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

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

- $\boldsymbol{\alpha}$ is a real-valued scaling factor known as **gain**

- $\boldsymbol{\beta}$ is a real-valued scaling factor known as **bias**

Why Use Both α, β?

- Consider two rescaled source samples of S rescaled to S'.

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

- Calculate the **contrast** (the absolute difference) between the source and destination, called **Δ**S and **Δ**S'.

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

- Now consider the relative change in contrast between the source and destination.

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

Why Use Both α, β?

- The relative change in contrast can be simplified as

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

- Thus, gain (**α**) controls the change in contrast.
- Whereas bias (**β**) *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

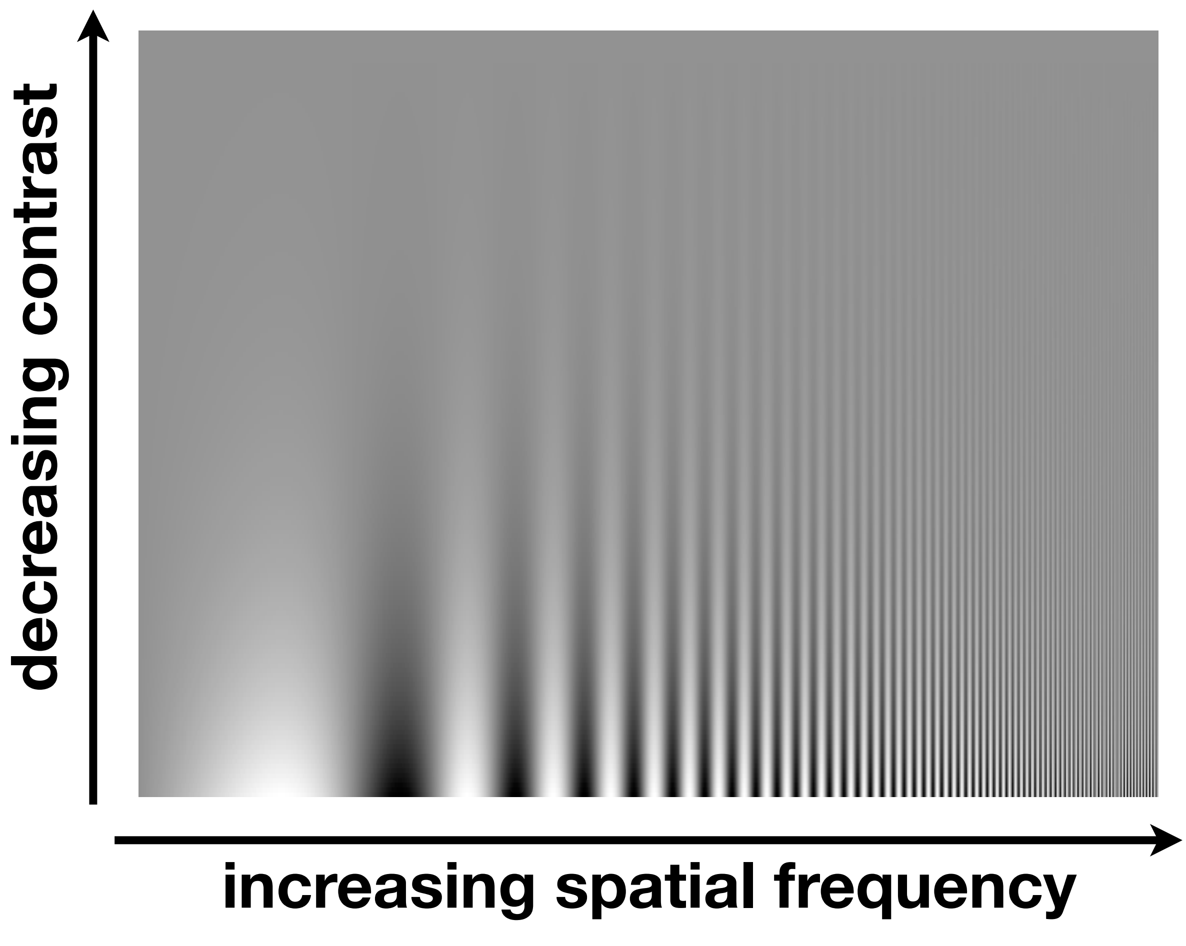
We want to be able to notice small details in the image (da)

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 α is much more important to the viewer that the bias β

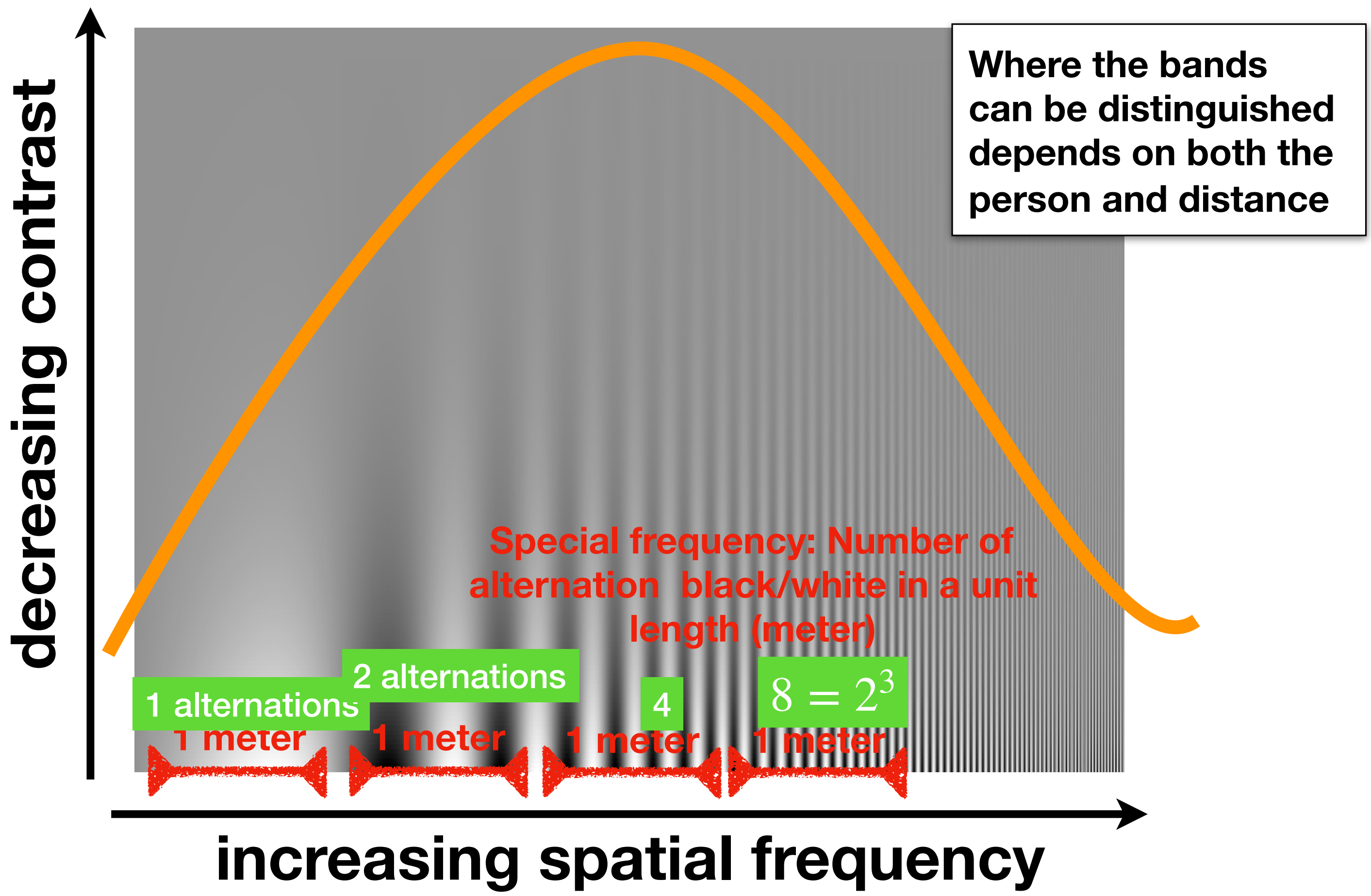
Contrast Sensitivity Function

Campbell-Robson Chart



Contrast Sensitivity Function

Campbell-Robson Chart



Contrast Sensitivities Vary by Channel

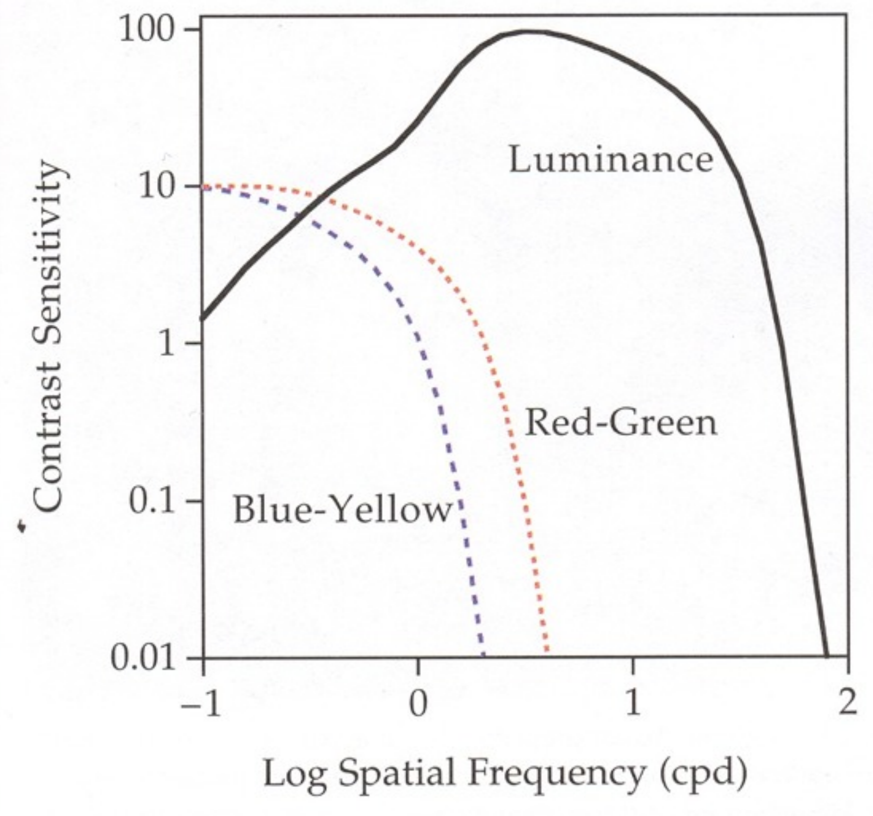


Figure 1-18. Spatial contrast sensitivity functions for luminance and chromatic contrast.

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.

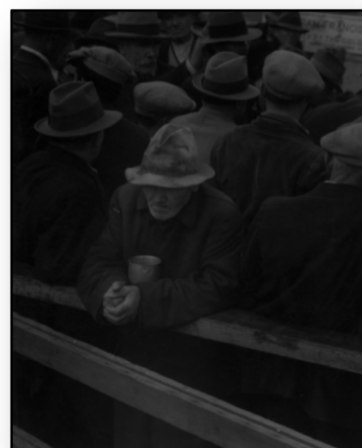
$$clamp(x, min, max) = \begin{cases} \text{min} & \text{if } \lfloor x \rfloor \leq \text{min}, \\ \text{max} & \text{if } \lfloor x \rfloor \geq \text{max}, \\ \lfloor x \rfloor & \text{otherwise.} \end{cases}$$

Rescaling Examples

gain = 1, bias = 55

gain = 1, bias = -55

gain = 2, bias=0

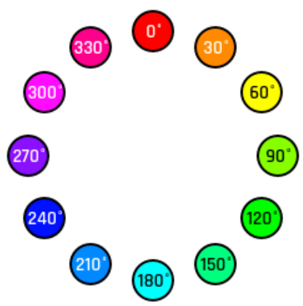


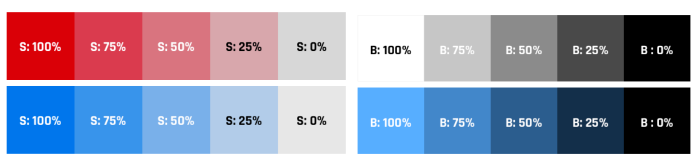
gain = .5, bias=0

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.

Credit “Learn Ui Design Blog”





- 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.

