Tone mapping

Digital Visual Effects, Spring 2007

Yung-Yu Chuang

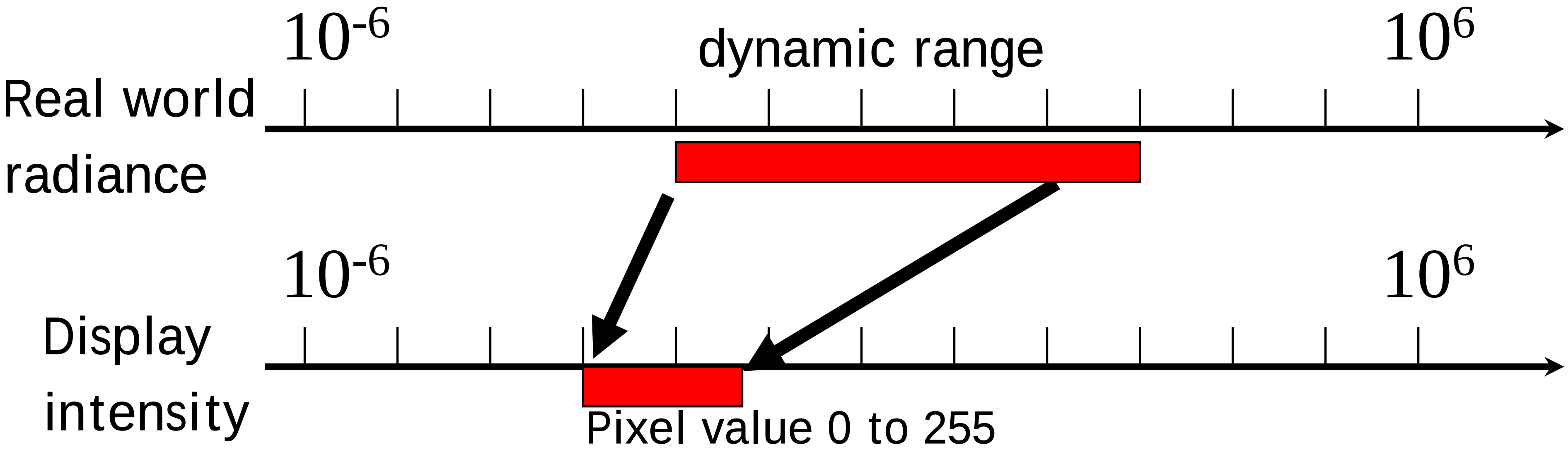
2007/3/13

with slides by Fredo Durand, and Alexei Efros

Tone mapping

- How can we display it?

  Linear scaling?, thresholding?



CRT has 300:1 dynamic range

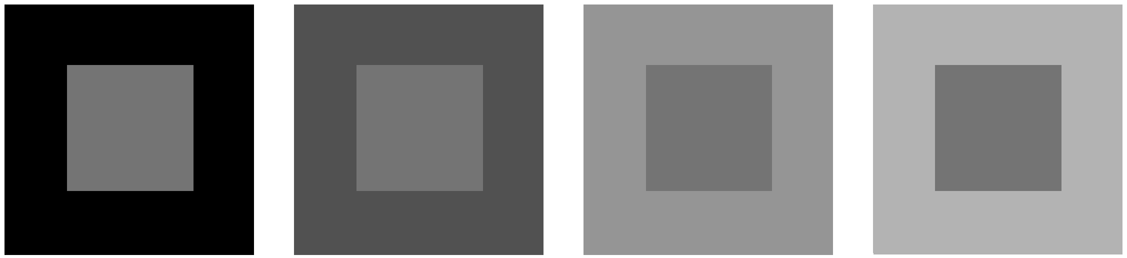
Preliminaries

- For color images

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \dfrac{R_w}{L_w} \\ L_d \dfrac{G_w}{L_w} \\ L_d \dfrac{B_w}{L_w} \end{bmatrix}$$

- Log domain is usually preferred.
- Gaussian filter. Sampling issues. Efficiency issues.

Eye is not a photometer!



- *"Every light is a shade, compared to the higher lights, till you come to the sun; and every shade is a light, compared to the deeper shades, till you come to the night."*

— John Ruskin, 1879

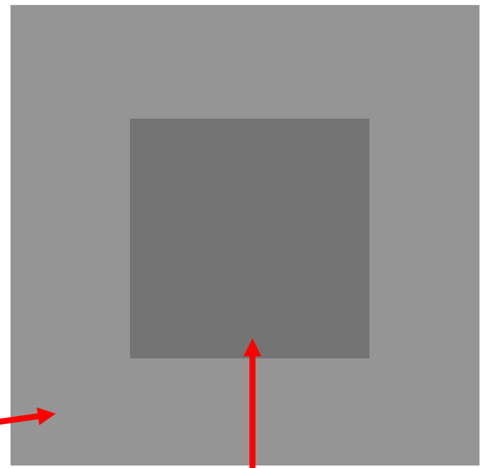
We are more sensitive to contrast

- Weber’s law

Just-noticeable Difference (JND)

$$\frac{\Delta I_b}{I_b} \sim 1\%$$

background intensity



flash

Global operator (Reinhart et al)

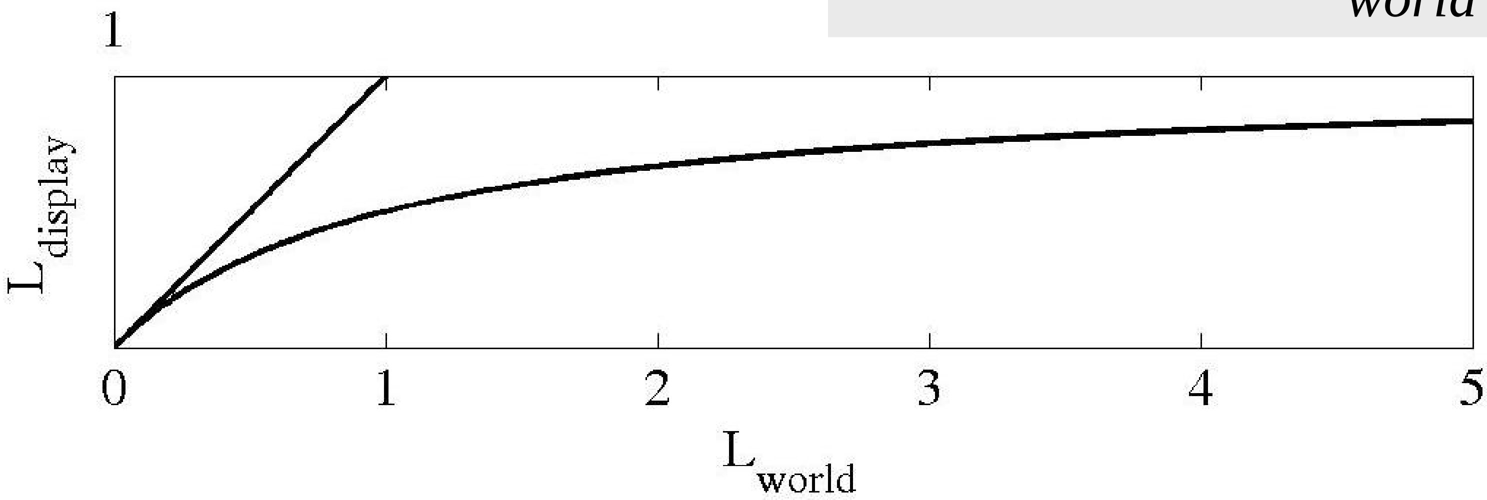
$$\bar{L} = \exp\left(\frac{1}{N}\sum_{x,y}\log(\delta + L(x,y))\right)$$

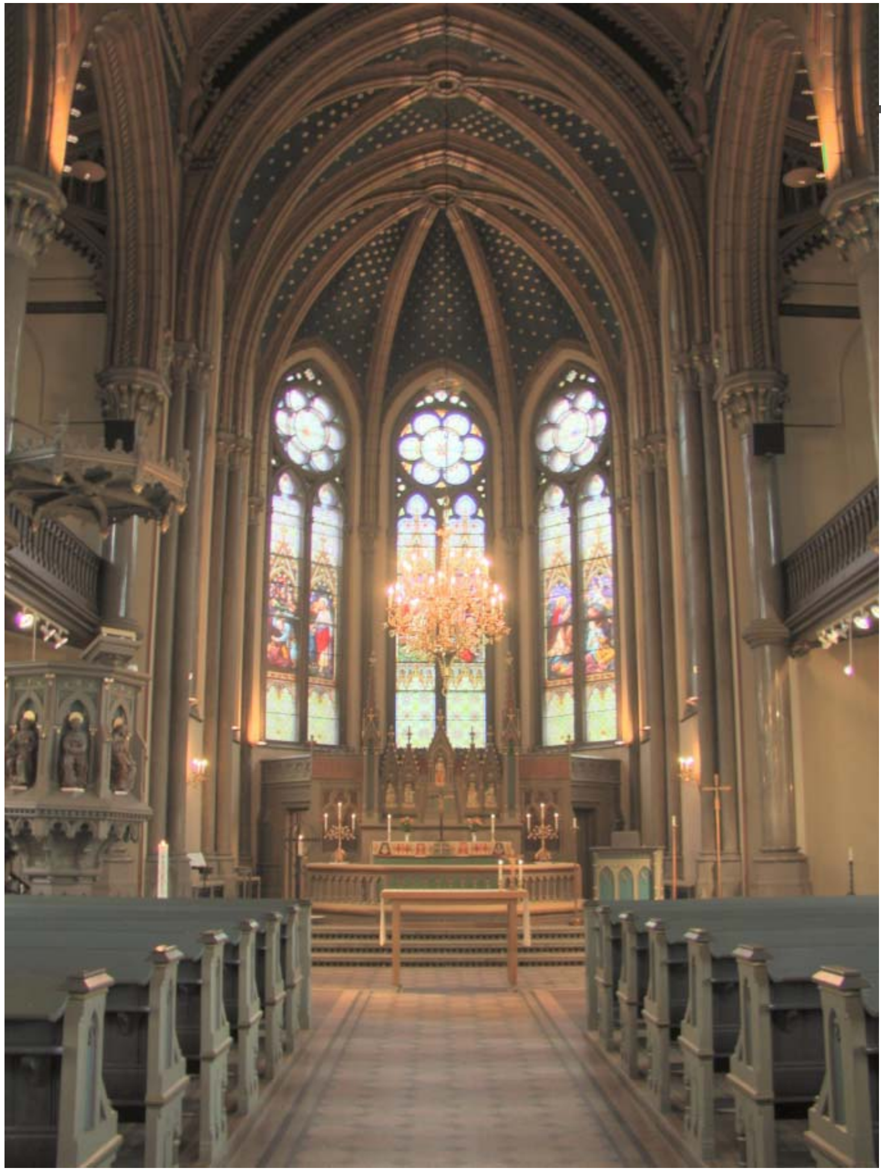
Approximation of scene's key (how light or dark it is). Map to 18% of display range for average-key scene

