

# Tone Reproduction

# First Example: Linear Rescaling

- Brightness (informally Luminosity) how much energy (in watts)
- **Rescaling** is a point processing technique that alters the **contrast** and/or **brightness** of an image.
- In photography, **exposure** is a measure of how much light is projected onto the imaging sensor.
  - **Overexposure**: more light than what the sensor can measure.
  - **Underexposure**: sensor is unable to detect the light.
- Images which are underexposed or overexposed can frequently be improved by brightening or darkening them.
- The contrast of an image can be altered to bring out the internal structure of the image.

# Rescaling Math

- Given a sample  $C_{in}$  of the source image, rescaling computes the output sample,  $C_{out}$ , using the scaling function

$$C_{out} = \alpha C_{in} + \beta$$

- $\alpha$  is a real-valued scaling factor known as **gain**
- $\beta$  is a real-valued scaling factor known as **bias**

# Why Use Both $\alpha$ , $\beta$ ?

- Consider two rescaled source samples of  $S$  rescaled to  $S'$ .
- Calculate the **contrast** (the absolute difference) between the source and destination, called  $\Delta S$  and  $\Delta S'$ .
- Now consider the relative change in contrast between the source and destination.

$$\begin{aligned}S'_1 &= \alpha S_1 + \beta, \\S'_2 &= \alpha S_2 + \beta.\end{aligned}$$

$$\begin{aligned}\Delta S' &= |S'_1 - S'_2|, \\ \Delta S &= |S_1 - S_2|.\end{aligned}$$

$$\frac{\Delta S'}{\Delta S} = \frac{|S'_1 - S'_2|}{|S_1 - S_2|}.$$

# Why Use Both $\alpha$ , $\beta$ ?

- The relative change in contrast can be simplified as

$$\begin{aligned}\frac{\Delta S'}{\Delta S} &= \frac{|(\alpha S_1 + \beta) - (\alpha S_2 + \beta)|}{|S_1 - S_2|} \\ &= \frac{|\alpha| \cdot |S_1 - S_2|}{|S_1 - S_2|} \\ &= |\alpha|.\end{aligned}$$

- Thus, gain ( $\alpha$ ) controls the change in contrast.
- Whereas bias ( $\beta$ ) *does not* affect the contrast
- Bias, however, controls the final **brightness** of the rescaled image. Negative bias darkens and positive bias brightens the image

# Sidebar: Relating Contrast Sensitivities to Signal Processing

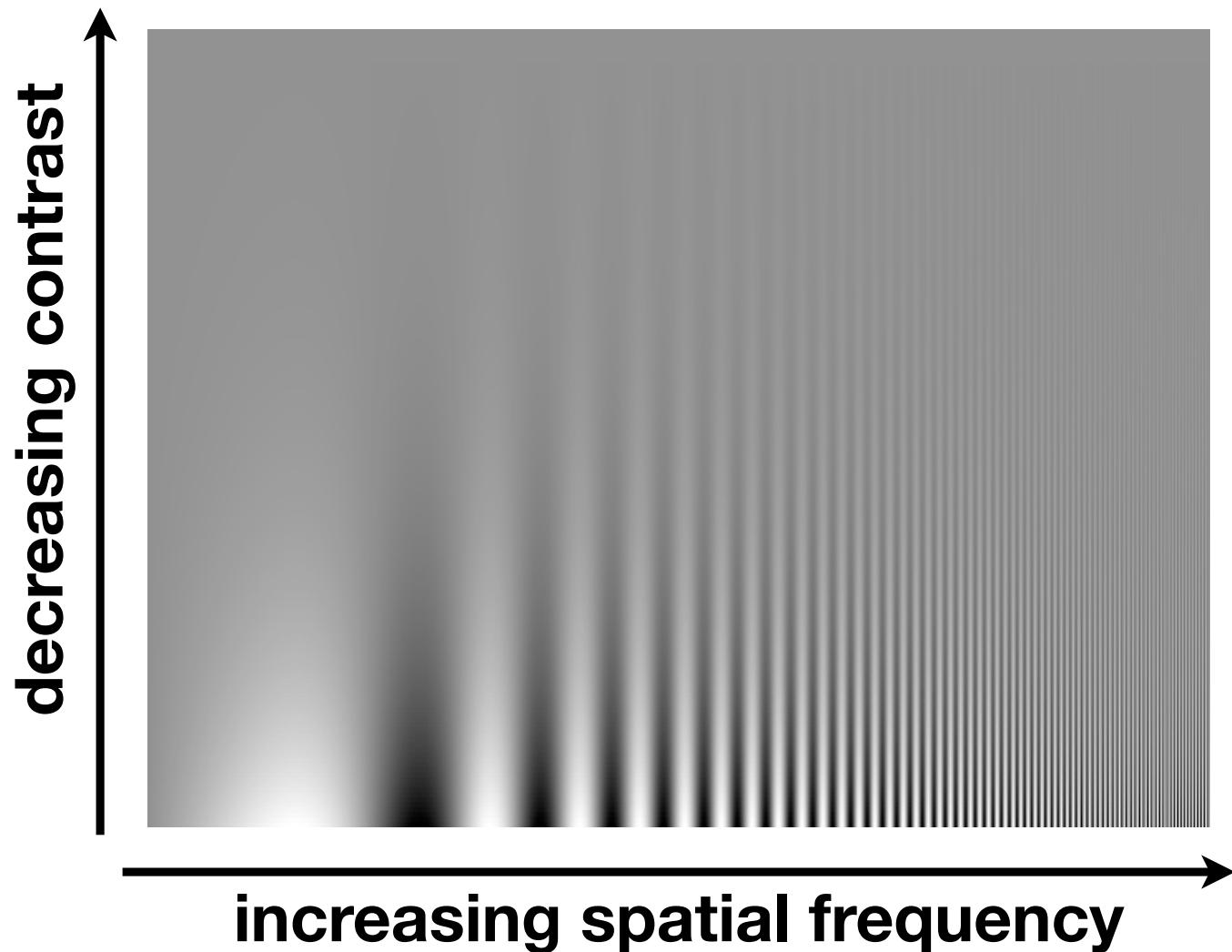
We want to be able to notice small details in the image (da)

**Will see: The contrast between a pixel and its neighborhood pixels matters much more than the absolute brightness of this pixel:**

**That is, the gain  $\alpha$  is much more important to the viewer than the bias  $\beta$**

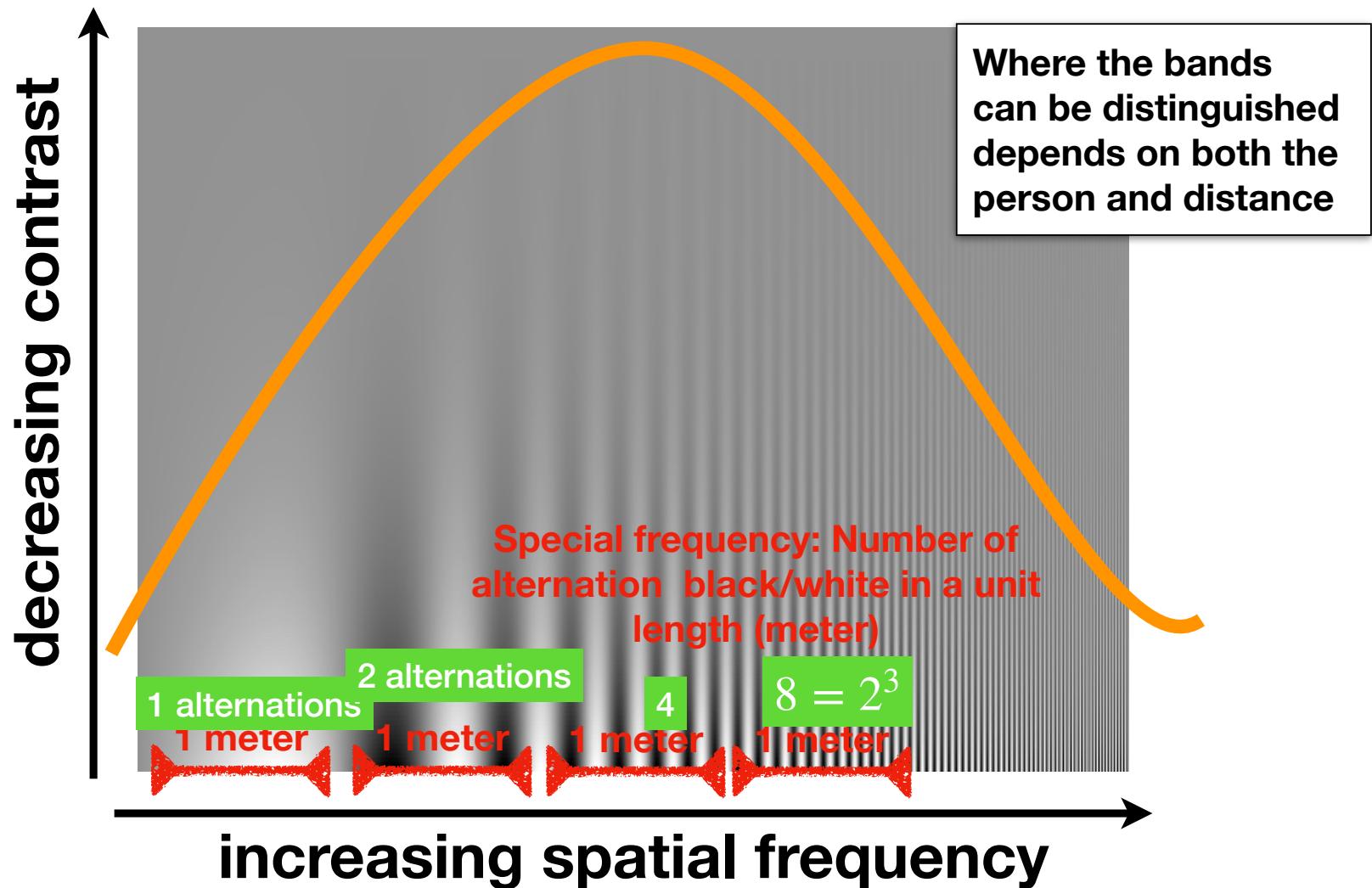
# Contrast Sensitivity Function

## Campbell-Robson Chart



# Contrast Sensitivity Function

## Campbell-Robson Chart



# Contrast Sensitivities Vary by Channel

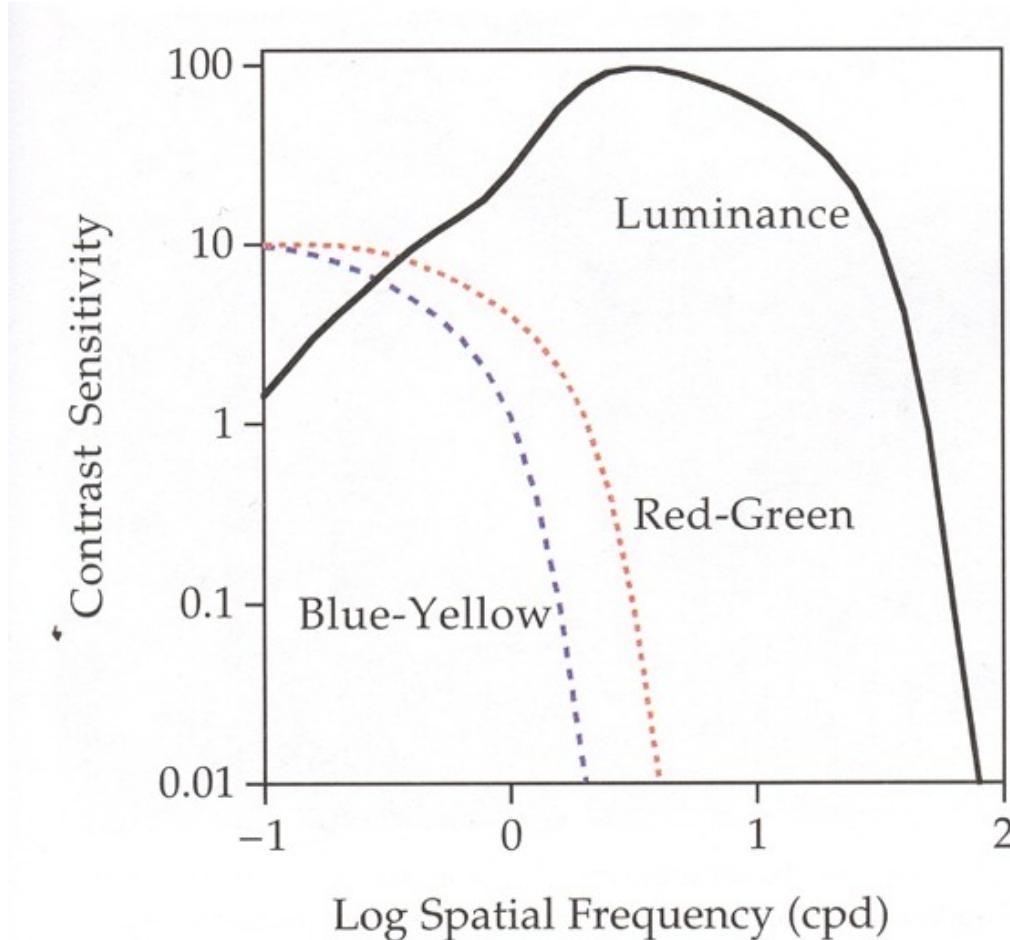


Figure 1-18. Spatial contrast sensitivity functions for luminance and chromatic contrast.

# Important: Clamping

- Rescaling may produce samples that lie outside of the output images (e.g. below 0 or above 255 in 8-bit images)
- **Clamping** the output values ensures that the output samples are truncated to the 8-bit dynamic range limit
- Note that clamping does ‘lose’ information, since it truncates.

$$clamp(x, min, max) = \begin{cases} \min & \text{if } \lfloor x \rfloor \leq \min, \\ \max & \text{if } \lfloor x \rfloor \geq \max, \\ \lfloor x \rfloor & \text{otherwise.} \end{cases}$$

# Rescaling Examples



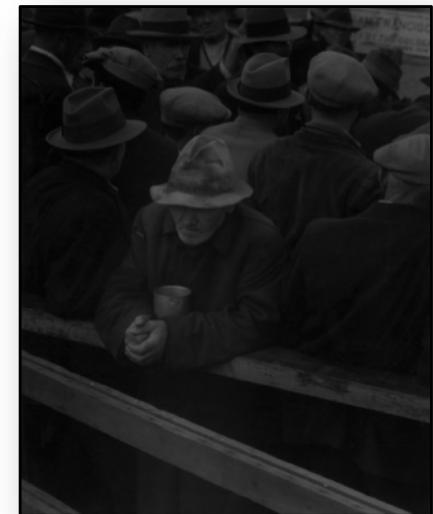
gain = 1, bias = 55



gain = 1, bias = -55



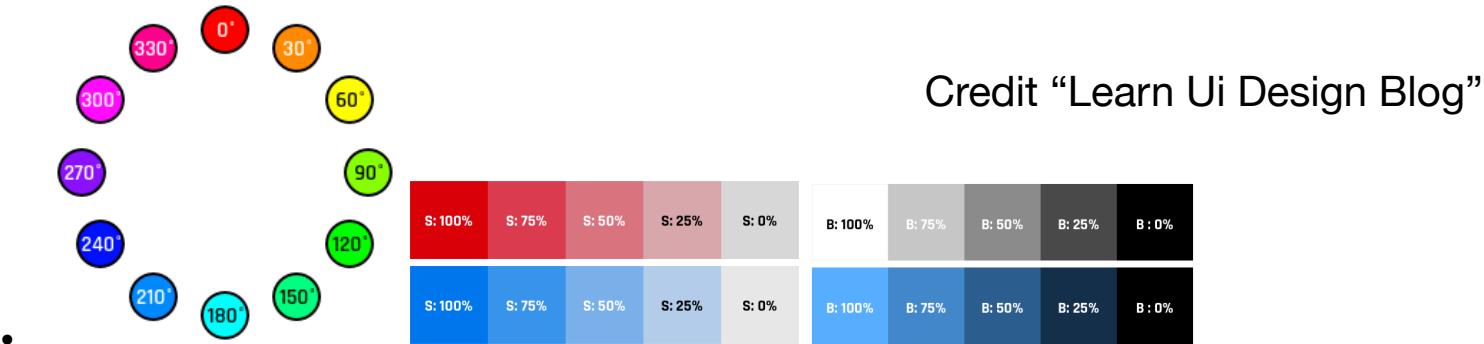
gain = 2, bias=0



gain = .5, bias=0

# Rescaling Color Images

- Often, it is desirable to apply different gain and bias values to each channel of a color image separately
  - Example: A color image that utilizes the HSB (Hue-Saturation-Brightness) color model.



