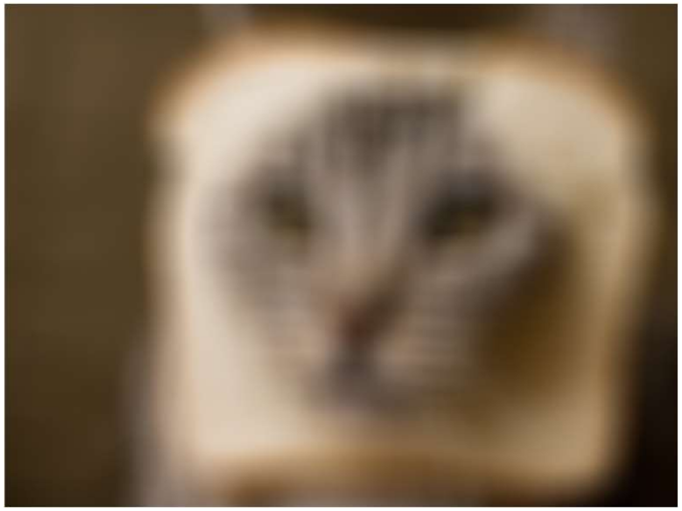
Additional Material

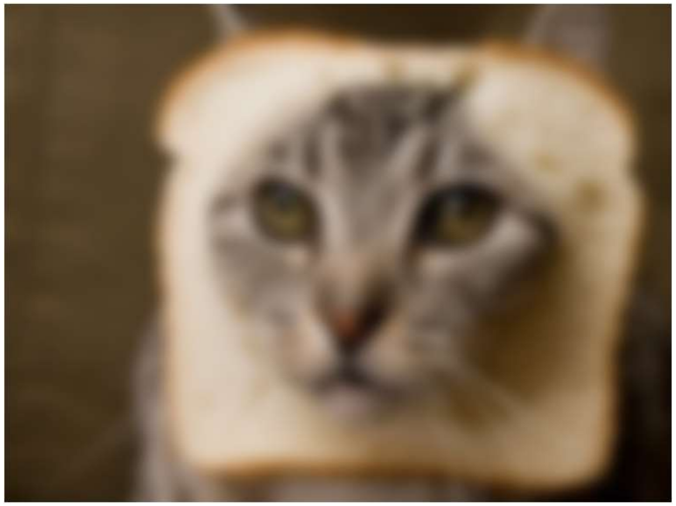
Bilateral Filter

Smoothing an image

- Box filter leads to slightly "blocky" appearance



Box Filter



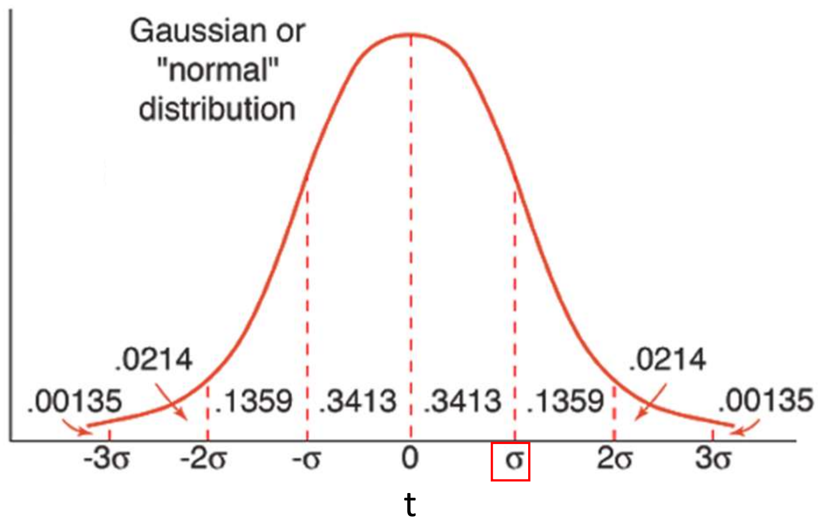
Gaussian Filter

- Reason can be explained by analyzing the image frequencies (out of scope)
- Gaussian filter is better at removing high-frequency content

Gaussian Filter

- Gaussian Kernel

$$G(t,\boxed{\sigma}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}}$$



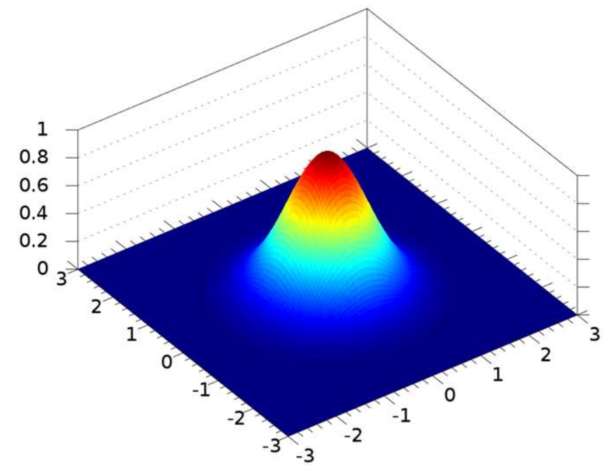
- The larger σ, the broader the function
- The smaller σ, the slimmer the function

Gaussian Filter

- 2D Gaussian Kernel

$$G((i,j),\sigma) = \boxed{\frac{1}{\sqrt{2\pi\sigma^2}}} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

Normalization factor
It makes sure that
integrating the whole
function results in a value
of 1.



Gaussian Filter – in practice

- Can you calculate a 3x3 kernel for $G((i,j),0.5)$?

$$G((i,j),\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

- What you will notice is that the sum of these 9 pixels is larger than 1.
- Hence, applying the filter several times,
  the image would become brighter in this case

Gaussian Filter – in practice

- A simple fix to this problem is to normalize:
- Divide the kernel by the sum of all elements

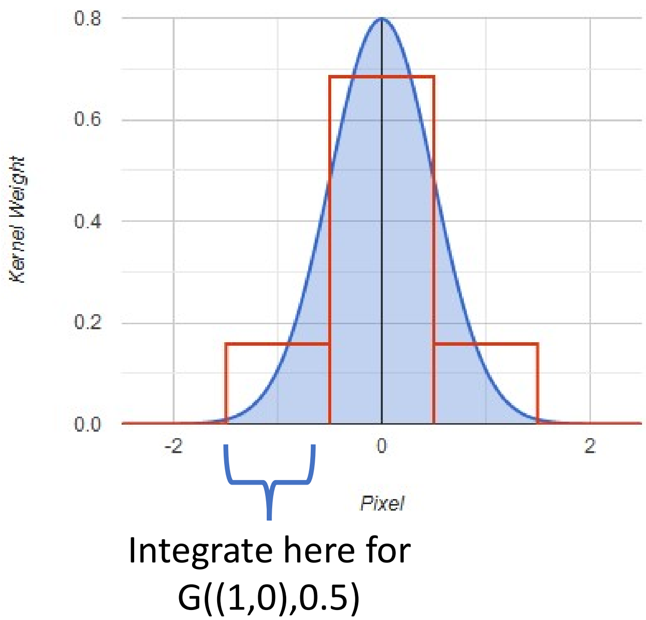
0.0	0.1	0.0
0.1	0.6	0.1
0.0	0.1	0.0

- Btw., if you normalize:

$$G((i,j),\sigma) = \cancel{\frac{1}{\sqrt{2\pi\sigma^2}}} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

Gaussian Filter – in practice

- A more precise solution:
- integrate the function over the pixels



http://dev.theomader.com/gaussian-kernel-calculator/

Gaussian Filter – in practice

How large should the filter be?

