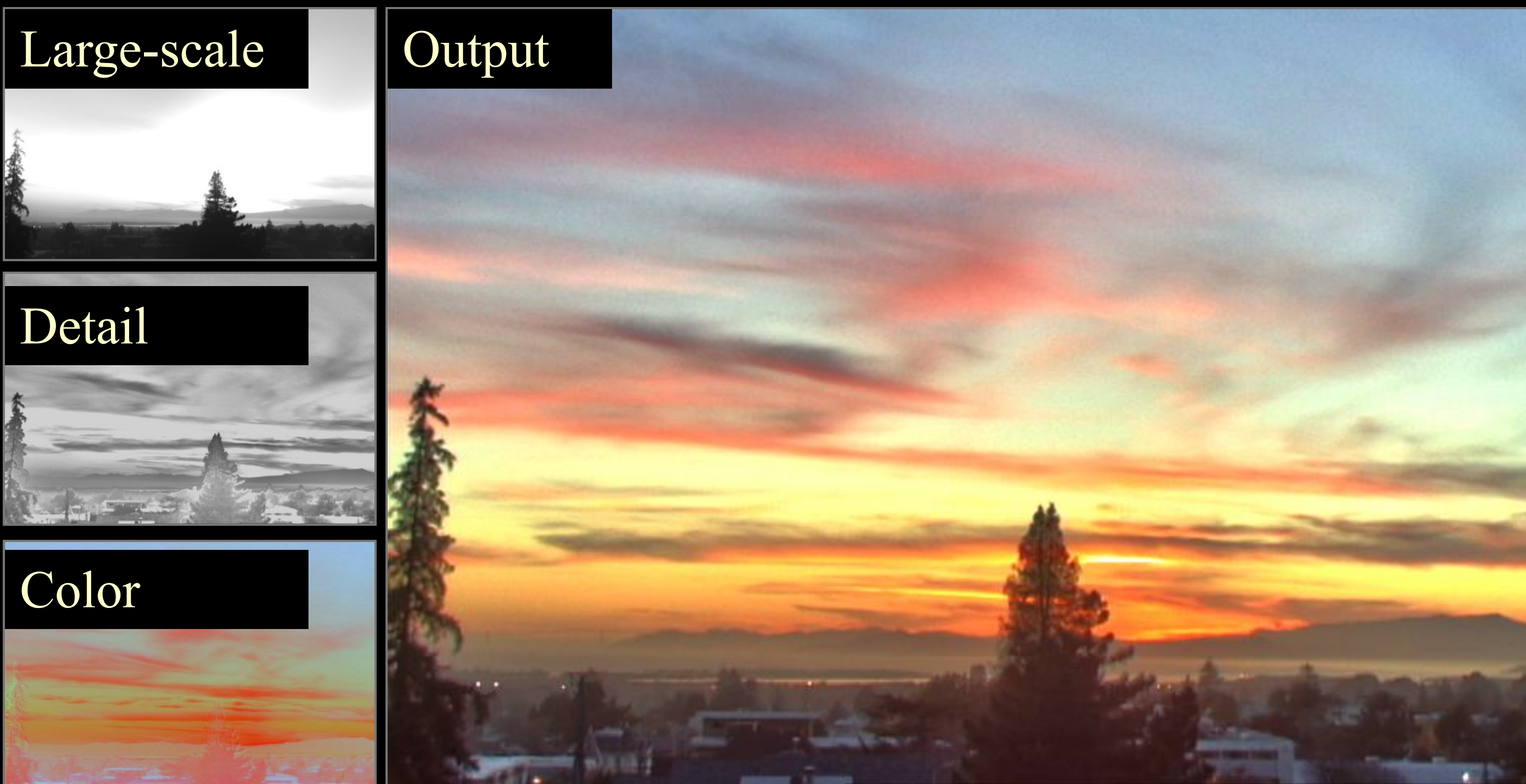
The halo nightmare

- For strong edges
- Because they contain high frequency



Our approach

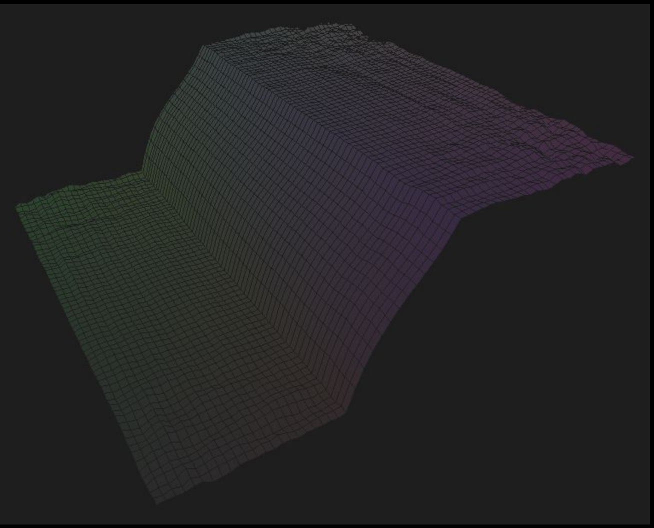
- Do not blur across edges
- Non-linear filtering