- Since all color information is contained in the H and S channels, it may be useful to adjust ONLY the brightness, encoded in channel B, without altering the color of the image in any way.
- Simply put, If we rescale an image without changing its color, make sure not to alter the ratio  $\frac{R}{B}$ ,  $\frac{R}{G}$ ,  $\frac{B}{G}$
- Rescaling the channels of a color image in a non-uniform manner is also possible rescaling each color channel separately.

# Human Perception

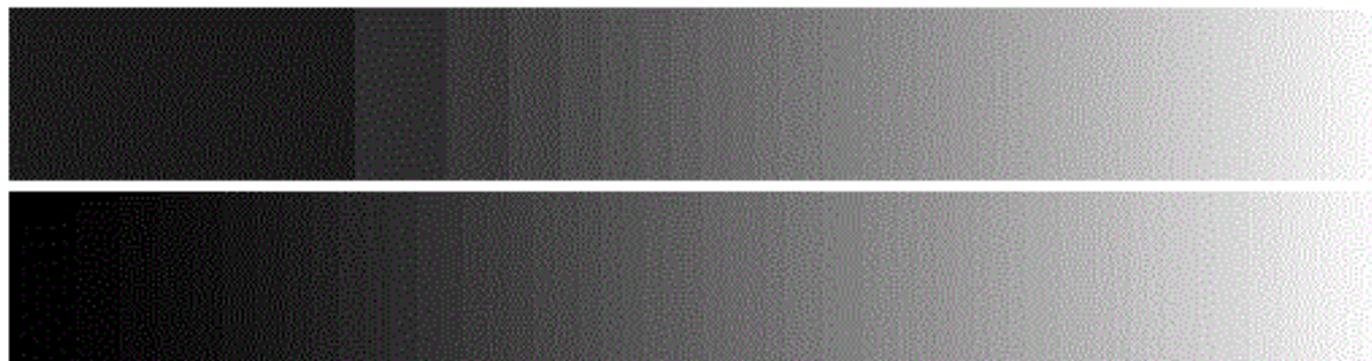
- Eye distinguishes color intensities as a function of the ratio between intensities.
- Consider  $I_1 < I_2 < I_3$ , for the step between  $I_1$  and  $I_2$  to look like the step from  $I_2$  to  $I_3$ , it must be that:

$$I_2 / I_1 = I_3 / I_2$$

- As opposed to the differences in contrast!  $I_2 - I_1 \neq I_3 - I_2$

# Perceived ( $I_p$ ) vs. Actual ( $I_a$ ) Intensity

- Perceived light actually behaves like  $I_p = (I_a)^\gamma$



Linear

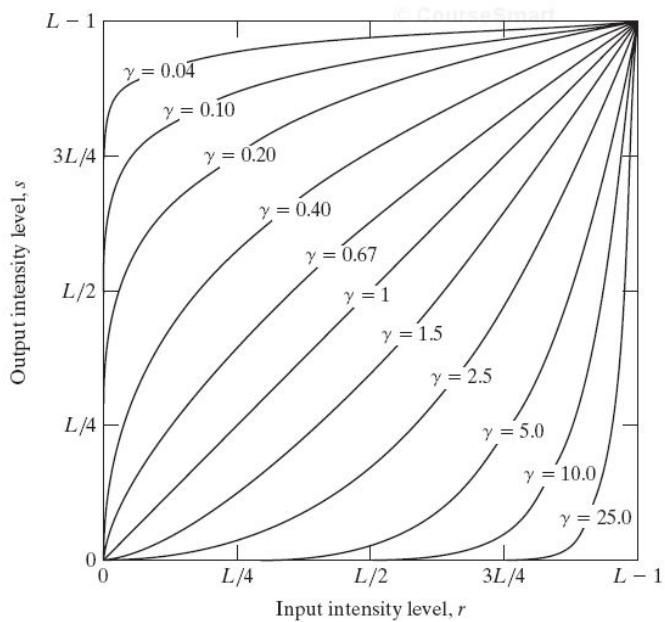
Gamma2.2

# Example: Gamma Correction

$$\mathbf{s} = \mathbf{r}^\gamma$$

a  
b  
c  
d

**FIGURE 3.9**  
(a) Aerial image.  
(b)–(d) Results of  
applying the  
transformation in  
Eq. (3.2-3) with  
 $c = 1$  and  
 $\gamma = 3.0, 4.0,$  and  
 $5.0$ , respectively.  
(Original image  
for this example  
courtesy of  
NASA.)



# Putting it all together: Gain, Bias, and Gamma

- $C_{out} = (\alpha C_{in} + \beta)^\gamma$
- $\alpha$  is known as **gain** (exposure)
- $\beta$  is known as **bias** (offset)
- $\gamma$  maps to a non-linear curve (**gamma** correction)

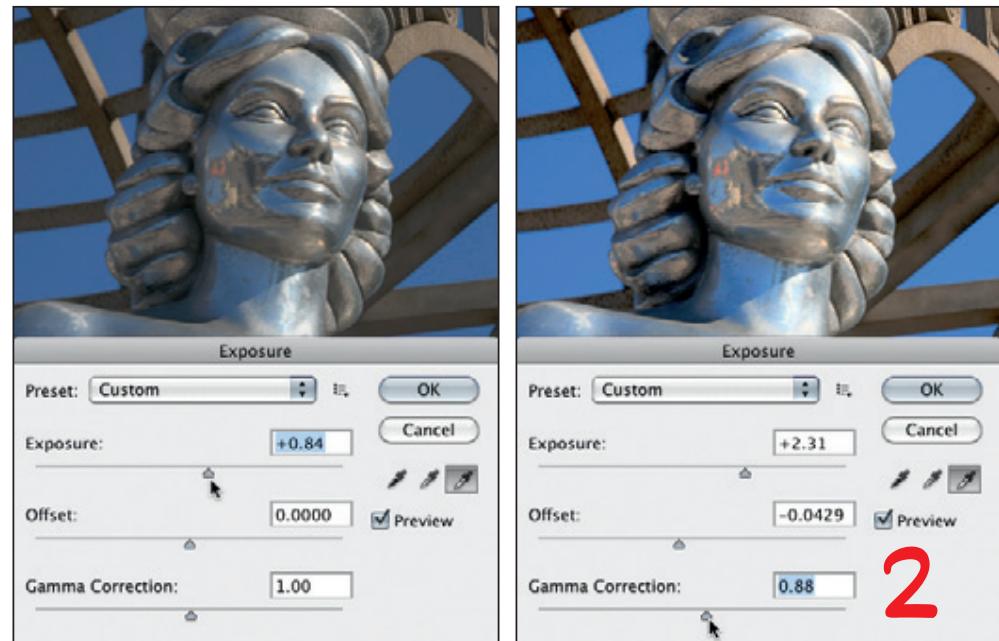
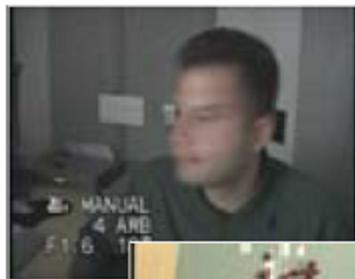


Image of Photoshop from  
Christian Bloch - The HDRI Handbook 2.0

# Dynamic Range

# The World is a High Dynamic Range (HDR)



1:1



1:1,500



1:25,000



1:400,000



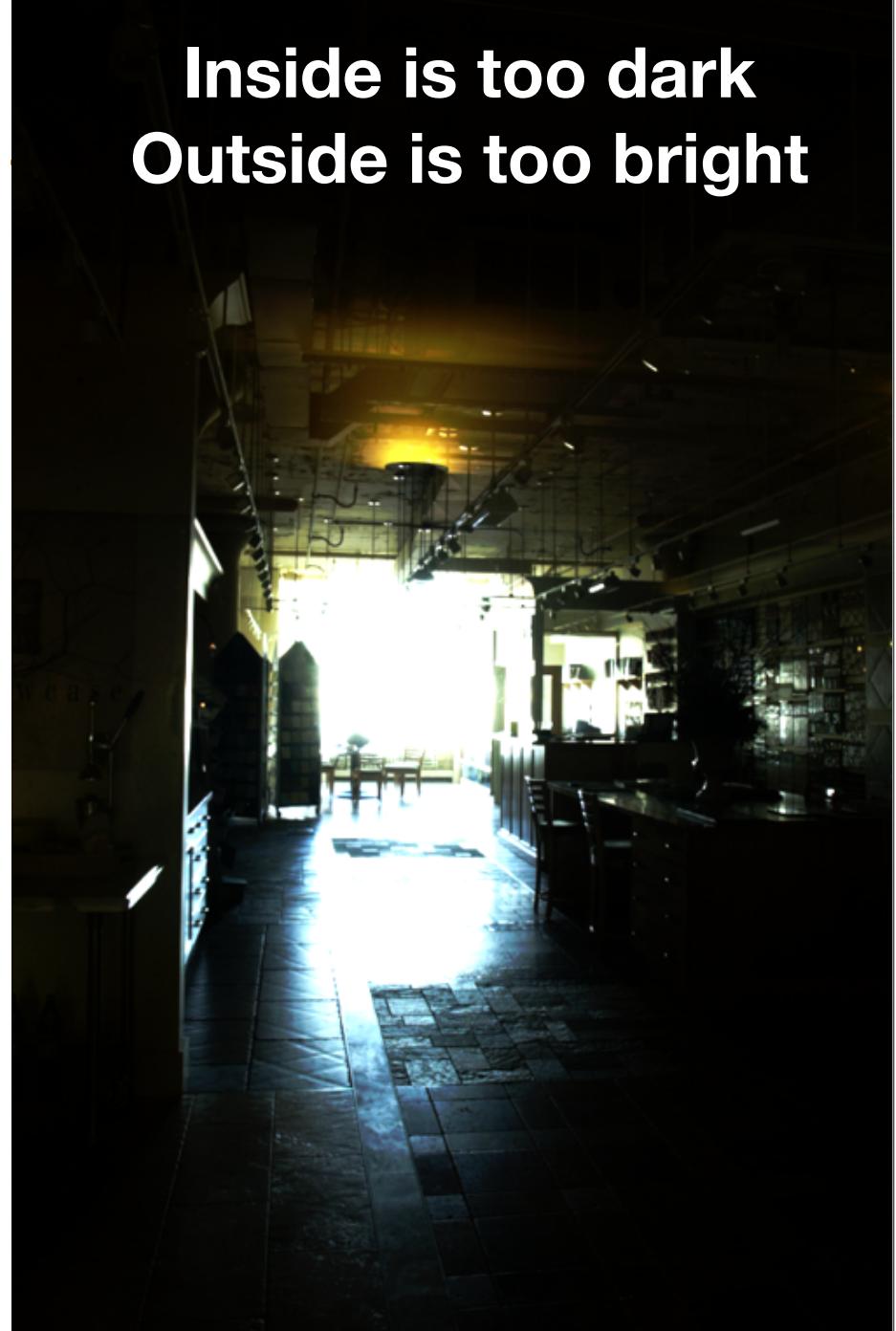
1:2,000,000,000

# Examples

**Sun overexposed  
Foreground too dark**

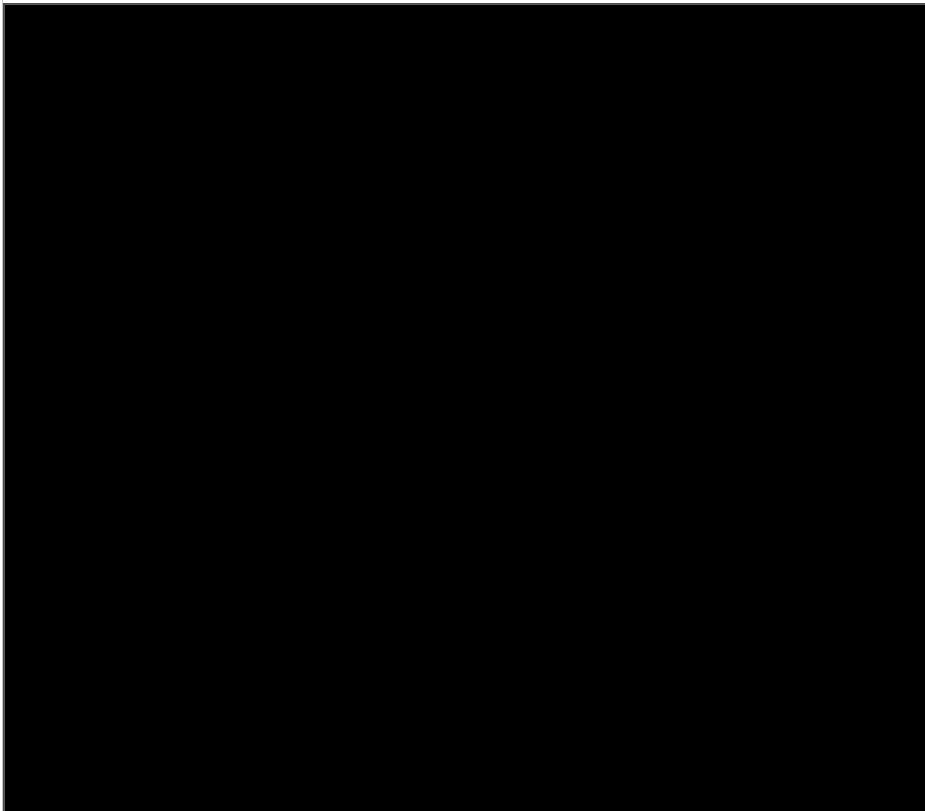


**Inside is too dark  
Outside is too bright**



# Dynamic Range in Displays?

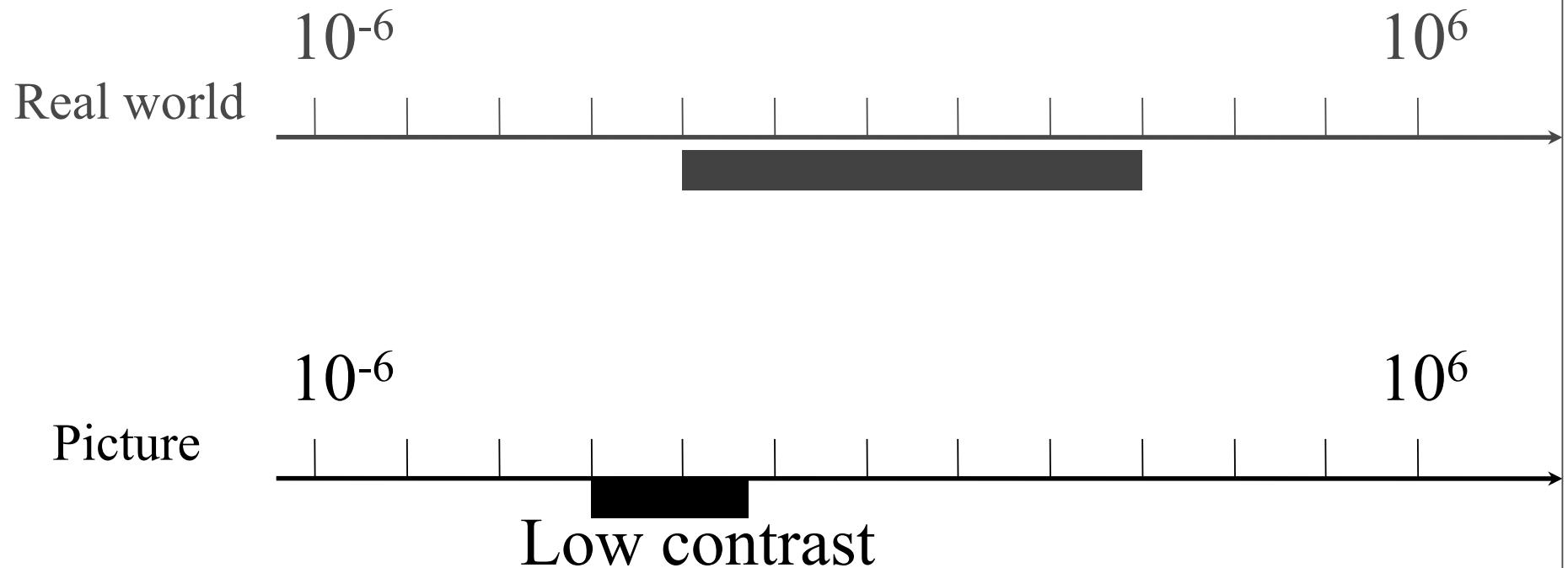
- Range of pure black vs. pure white?