- The number of pixels in the filter is linked to sigma

- Rule of thumb: round to odd (6* σ) as a filter size
- e.g., σ = 0.5 should be 3 pixels.

Gaussian Filter – in practice

- Gaussian filter (and Box filter as well) is separable
- Separable means that 2 convolutions with a 1D kernel can be performed instead of 1 convolution with a 2D kernel:
  Gaussian along the X axis Gx, followed by a Gaussian along Y axis Gy

$$((I * Gx) * Gy) = \big(I * (Gx * Gy)\big) = I * G$$

- This is efficient because of linear cost instead of quadratic cost

Example of separability for the Box filter:

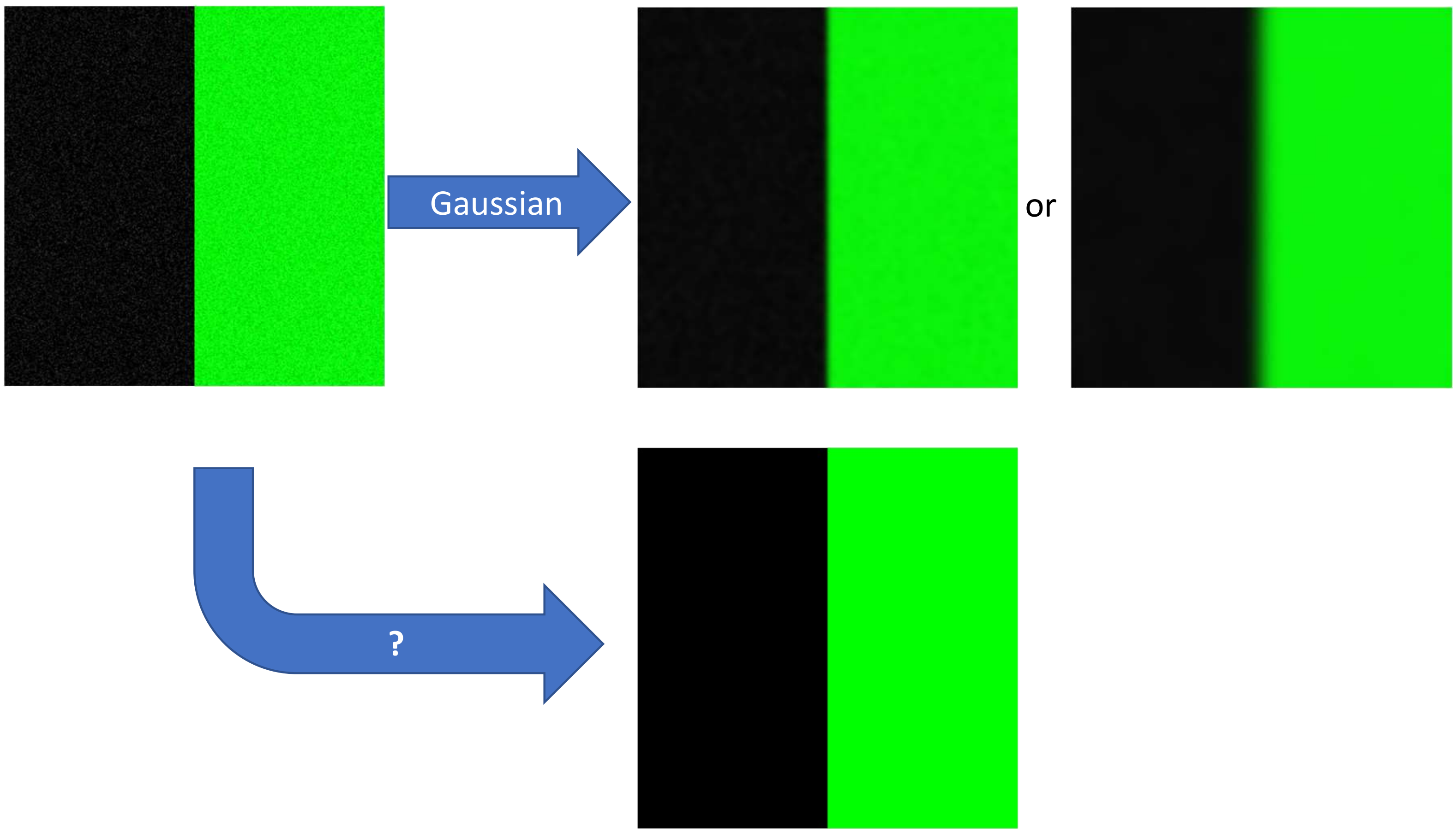
$$\overbrace{\begin{bmatrix}1/3\\1/3\\1/3\end{bmatrix} * \begin{bmatrix}1/3 & 1/3 & 1/3\end{bmatrix}}^{\text{6 elements to filter with}} = \overbrace{\begin{bmatrix}1/9 & 1/9 & 1/9\\1/9 & 1/9 & 1/9\\1/9 & 1/9 & 1/9\end{bmatrix}}^{\text{9 elements to filter with}}$$

Gaussian Filter

- Rem

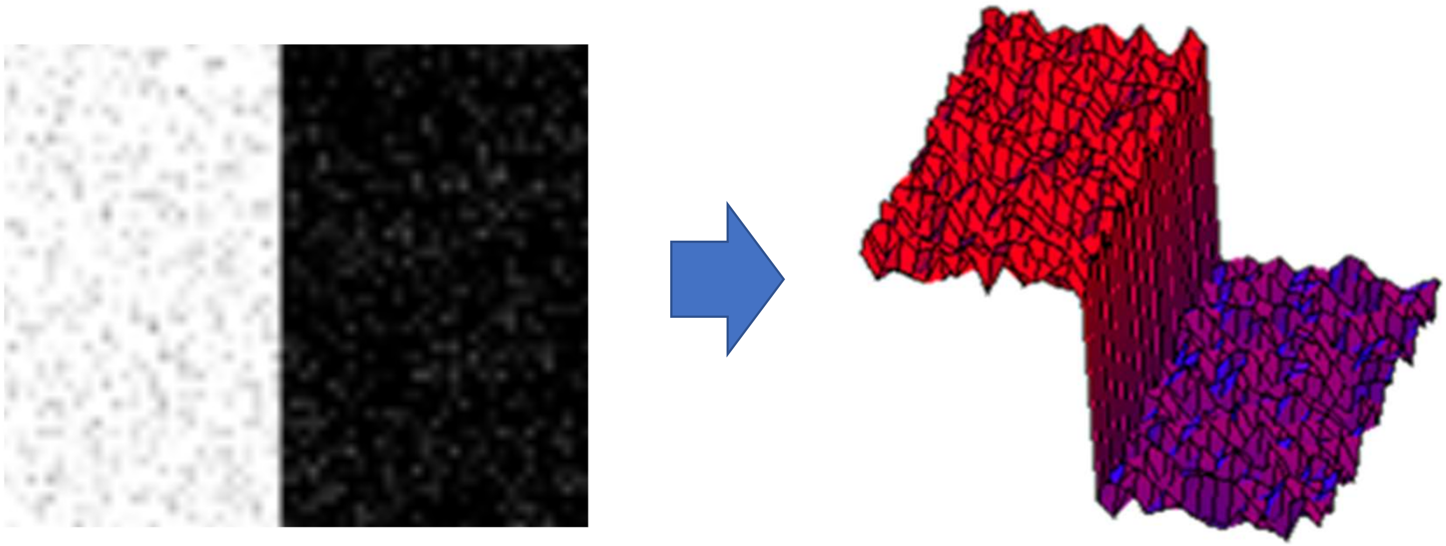
Smarter Filters?