User-specified; high key or low key

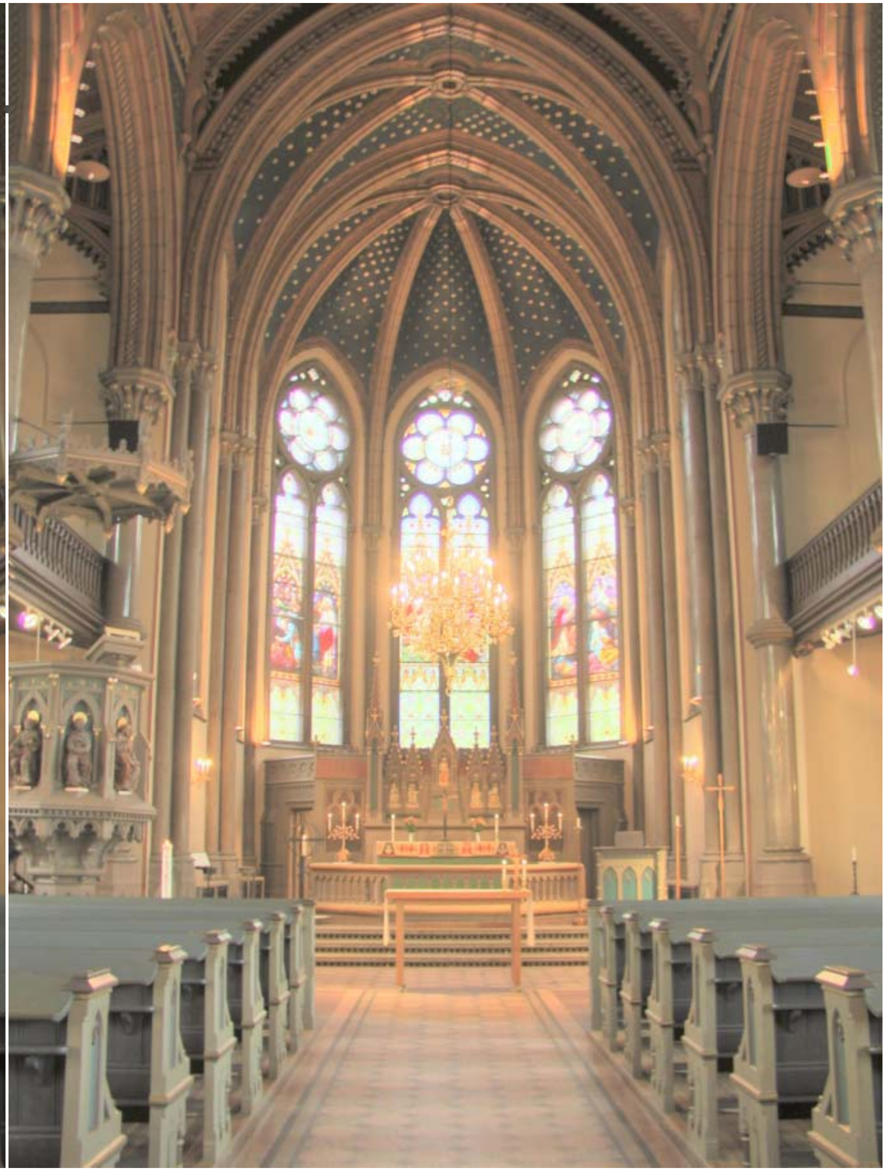
$$L_w(x,y) = \frac{a}{\bar{L}} L(x,y)$$

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





low key (0.18)



high key (0.5)

Frequency domain

- First proposed by Oppenheim in 1968!
- Under simplified assumptions,

image = illuminance * reflectance

low-frequency attenuate more

high-frequency attenuate less



Oppenheim

- Taking the logarithm to form density image
- Perform FFT on the density image
- Apply frequency-dependent attenuation filter

$$s(f) = (1-c) + c\frac{kf}{1+kf}$$

- Perform inverse FFT
- Take exponential to form the final image

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

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

A typical photo

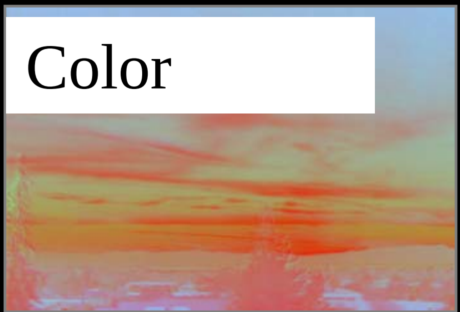
- Sun is overexposed
- Foreground is underexposed

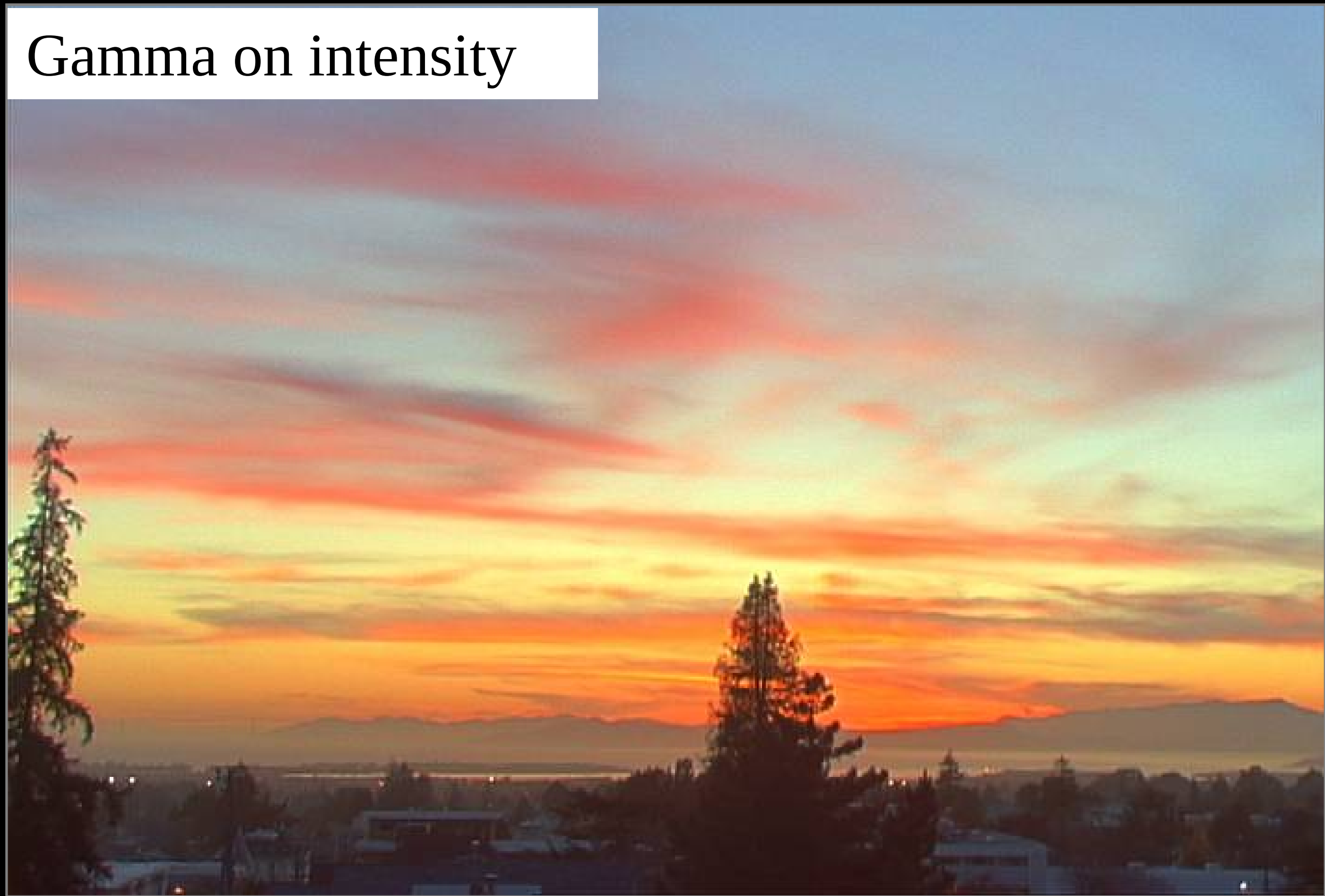
Gamma compression

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

Gamma compression on intensity

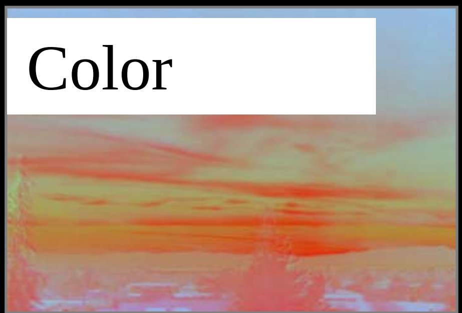
- Colors are OK, but details (intensity high-frequency) are blurred