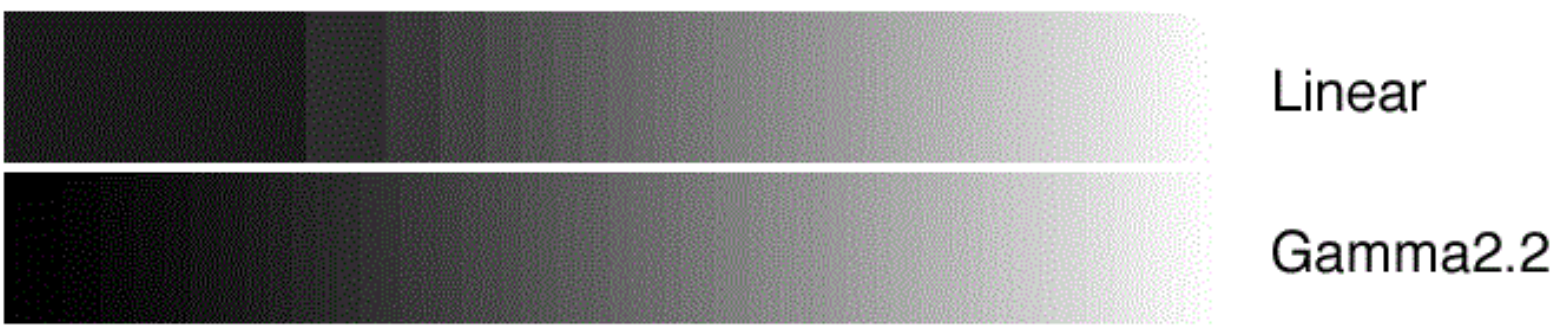
- Consider $I_1 < I_2 < I_3$, for the step between I_1 and I_2 to look like the step from I_2 to I_3, it must be that:

$$I_2 / I_1 = I_3 / I_2$$

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

Perceived (I_p) vs. Actual (I_a) Intensity

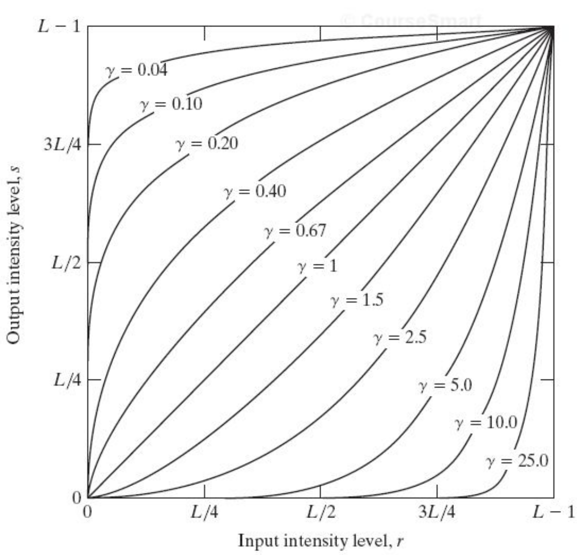
- Perceived light actually behaves like $I_p = (I_a)^\gamma$



http://www.anyhere.com/gward/hdrenc/

Example: Gamma Correction

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



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

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

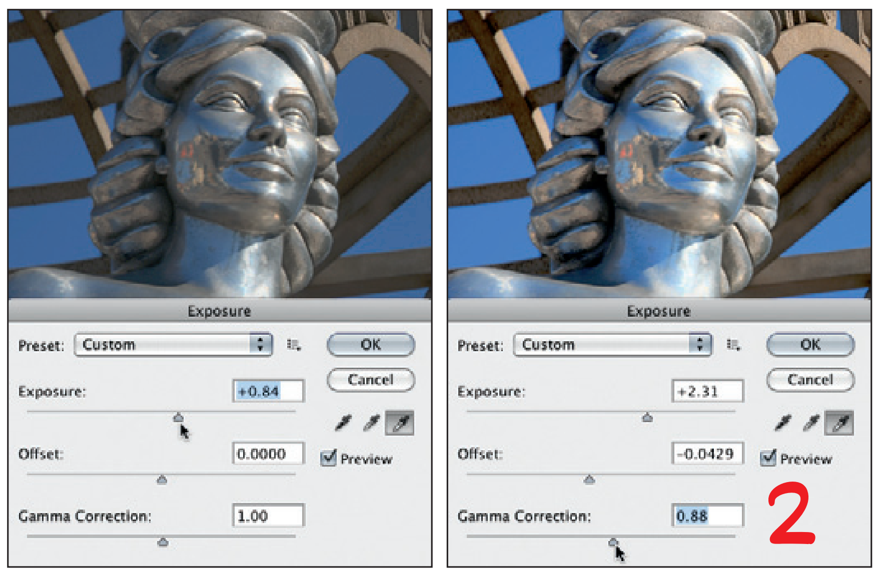
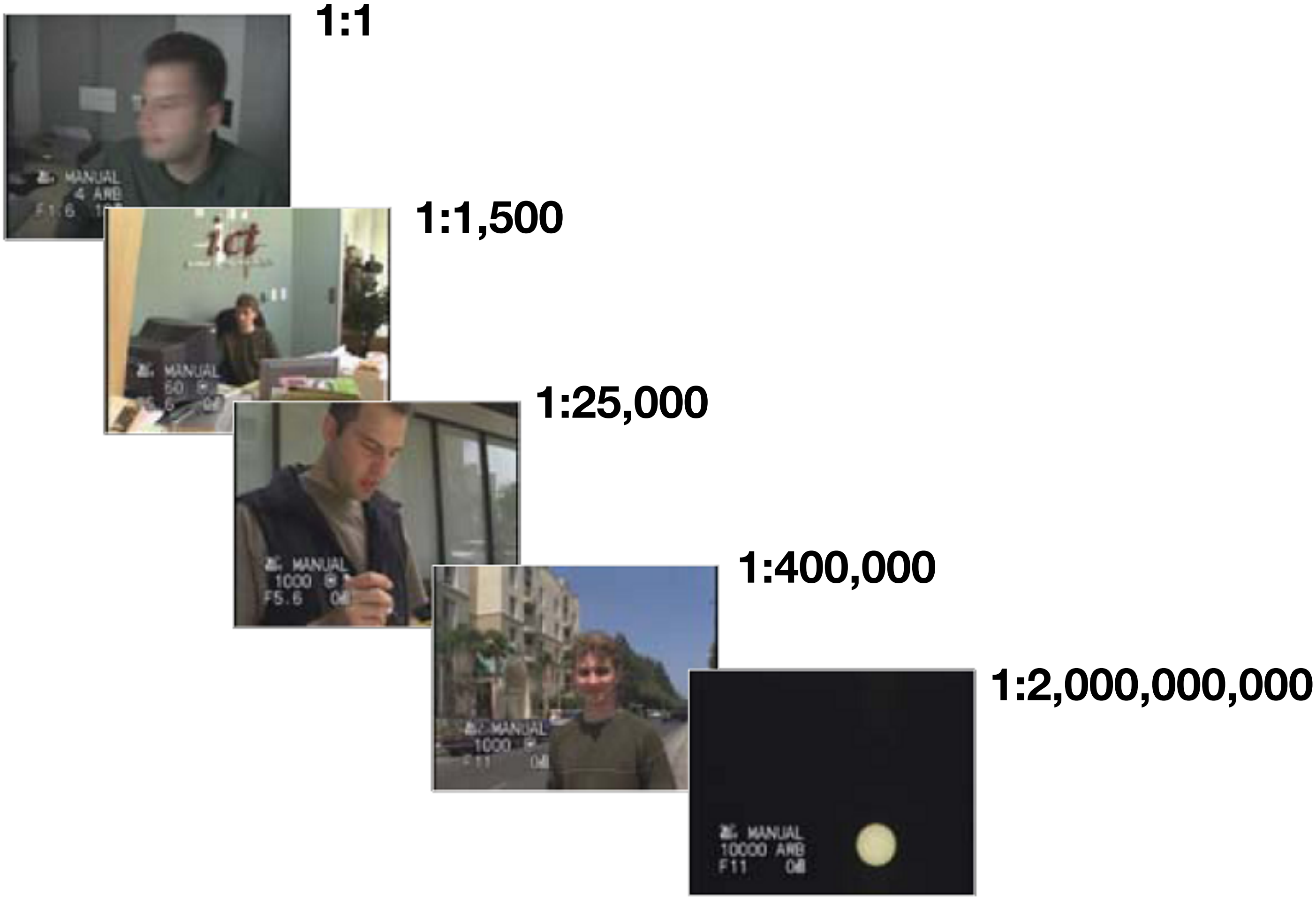


Image of Photoshop from Christian Bloch - The HDRI Handbook 2.0

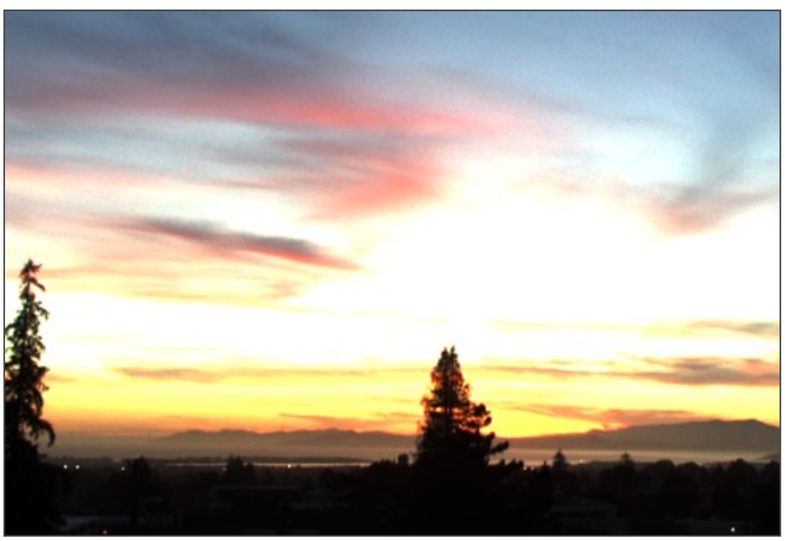
Dynamic Range

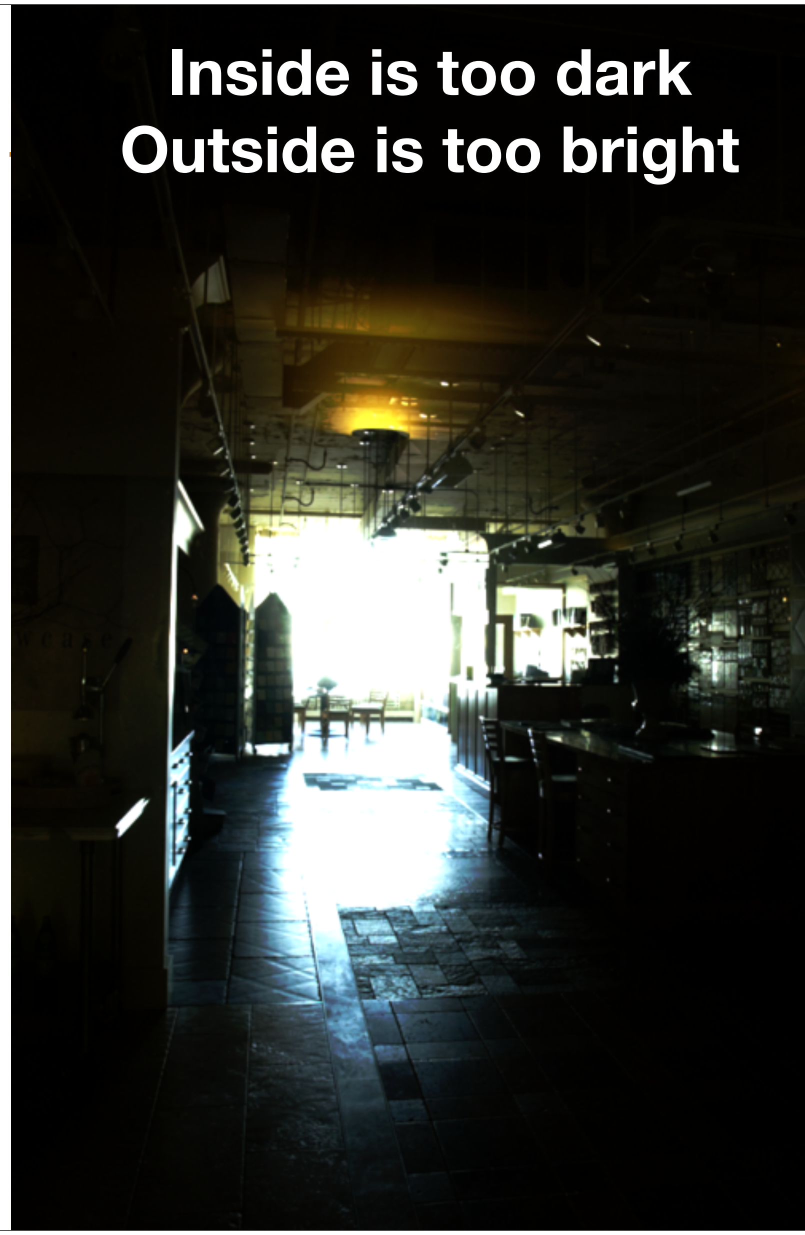
The World is a High Dynamic Range (HDR)