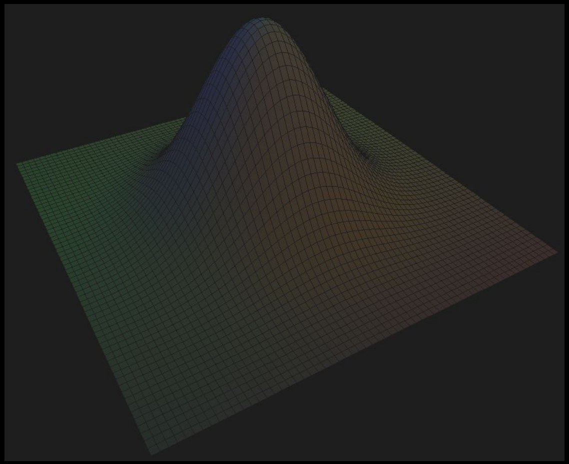


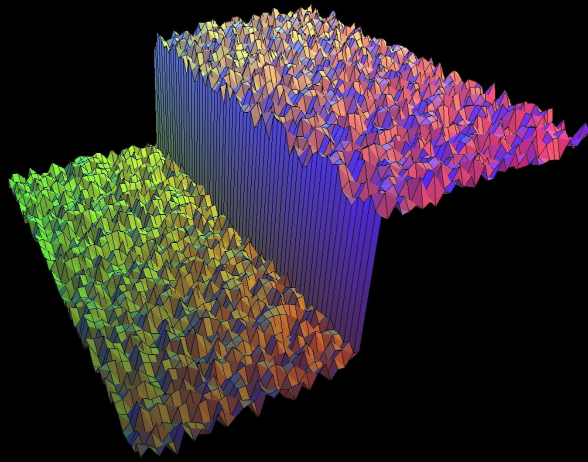
Start with Gaussian filtering

- Here, input is a step function + noise

$$J \quad = \quad f \quad \otimes \quad I$$





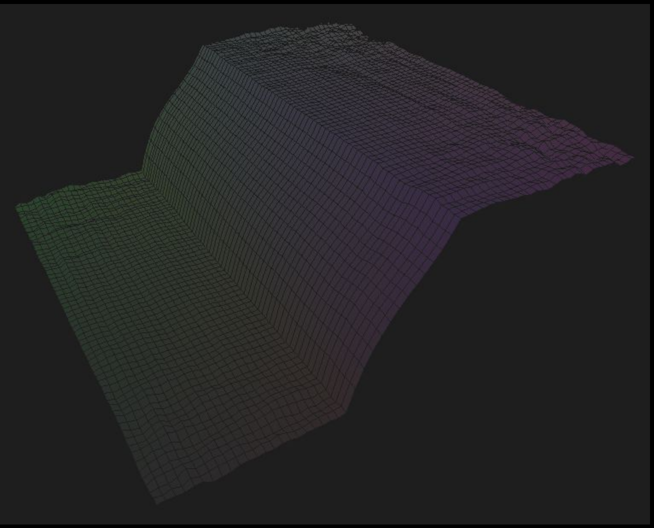


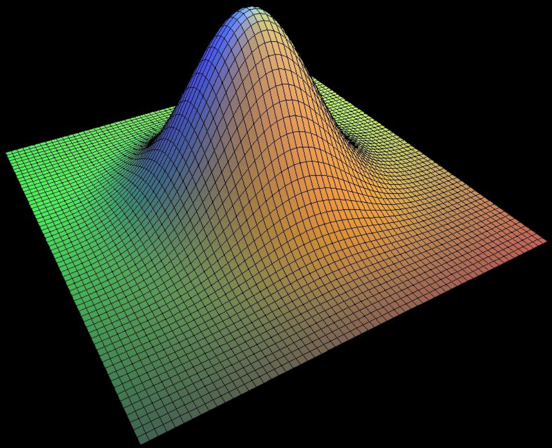
output input

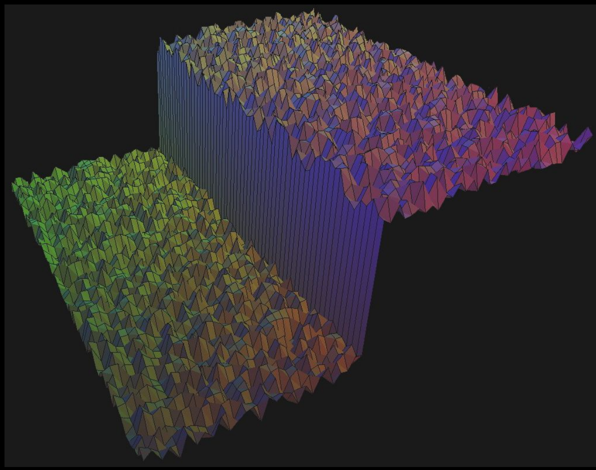
Start with Gaussian filtering

- Spatial Gaussian f

$$J = f \otimes I$$





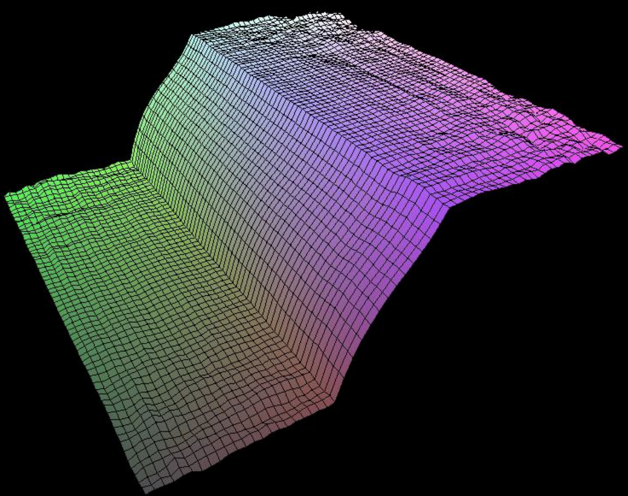


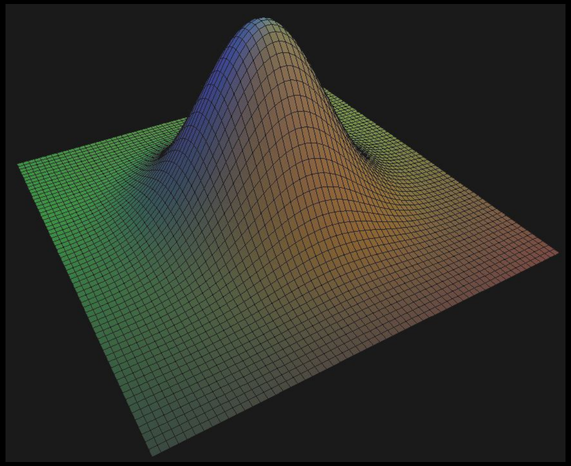
output ⟸ input

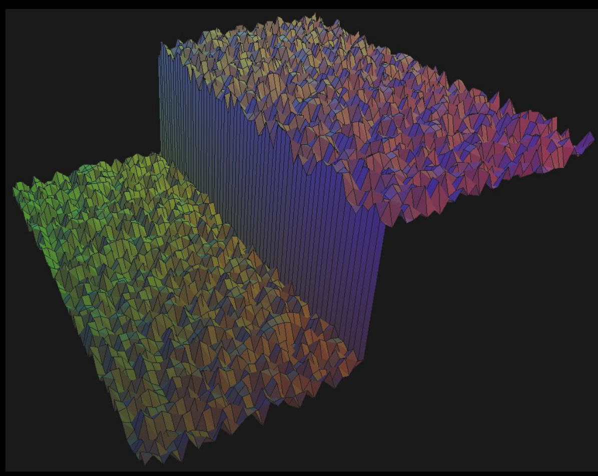