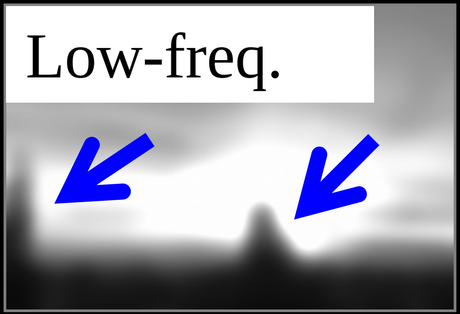
Chiu et al. 1993

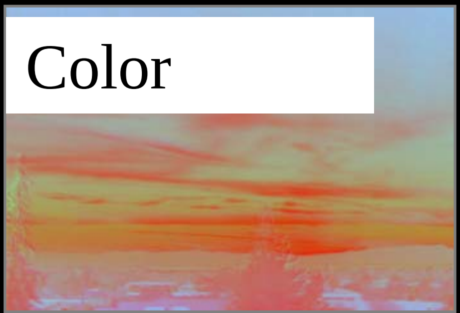
- Reduce contrast of low-frequencies
- Keep high frequencies

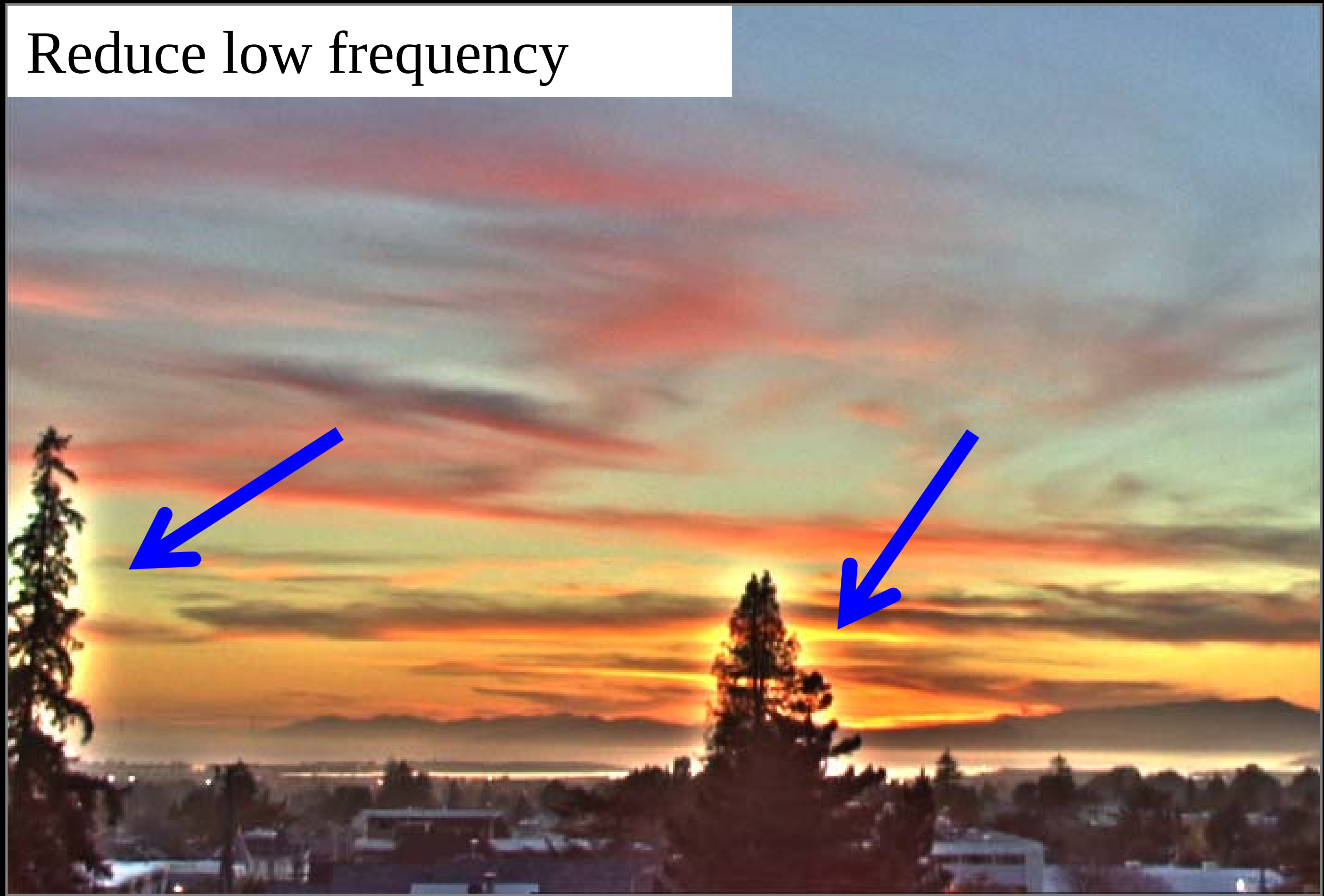


The halo nightmare

- For strong edges
- Because they contain high frequency

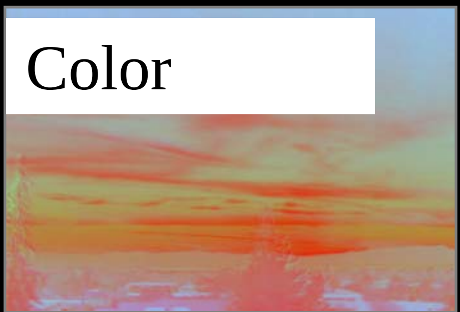


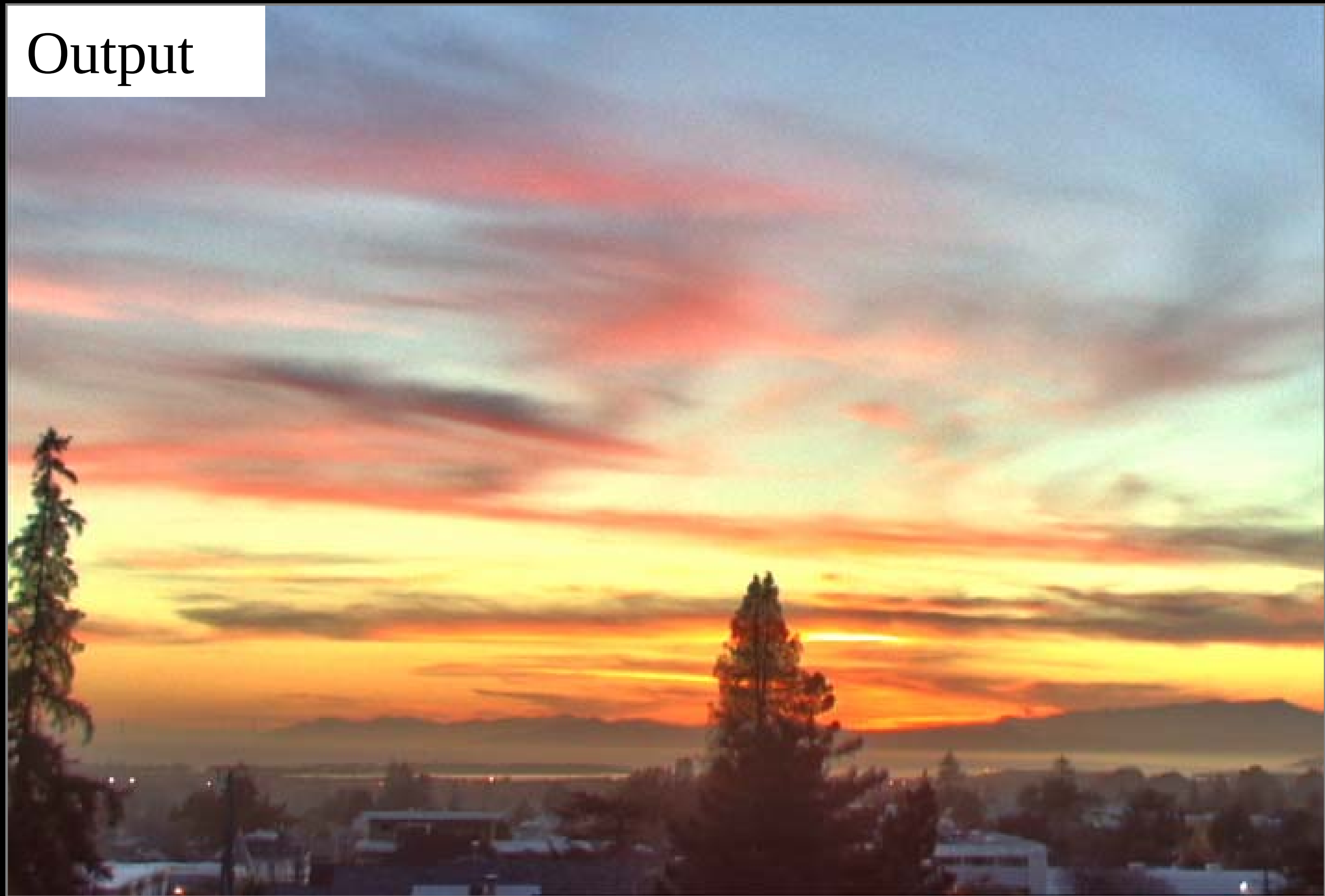




Durand and Dorsey

- Do not blur across edges
- Non-linear filtering





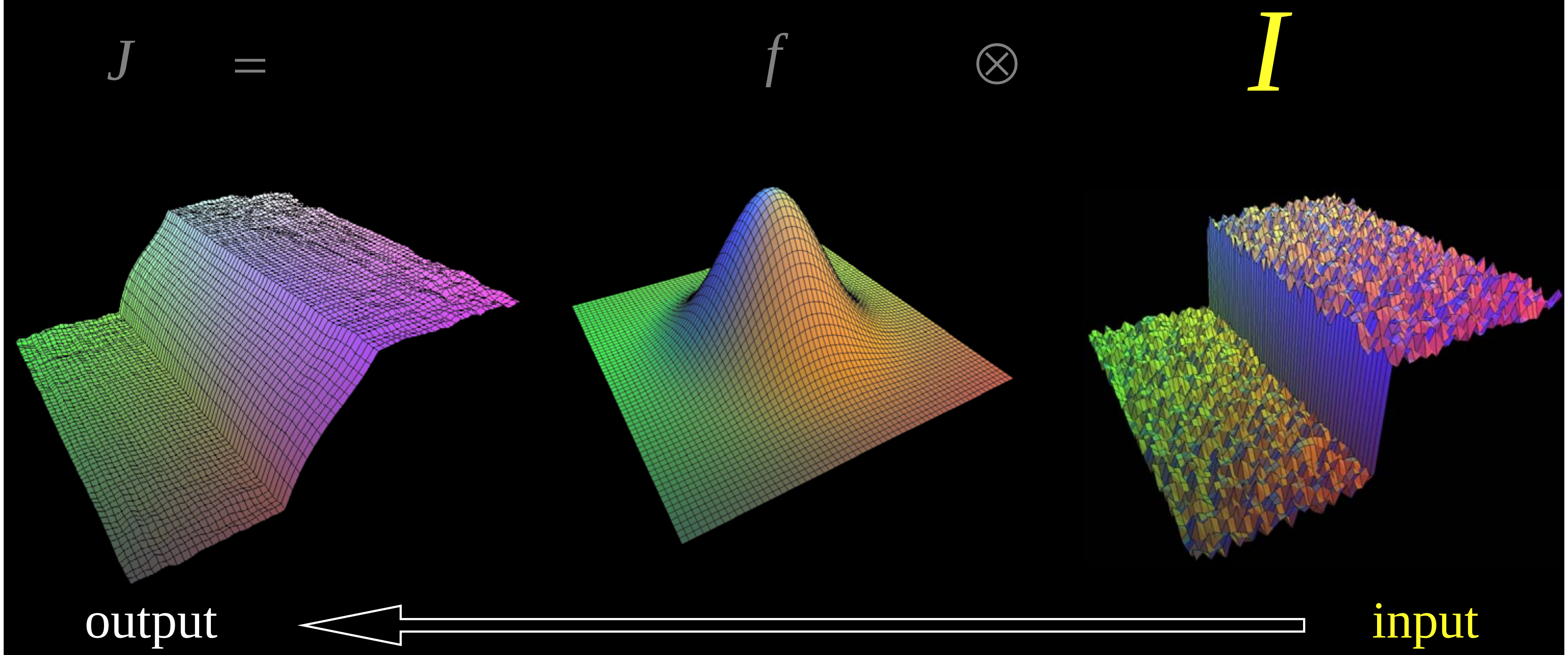
Edge-preserving filtering

- Blur, but not across edges

- Anisotropic diffusion [Perona & Malik 90]
  - Blurring as heat flow
  - LCIS [Tumblin & Turk]
- Bilateral filtering [Tomasi & Manduci, 98]