Examples

Sun overexposed
Foreground too dark



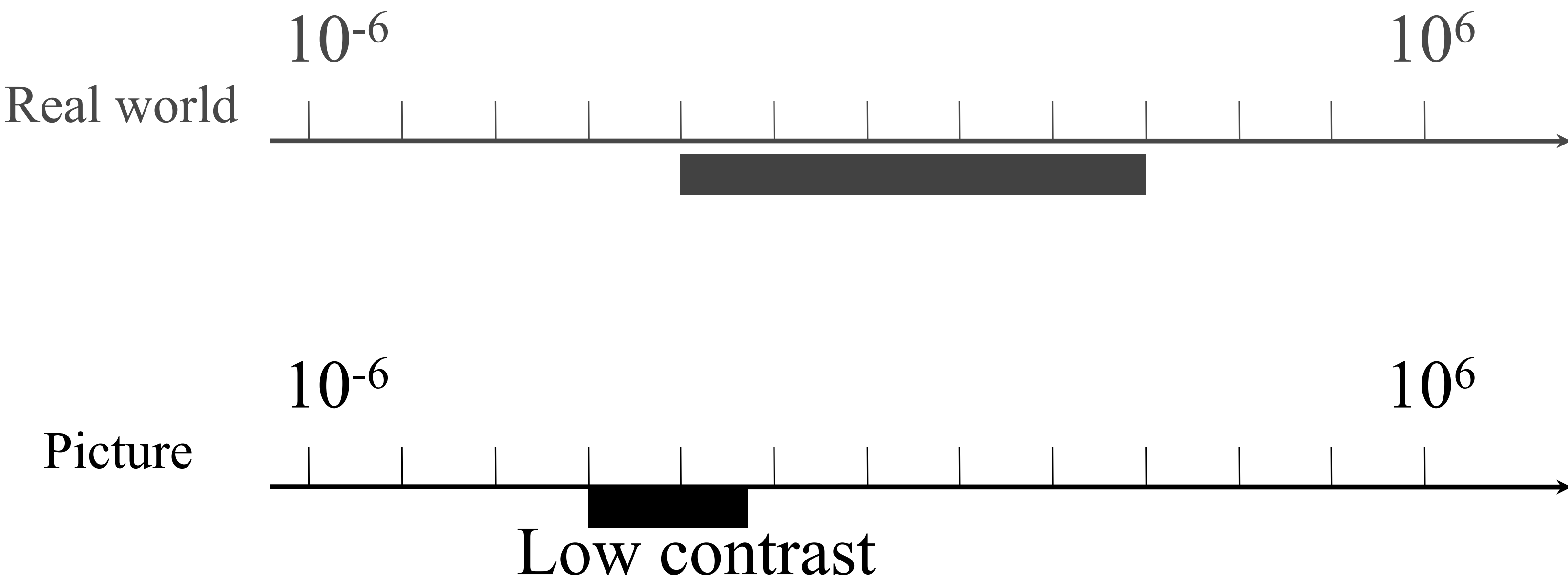


Dynamic Range in Displays?

- Range of pure black vs. pure white?

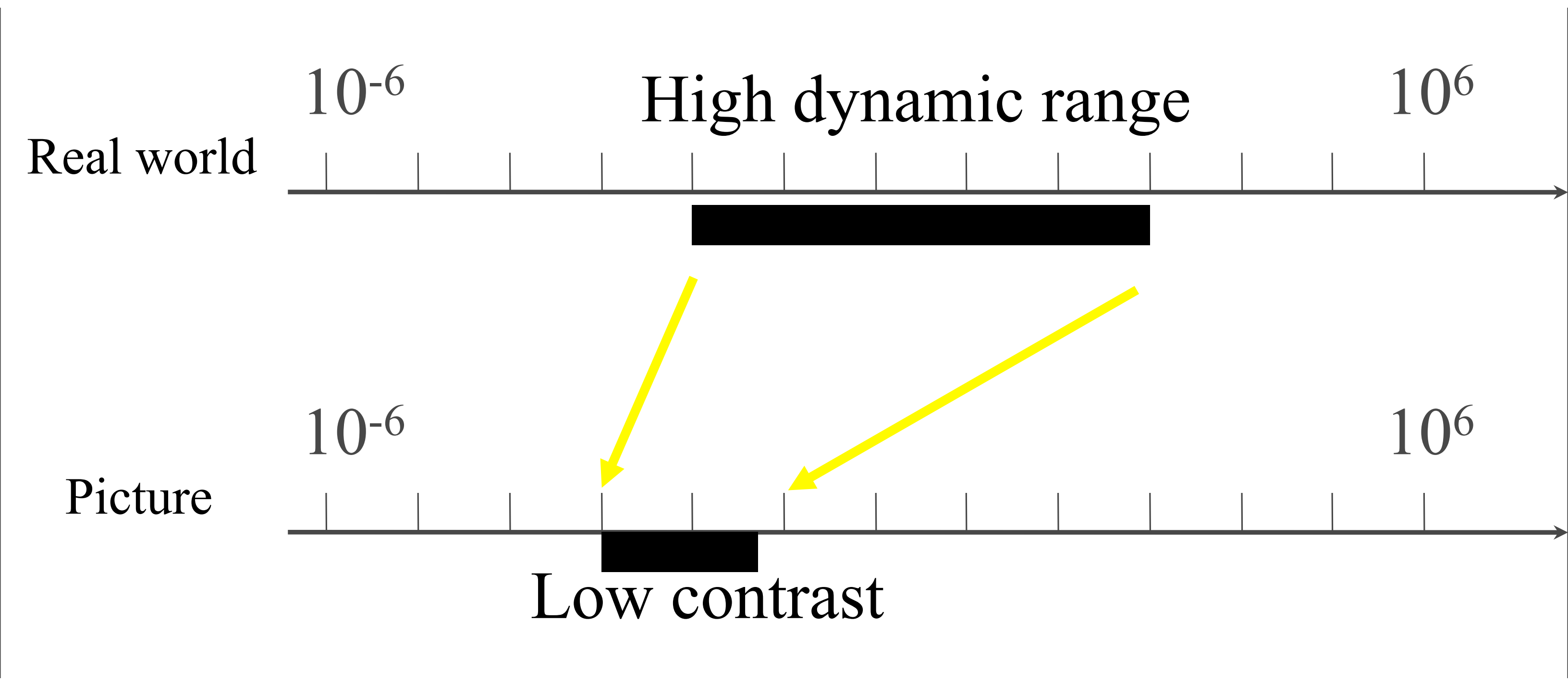
Dynamic Range in Displays?

- **Typically 1: 20 or 1:50**
  - Black is ~ 50x darker than white
- **Max 1:500**



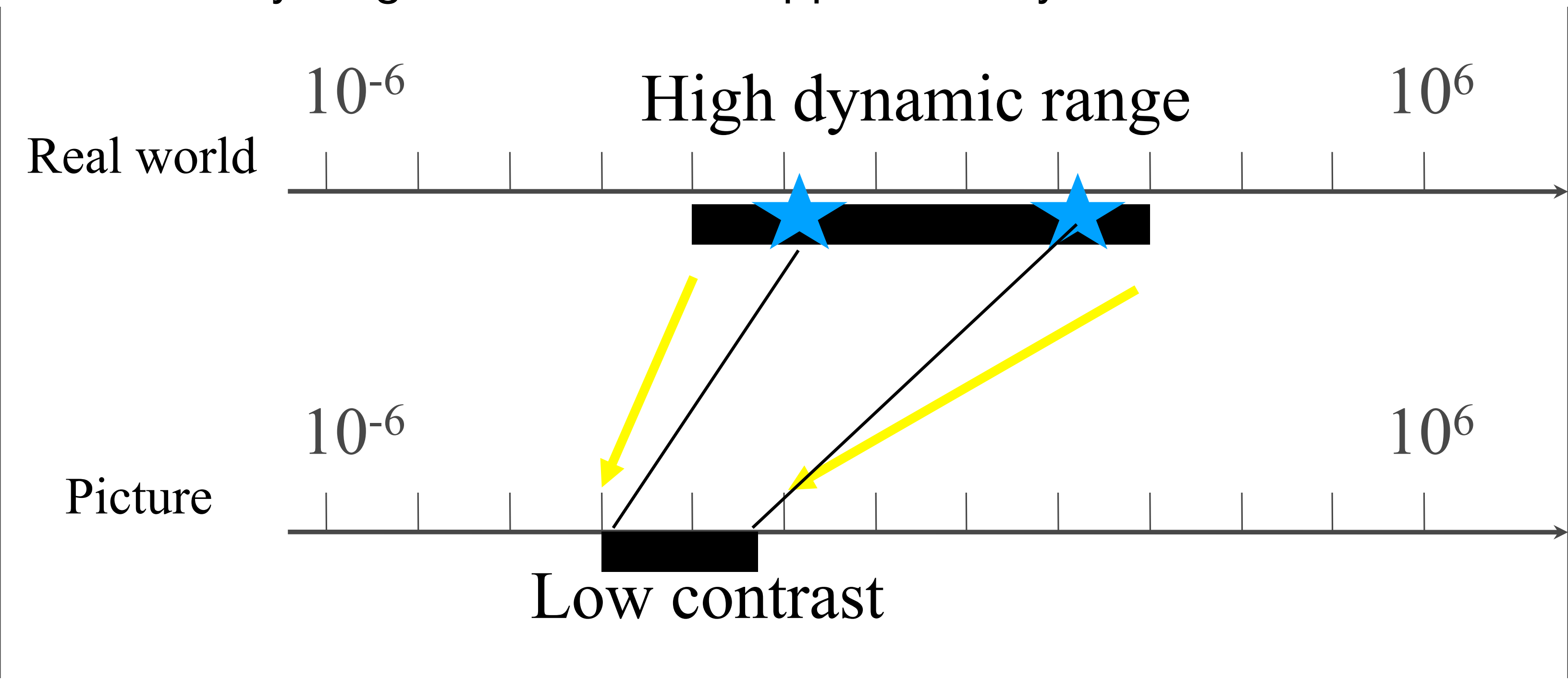
Problem: Displaying the Information

- 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.

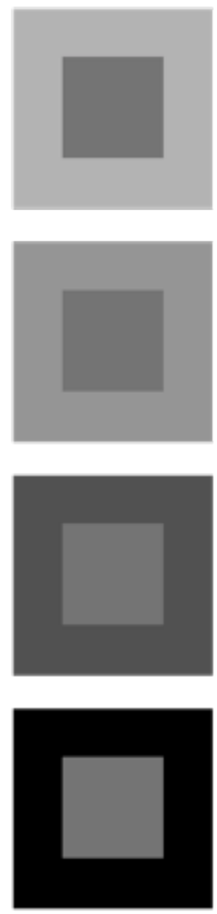


Rescaling Color Images

- 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.

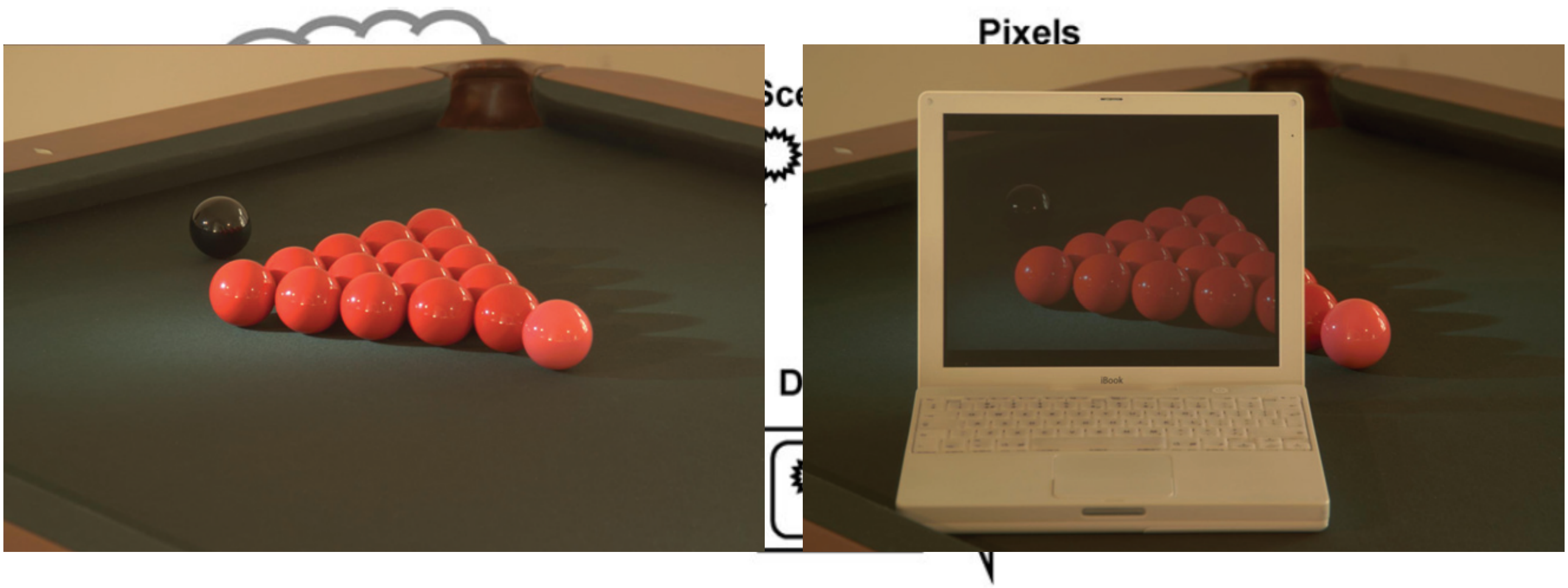
Eyes and Dynamic Range

- 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.