Start with Gaussian filtering

- Output is blurred

$$J = f \otimes I$$





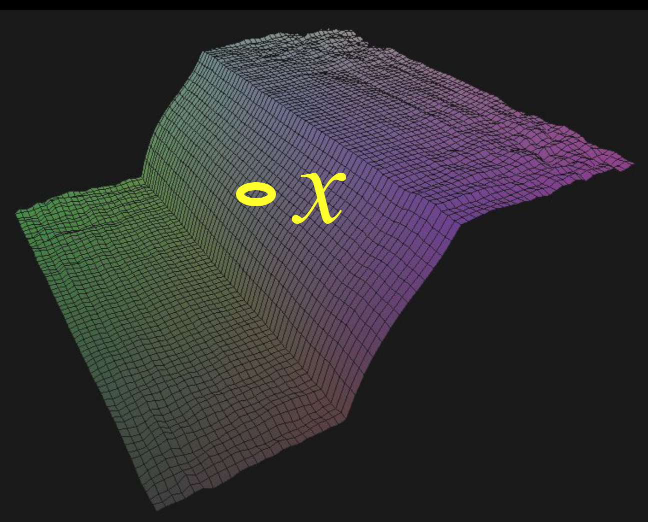


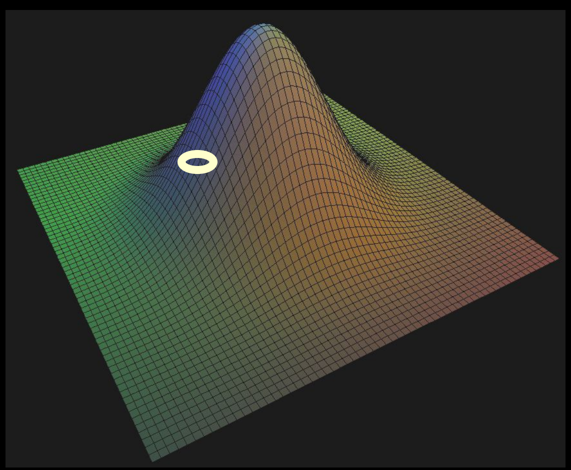
output input

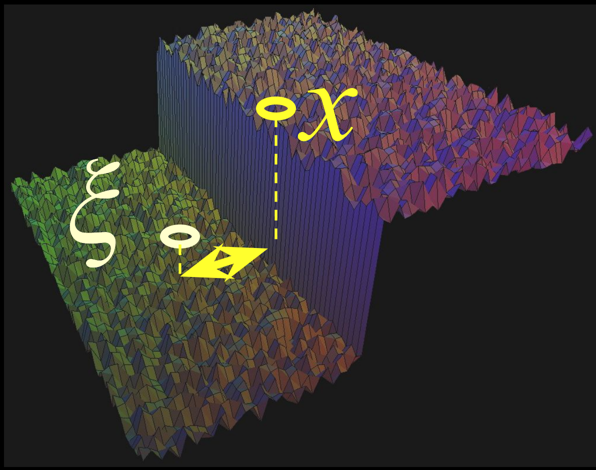
Gaussian filter as weighted average

- Weight of ξ depends on distance to x

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





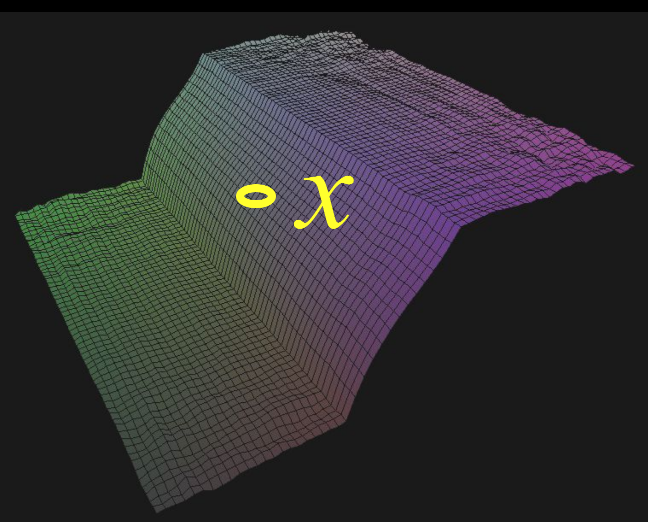


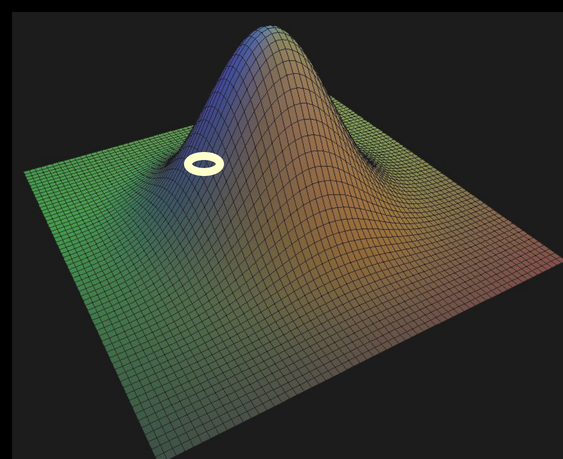
output ⟵ input

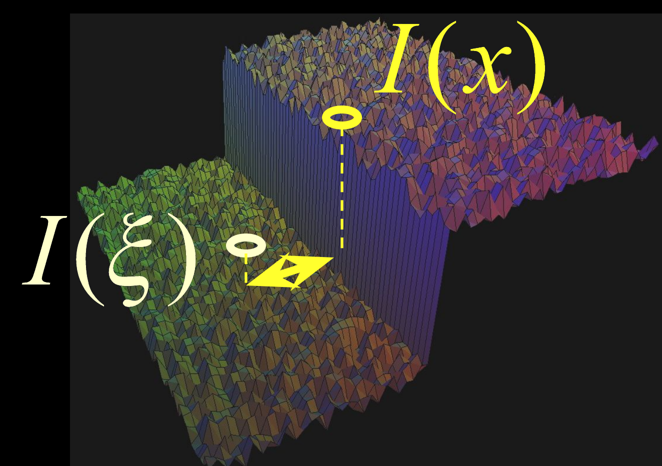
The problem of edges