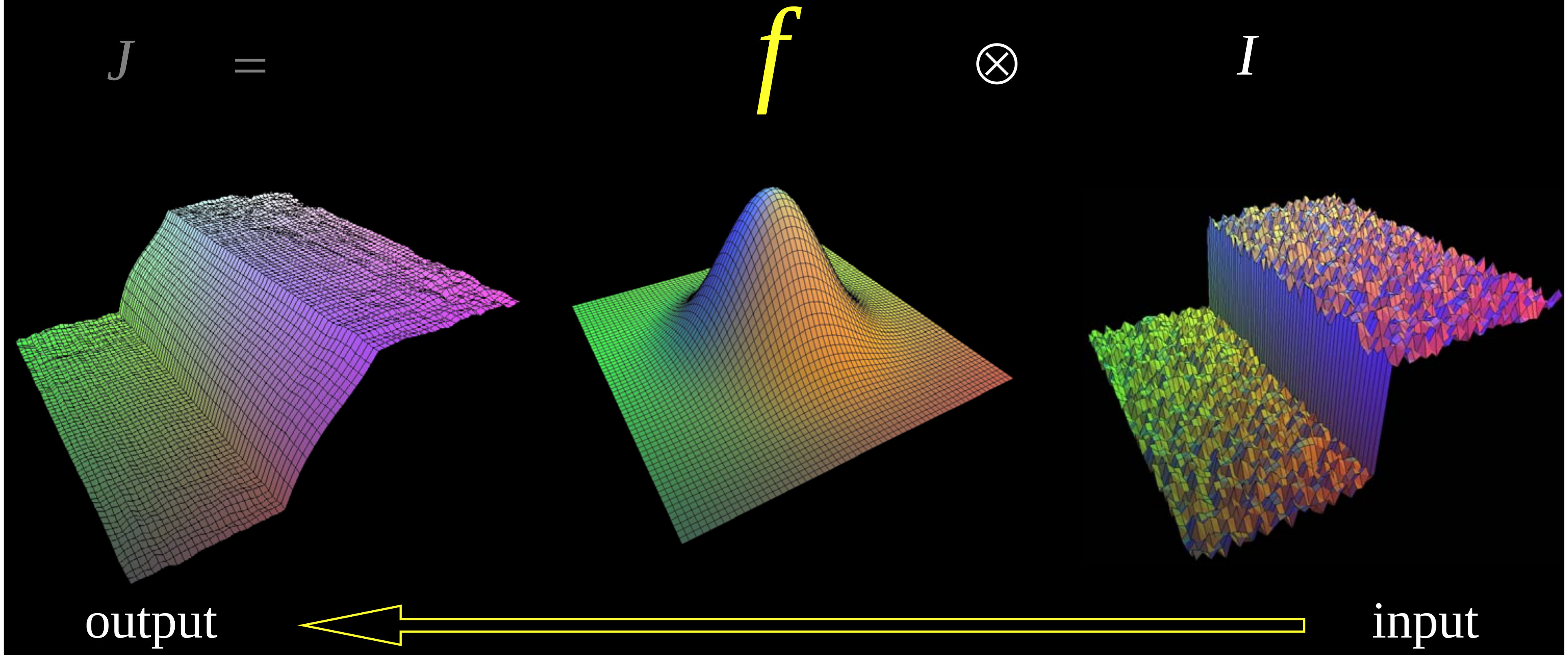
Start with Gaussian filtering

- Here, input is a step function + noise



Start with Gaussian filtering

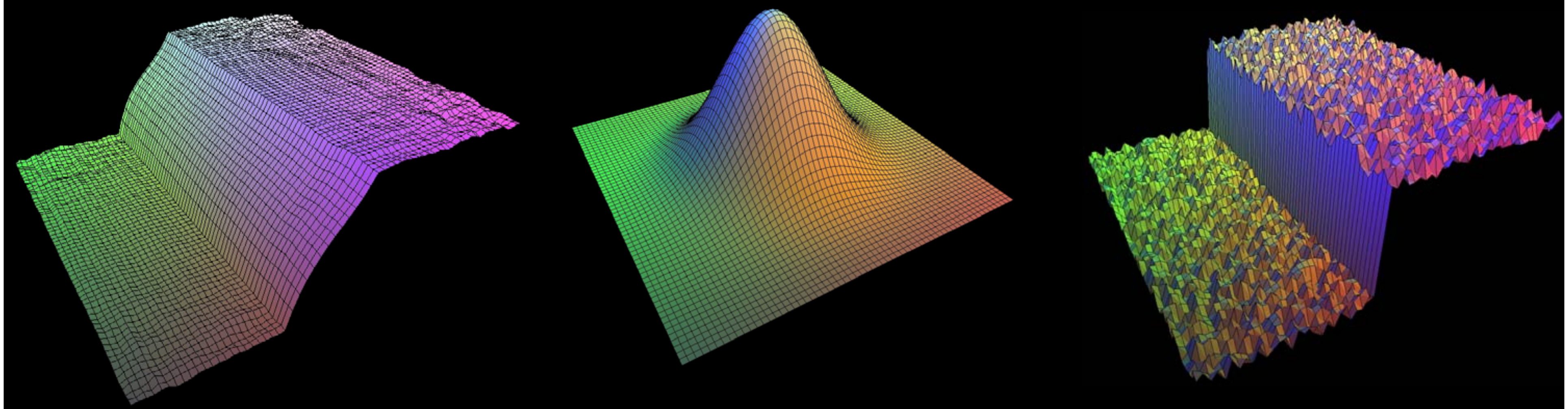
- Spatial Gaussian f



Start with Gaussian filtering

- Output is blurred

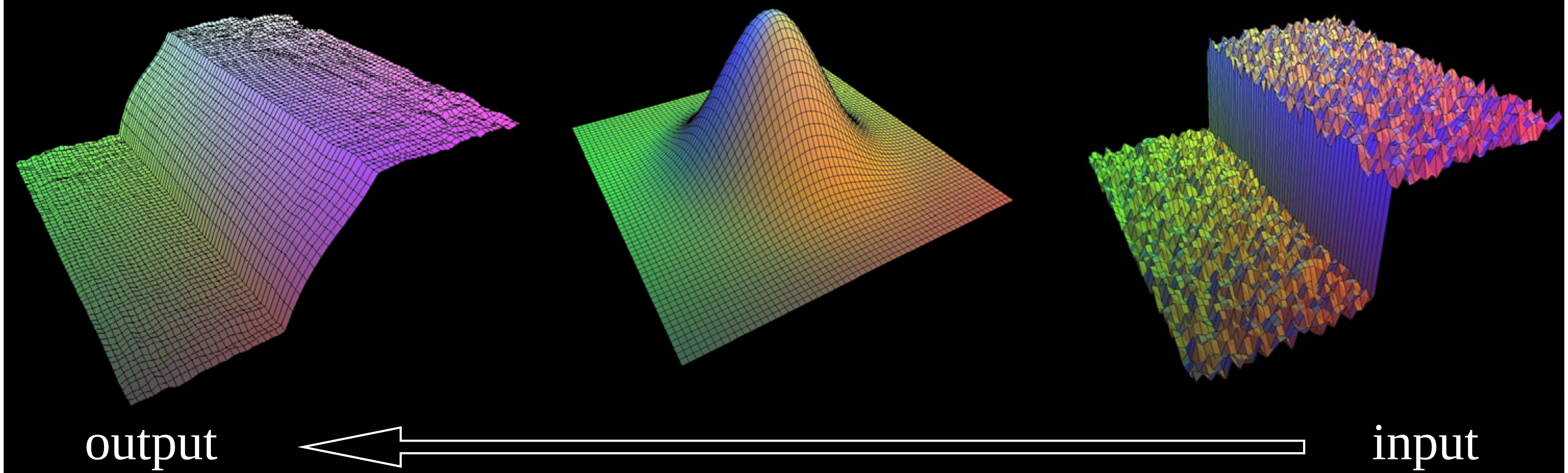
$$J = f \otimes I$$



output ⟵ input

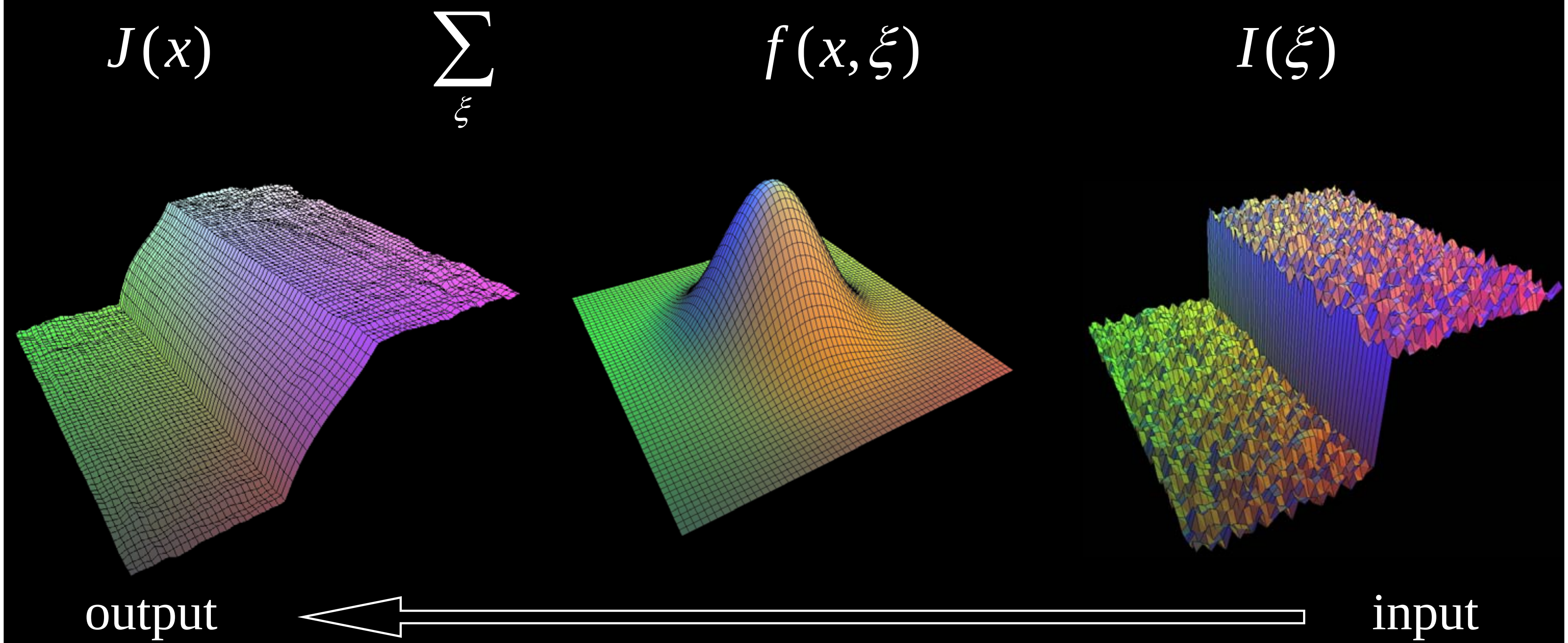
Gaussian filter as weighted average

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



The problem of edges

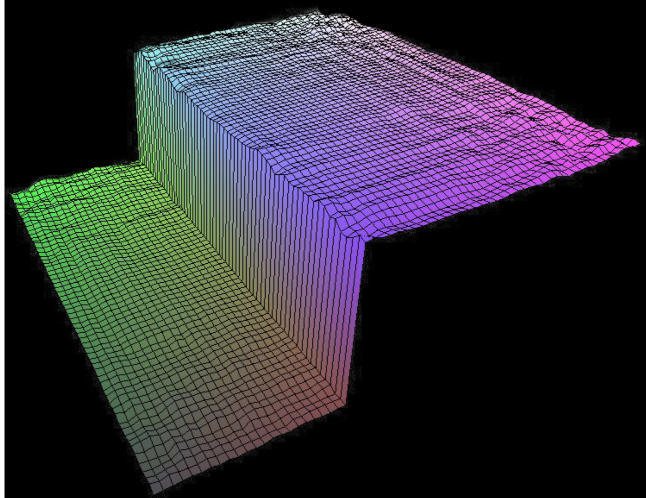
- Here, $I(\xi)$ "pollutes" our estimate J(x)
- It is too different