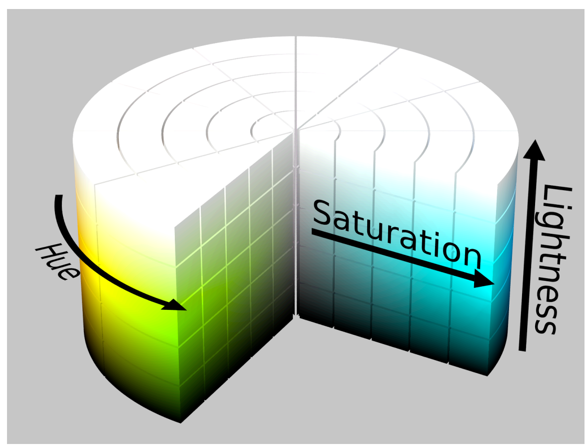
Tone Mapping

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

$$L(p_1, out) = L(p_1, \textbf{Source})/10 + 0.2$$
$$L(p_2, out) = L(p_1, \textbf{Source})/3 + 0.5$$



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. $L_w(p) = \frac{1}{61}(20.0R(p) + 40G(p) + B(p))$
  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** α, β
  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 operaions on L it is actually recommended to perform them on $\log(L)$ domain. Before using the values That is, use 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

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

What if rescaling and gamma-correction are not sufficient

- A useful collections 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 is slow: Big regions are “high”, big regions are “low” (low spatial frequency)
- Note that this does not mean the 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

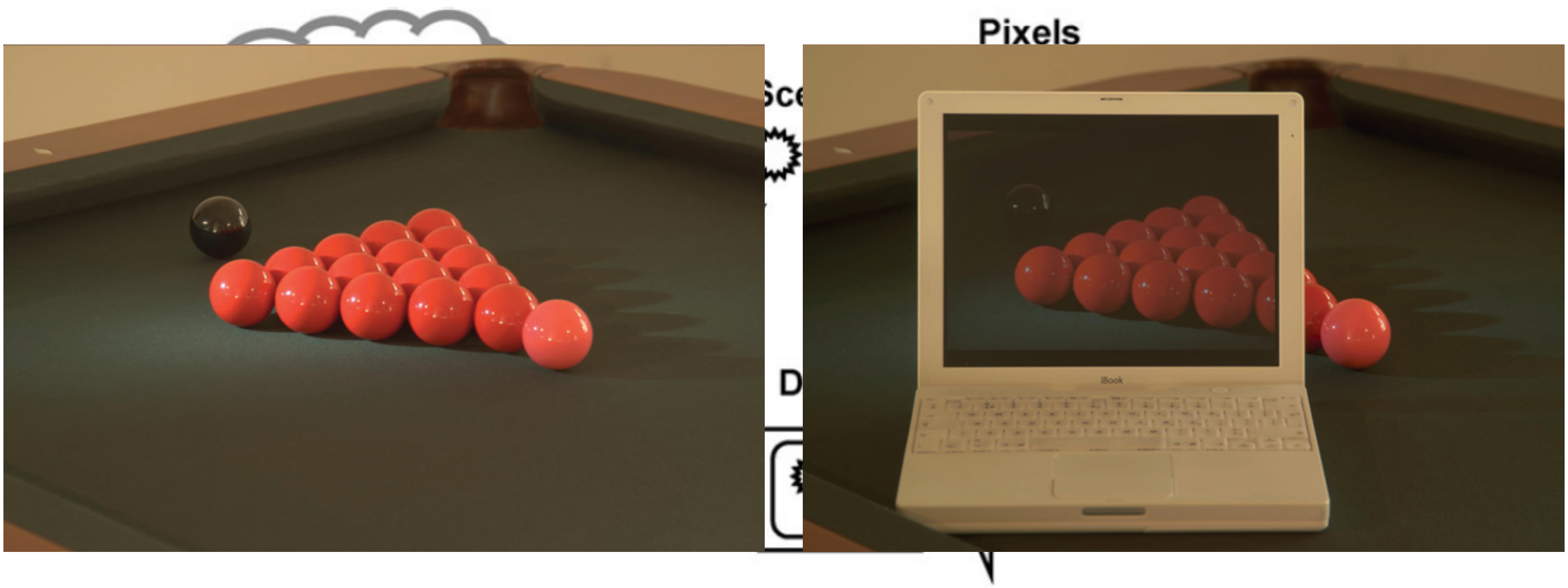
From Durand and Dorsey. No single global exposure can preserve both the colors of the sky and the details of the landscape, as shown on the rightmost images.

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.

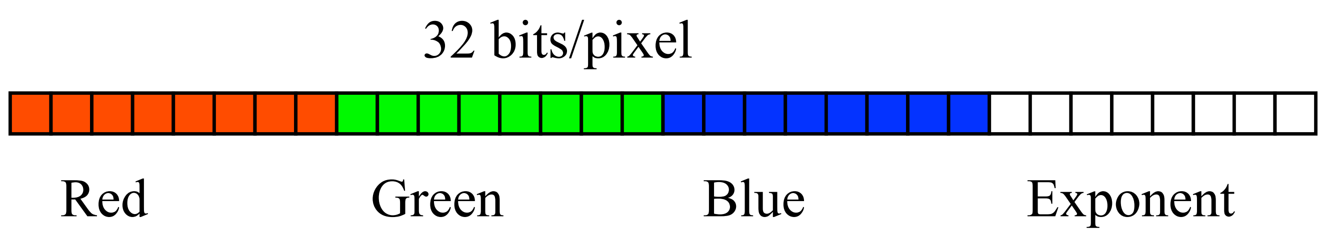
Question: But why do we need more than 100 levels of intensity (luminance) if in the input file we only have 256 values of intensities (RGB) ?

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)



(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

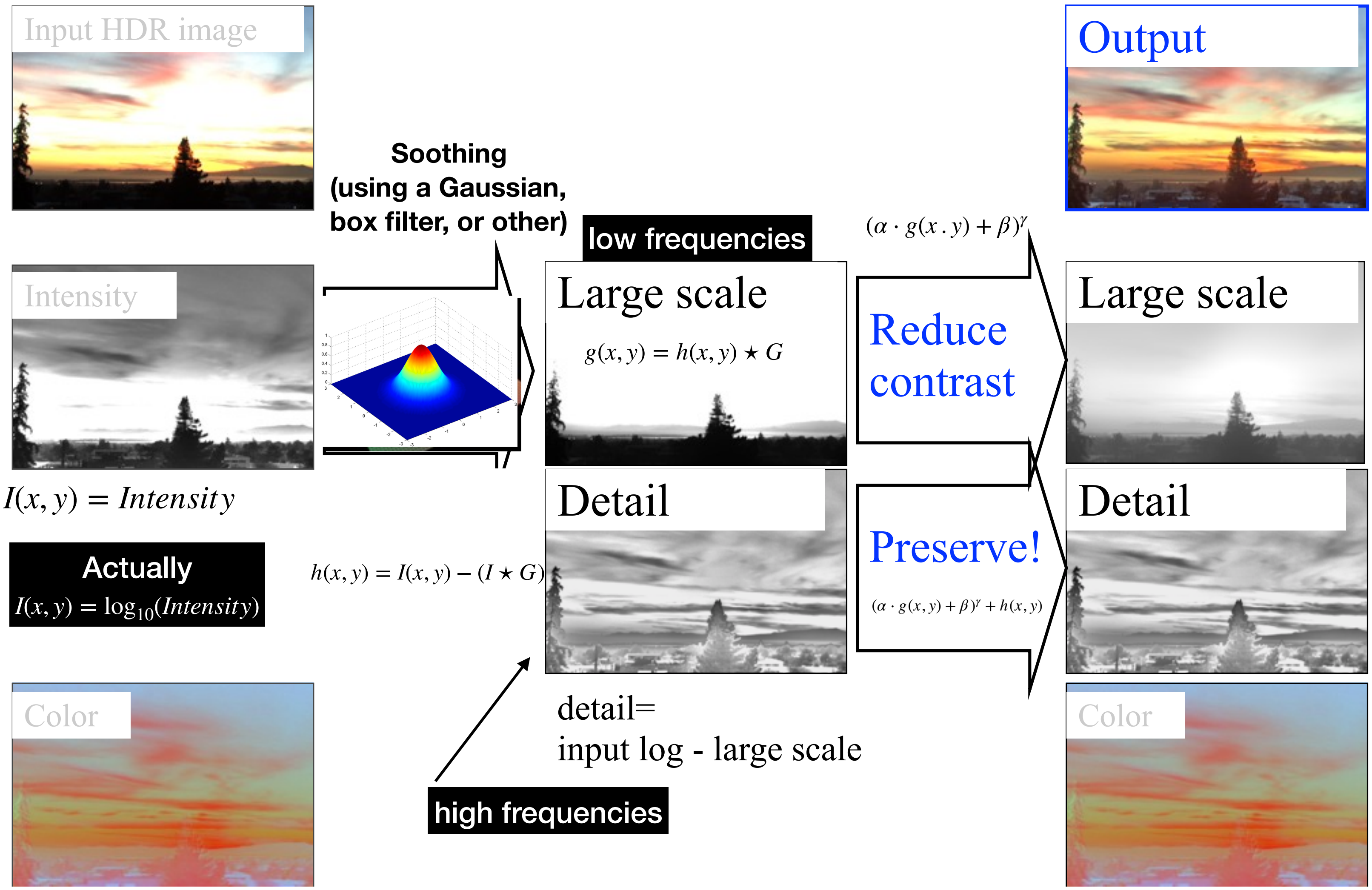
Ward, Greg. "Real Pixels," in Graphics Gems IV, edited by James Arvo, Academic Press, 1994

Non-local tone mapping

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

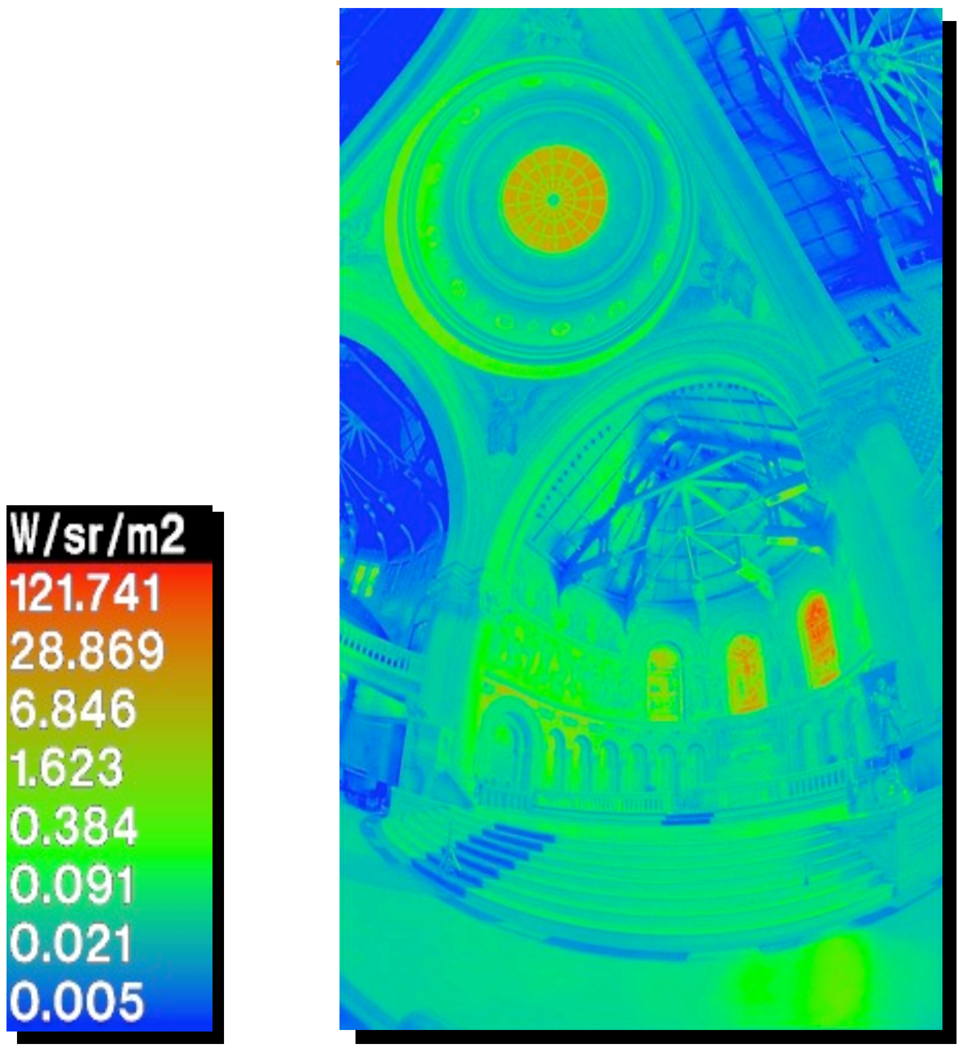
But also depends on luminosity of pixels elsewhere