# Dynamic Range in Displays?

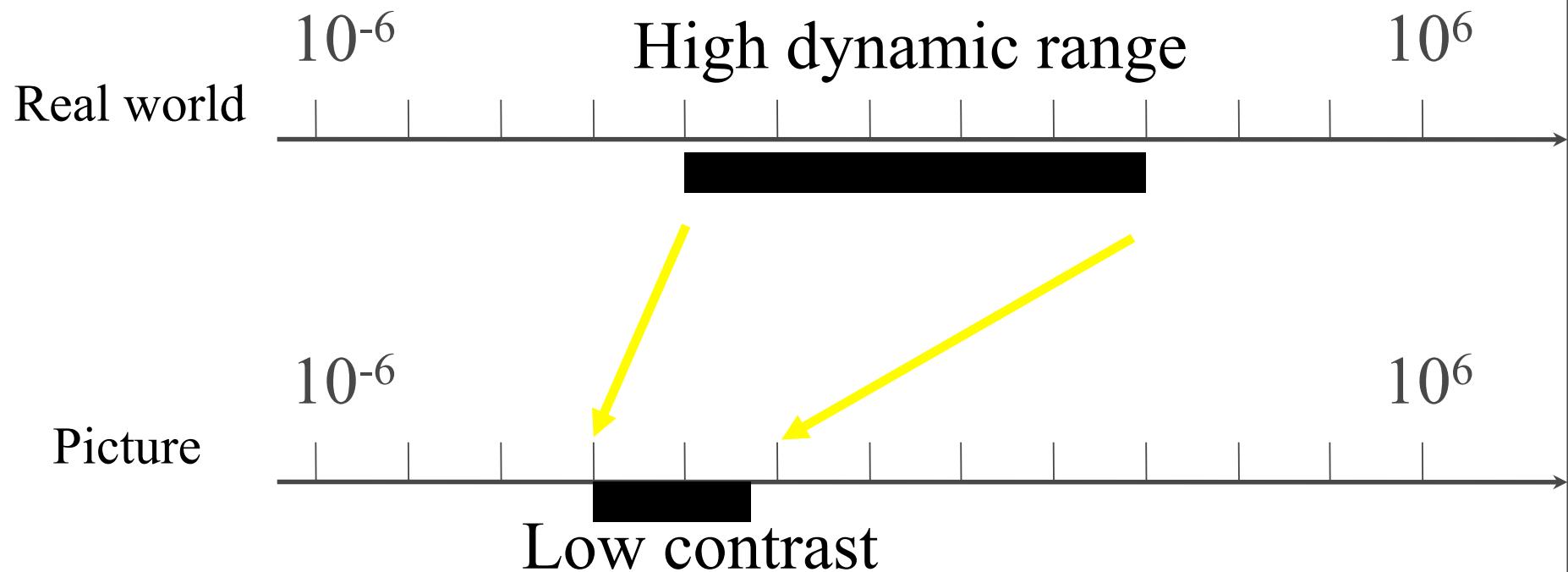
- Typically 1: 20 or 1:50
  - Black  is  $\sim 50x$  darker than white 

- Max 1:500



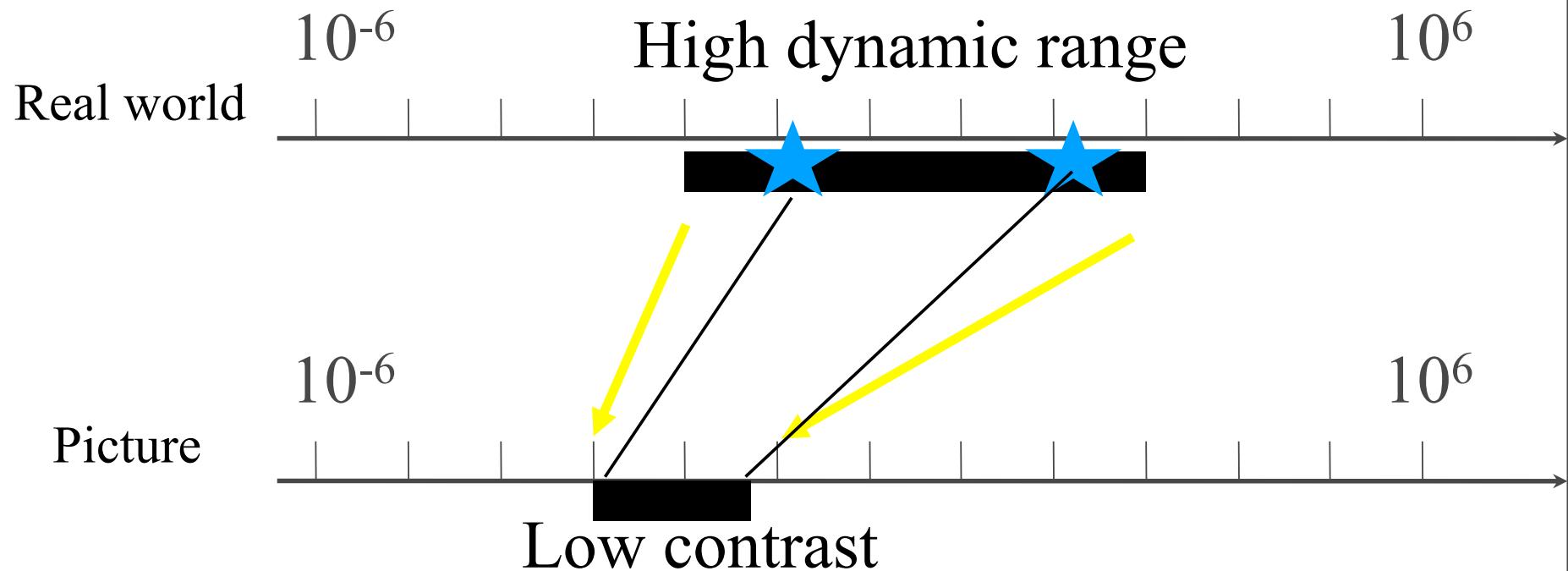
# Problem: Displaying the Information

- Problem: How should we map scene radiances (up to 1:100,000) to display radiances (only around 1:100) to produce a satisfactory image?
- Goal: match limited contrast of the display medium while preserving details
- Solution: **Tone Mapping**



# First solution: Linear mapping

- We will find the pixels with min and max intensity in the input image.
- Map them to the min and max intensities of the display
- Everything in between is mapped linearly.

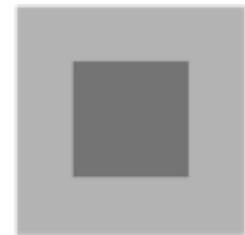


# Rescaling Color Images

- Often, it is desirable to apply different gain and bias values to each channel of a color image separately
  - Example: A color image that utilizes the HSB color model. Since all color information is contained in the H and S channels, it may be useful to adjust ONLY the brightness, encoded in channel B, without altering the color of the image in any way.
- Rescaling the channels of a color image in a non-uniform manner is also possible rescaling each color channel separately.

# Eyes and Dynamic Range

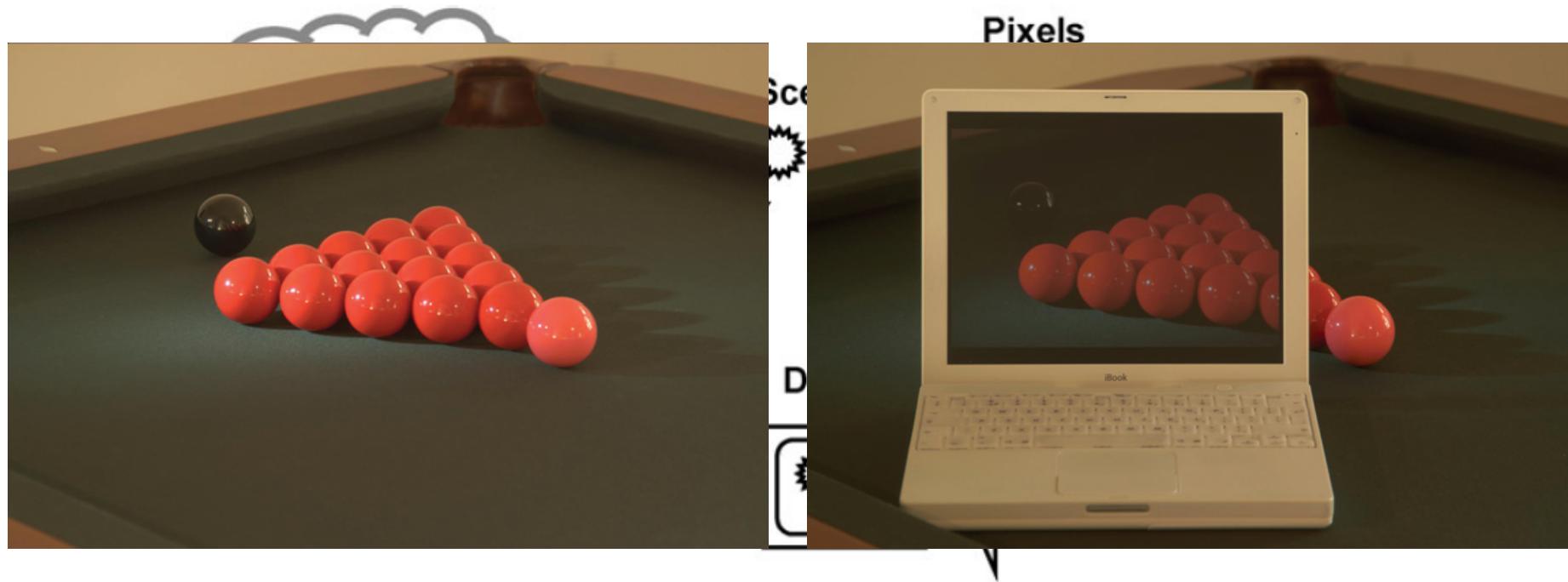
- We're sensitive to change (multiplicative)
  - A ratio of 1:2 is perceived as the same contrast as a ratio of 100 to 200
  - Use the log domain as much as possible
- But, eyes are **not** photometers
  - Dynamic adaptation (very local in retina)
  - Different sensitivity to spatial frequencies



Headlights  
are ON in  
both  
photos



# Approach: Visual Matching



- We do not need to reproduce the true radiance as long as it gives us a visual match.

# Tone Mapping

**Point operations** - performing the same operation (with the same constant) on the each pixel

**Global operations** - we could use different constant for different pixels

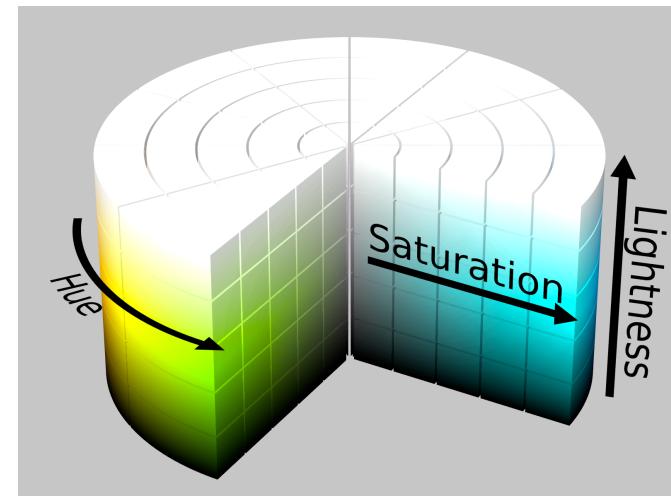
**Silly example**

For  $L(p_1)$  Lightness of  $p_1$

$$L(p_1, \text{out}) = L(p_1, \text{Source})/10 + 0.2$$

$$L(p_2, \text{out}) = L(p_1, \text{Source})/3 + 0.5$$

However, hw2 does not require the usage of a color scheme beside RGB  
See next slide



# Can we just scale? Maybe!

- For a color image, try to convert the input (world) luminance  $\mathbf{L}_w$  to a target display luminance  $\mathbf{L}_d$ .
- The input Luminance at pixel p is

$$1. \quad L_w(p) = \frac{1}{61}(20.0R(p) + 40G(p) + B(p))$$

2. Compute the large luminance at this pixel  $L_w(p)$

$$L_w(p) = \frac{1}{61}(20.0R_w(p) + 40G_w(p) + B(p))$$

3. Change  $L$ , for example by scaling  $L_d = \alpha L_w + \beta$ , (the user decided about the values of  $\alpha, \beta$ )

4. Upgrade the RGB representation of this pixel by multiplying each pixel by the value  $L_d/L_w$  - see formula on the right side of the slide.

- When performing operations on  $L$  it is actually recommended to perform them on  $\log(L)$  domain. Before using the values That is, use  $\ln$  in particular, it works best in the log and/or exponential domains
- $\log_{10}(x)=1+\log_{10}(y)$  means  $x=10y$
- The base of the log is not important, as long as we are consistent in the mapping

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

# What if rescaling and gamma-correction are not sufficient

- A useful collection of algorithms use the idea that in multiple scenarios, the highlights-illuminated regions
- Applicable for regions where changes from high brightness to low brightness occurs slowly: Big regions are “high”, big regions are “low” (low spatial frequency)
- Note that this does not mean the “high” or “low” regions are short in details.
- Examples: Sky, windows during day light
- Or the way chatGPT rephrased it:
  - A collection of algorithms that is useful in many scenarios uses the idea that in regions where there is a slow change from bright to dark, large regions can be identified as either “high” or “low” brightness.
  - This is because these regions have low spatial frequency. However, this does not mean that the “high” or “low” regions are lacking in detail. Examples of this include the sky and windows during daylight hours.

# Without HDR + Tone Mapping





From Durand and Dorsey. No single global exposure can preserve both the colors of the sky and the details of the landscape, as shown on the rightmost images.