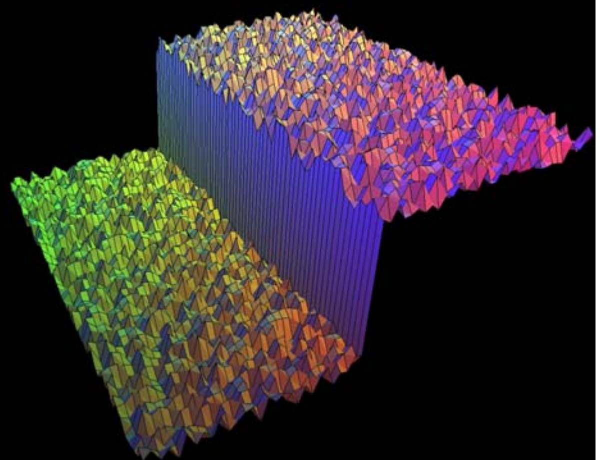


Principle of Bilateral filtering

- [Tomasi and Manduchi 1998]
- Penalty g on the intensity difference

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



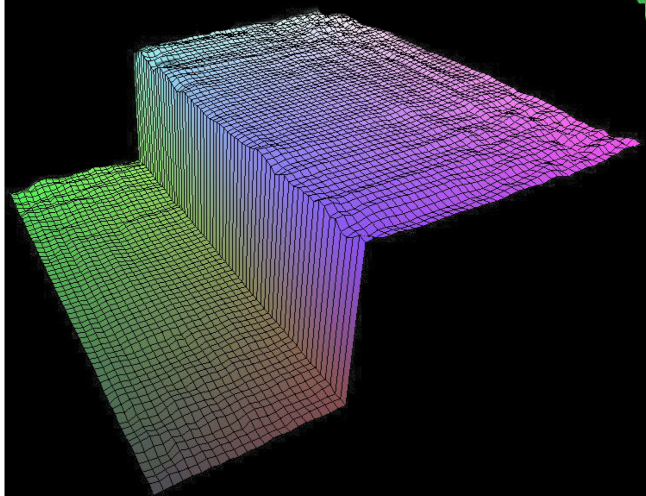


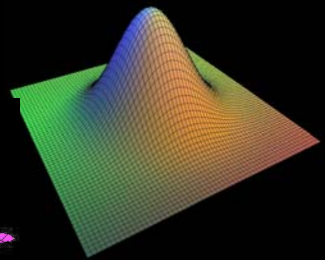
output ⟵ input

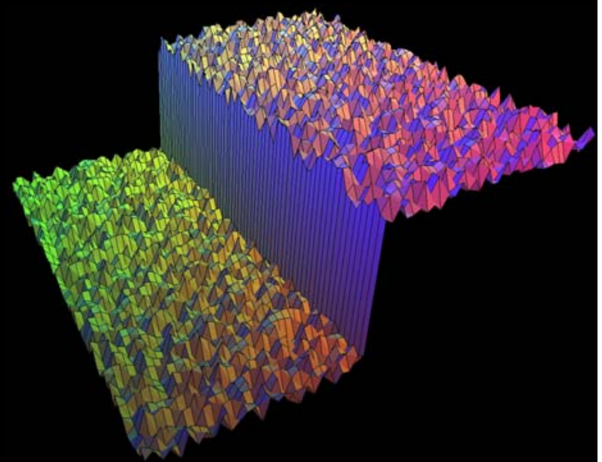
Bilateral filtering

- [Tomasi and Manduchi 1998]
- Spatial Gaussian f

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





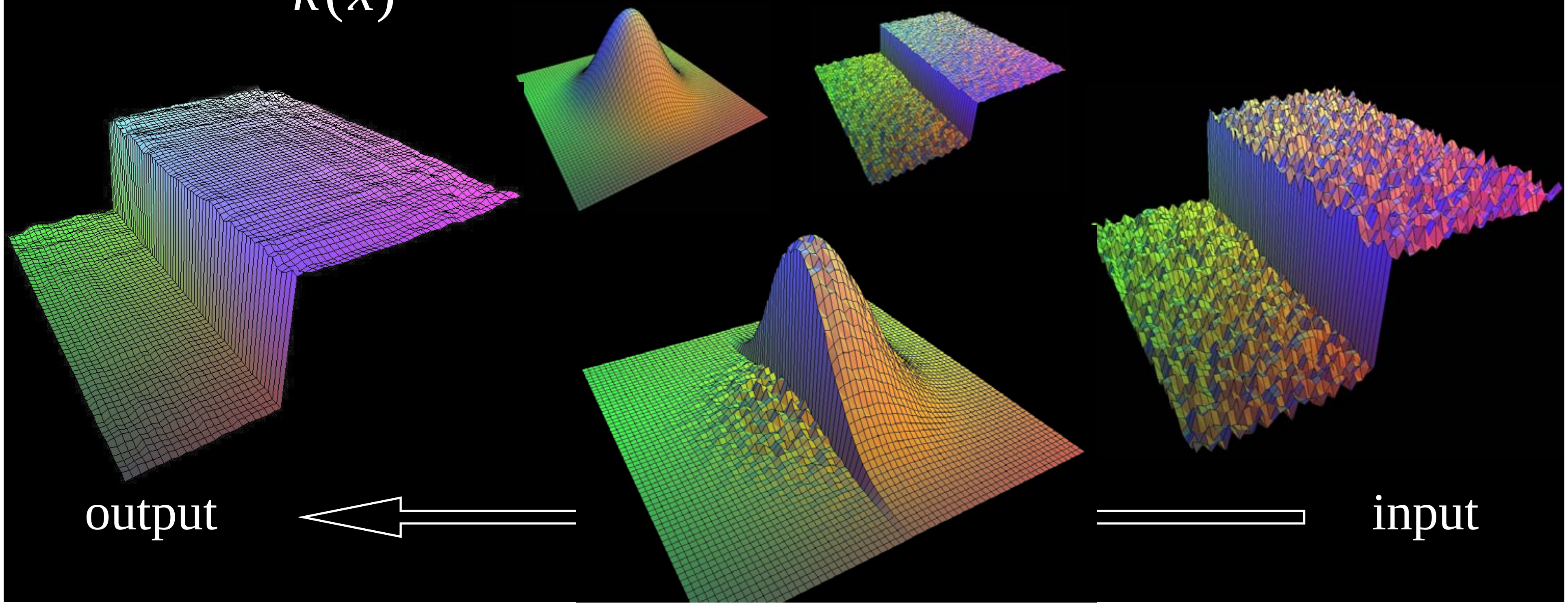


output ⟵ input

Bilateral filtering

- [Tomasi and Manduchi 1998]
- Spatial Gaussian f
- Gaussian g on the intensity difference