Smarter Filters?



Visualization

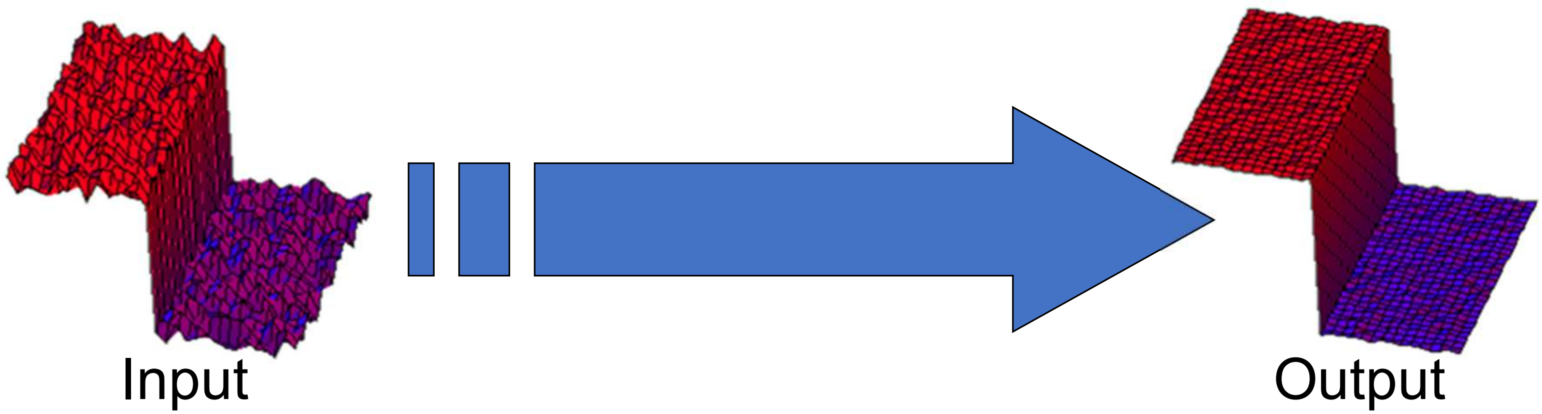
- To ease understanding, we will illustrate images as a “terrain”



Bilateral Filter

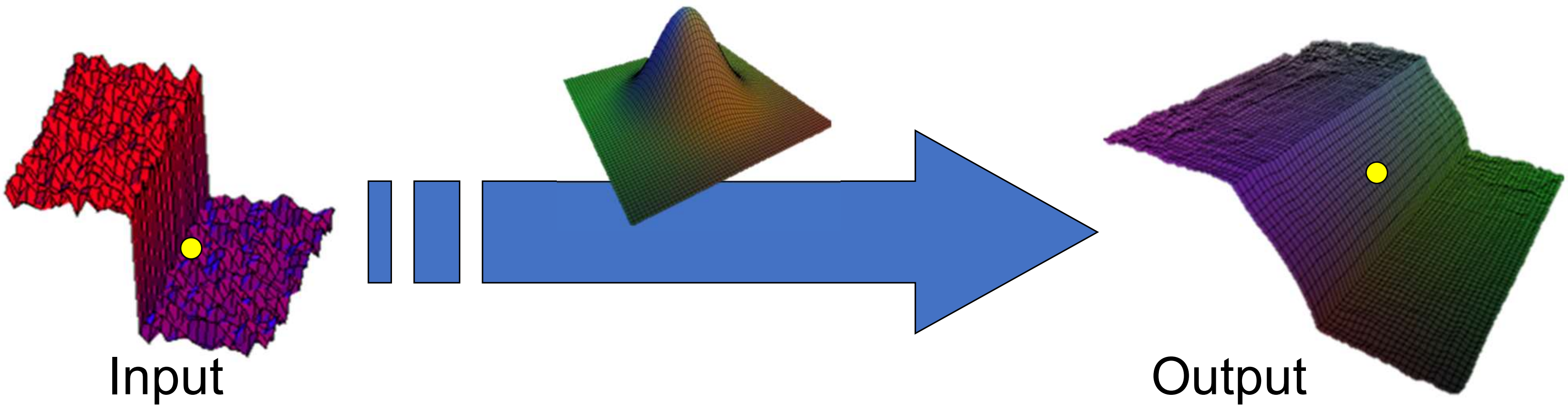
- Bilateral filter – edge preserving filter