# With HDR + Tone Mapping



# Before



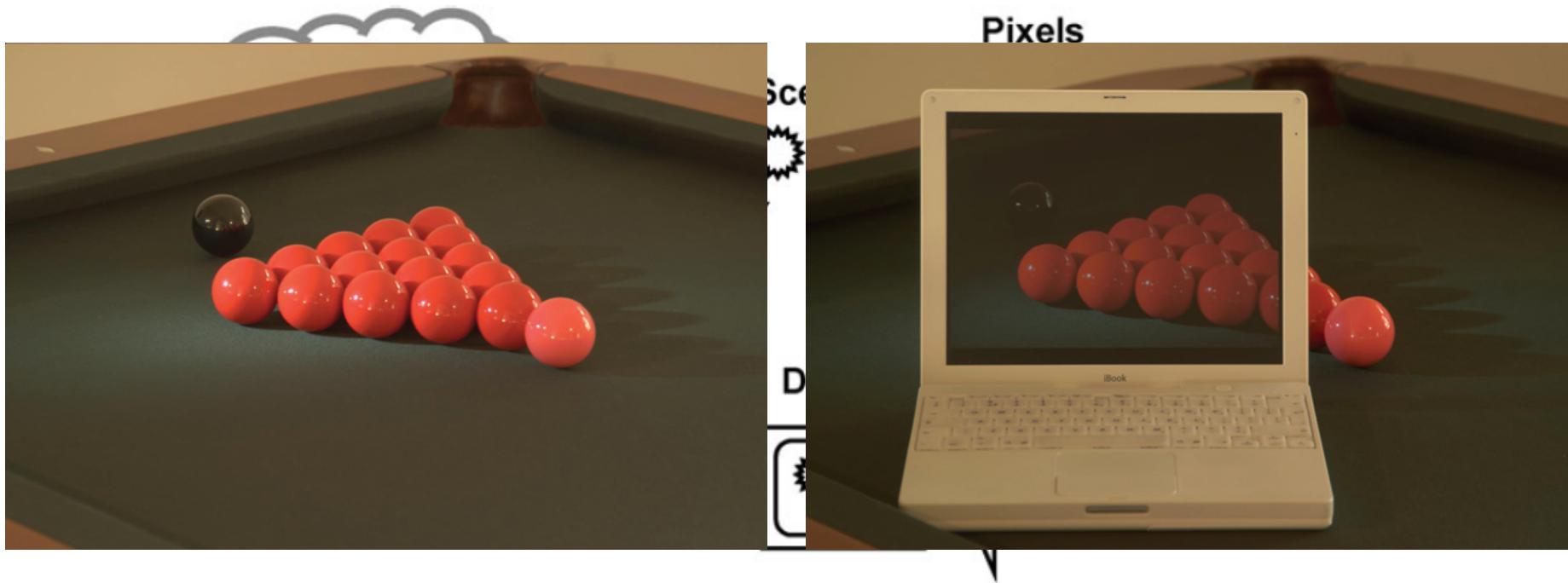
<http://abduzeedo.com/20-beautiful-hdr-pictures>

# Not All Tone Mapping Produces Extreme Results, Sometimes Just Beautiful Ones

Check (recommended)

<http://luminous-landscape.com/essays/hdr-plea.shtml>

# Approach: Visual Matching



- We do not need to reproduce the true radiance as long as it gives us a visual match.

Question: But why do we need more than 100 levels of intensity (luminance) if in the input file we only have 256 values of intensities (RGB) ?

Answer: Not all file format has so few levels.

Even PPM could have 2 bytes per channel, so  $256^2=65536$  levels per channel.

Other formats gives much wider range:

# Radiance RGBE Format (.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

## Non-local tone mapping

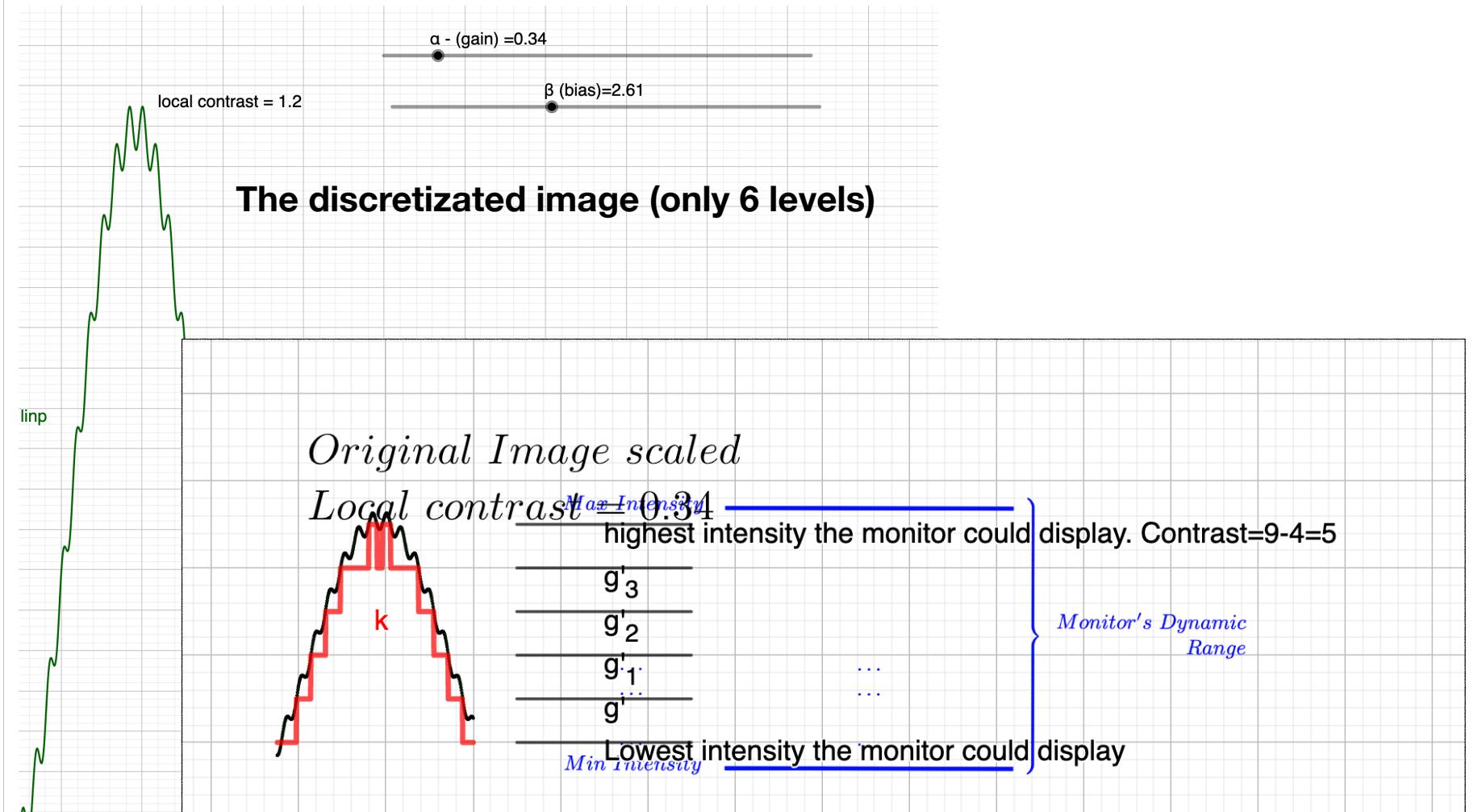
For a pixel  $p$ , the target value of  $L_d(p)$  depends not  
only on  $p$

But also depends on luminosity of pixels elsewhere

**Given - an image with a large dynamic range (e.g. from image of outdoor scene  
Need to compress it to much smaller dynamic range (monitor)  
but avoid decreasing local contrast**

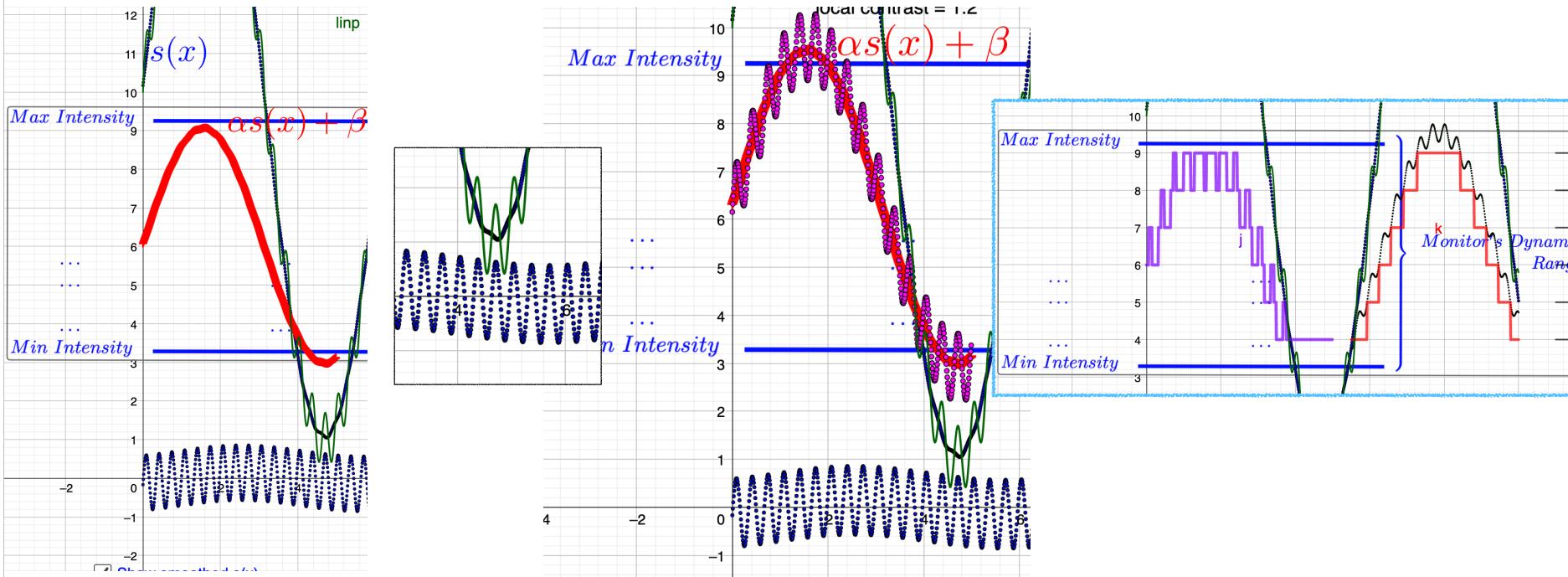
**In the GG demo, the input image represents the input image**

**Option 1 - compress  $C_{out}(p_i) = \alpha C_{in}(p_i) + \beta$**

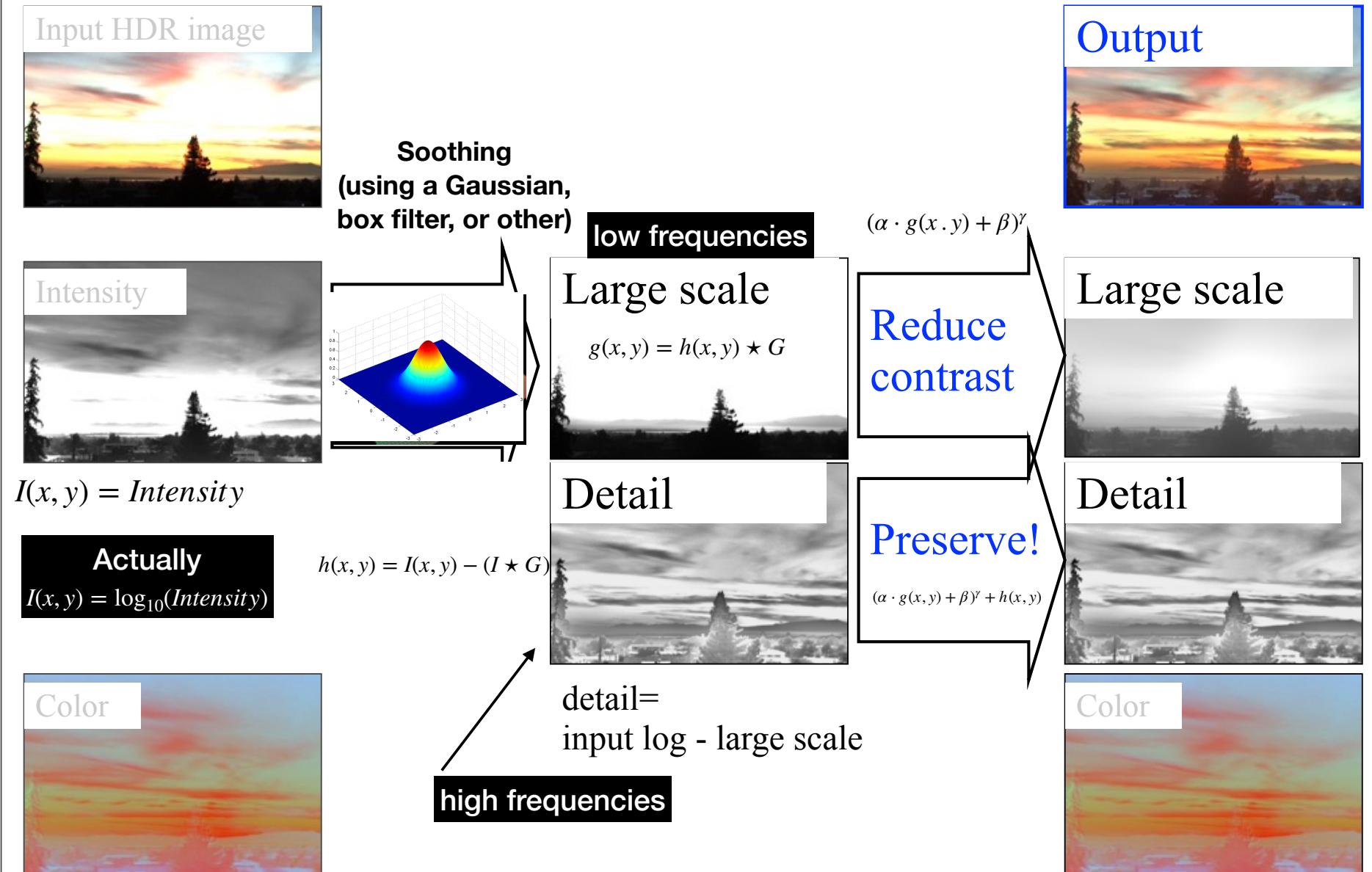


## A better idea

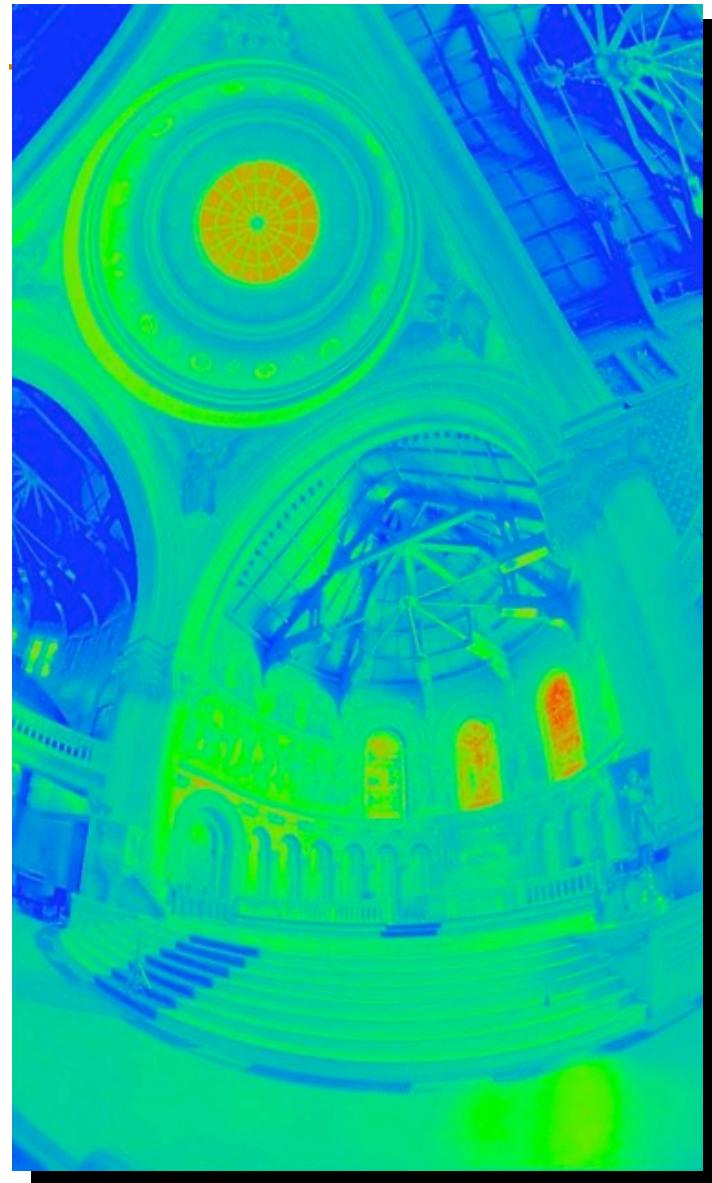
- $f(x)$  = Brightness level at pixel  $x$
- Compute  $s(x)$  - a low pass filter:
  - For example, compute  $s(x) = (f(x-1) + f(x) + f(x+1))/3$  (mean filter)
  - better idea: Use **convolution** with a tent or a gaussian
- in  $s(x)$ , details that change frequently (high special frequencies) are averaged.
- computer  $h(x) = f(x) - s(x)$ . Here only the details that have high special frequencies appear.
- Hopefully, the contrast (max intensity - min intensity) are no nearly as large as the contrast in  $s(x)$
- Rescale  $s(x)$  by using the gain and bias ( $\alpha, \beta$ ) such that it fits the monitor dynamic range.  
 $s(x) \rightarrow \alpha \cdot s(x) + \beta$ . Usually  $\alpha < 1$ .
- Place back the missing details. The output is  $f_{out}(x) = \alpha s(x) + \beta + h(x)$