Recap

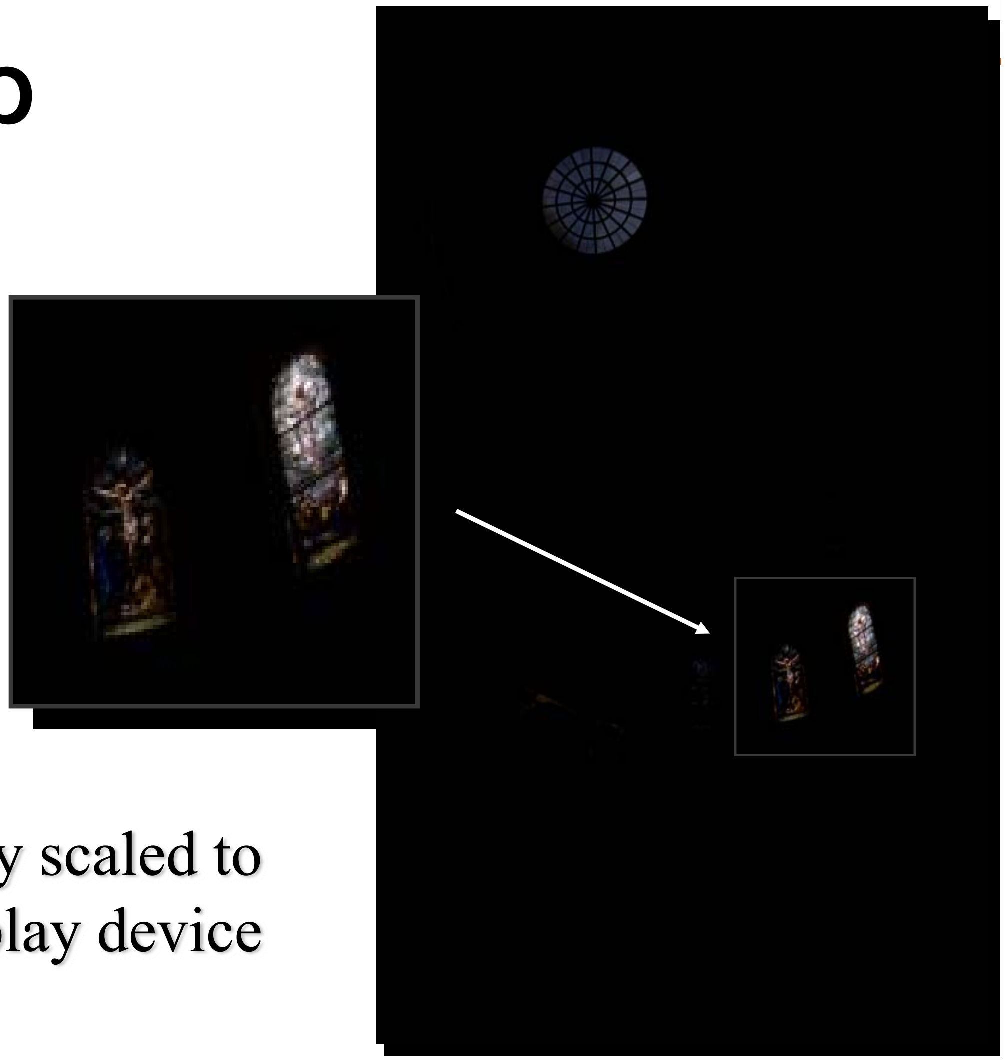


Input HDR image
Output
Soothing
(using a Gaussian,
box filter, or other)
low frequencies
$(\alpha \cdot g(x . y) + \beta)^{\gamma}$
Intensity
Large scale
$g(x, y) = h(x, y) \star G$
Reduce
contrast
Large scale
$I(x, y) = Intensity$
Detail
Detail
Preserve!
Actually
$I(x, y) = \log_{10}(Intensity)$
$h(x, y) = I(x, y) - (I \star G)$
$(\alpha \cdot g(x, y) + \beta)^{\gamma} + h(x, y)$
Color
Color
detail=
input log - large scale
high frequencies

The Radiance Map

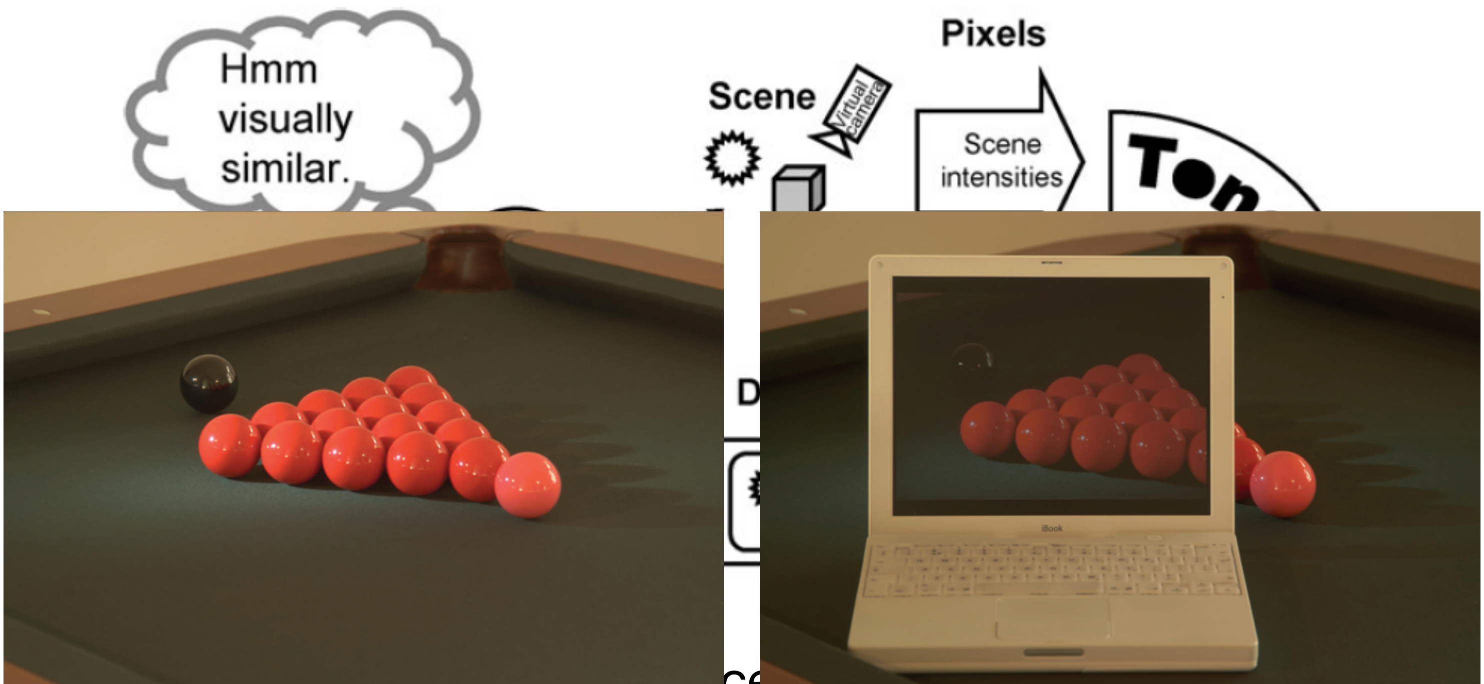


The Radiance Map



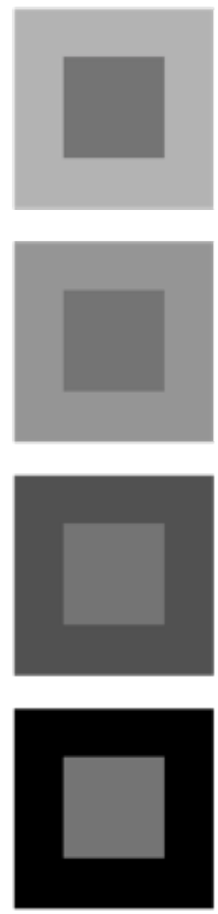
Linearly scaled to display device

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

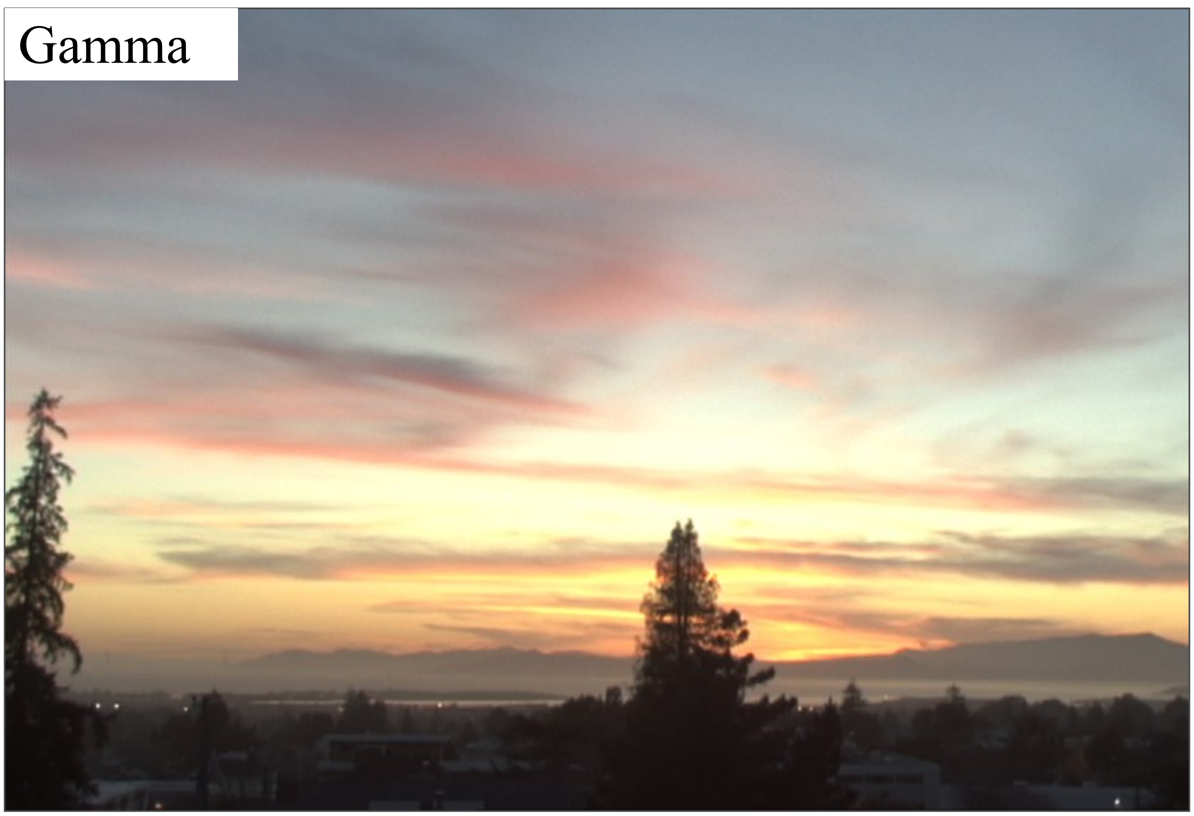


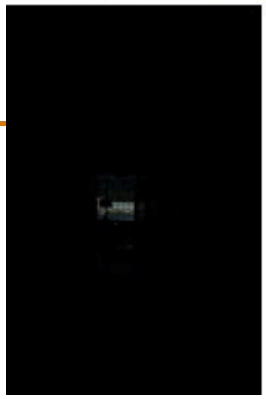
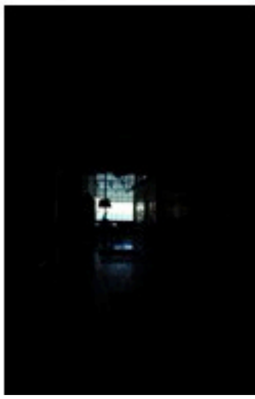
- We're sensitive to change (multiplicative)
  - A ratio of 1:2 is perceived as the same contrast as a ratio of 100 to 200
  - Use the log domain as much as possible
- But, eyes are **not** photometers
  - Dynamic adaptation (very local in retina)
  - Different sensitivity to spatial frequencies