- Here, $I(\xi)$ “pollutes” our estimate J(x)
- It is too different

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







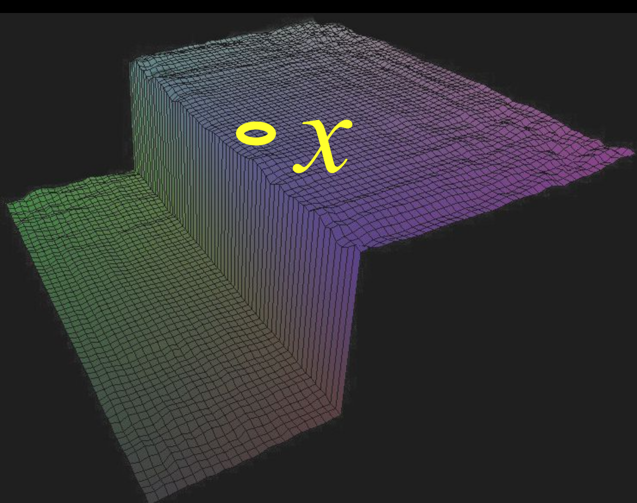
output ⟵ input

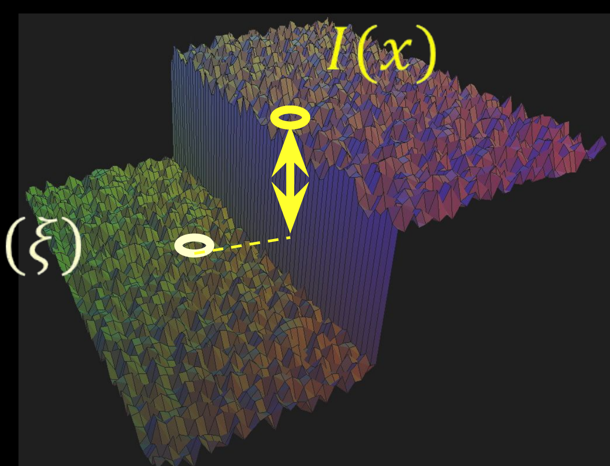
Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- Penalty g on the intensity difference

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





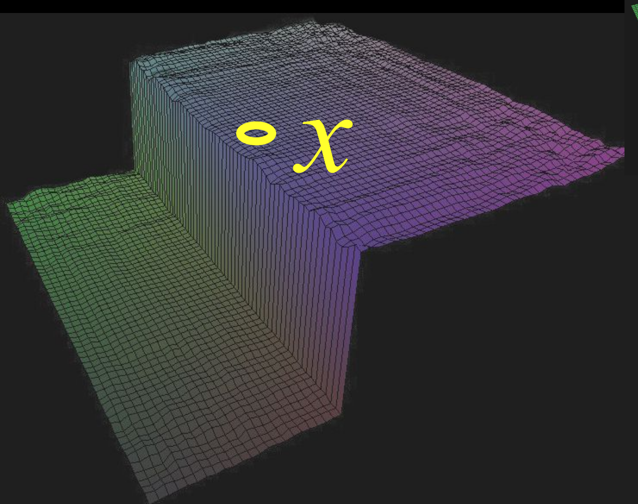
output ⟵ input

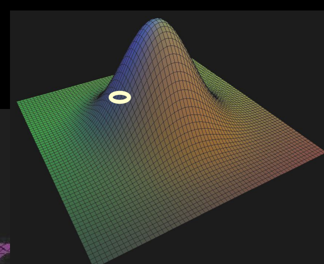
Bilateral filtering

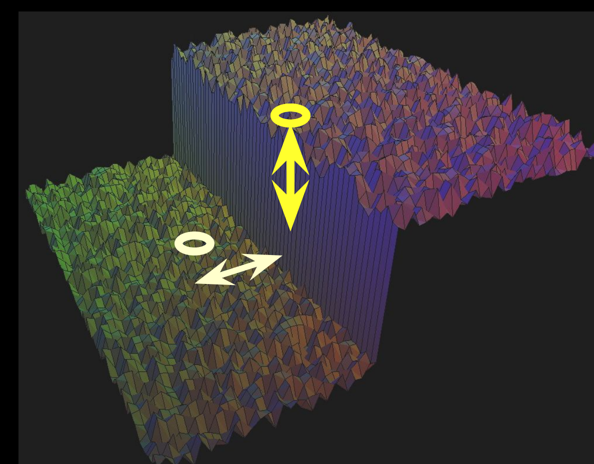
[Tomasi and Manduchi 1998]