Smith and Brady 1997; Tomasi and Manducci 1998; Durand et al. 2002



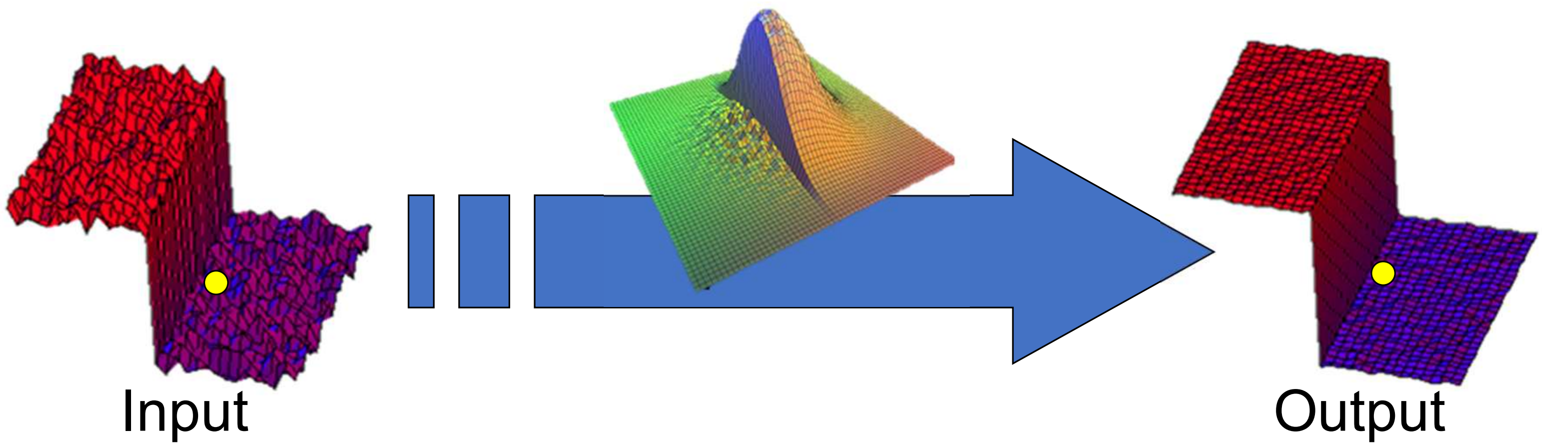
Bilateral Filter: Motivation

- Gaussian



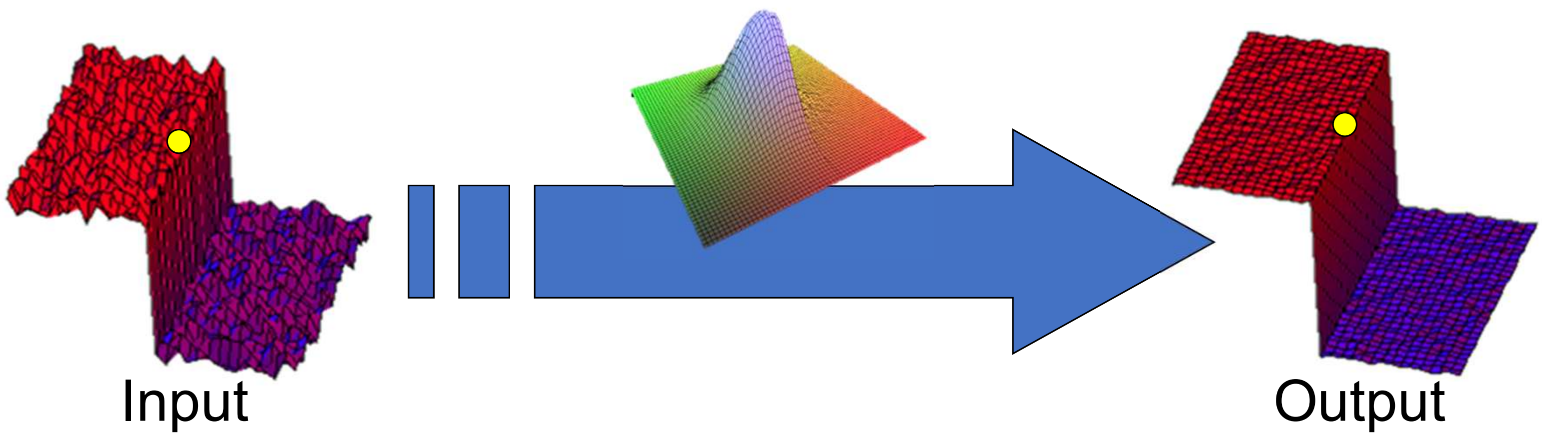
Bilateral Filter: Illustration

- Bilateral filter – edge preserving filter
  depending on the center pixel, the filter looks different

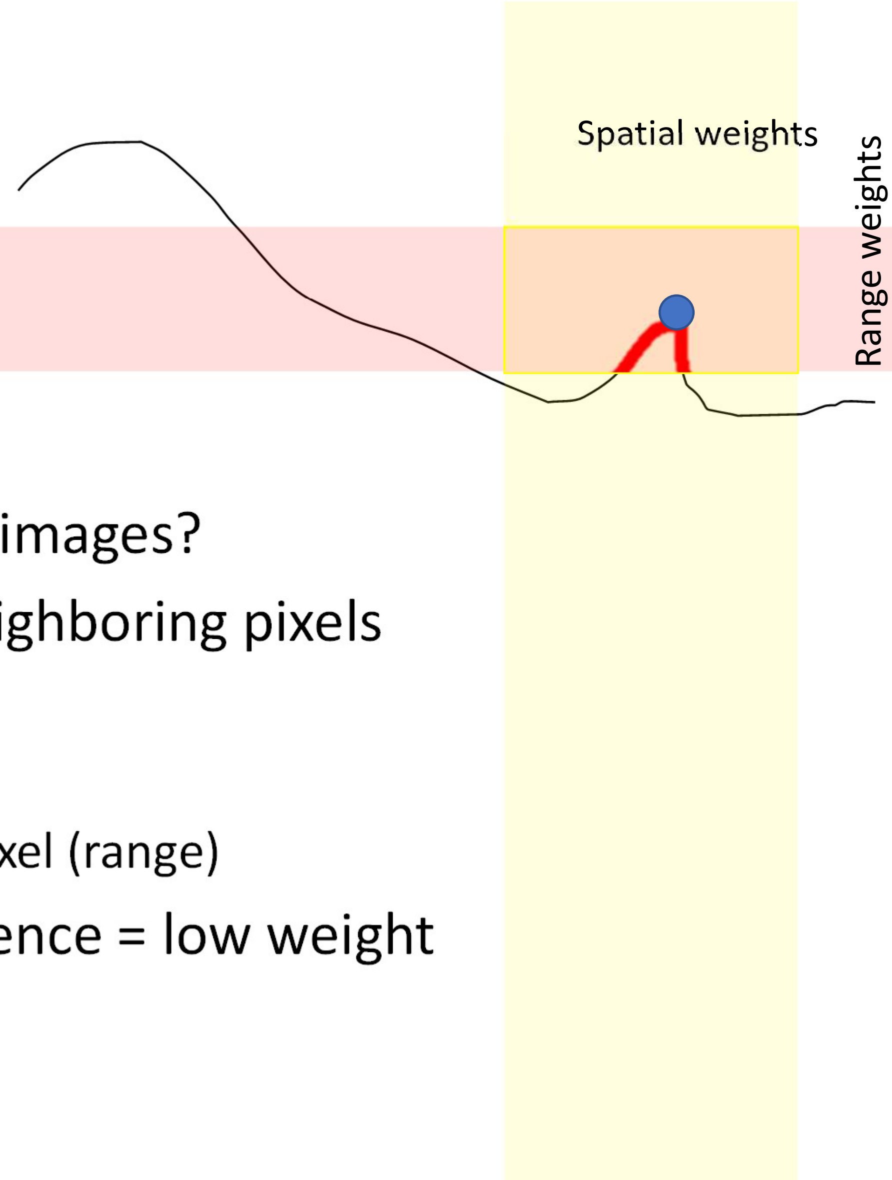


Bilateral Filter: Illustration

- Bilateral filter – edge preserving filter
  depending on the center pixel, the filter looks different