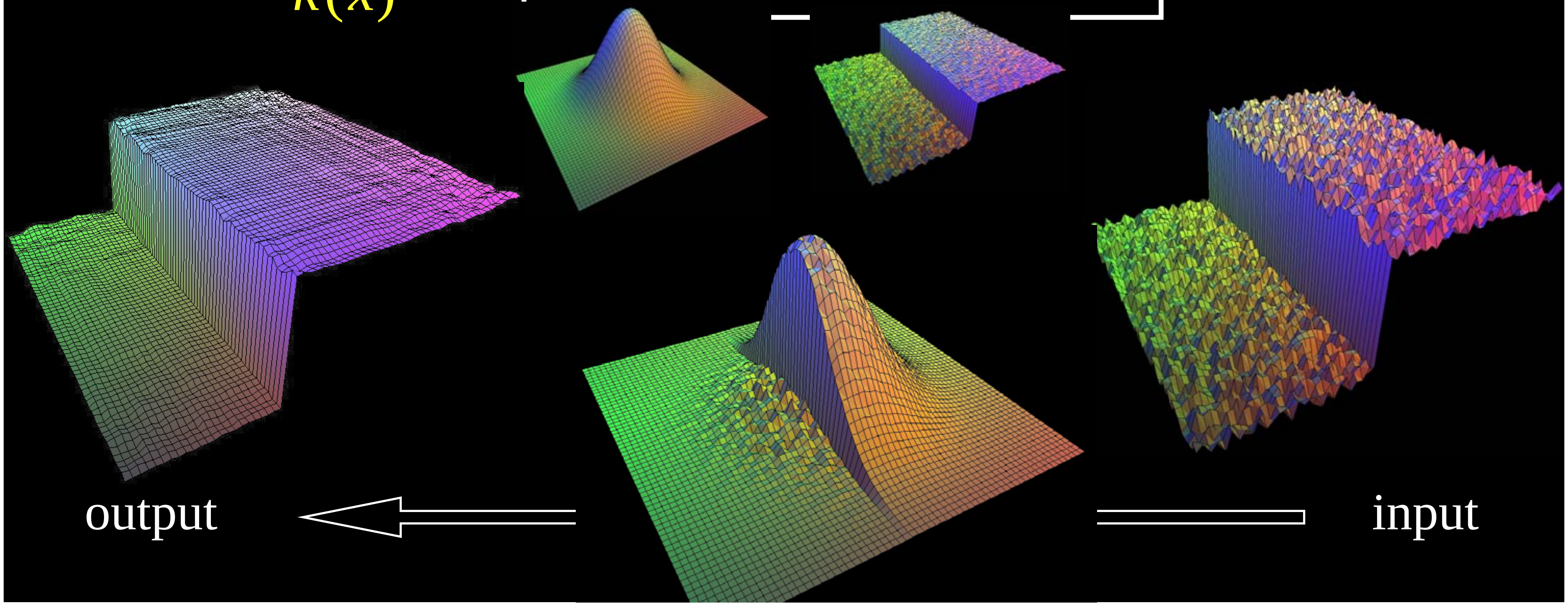
$$J(x) \quad \frac{1}{k(x)} \quad f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



Normalization factor

- [Tomasi and Manduchi 1998]
- k(x)= $\sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x))$

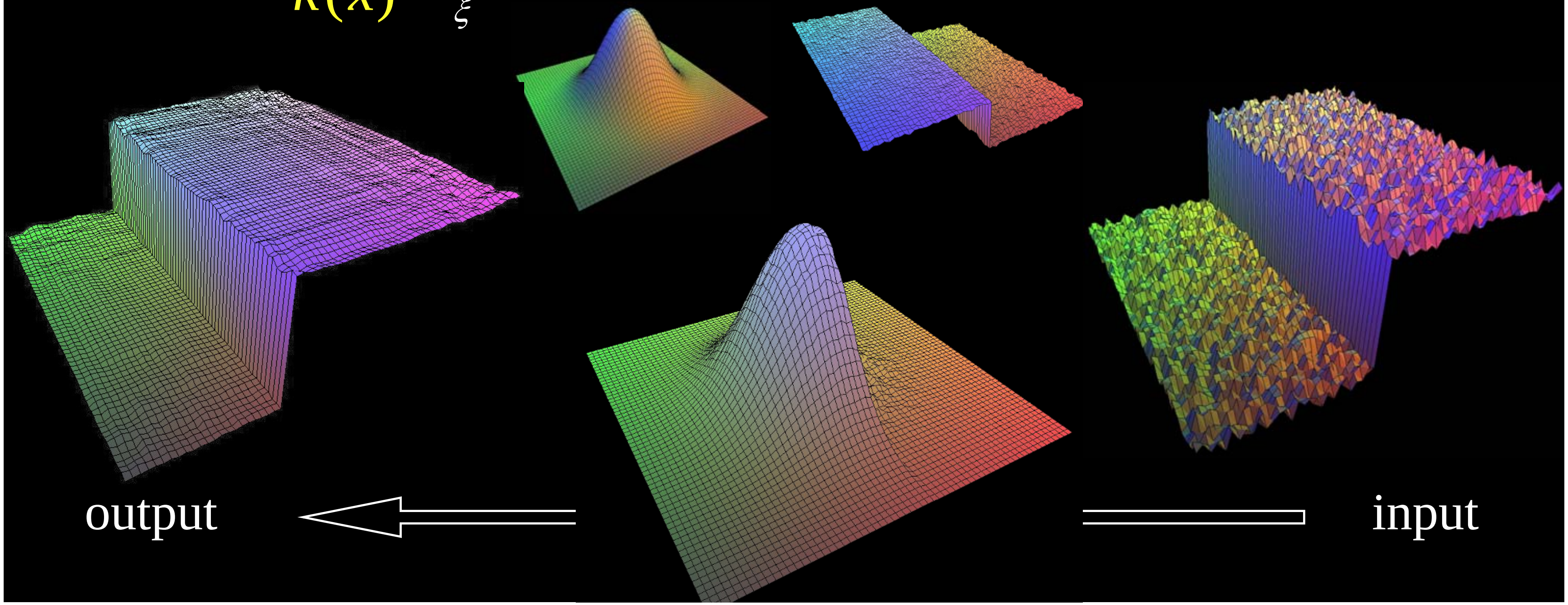
$$J(x) \quad \frac{1}{k(x)} \quad f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



Bilateral filtering is non-linear

- [Tomasi and Manduchi 1998]
- The weights are different for each output pixel

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



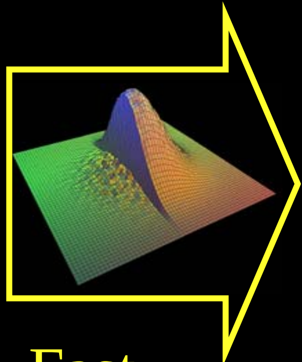
Contrast reduction

Input HDR image

Contrast
too high!