# Recap



# The Radiance Map

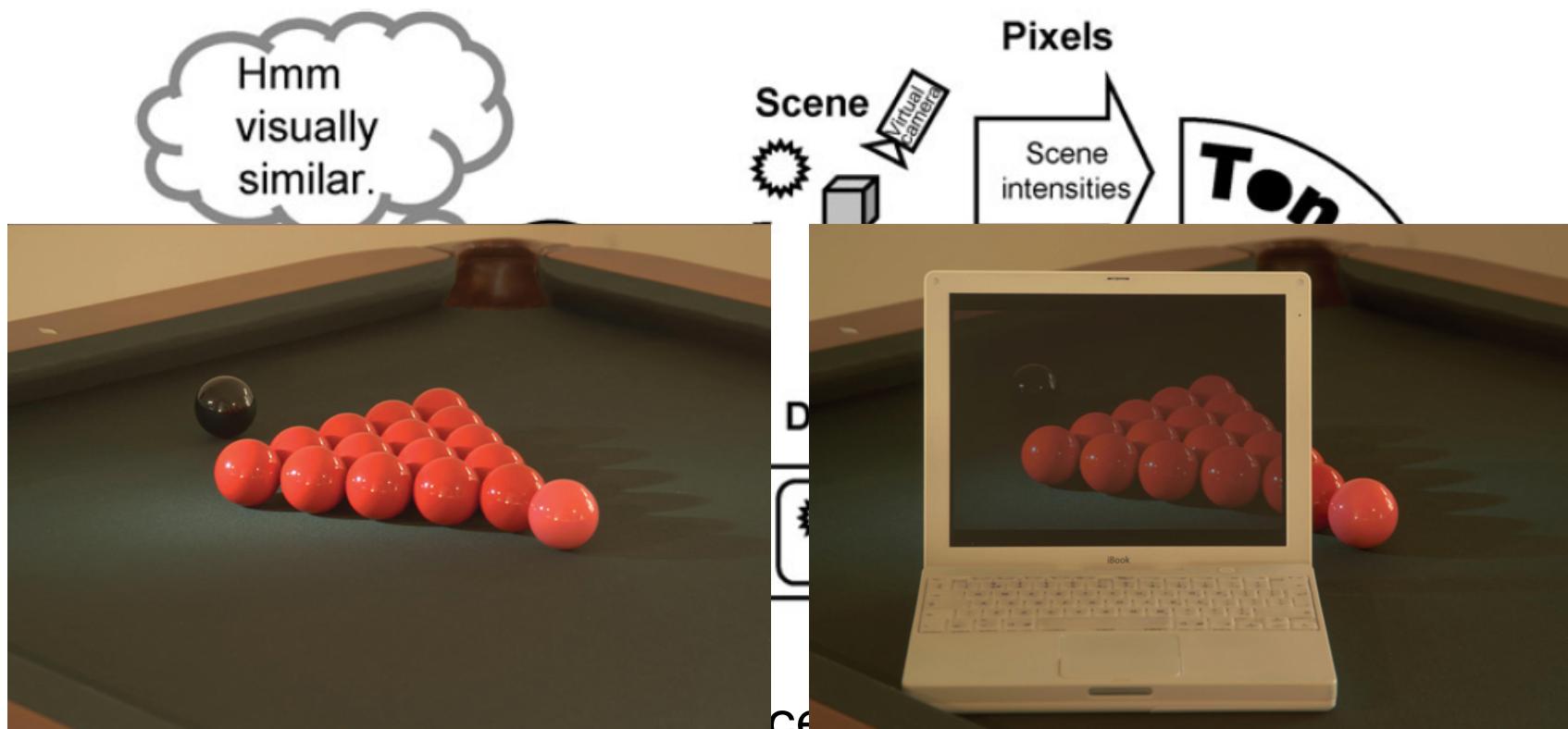


# The Radiance Map



Linearly scaled to  
display device

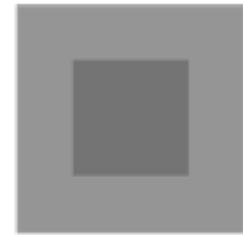
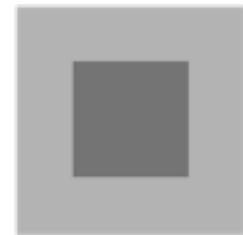
# Approach: Visual Matching



We do not need to reproduce the true radiance as long as it gives us a visual match.

# Eyes and Dynamic Range

- We're sensitive to change (multiplicative)
  - A ratio of 1:2 is perceived as the same contrast as a ratio of 100 to 200
  - Use the log domain as much as possible
- But, eyes are **not** photometers
  - Dynamic adaptation (very local in retina)
  - Different sensitivity to spatial frequencies



Headlights  
are ON in  
both  
photos



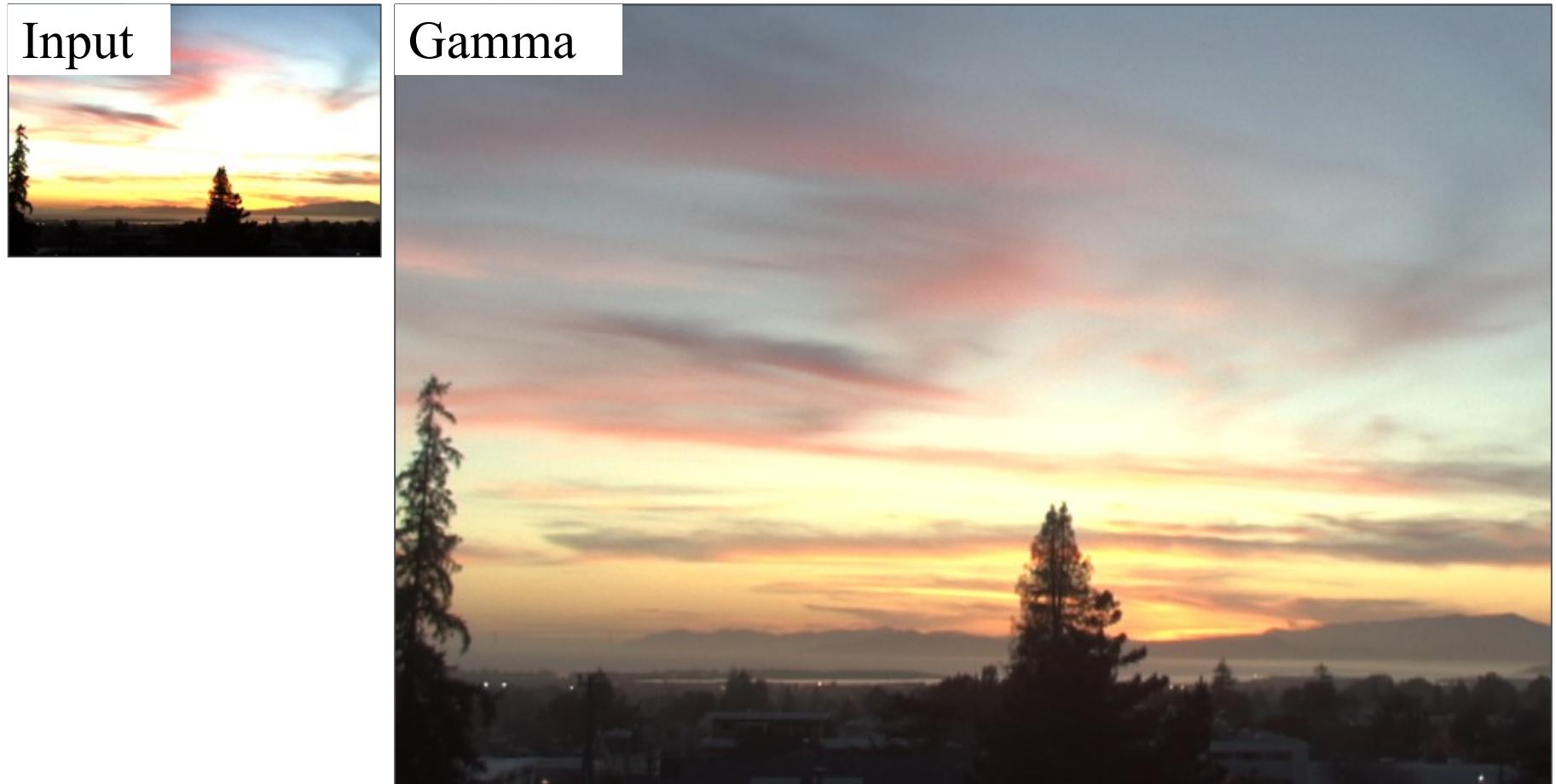
# Can we just scale? Maybe!

- For a color image, try to convert the input (world) luminance  $\mathbf{L}_w$  to a target display luminance  $\mathbf{L}_d$
- This type of scaling works (sometimes). In particular, it works best in the log and/or exponential domains
- $\log_{10}(x)=1+\log_{10}(y)$  means  $x=10y$
- The base of the log is not important, as long as we are consistent in the mapping

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

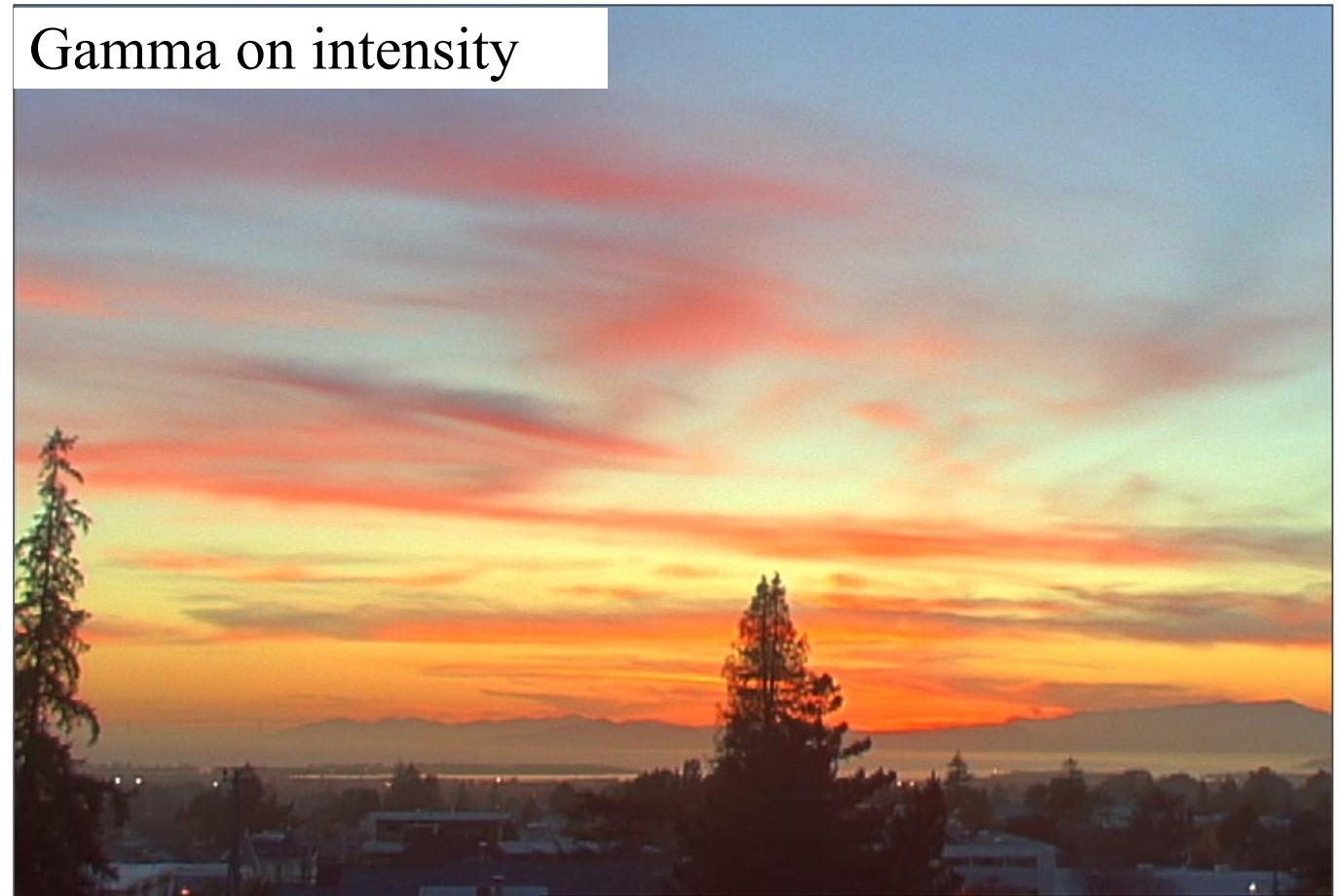
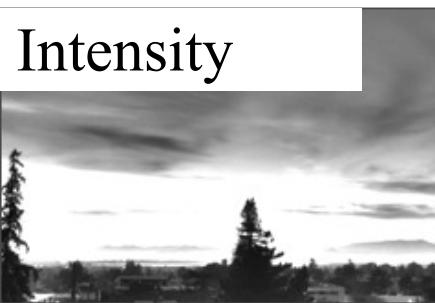
# What scale value to use? How about Gamma compression

- $C_{\text{out}} = C_{\text{in}}^{\gamma}$ , where  $0 < \gamma < 1$  applied to each R,G,B channel
- $C_{\text{out,red}} = C_{\text{in,red}}^{\gamma}$  ;  $C_{\text{out,green}} = C_{\text{in,green}}^{\gamma}$   $C_{\text{out,blue}} = C_{\text{in,blue}}^{\gamma}$
- Colors are washed out, why?