Headlights are ON in both photos

What scale value to use?
How about Gamma compression

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





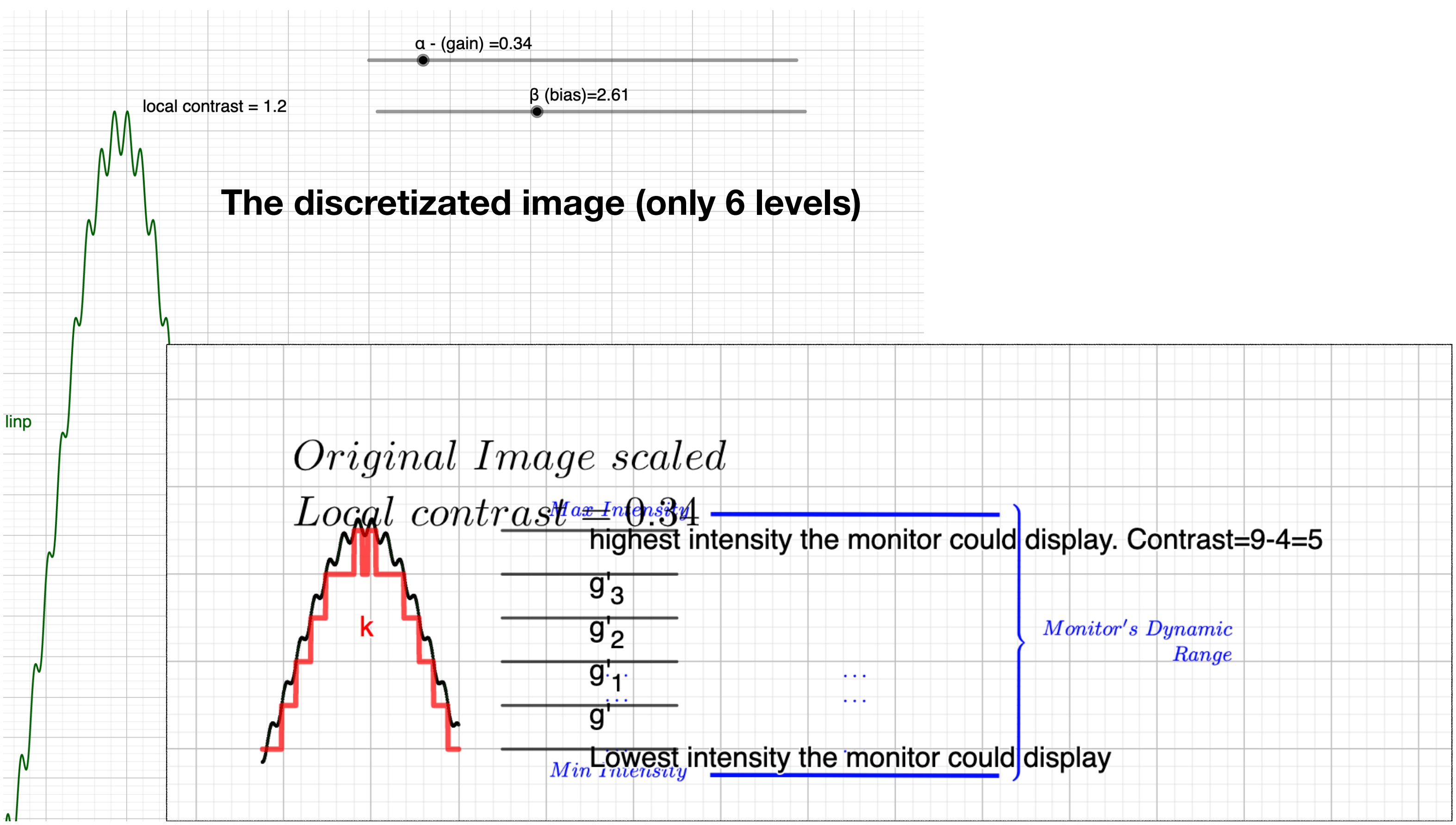
Oppenheim 1968, Chiu et al. 1993

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

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

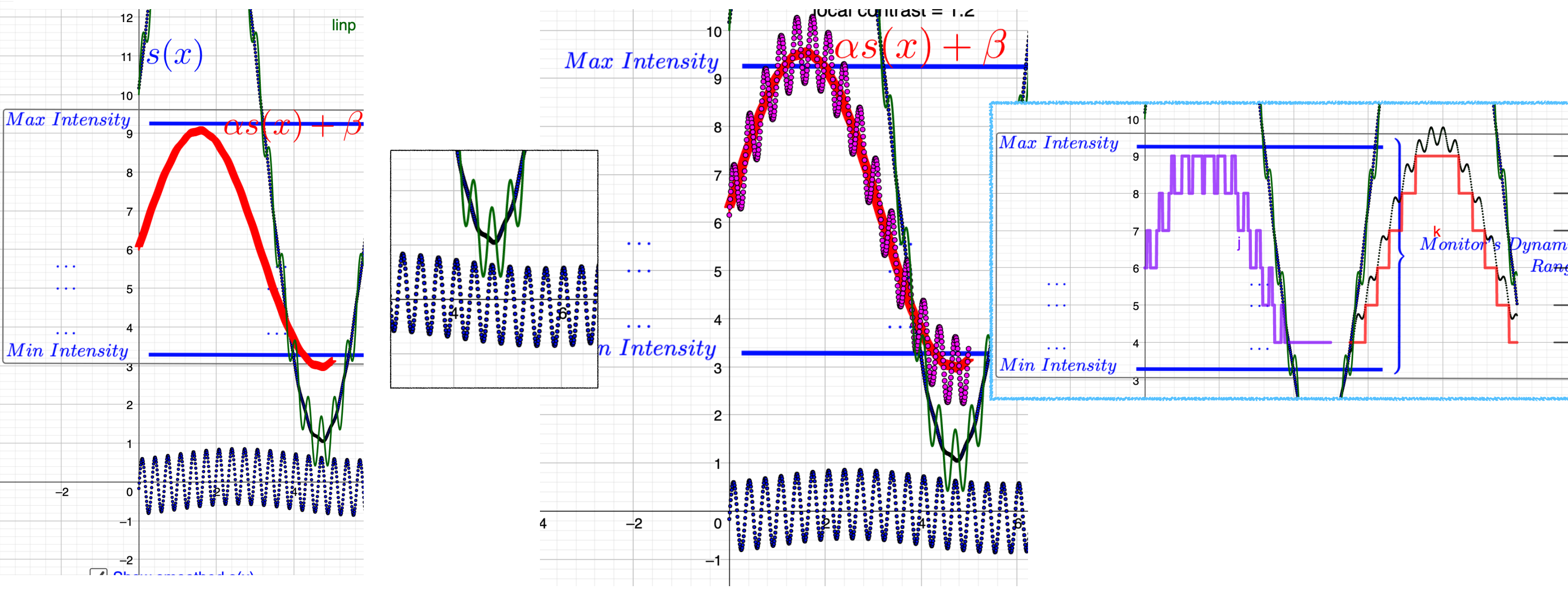
In the GG demo, the input image represents the input image

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



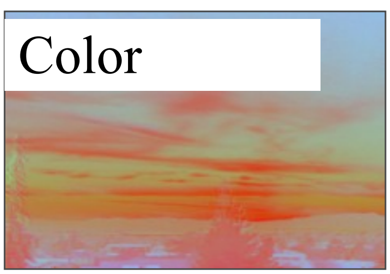
A better idea

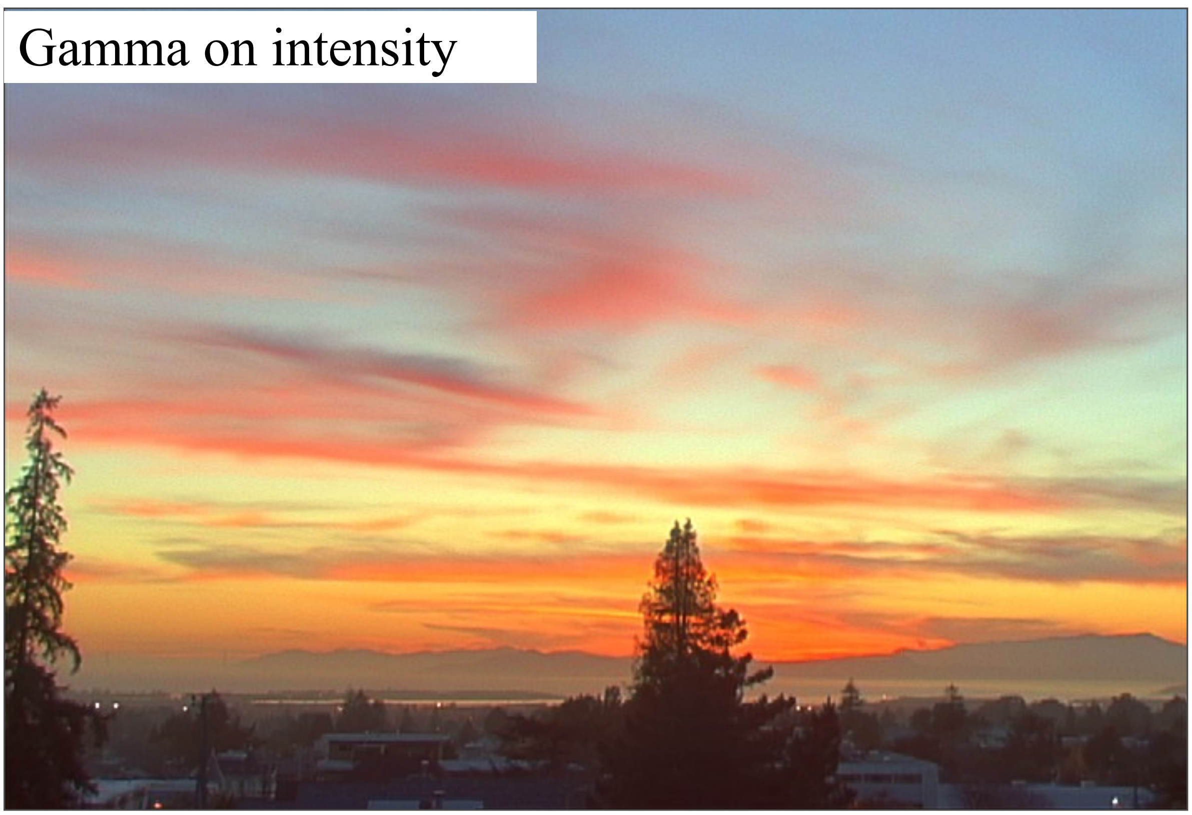
- $f(x) =$ **Brightness level at pixel** x
- Compute $s(x)$ - a low pass filter:
  •For example, compute $s(x) = \left(f(x-1) + f(x) + f(x+1)\right) / 3$ (mean filter)
  •better idea: Use **convolution** with a tent or a gaussian
- in $s(x)$, details that change frequently (high special frequencies)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



Gamma compression on Intensity

- Colors ok, but details in intensity are blurry





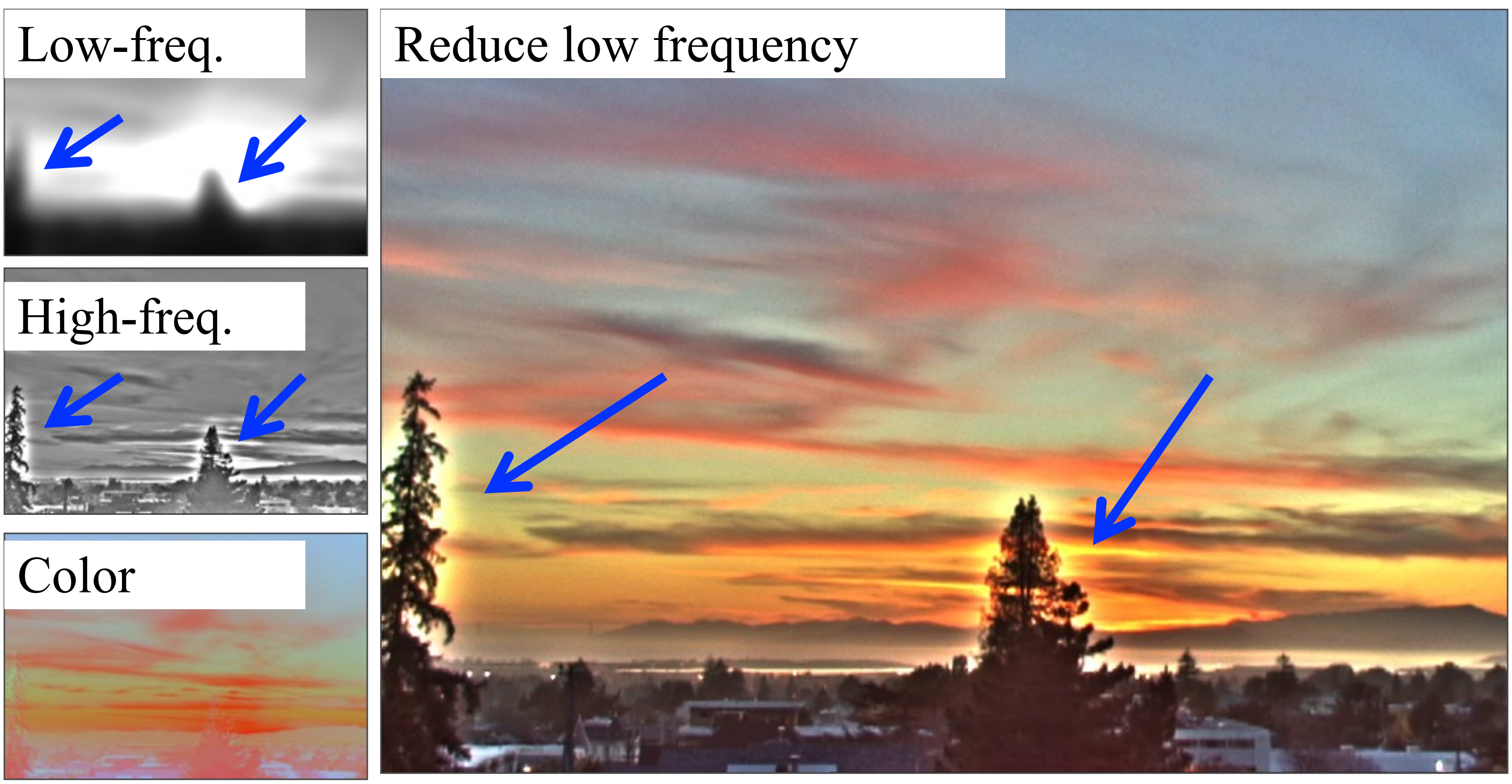
Can we just scale? Maybe!