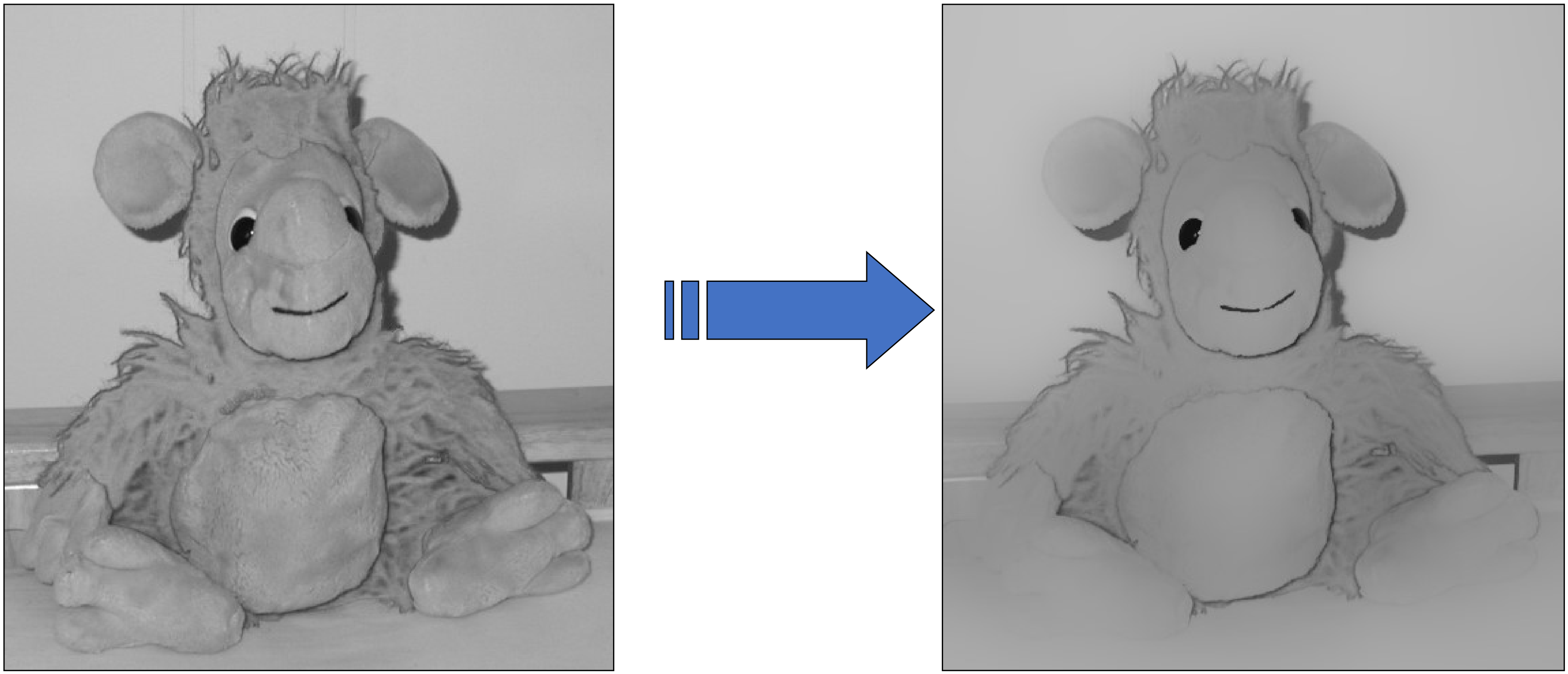


Bilateral Filter: Definition



- How does the filter work on grayscale images?
- Weighted average over intensity of neighboring pixels
- Weights based on
  - Distance from center (spatial)
  - Difference in intensity from the center pixel (range)
- Large distance or large intensity difference = low weight

Bilateral Filter: Example



Bilateral Filter: Formal Definition

- f, g are filters (spatial and range respectively), typically Gaussians
- I is the image to be filtered

$$J(x) = \frac{1}{k(x)} \sum_y \overbrace{f(x-y)}^{\text{Penalizes spatial distance (distance of position y to position x)}} \underbrace{g\big(I(x) - I(y)\big)}_{\text{Penalizes value distance (distance of value at y to value at x)}} I\,(y)$$

$$k(x) := \sum_y f(x-y) g\big(I(x) - I(y)\big)$$

- k is a normalization factor (compare discussion for Gaussian)

Bilateral Filter Application: Tone Mapping

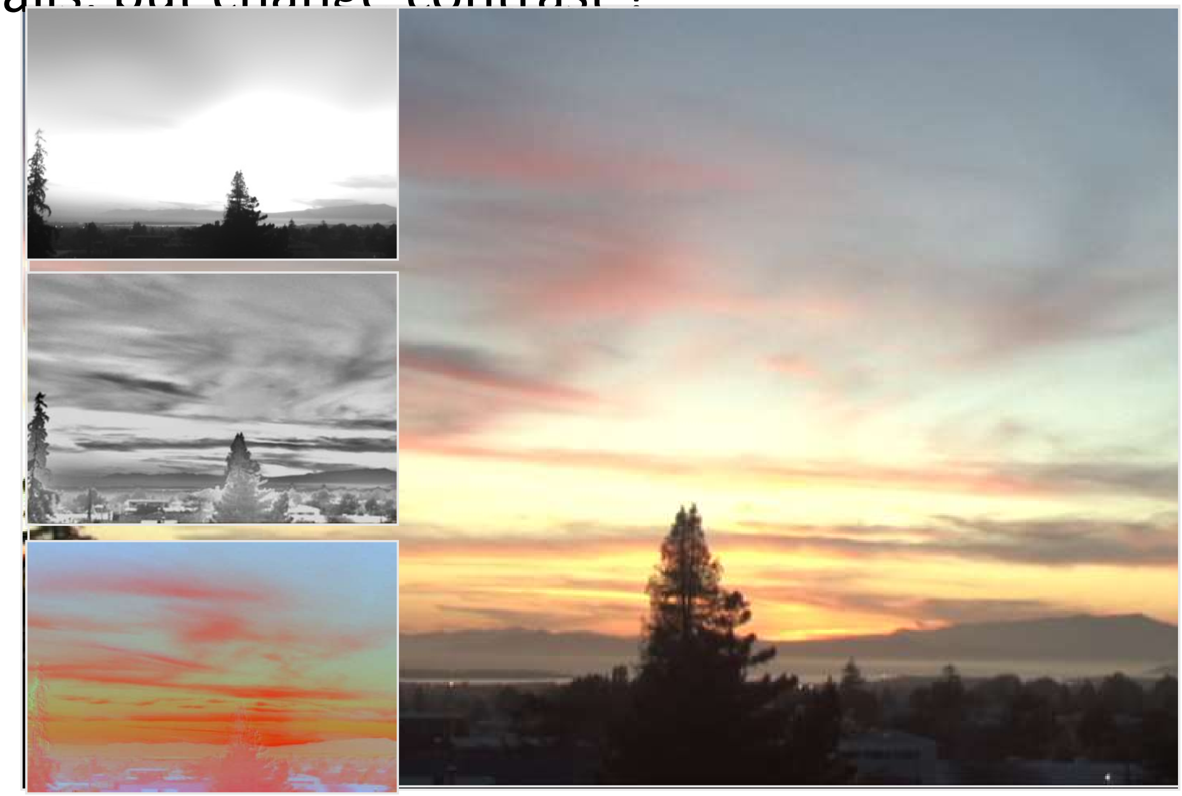
Contrast Reduction!