- Spatial Gaussian f

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







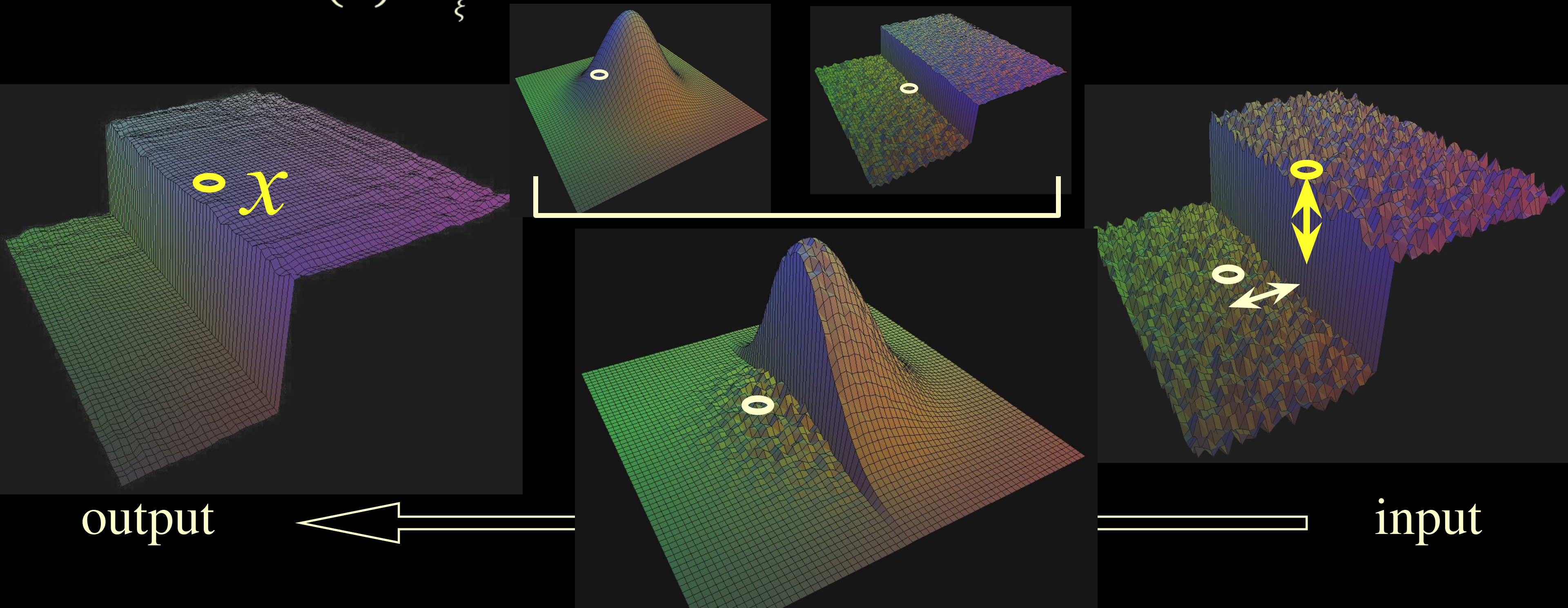
output input

Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian f
- Gaussian g on the intensity difference

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

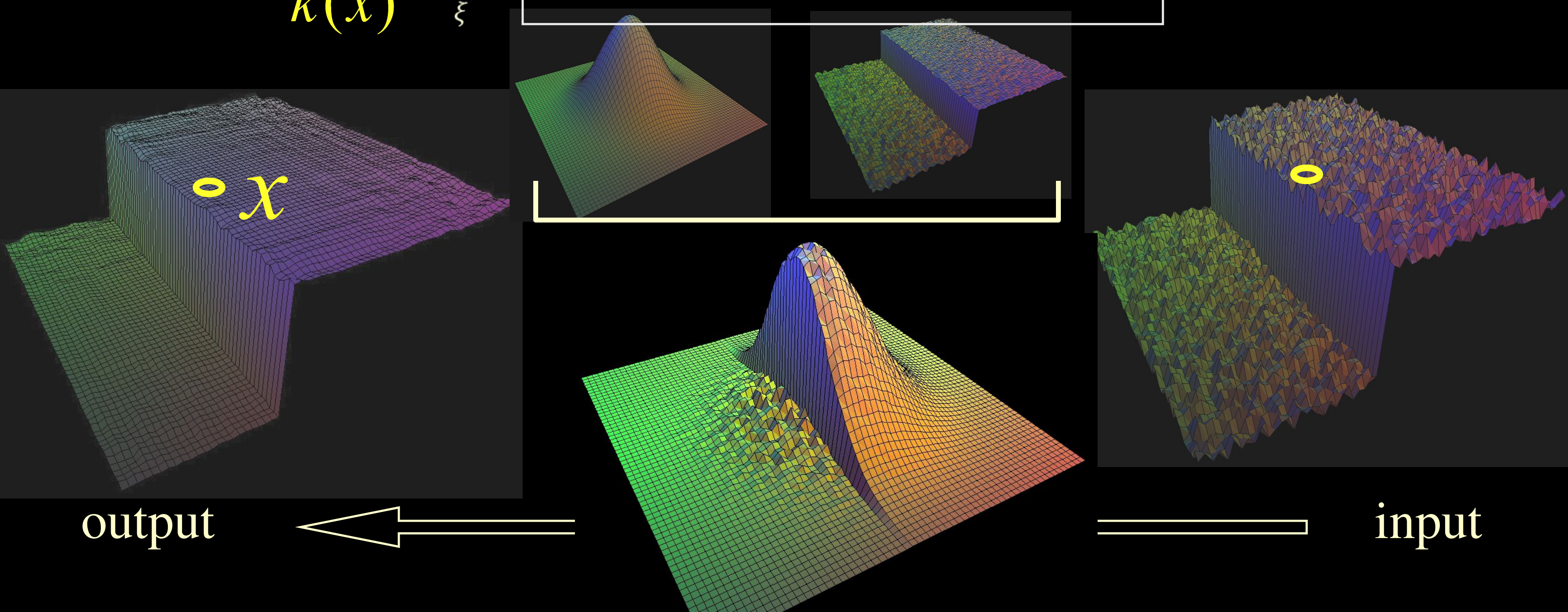


Normalization factor

[Tomasi and Manduchi 1998]

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

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

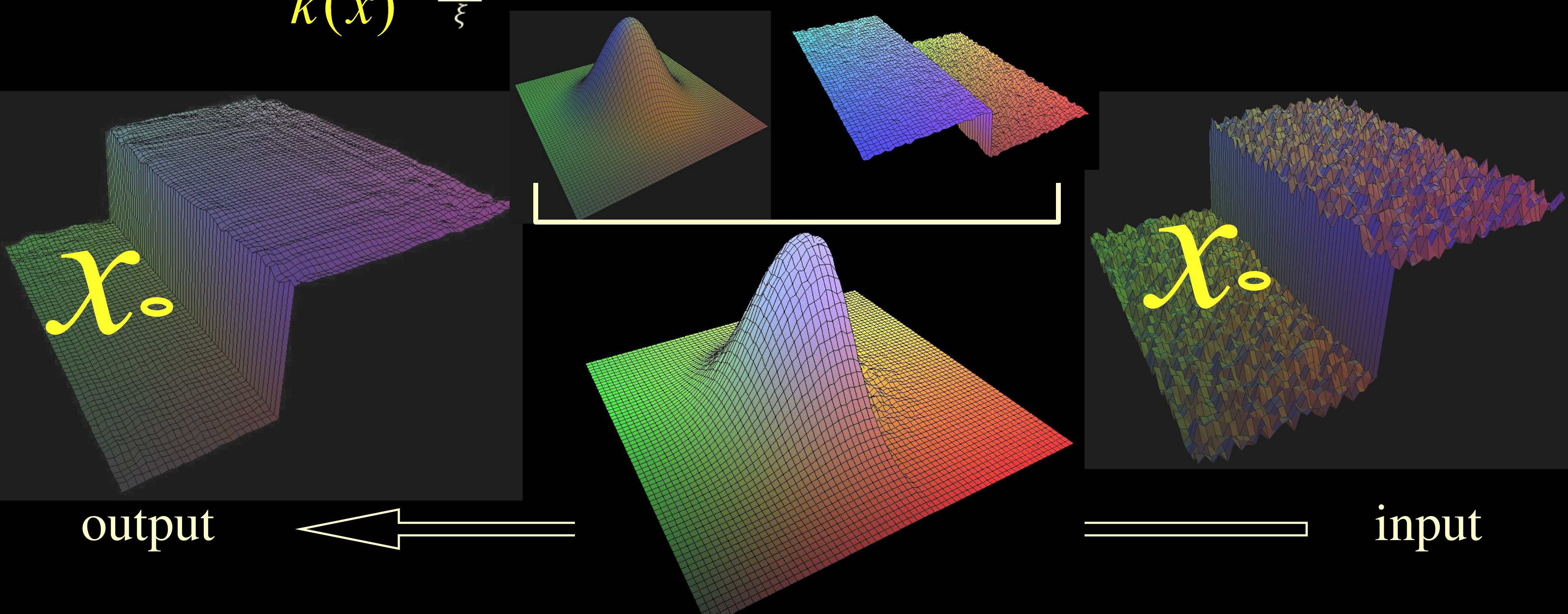


Bilateral filtering is non-linear

[Tomasi and Manduchi 1998]