# Gamma compression on Intensity

- Colors ok, but details in intensity are blurry



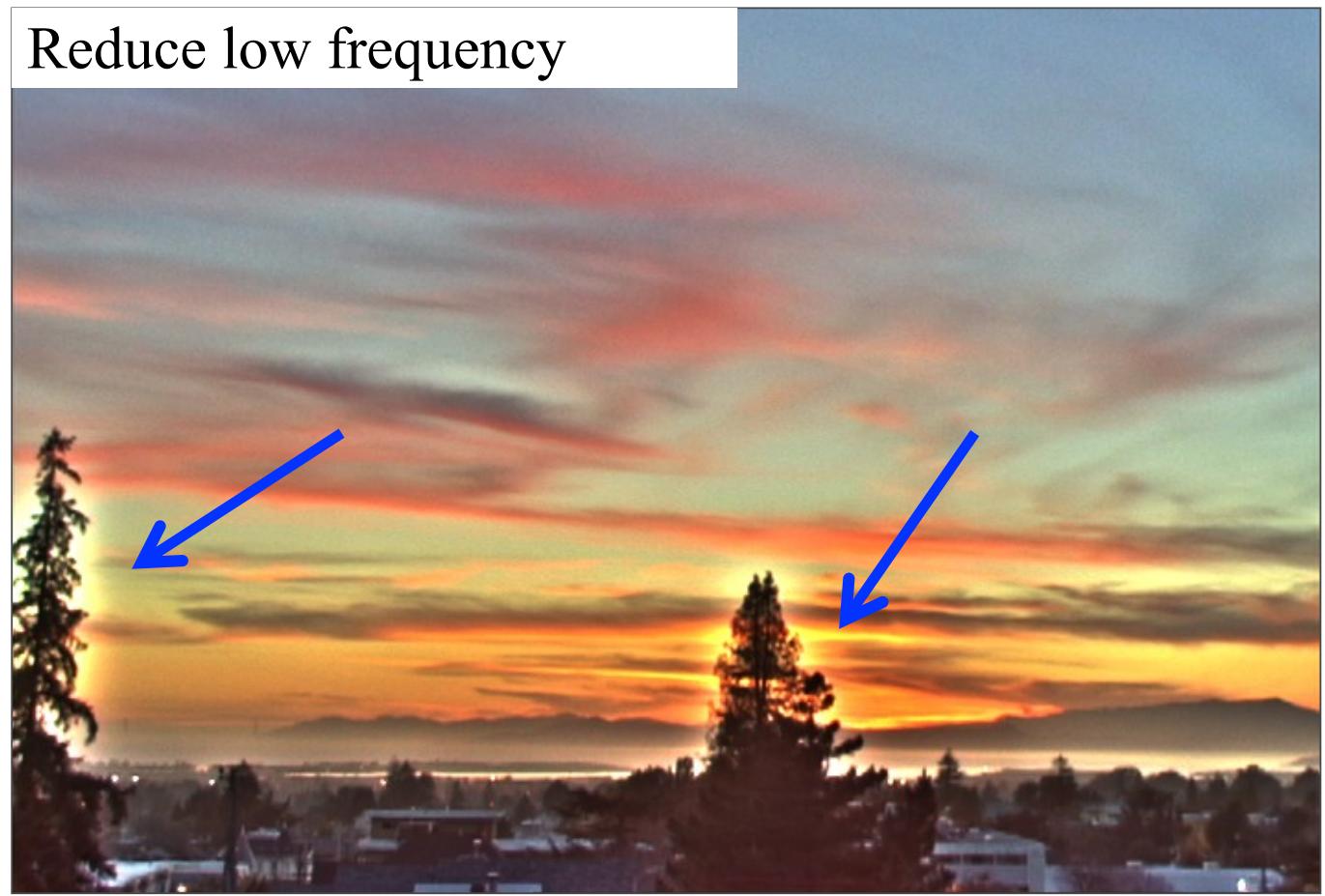
# Oppenheim 1968, Chiu et al. 1993

- Reduce contrast of low-frequencies
- Keep mid and high frequencies



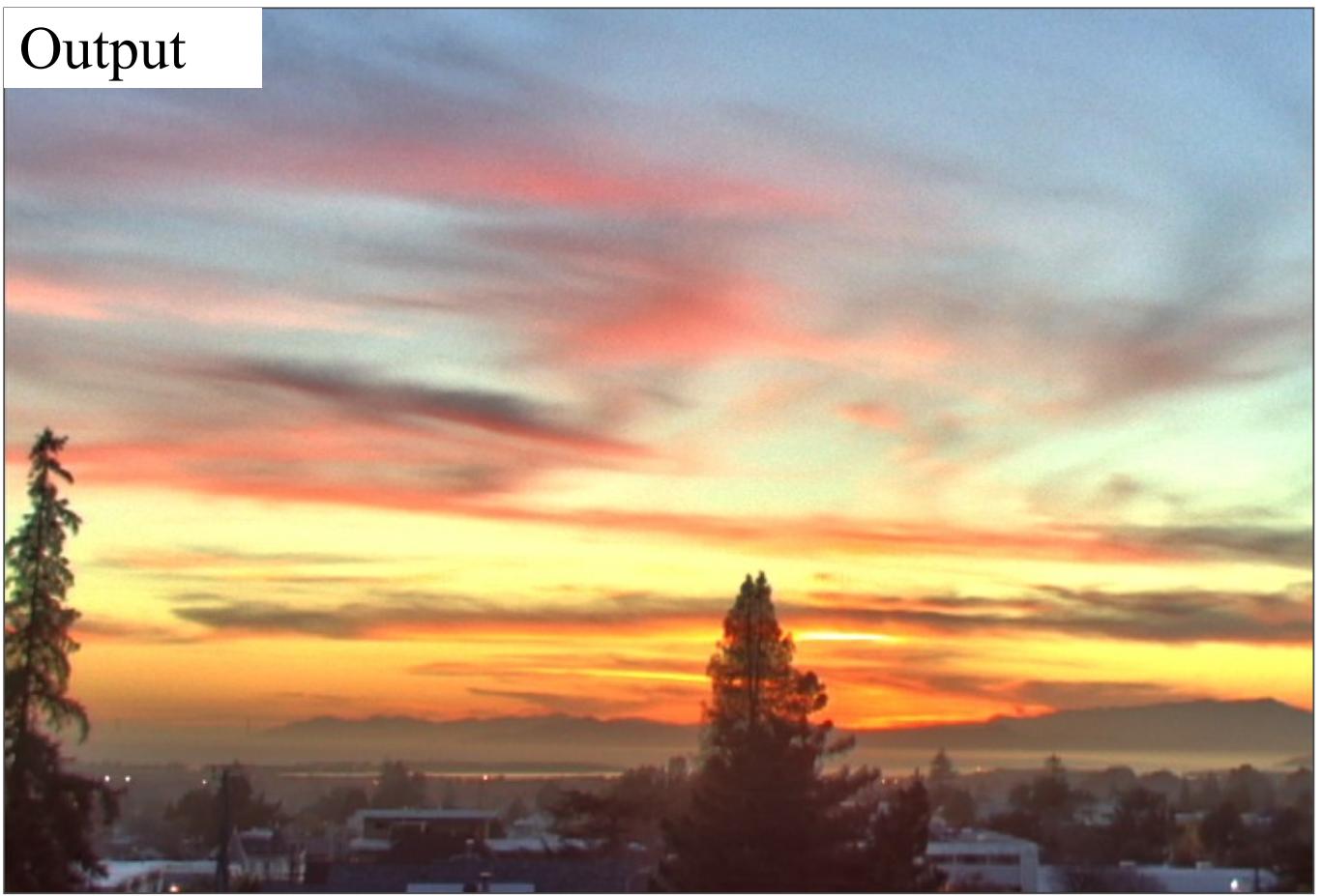
# The halo nightmare

- For strong edges
- Because they contain high frequency



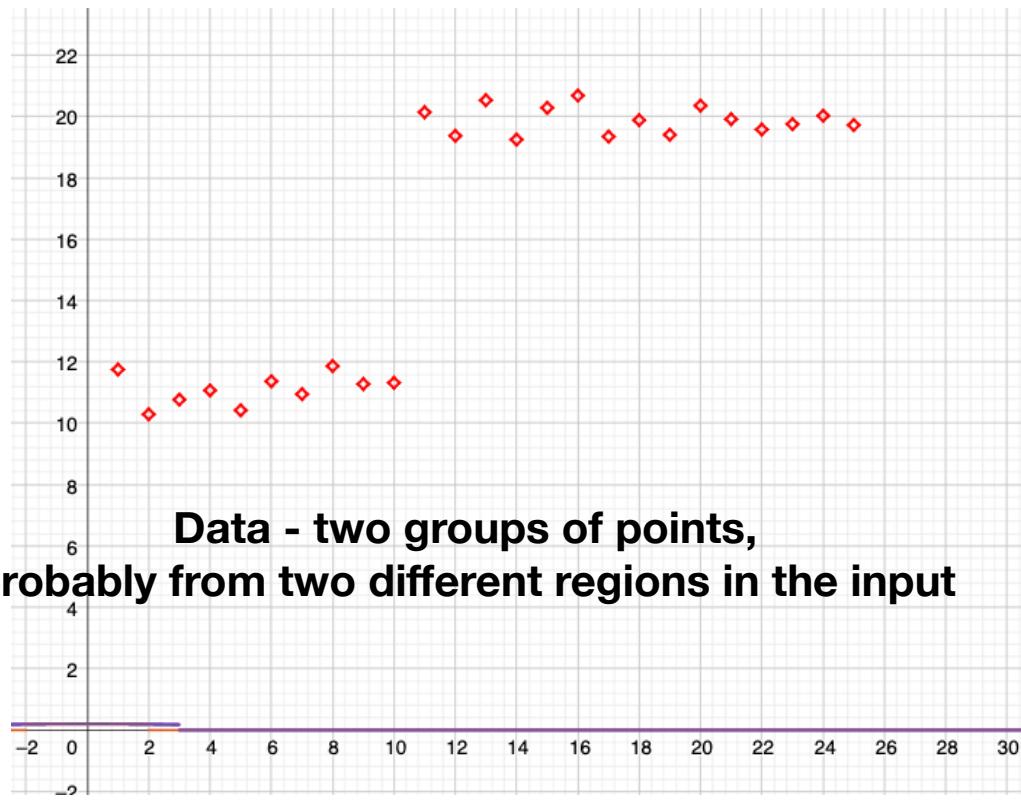
# Our approach

- Do not blur across edges
- Non-linear filtering



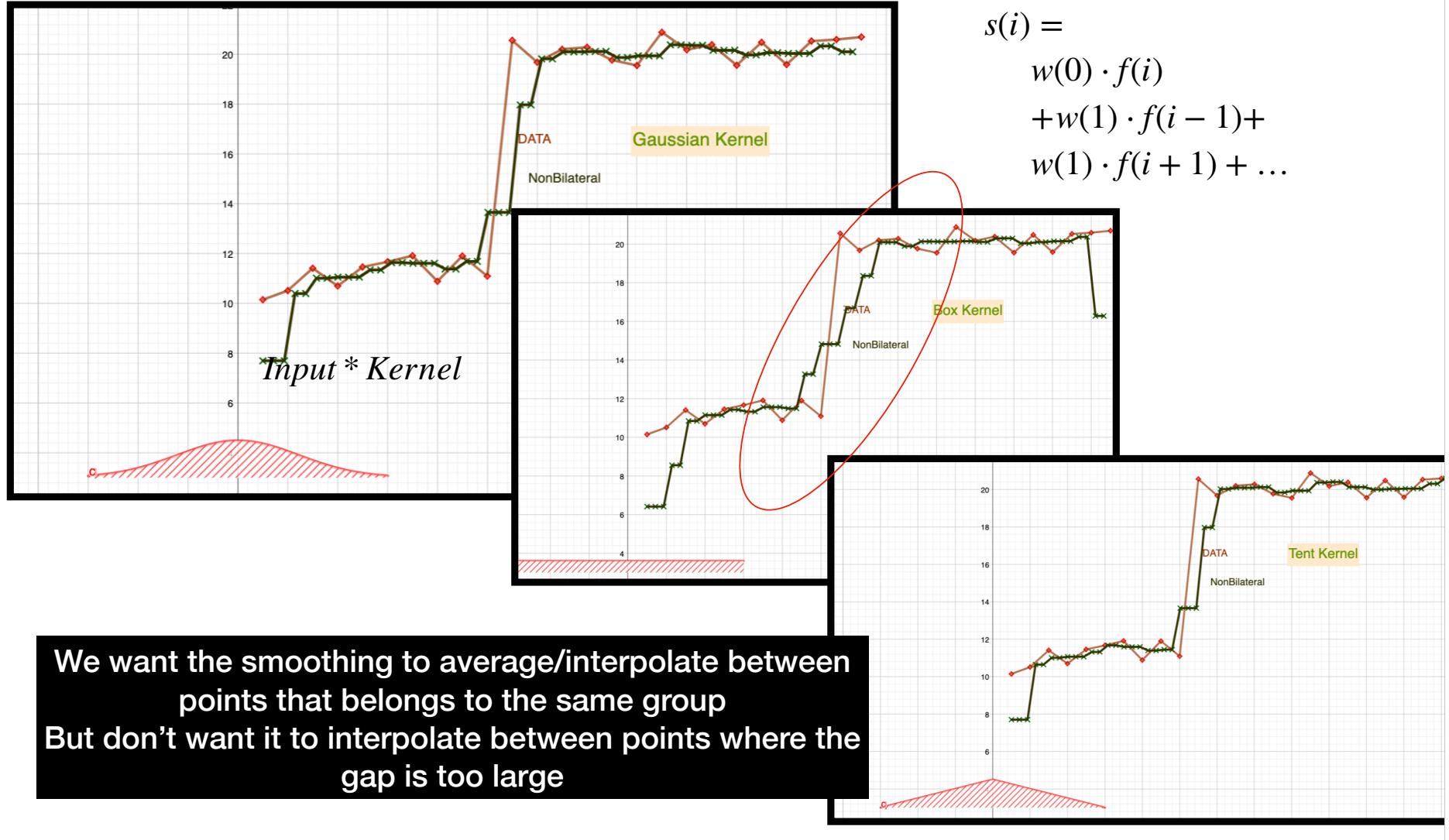
# Bilateral Filtering

To understand what are bilateral filtering, we need to revisit smoothing operations



# Bilateral Filtering

To understand what are bilateral filtering, we need to revisit smoothing operations



## How to smooth while persevering boundaries

In the convolution when compute  $s(i)$ , only sum values of  $f(x)$  where  $|f(x) - f(i)| \leq 3$

Instead of setting  $s(i) = \sum_{j=-3}^3 w(j) \cdot f(i+j)$ , the process to compute  $g(i)$  is slightly longer

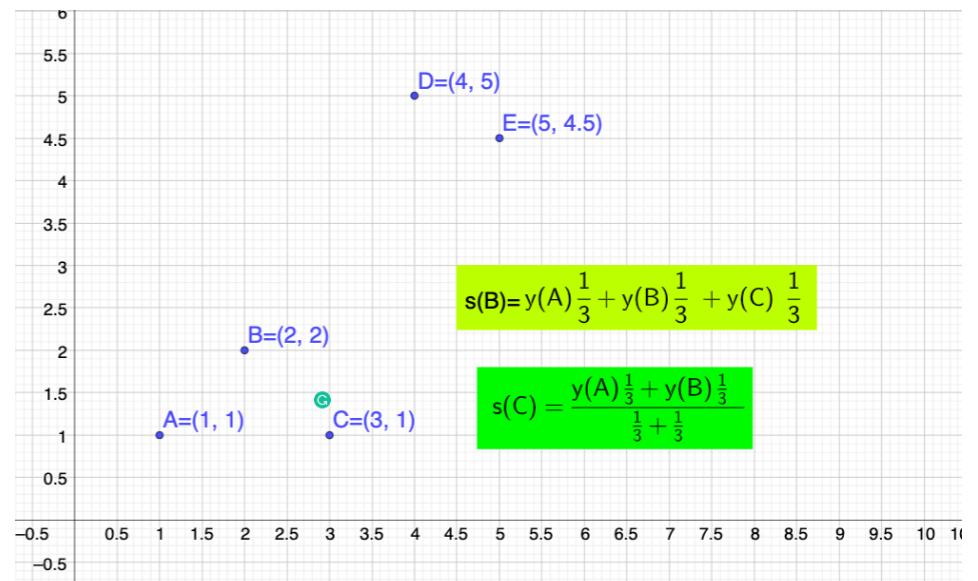
**Compute first**

$$s_0 = \sum \{ f(i+j) \cdot w(j) \mid |f(i) - f(i+j)| \leq 3 \text{ and } |j| \leq 3 \}$$