How to preserve details, but change contrast ?

Large-scale

Detail

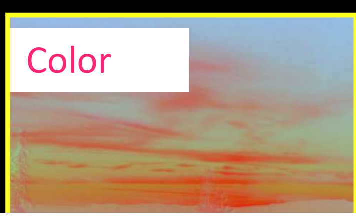
Color



HDR Contrast Reduction [Durand & Dorsey 2002]

Contrast too high!

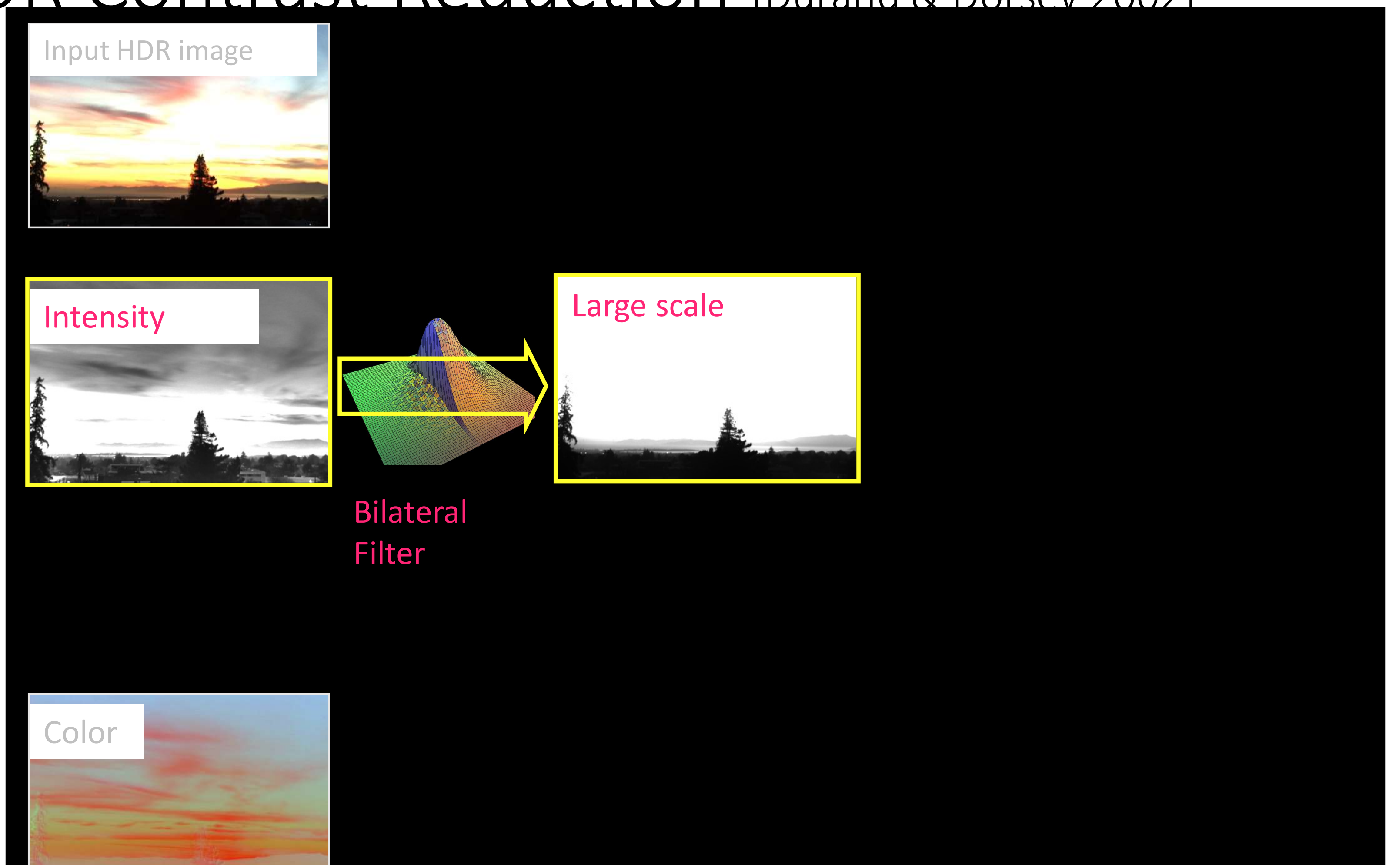
HDR Contrast Reduction [Durand & Dorsey 2002]