- The weights are different for each output pixel

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

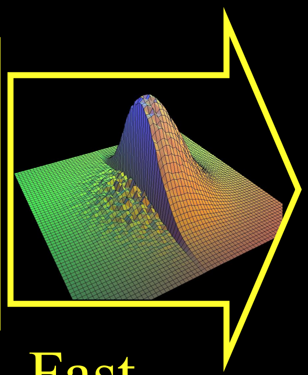


Contrast reduction

Contrast
too high!

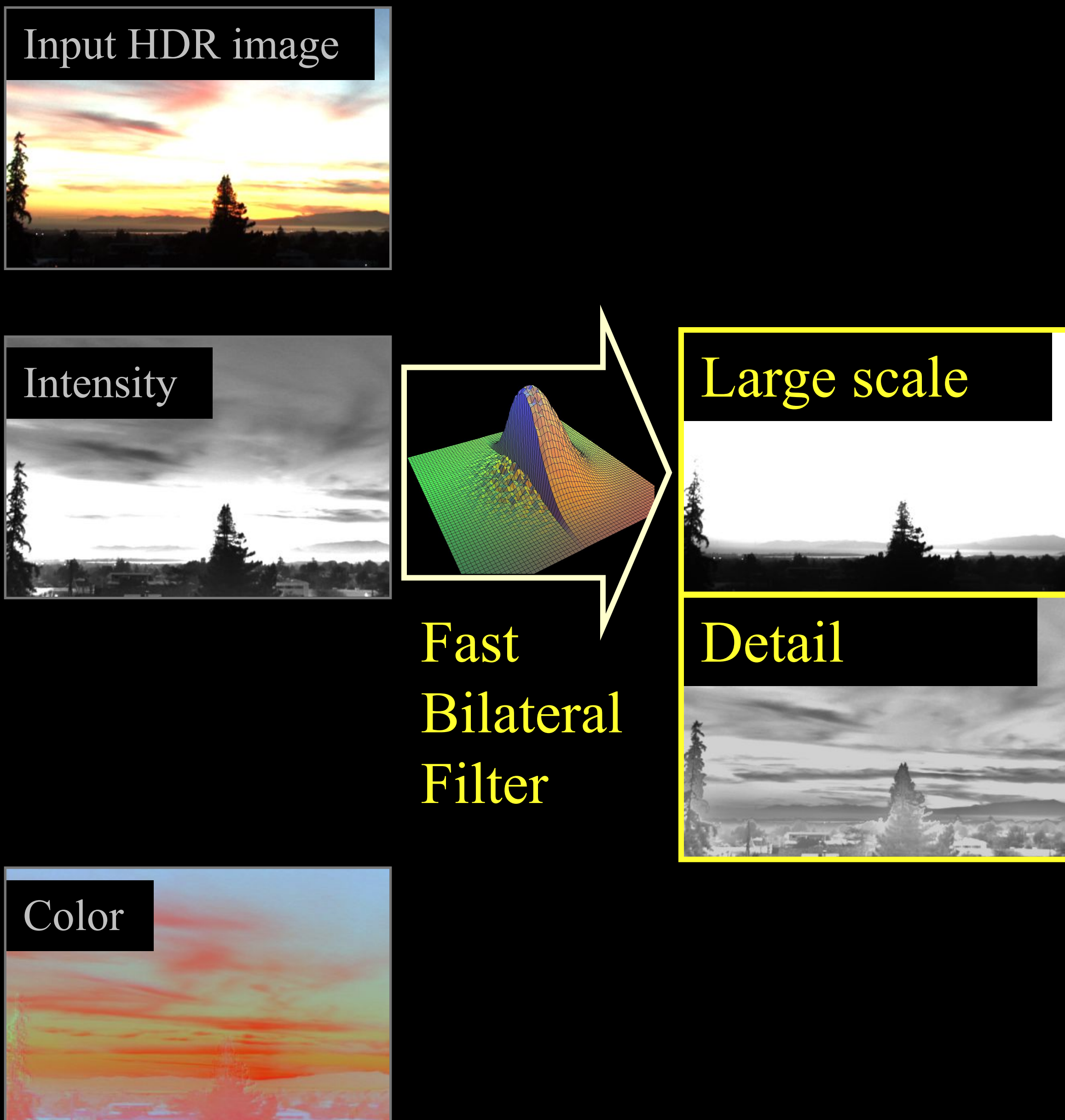