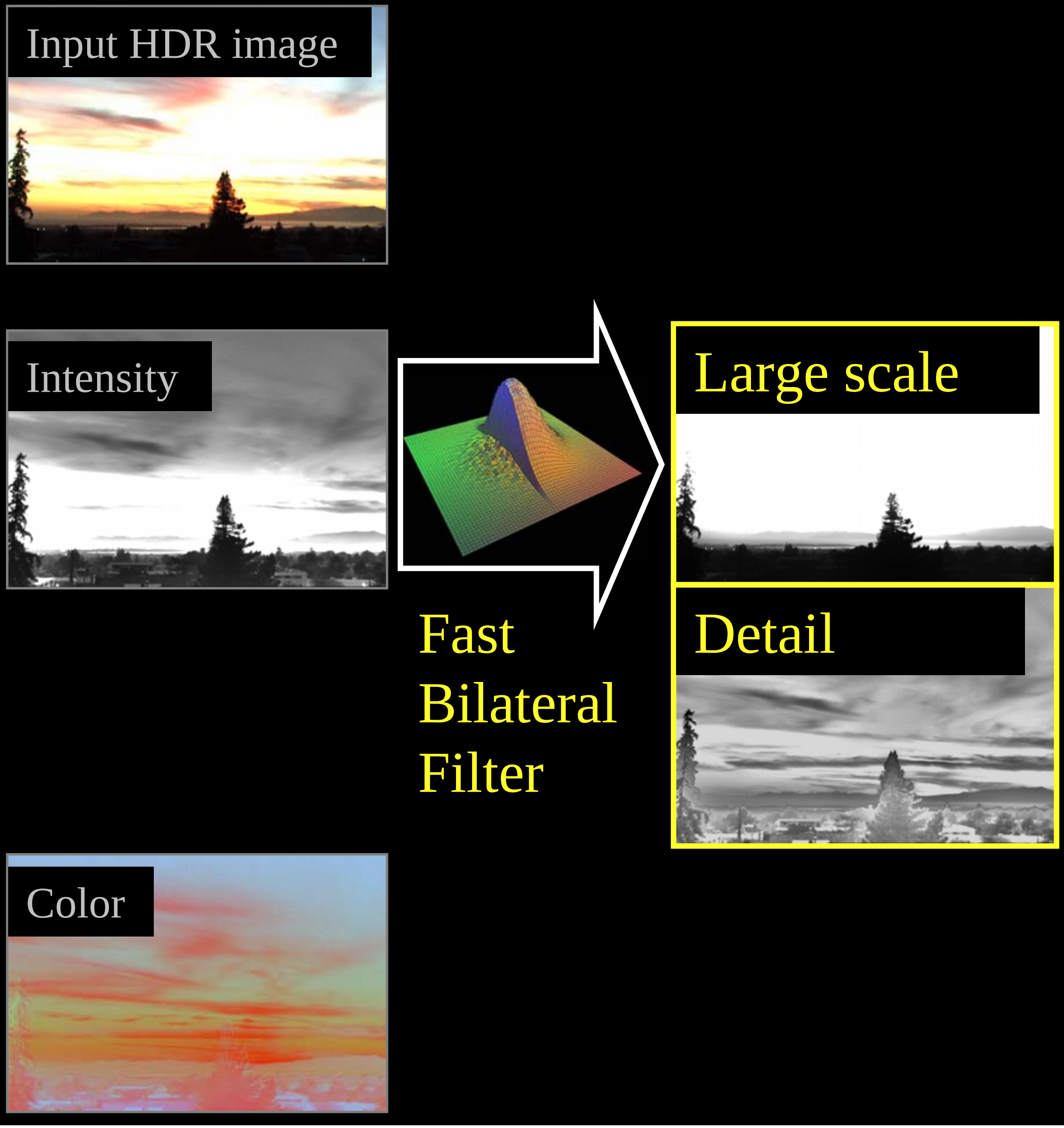
Contrast reduction

Contrast reduction

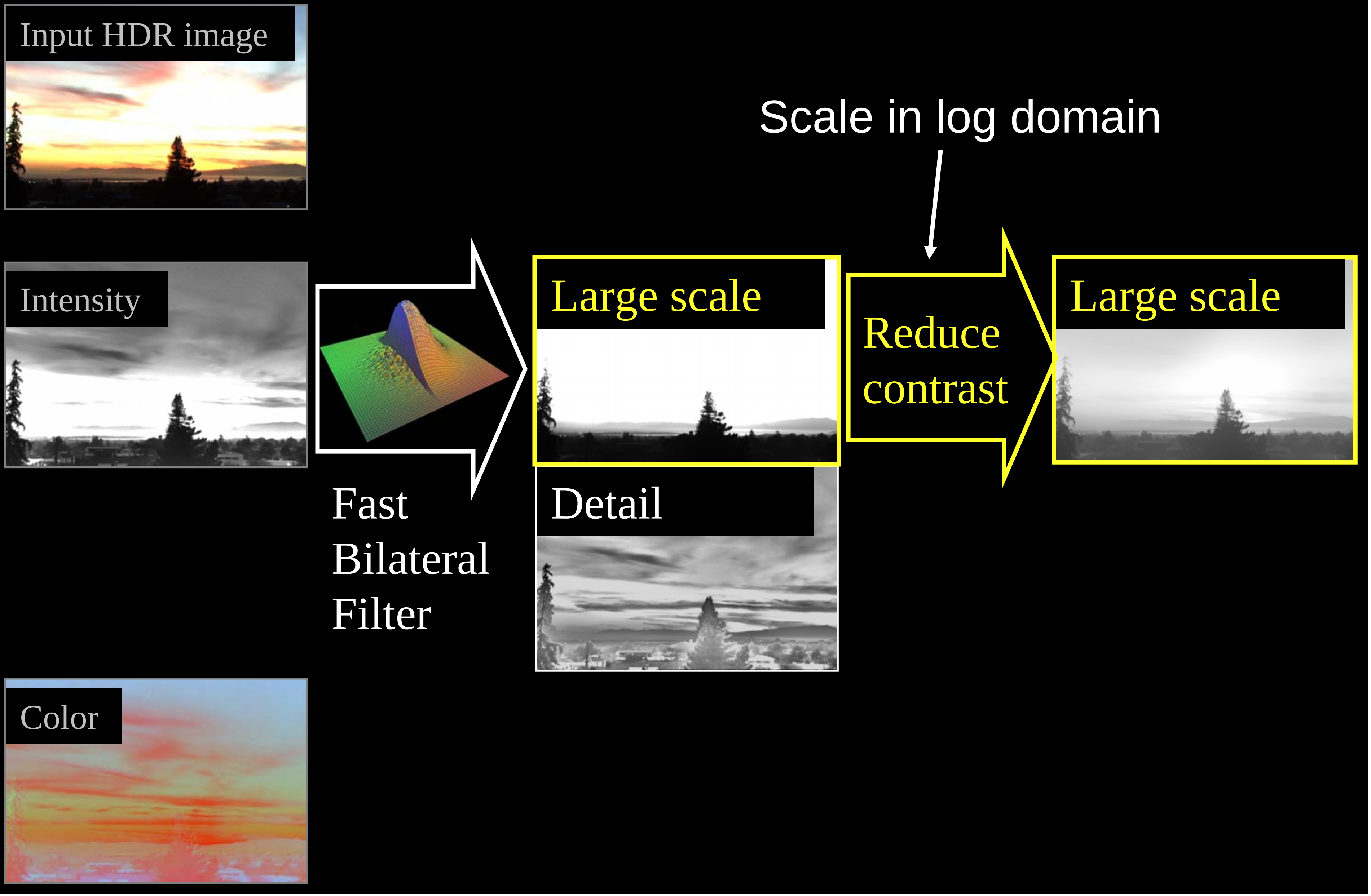


Fast
Bilateral
Filter

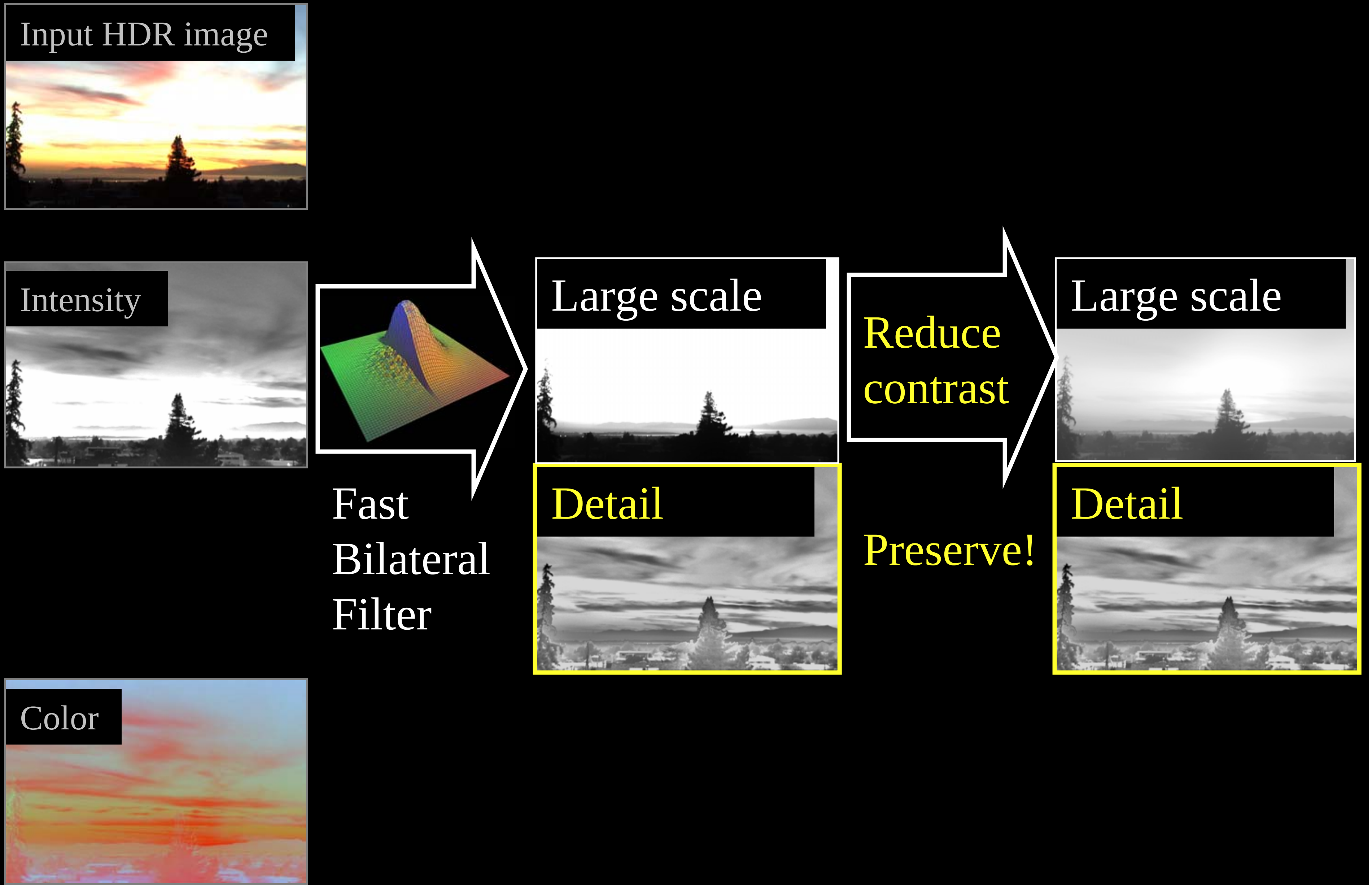
Contrast reduction