$$w_0 = \sum \{ w(j) \mid |f(i) - f(i+j)| \leq 3 \text{ and } |j| \leq 3 \}$$

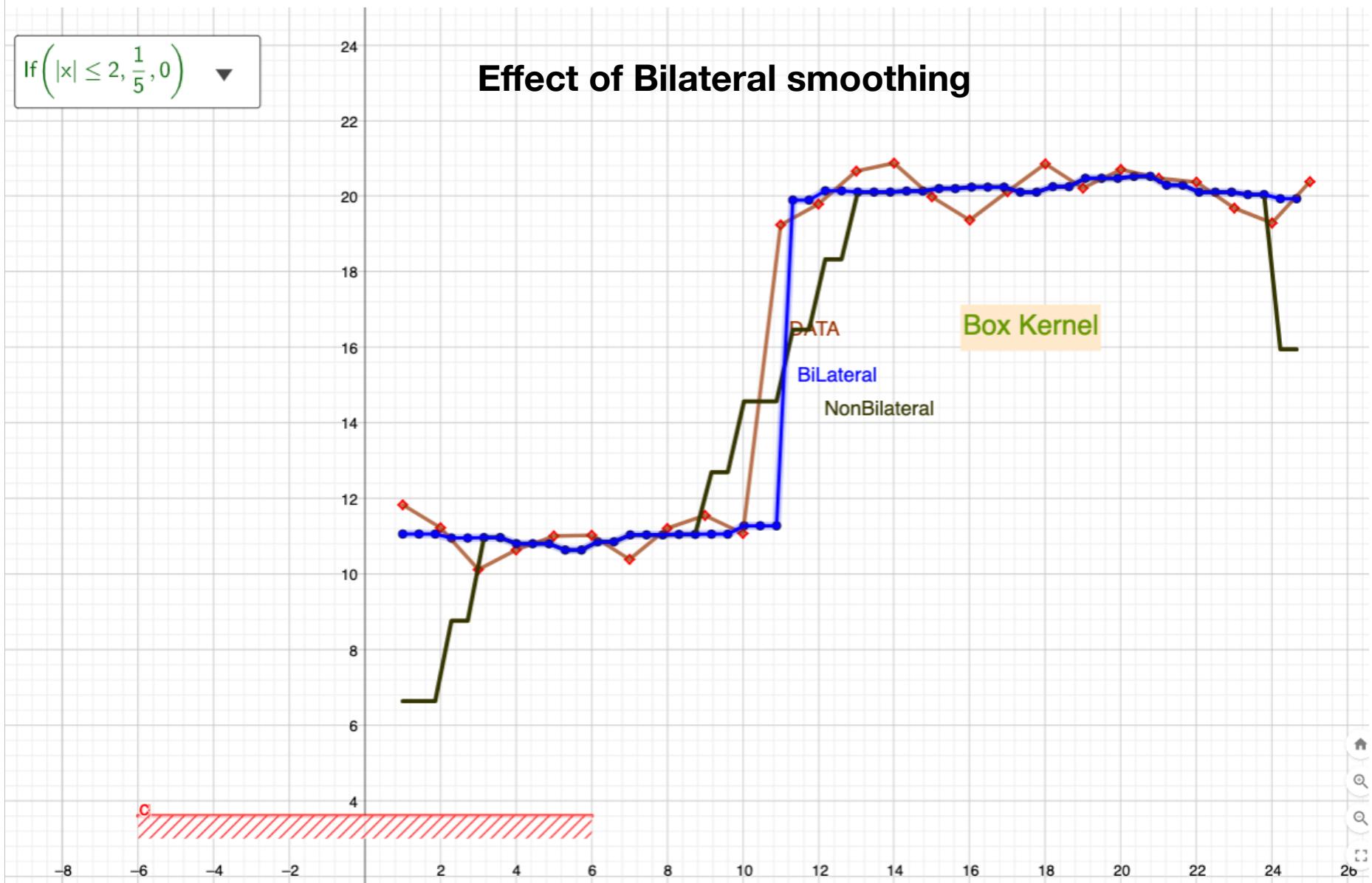
and set  $s(i) = \frac{s_0}{w_0}$

**Example**  $w(0) = w(1) = w(-1) = \frac{1}{3}$

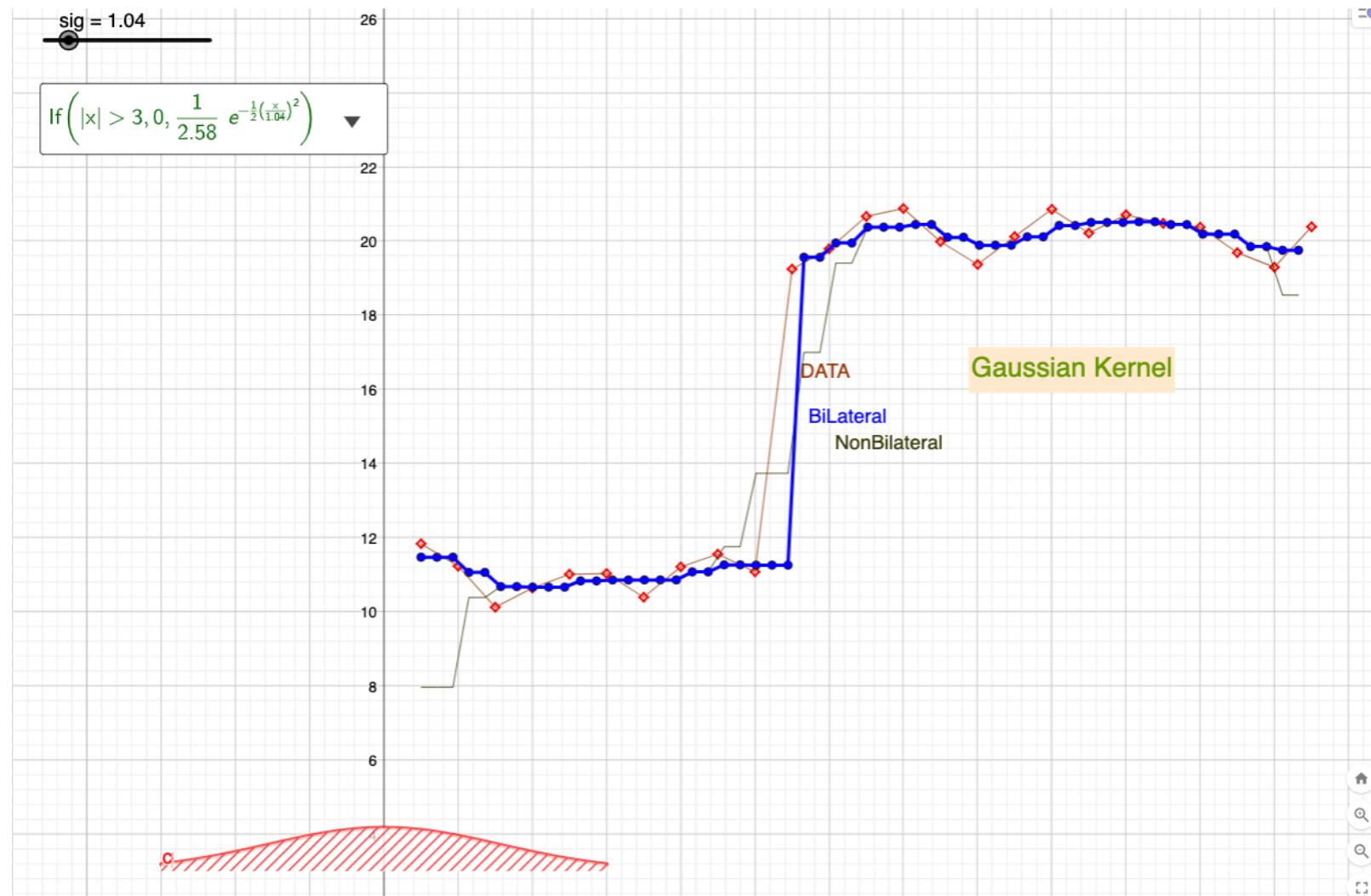


$$\text{If } (|x| \leq 2, \frac{1}{5}, 0)$$

## Effect of Bilateral smoothing

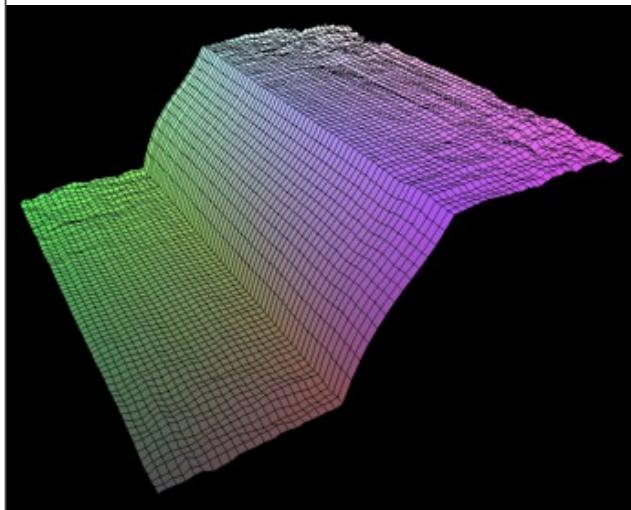


## Effect of Bilateral smoothing

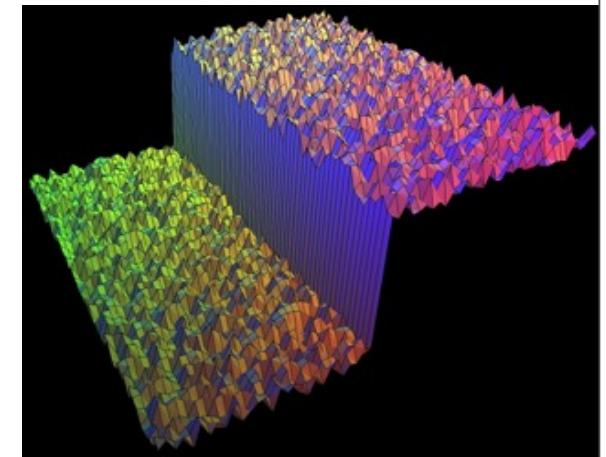
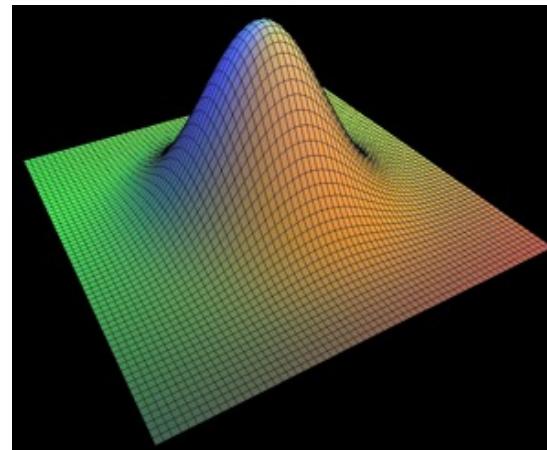
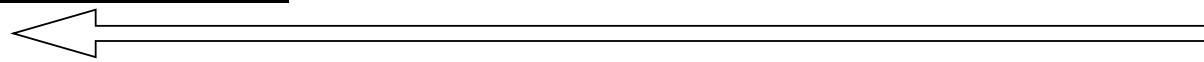


# Start with Gaussian filtering

- Here, input is a step function + noise

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

output

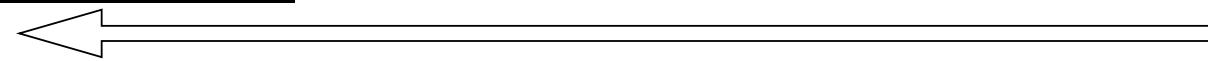
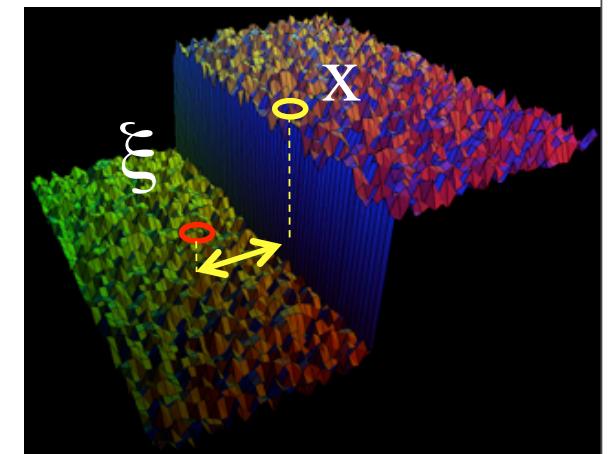
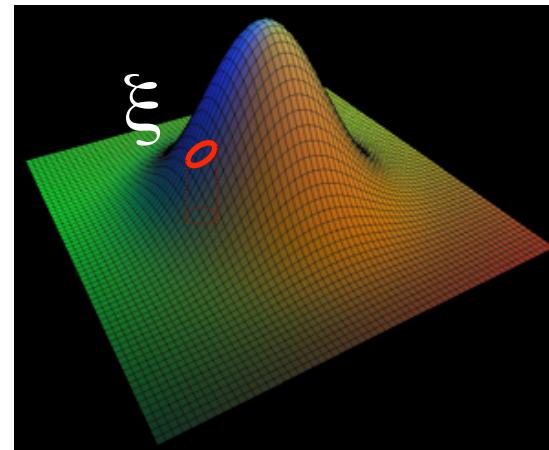
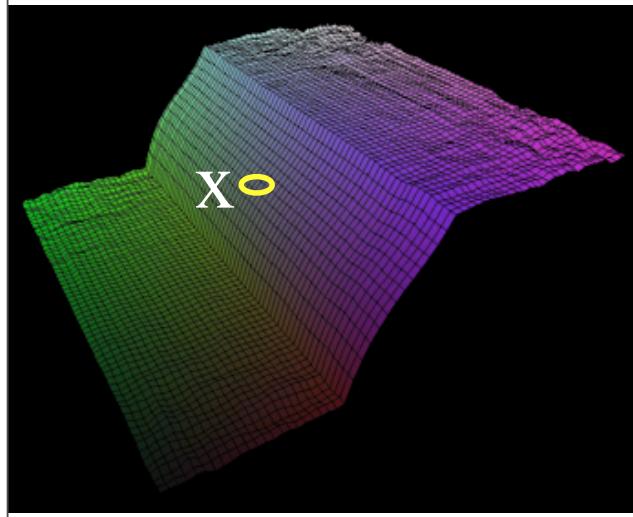


input

# Gaussian filter as weighted average

- Weight of  $\xi$  depends on distance to  $x$

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



# The importance of convex combinations

**weight function**

$$J(\mathbf{x}) = \sum \underbrace{f(\mathbf{x}, \xi)}_{I(\xi)}$$

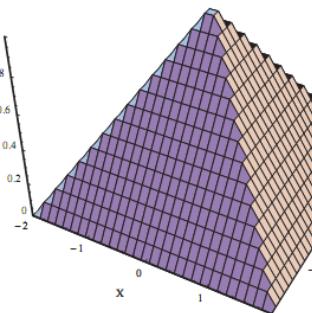
**Intensity**

x is the point where we need the answer  
 ξ is nearby point

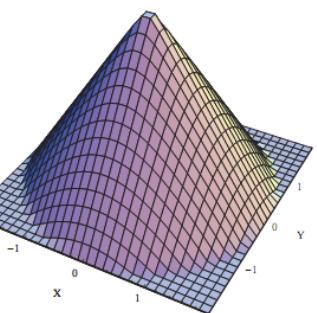
When we smooth, or interpolate we usually use weighted average.

$$G(\mathbf{x}, \xi.x, \xi.y) = \frac{1}{2\pi\sigma} e^{-\frac{1}{2\sigma^2}(x-\xi.x)^2 + (y-\xi.y)^2}$$

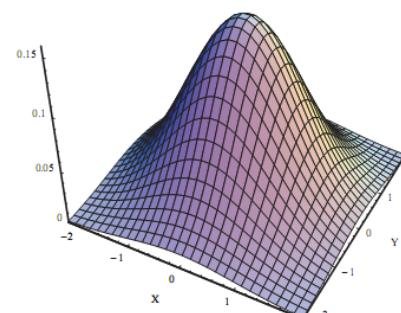
Which functions could  $f(\mathbf{x}, \xi)$  be -



(a) Pyramid.



(b) Cone.