Contrast reduction

Contrast reduction

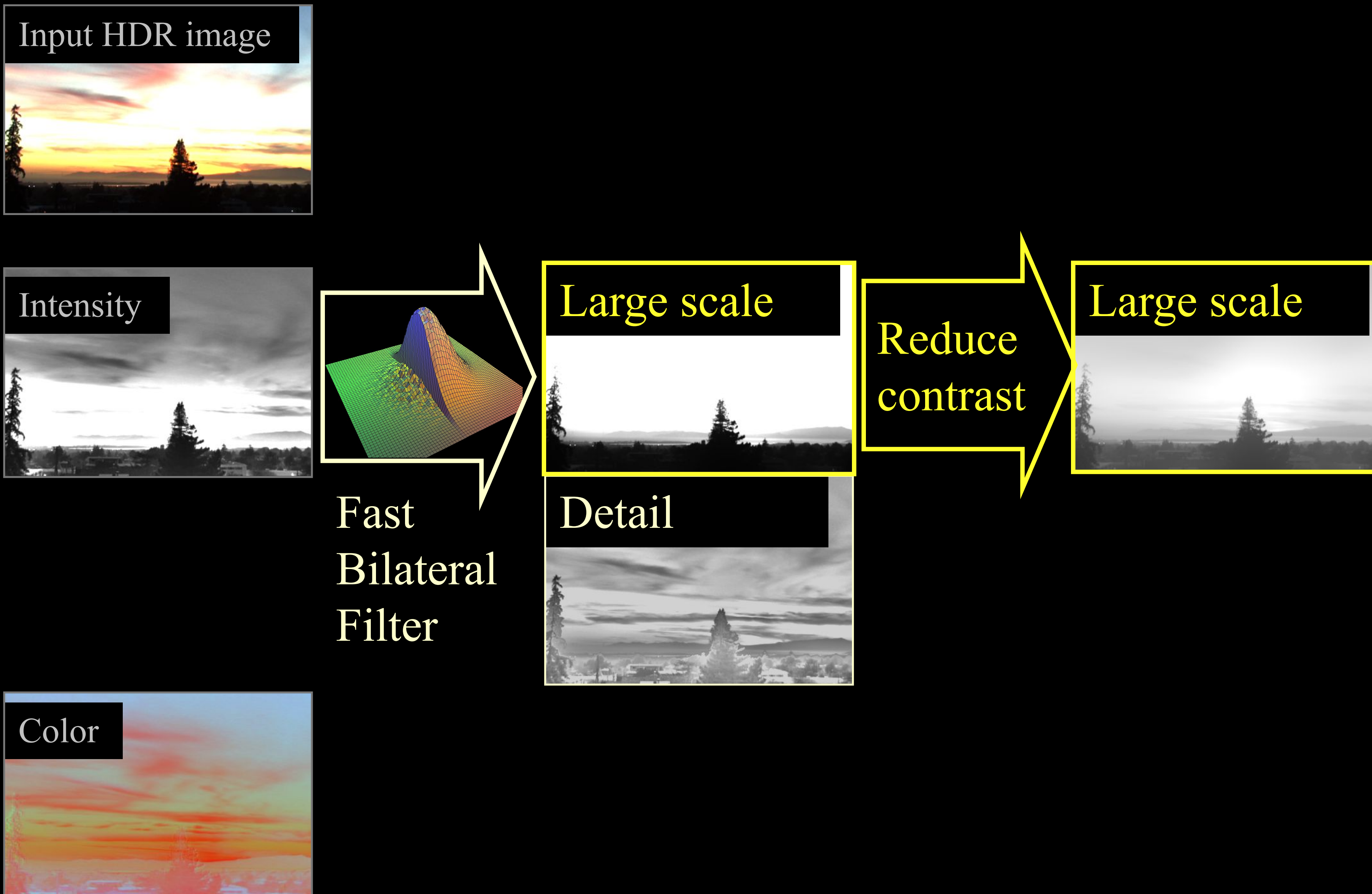


Fast
Bilateral
Filter

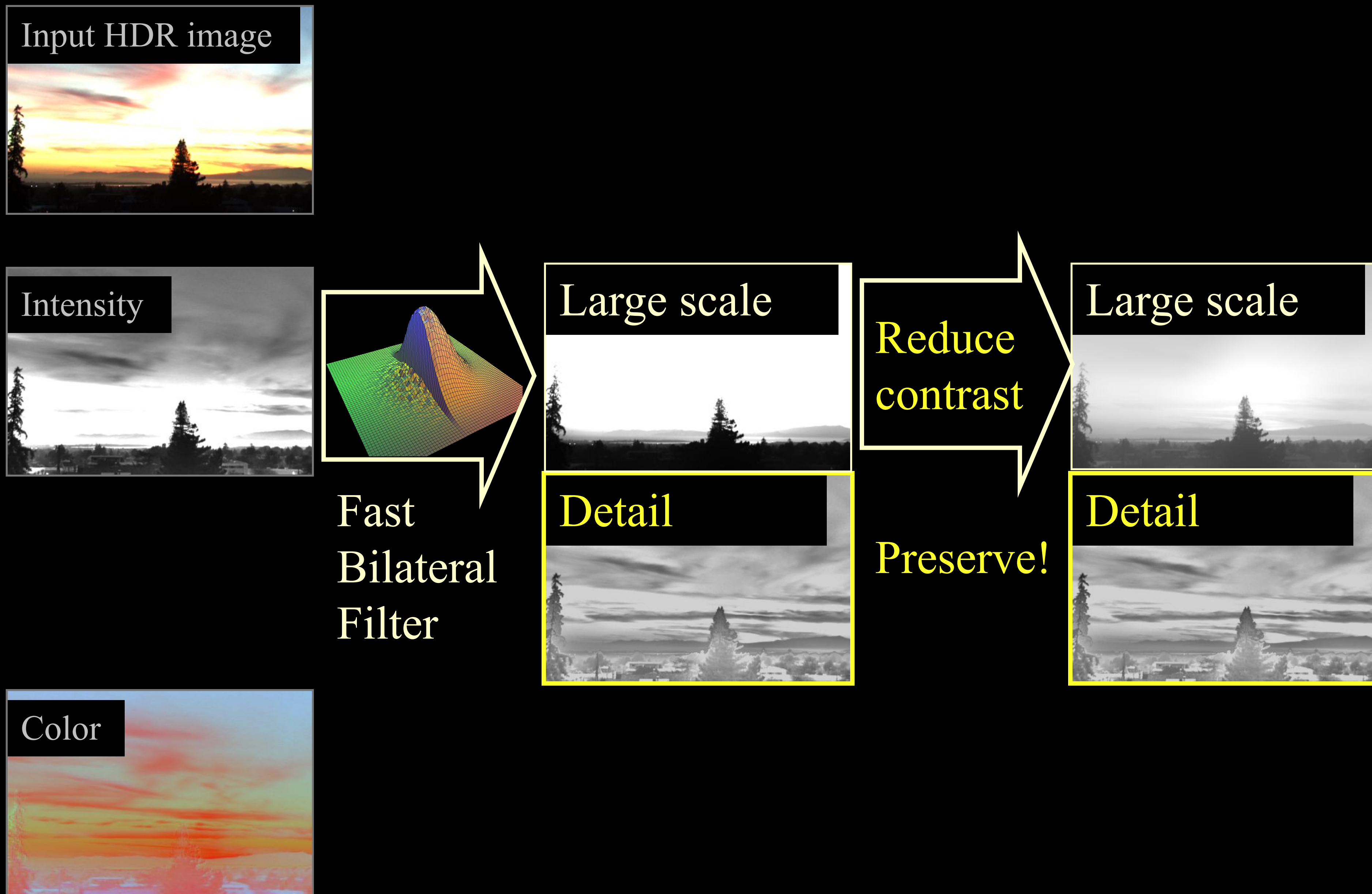
Contrast reduction



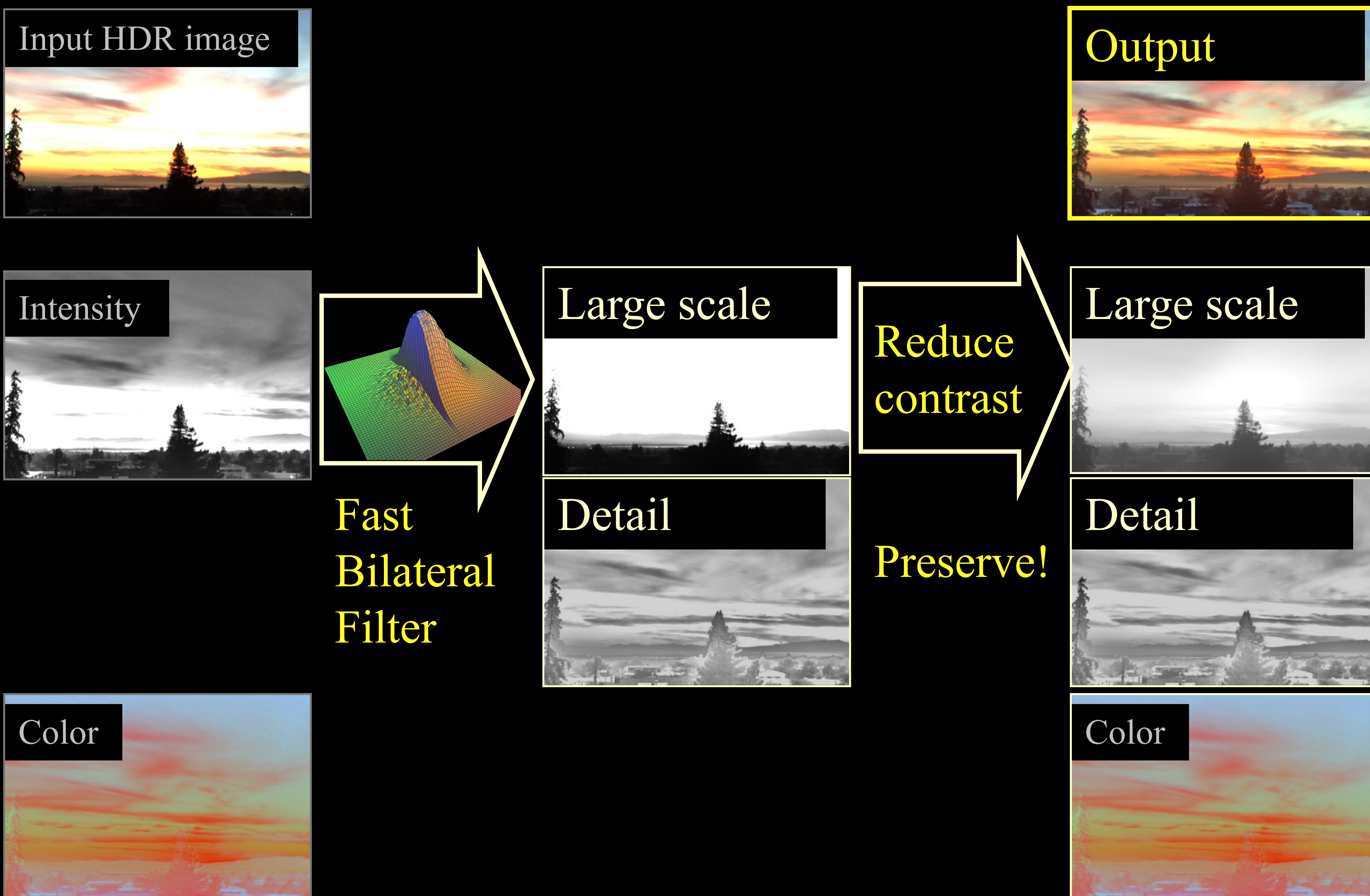
Contrast reduction