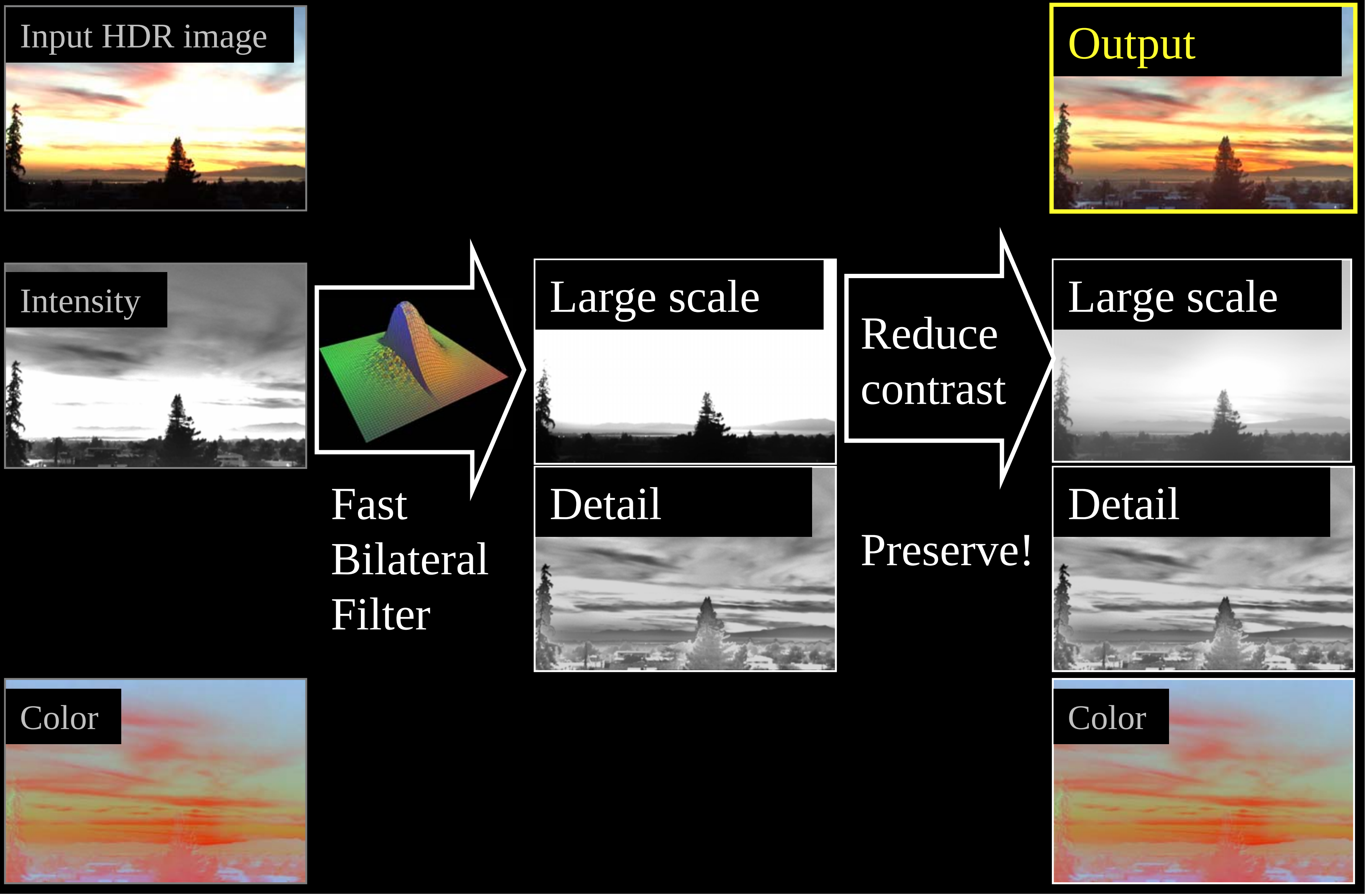
Contrast reduction

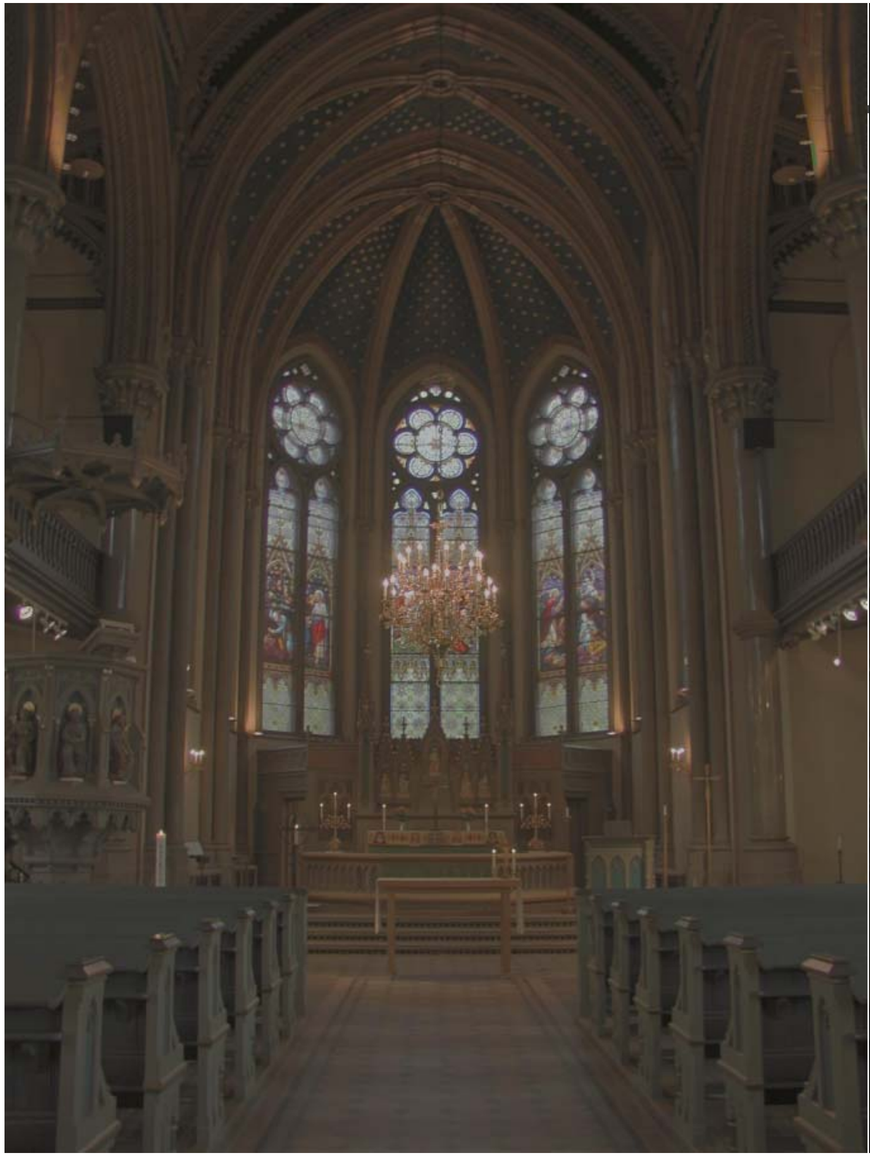


Contrast reduction

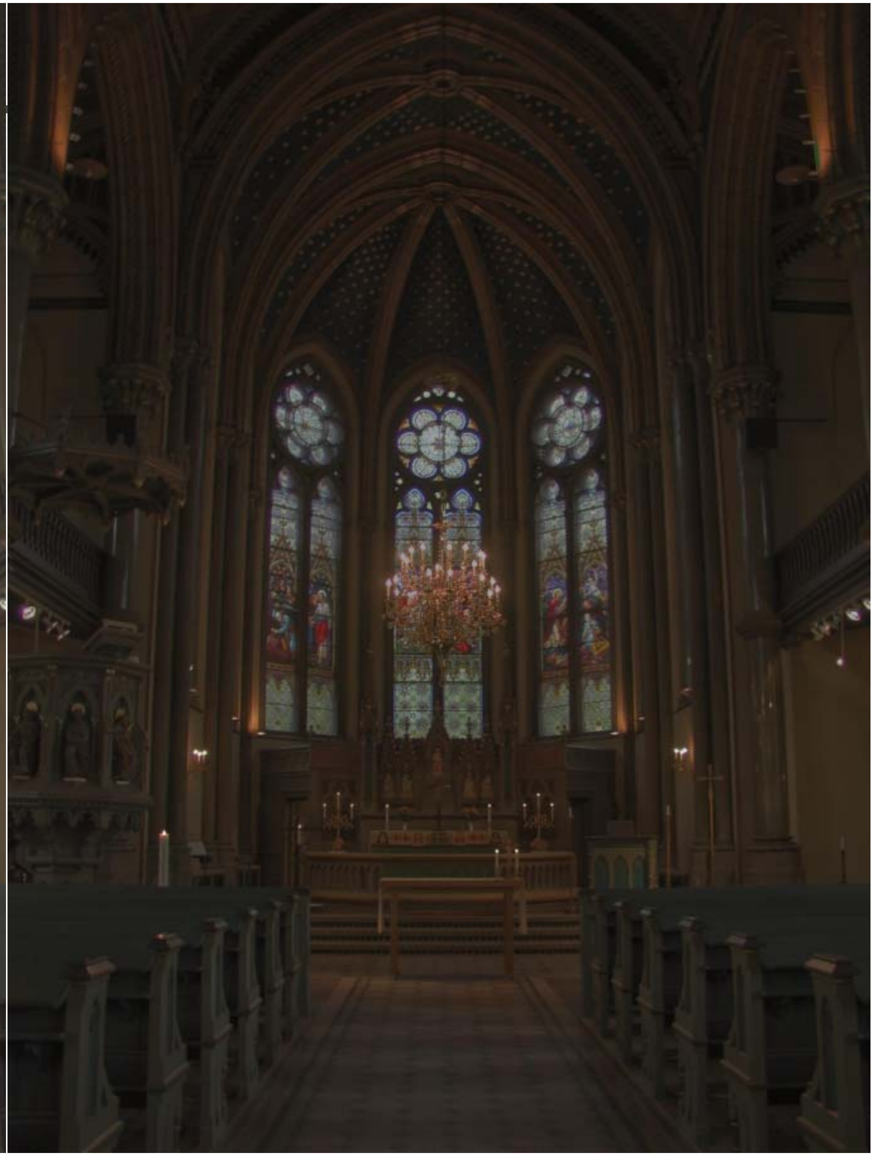


Contrast reduction





Oppenheim



bilateral