- 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

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

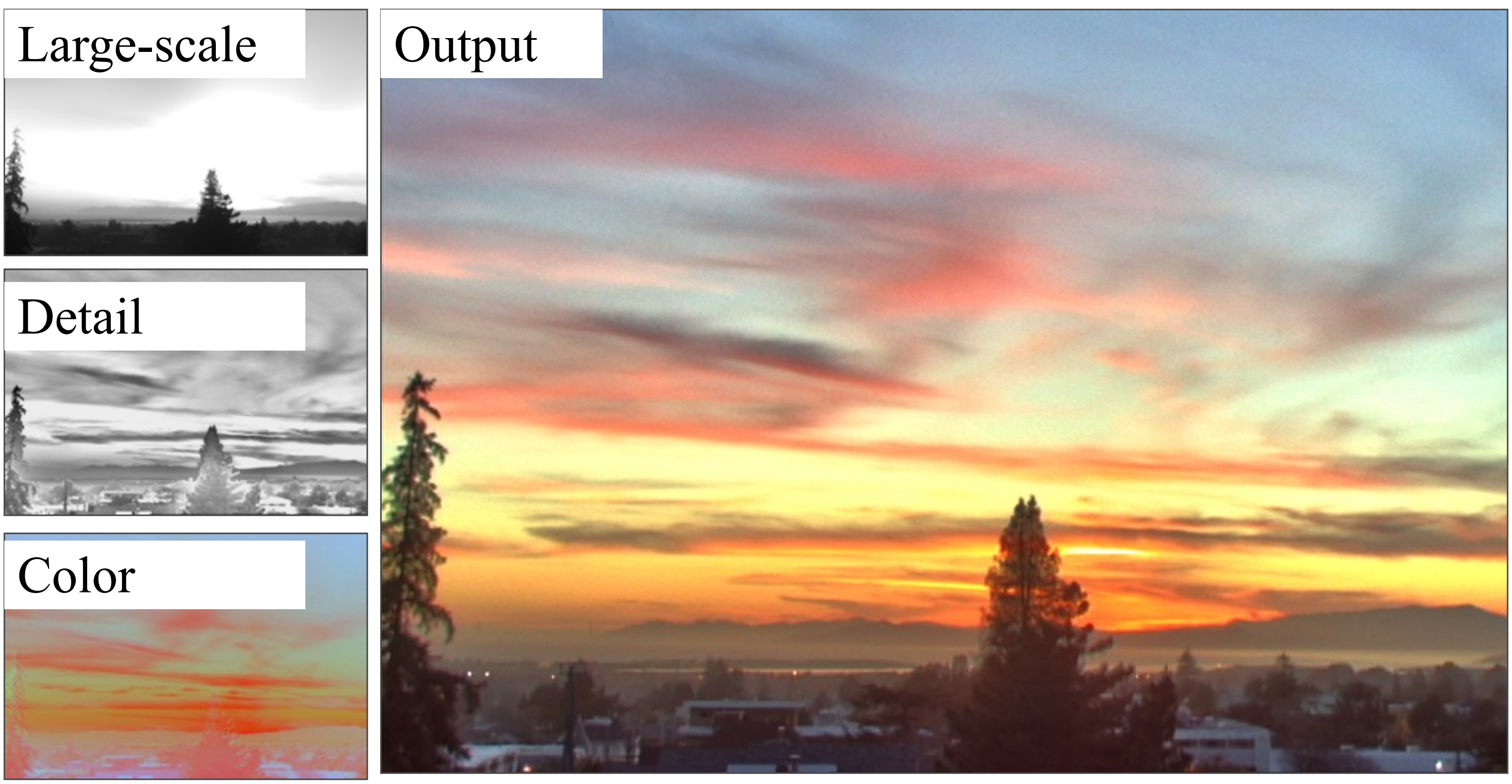
The halo nightmare

- **For strong edges**
- **Because they contain high frequency**



Our approach

- **Do not blur across edges**
- **Non-linear filtering**



The importance of convex combinations

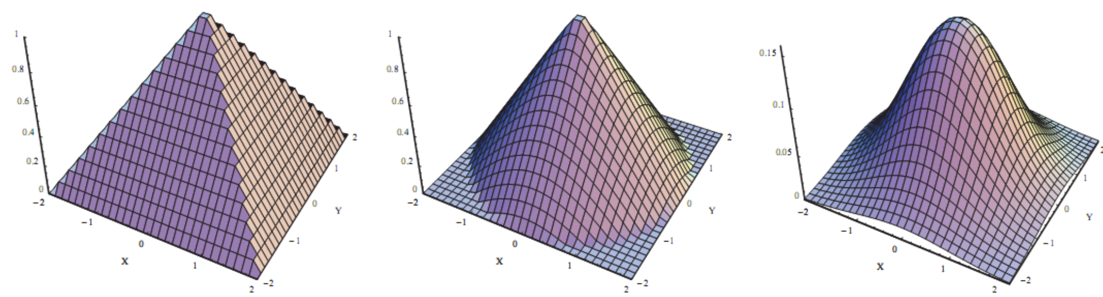
$$J(\mathbf{x}) = \sum \overbrace{f(\mathbf{x}, \xi)}^{\textbf{weight function}} \underbrace{I(\xi)}_{\textbf{Intensity}}$$

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

When we smooth, or interpolate we usually use wighted 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(x,ξ) be -



(a) Pyramid. (b) Cone. (c) Gaussian.

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

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

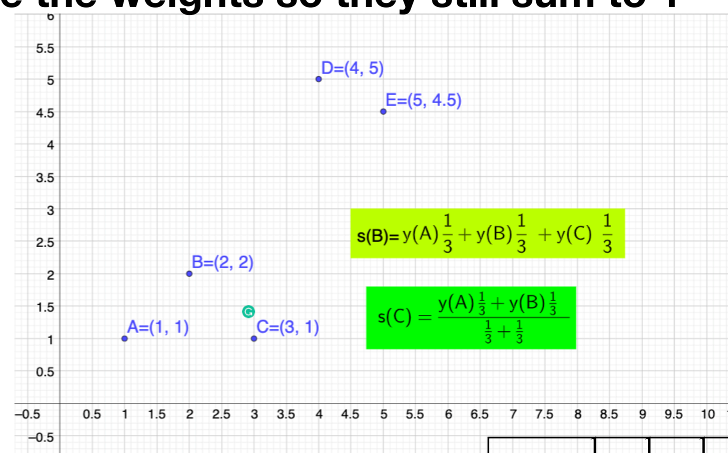
We will try to make sure that sum of weights =1 (this is called **convex combination**)
See example of bilinear interpolation on the whiteboard

We could avoid averaging points which are too far away. However, we need to normalize the weights so they still sum to 1

$N(p) = p's\ neighborhood$. **Pixels near** p.

Non-bilateral: $s(x) = \sum_{p' \in N(p)} w(p - p')L(p')$

$$w(p - p') = \begin{cases} 1\ if & |p - p'| \le 3 \\ 0 & otherwise\ . \end{cases}$$



Note: Sum of weights =1

$$k(B) = \frac{1}{3} + \frac{1}{3} + \frac{1}{3} = 1$$

$$k(C) = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bilateral smoothing: Redefine $N(p)$

$N'(p)$ = **pixel in** $N(p)$, **for which** $|L(p) - L(p')| \le 3$

$$k(p) = \sum_{p' \in N'(p)} w(p - p')$$

$$s(x) = \sum_{p' \in N'(p)} \frac{w(p - p')}{k(p)} L(p')$$

Same in two steps : Define $g(z) = \begin{cases} 1\ if & |z| \le 3 \\ 0 & otherwise\ . \end{cases}$

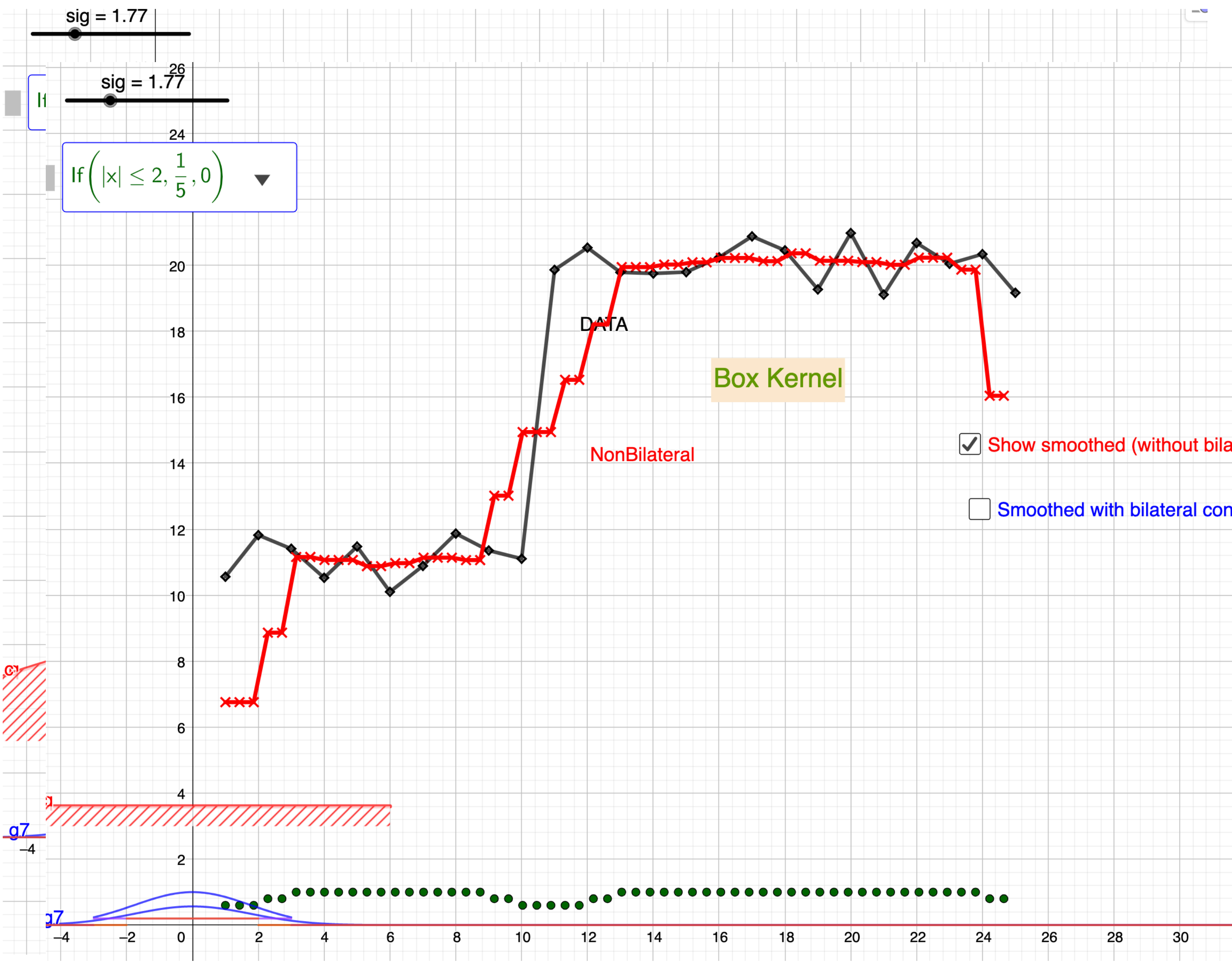
bilateral:

$$k(p) = \sum_{p' \in N(p)} g\big(L(p) - L(p')\big) w(p - p')$$

note: $k(p)$ = **summing weights only for pixels relevant to** p

$$s(x) = \sum_{p' \in N(p)} \frac{w(p - p')}{k(p)} g(L(p) - L(p')) \cdot L(p')$$

In non-bilateral kernel - the average blues the boundaries



sig = 1.77
sig = 1.77
$\text{If}\left(|x| \leq 2, \frac{1}{5}, 0\right)$
DATA
Box Kernel
NonBilateral
Show smoothed (without bilat
Smoothed with bilateral conv
g7