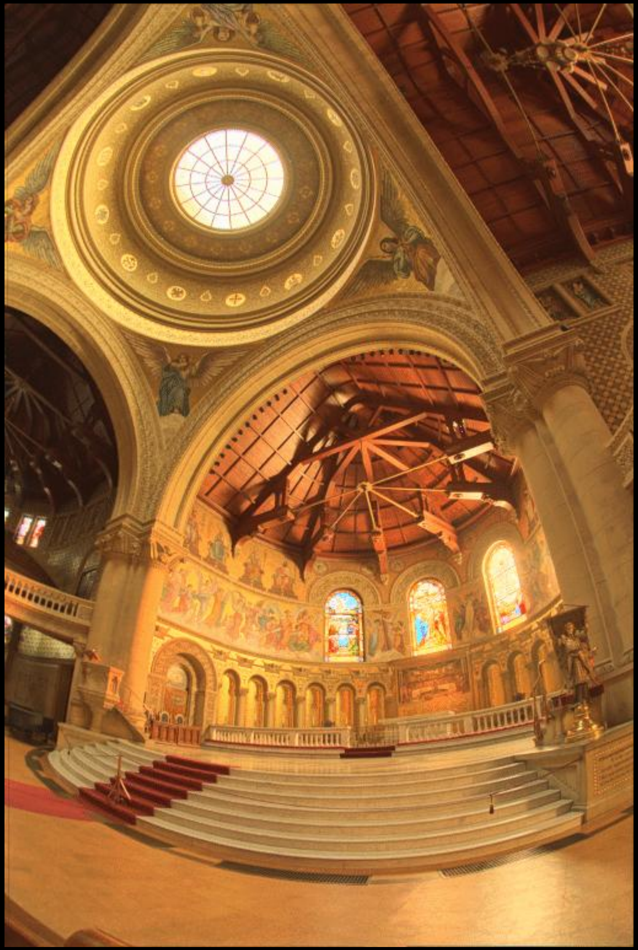
Contrast reduction



Contrast reduction



Informal comparison



Photographic
[Reinhard et al.]

Bilateral
[Durand et al.]

Informal comparison

Photographic
[Reinhard et al.]

Bilateral
[Durand et al.]