(c) Gaussian.

$$f(\mathbf{x}, \xi) = \max \left\{ 0, \frac{3}{\alpha^2} - \frac{3}{\alpha^3} \max(|x - \xi.x|, |y - \xi.y|) \right\}$$

$\alpha$  is the **width** of the base of the pyramid. So in Fig(a),  $\alpha=4$

We will try to make sure that sum of weights =1 (this is called **convex combination**)

See example of bilinear interpolation on the whiteboard

# The problem of edges

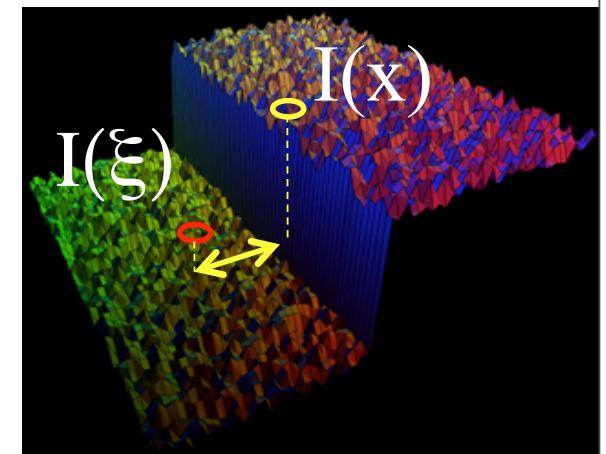
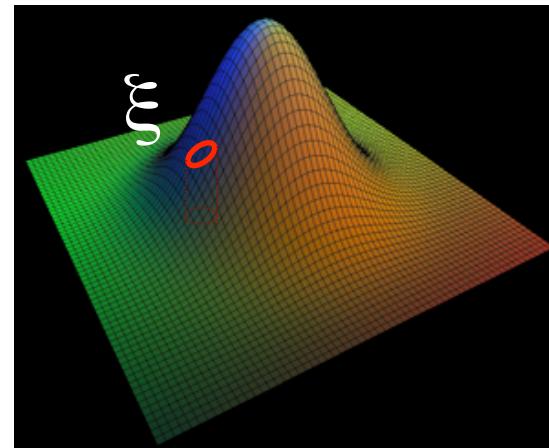
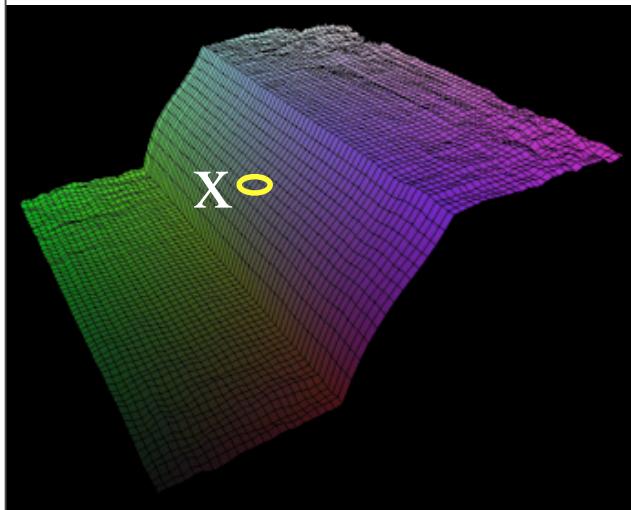
- Here,  $I(\xi)$  “pollutes” our estimate  $J(x)$
- It is too different

- To correct it, we will have to change the averaging.

- Remember that during the smoothing, we will first sum  $\sum I(\xi)$  for all point  $\xi$  near  $x$ .

- To resolve the **halo problem**, We will avoid summing  $I(\xi)$  if  $I(\xi)$  is very far from  $I(x)$

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



←

input

# Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- Penalty **g** on the intensity difference

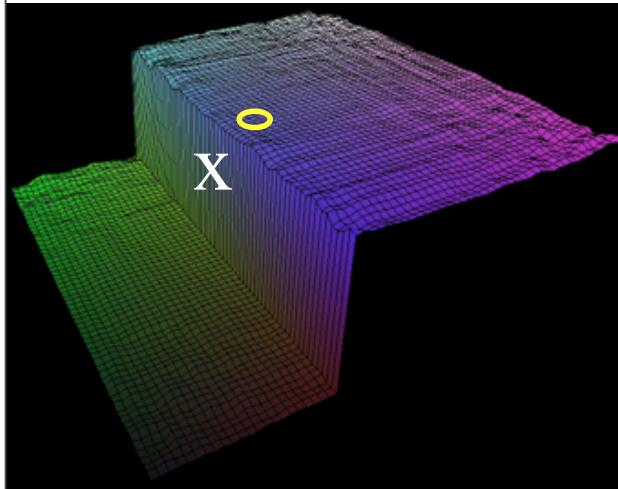
Remember that the sum of weights must be 1.

What to do if we skip some terms ?

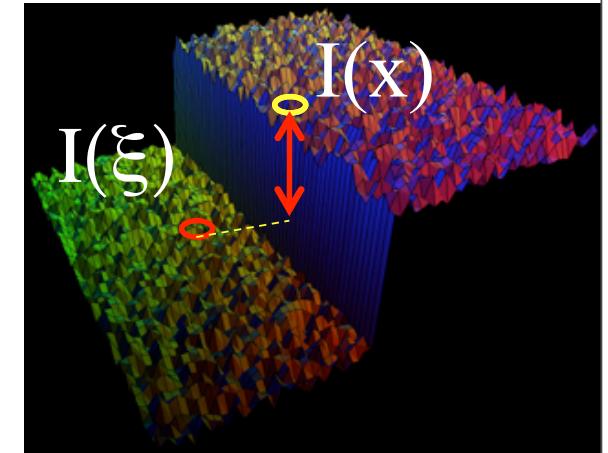
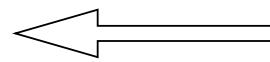
We will divide the total sum by  $k(x)$

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

$g(|a - b|) = 1$  if  $a$  is close to  $b$ , and zero otherwise



output



input

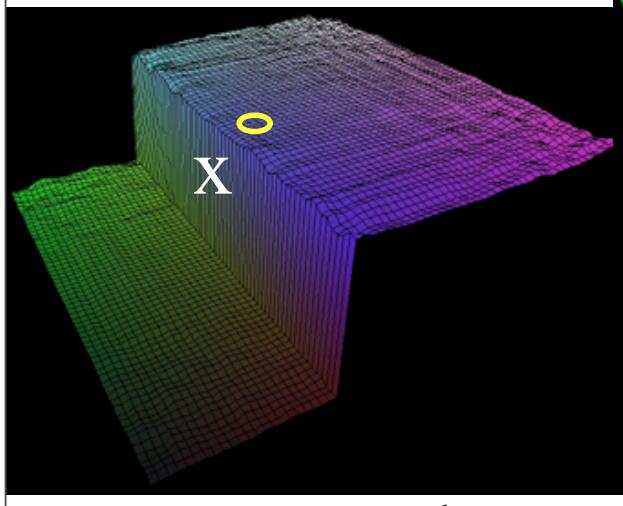
# Bilateral filtering

[Tomasi and Manduchi 1998]

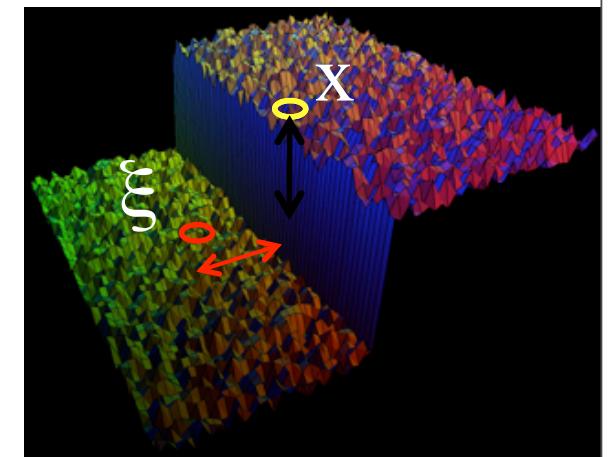
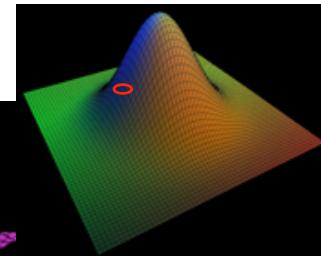
- Spatial Gaussian f

Remember that the sum of weights must be 1.  
 What to do if we skip some terms ?  
 We will divide the total sum by  $k(x)$

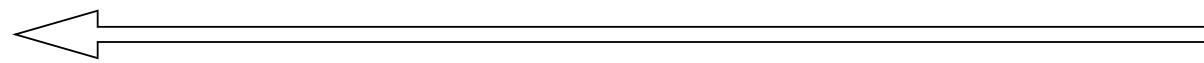
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi) g(I(\xi) - I(x)) \cdot I(\xi)$$



output



input

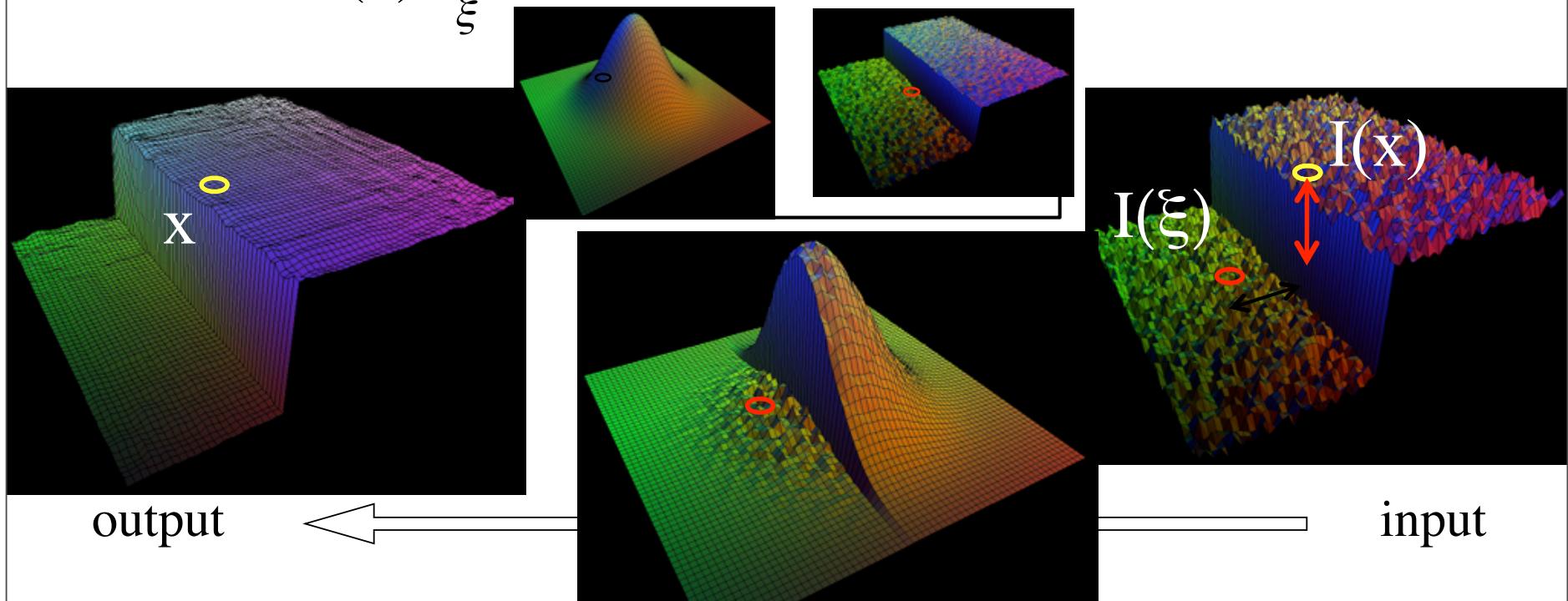


# Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian  $f$
- Gaussian  $g$  on the intensity difference

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

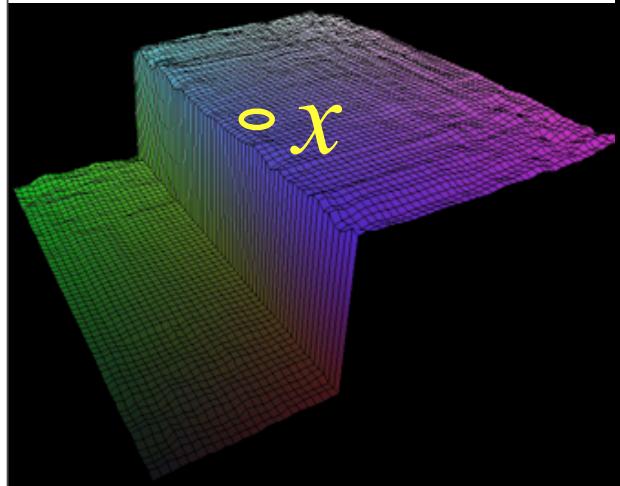


# Normalization factor

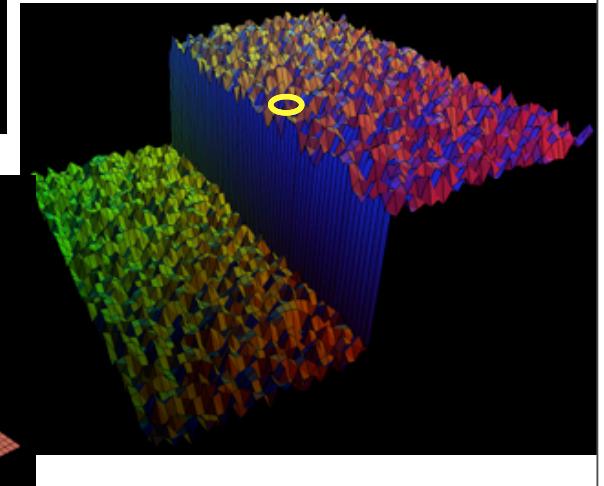
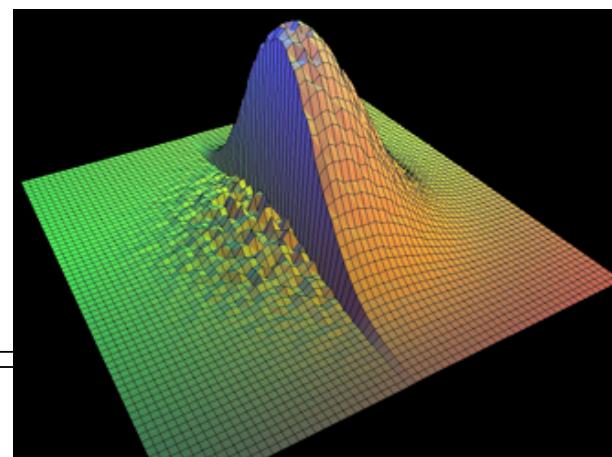
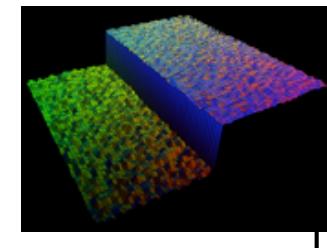
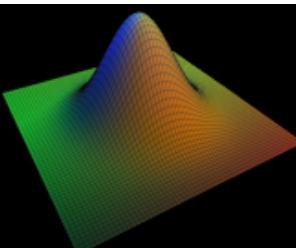
[Tomasi and Manduchi 1998]

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

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



output



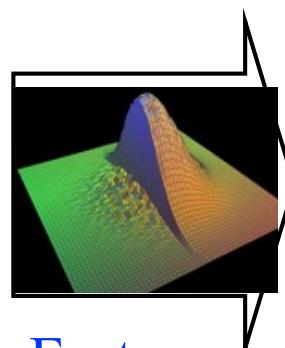
input

# Log domain

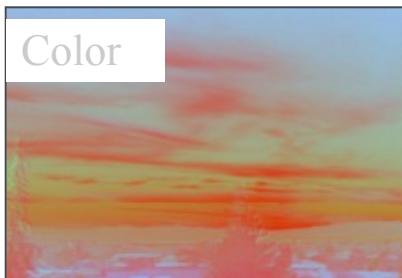
---

- Very important to work in the log domain
- Recall: humans are sensitive to multiplicative contrast
- With log domain, our notion of “strong edge” always corresponds to the same contrast

# Recap



Fast  
Bilateral  
Filter  
**IN LOG**



detail=  
input log - large scale

**Reduce  
contrast**

**Preserve!**