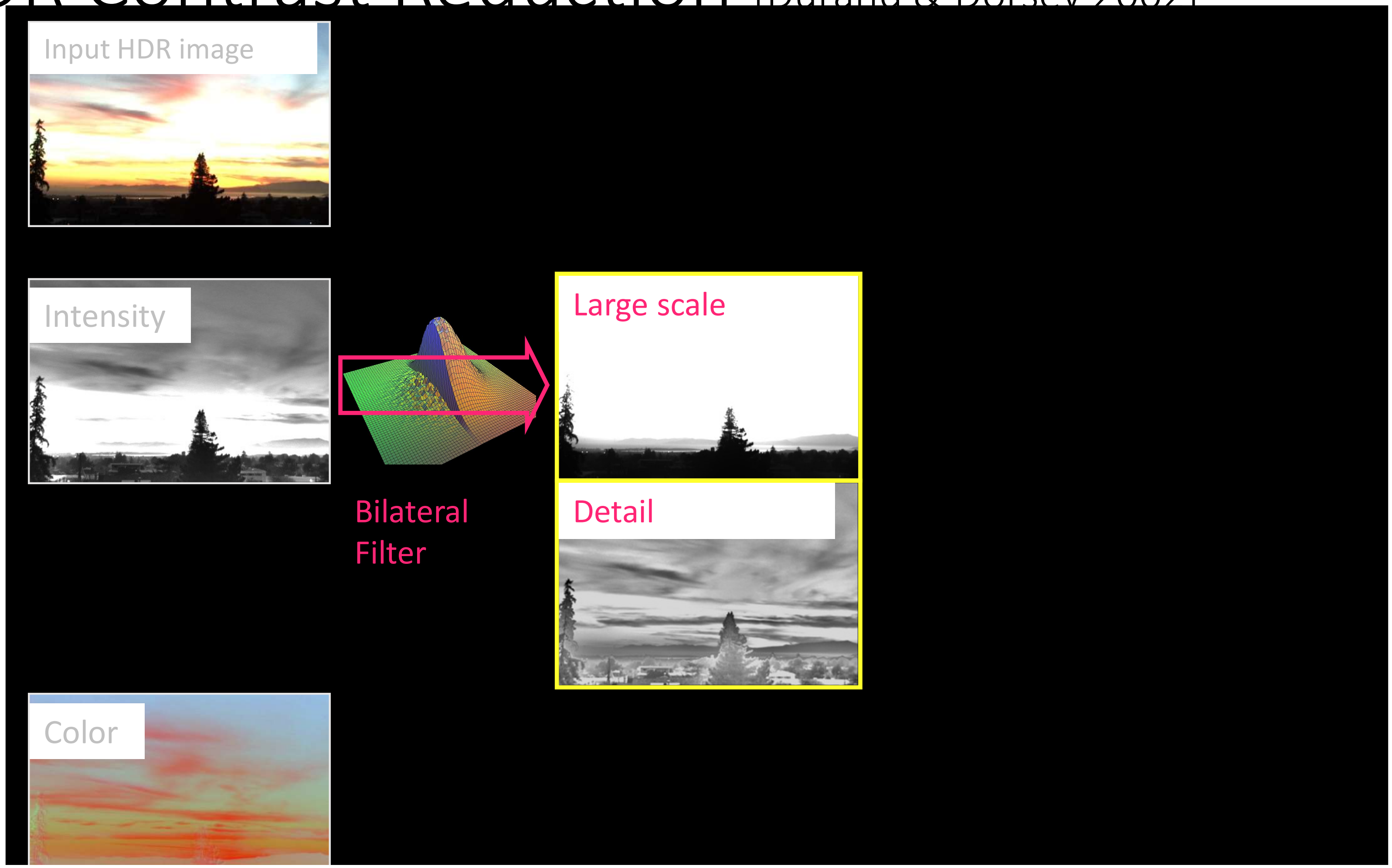
HDR Contrast Reduction: Intensity/Color

- Input Image pixel (R,G,B)
- Intensity=ColorToGrayscale(R,G,B)
- Color= (R,G,B)/Intensity

HDR Contrast Reduction [Durand & Dorsey 2002]



HDR Contrast Reduction [Durand & Dorsey 2002]



HDR Contrast Reduction: Detail

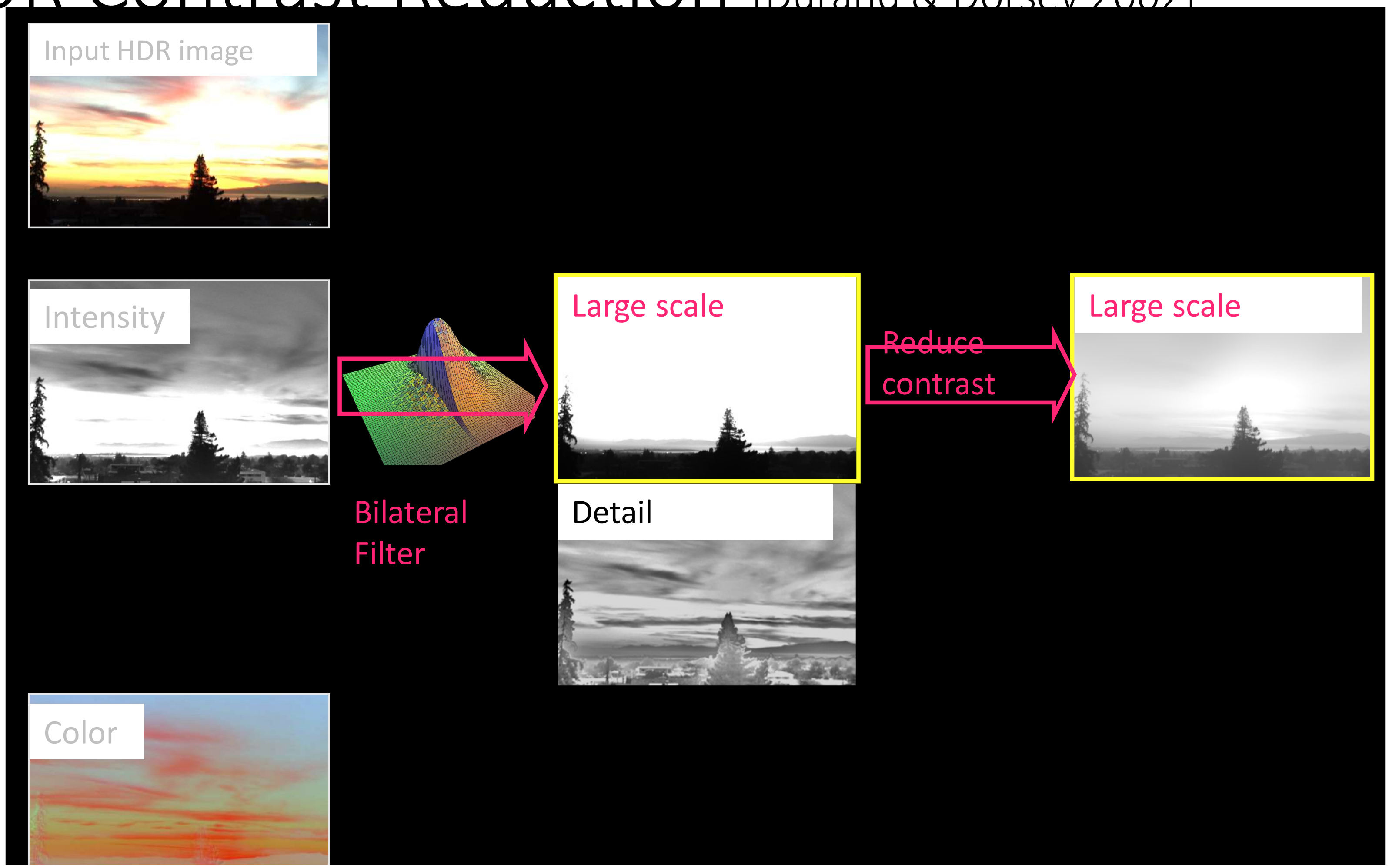
/

=

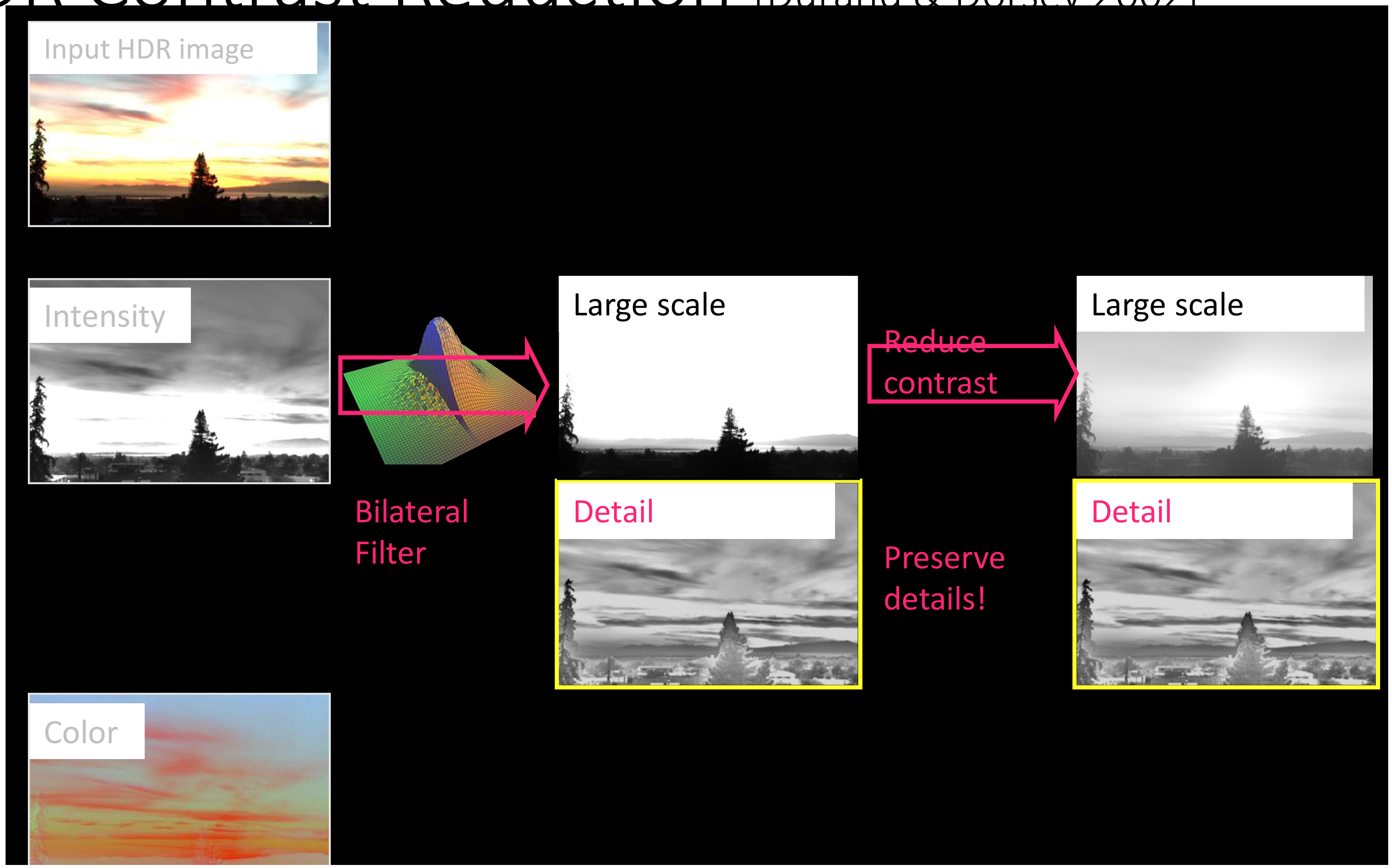
Intensity | Large-scale | Detail

Operation can be inverted:
Large scale * Detail = Intensity

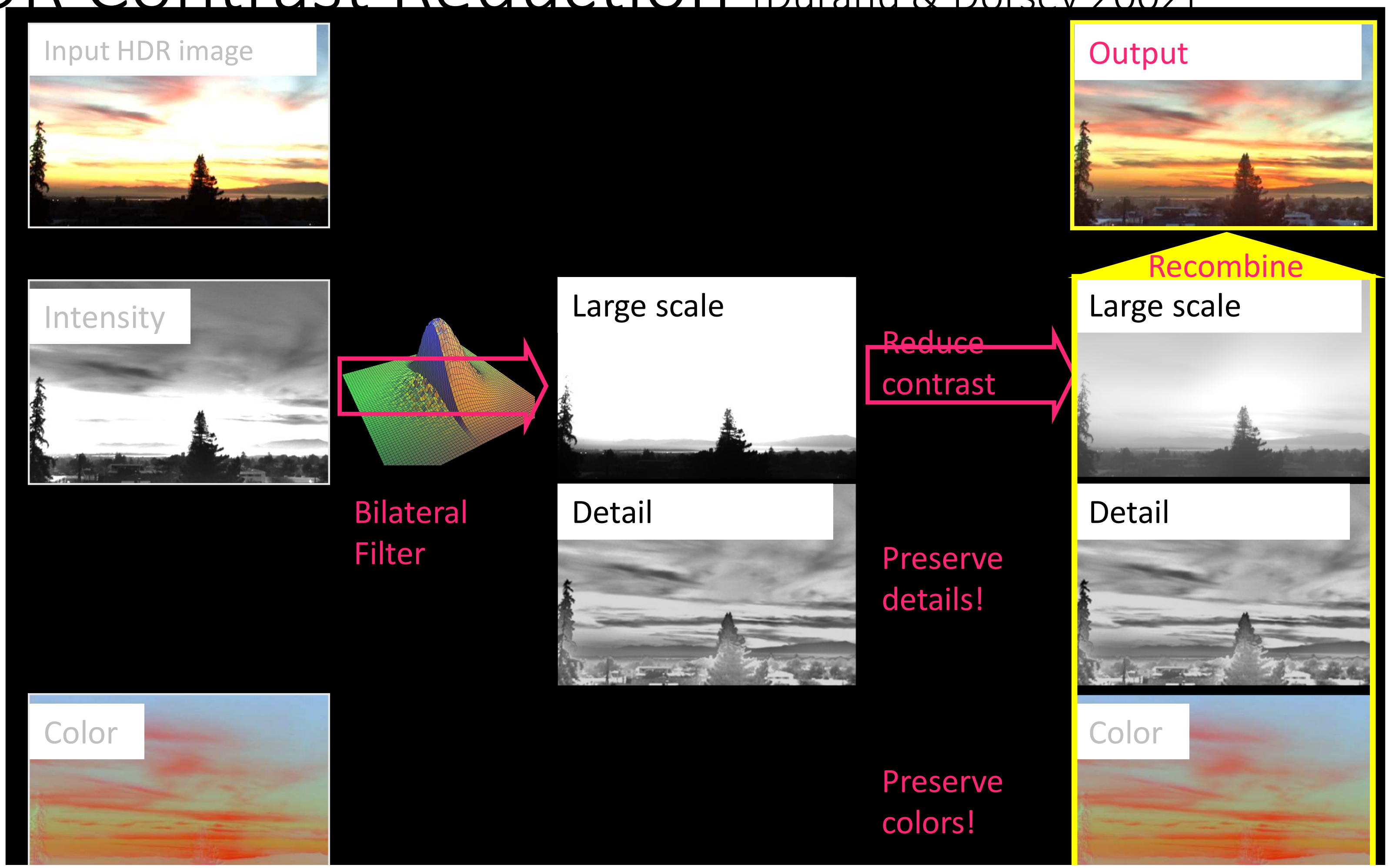
HDR Contrast Reduction [Durand & Dorsey 2002]



HDR Contrast Reduction [Durand & Dorsey 2002]



HDR Contrast Reduction [Durand & Dorsey 2002]



HDR Contrast Reduction [Durand & Dorsey 2002]