$N(p)$ = **pixels near** p

non-bilateral:

$$s(x) = \sum_{p' \in N(p)} w(p - p')L(p')$$

bilateral smoothing: Redefine $N(p)$

$N(p)$ = **pixel near p, for which** $|L(p) - L(p')| \leq 3$

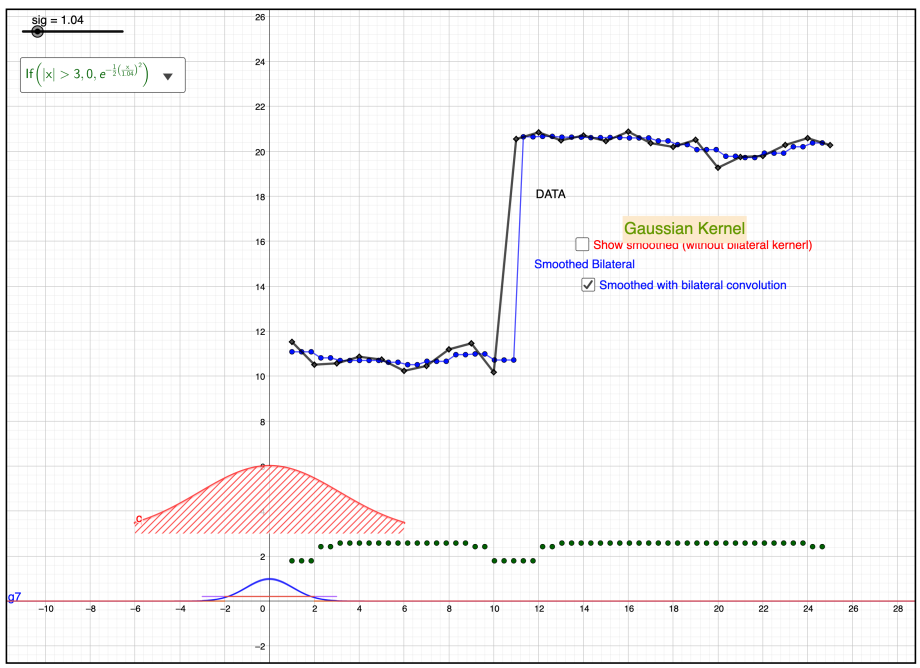
non-bilateral:

$$k(p) = \sum_{p' \in N(p)} w(p - p')$$

$$s'(x) = \sum_{p' \in N(p)} w(p - p')L(p')$$

$$k(p) = \sum_{p' \in N(p)} w(p - p')$$

$$s(p) = s'(v)/$$



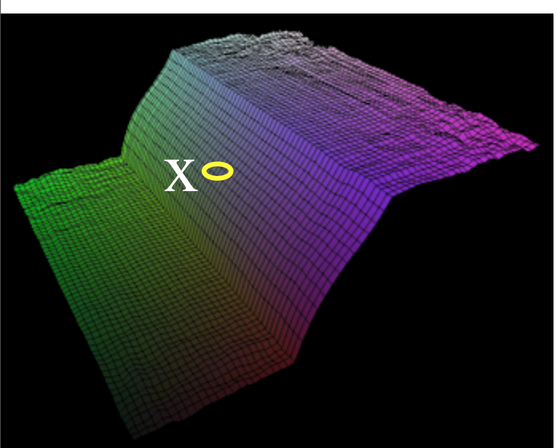
Gaussian filter as weighted average

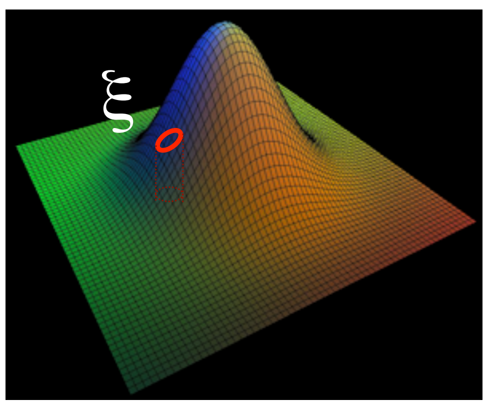
- **Weight of ξ depends on distance to x**

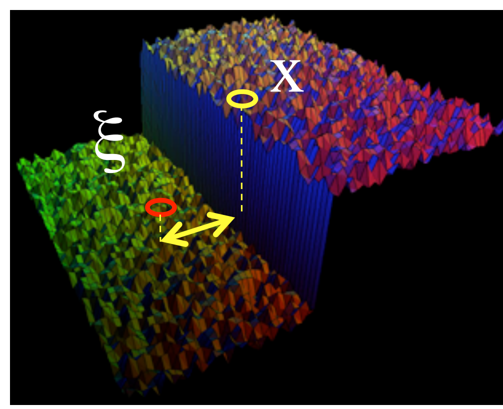
Non Bilateral

Averaged (smoothed) value at pixel x		Weight - e.g. $1/distance(x,\xi)$	Intensity at ξ

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





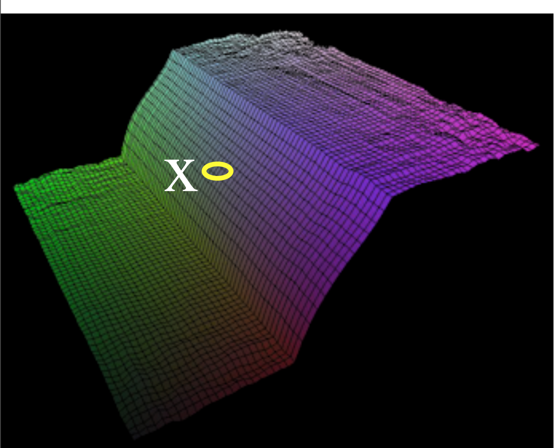


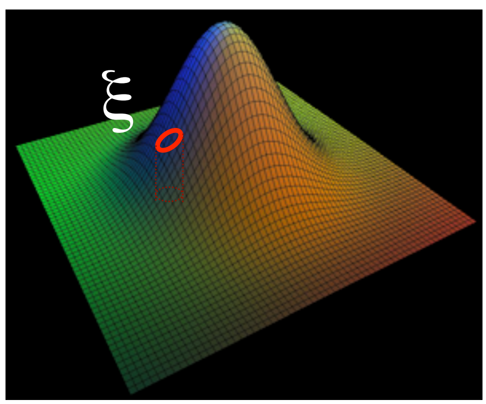
output ⟵ input

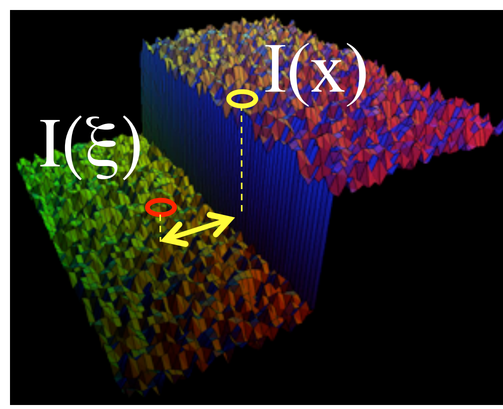
The problem of edges

- **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 filtering

[Tomasi and Manduchi 1998]

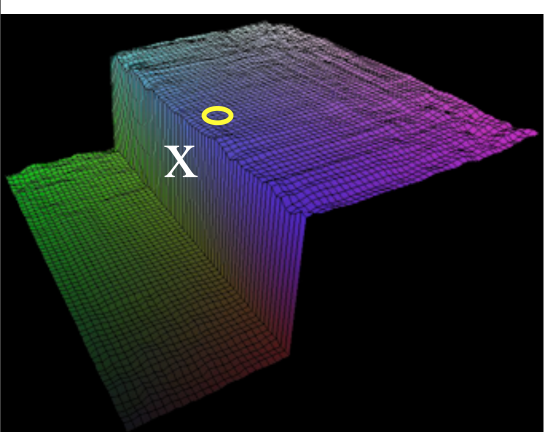
- **Penalty g on the intensity difference**

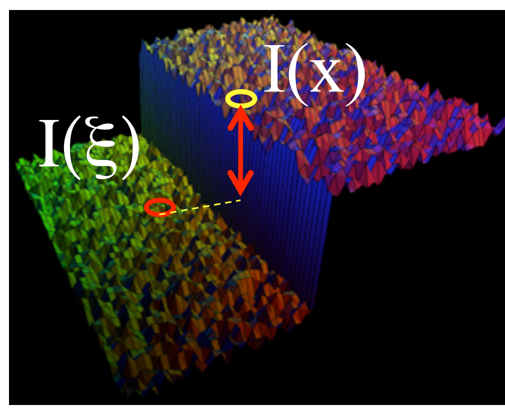
If $I(\xi) - I(x)$ is large, then $g\big(I(\xi) - I(x)\big) \approx 0$

Remember that the sum of weights $\sum f(x,\xi)$ must be 1.
What to do if we skip some terms ? (that is We will divide the total sum by k(x) - see next slide

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

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





output input

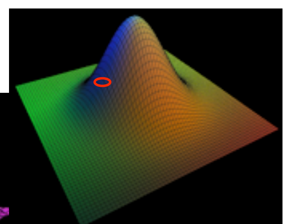
Bilateral filtering

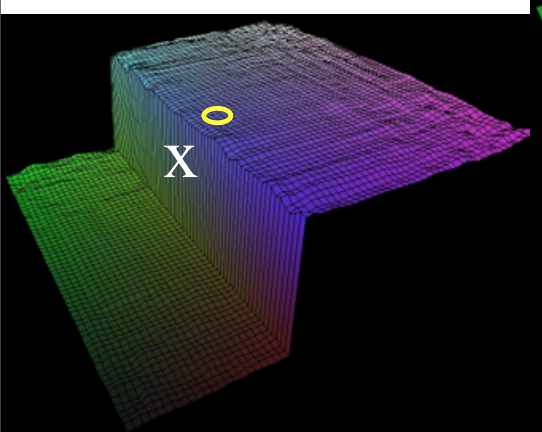
[Tomasi and Manduchi 1998]

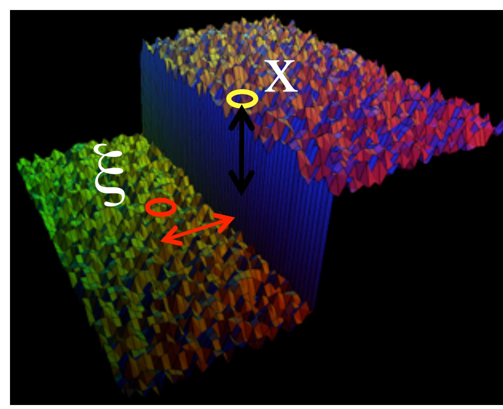
- **Spatial Gaussian f**

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)

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







output

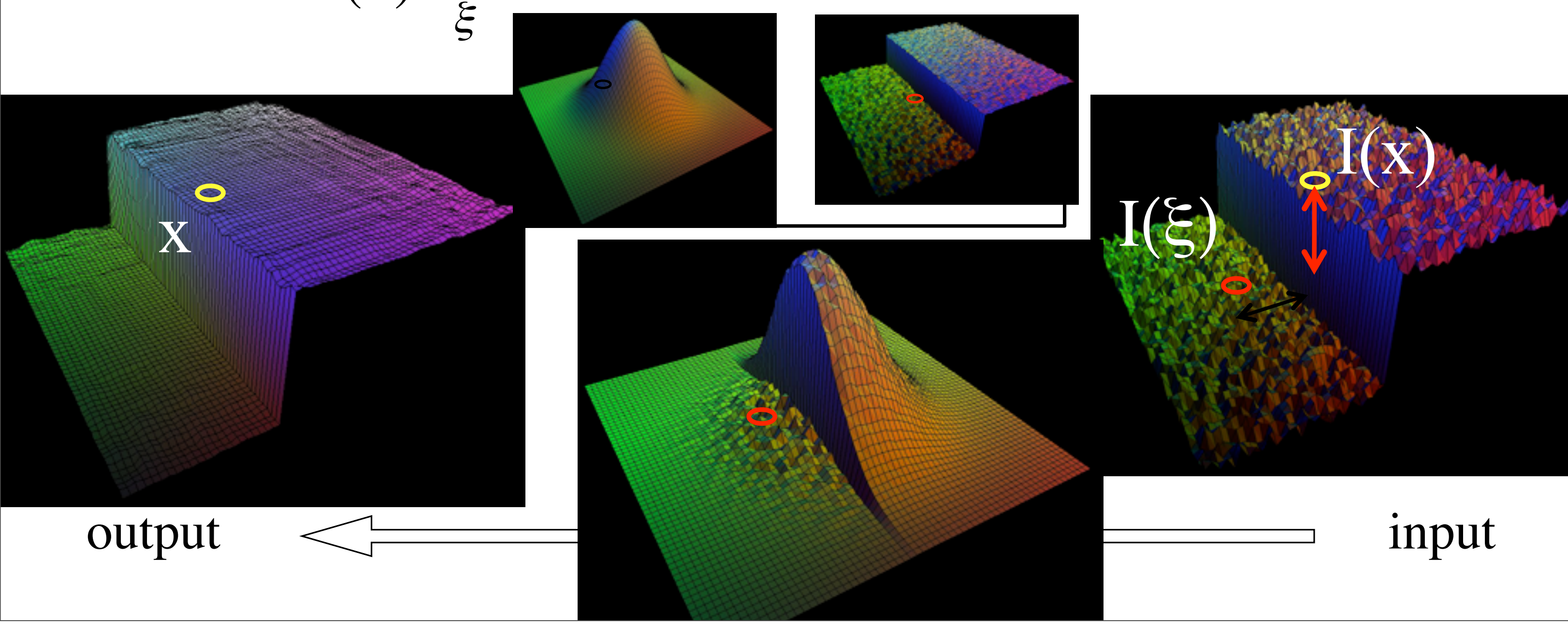
input

Bilateral filtering

[Tomasi and Manduchi 1998]

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

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

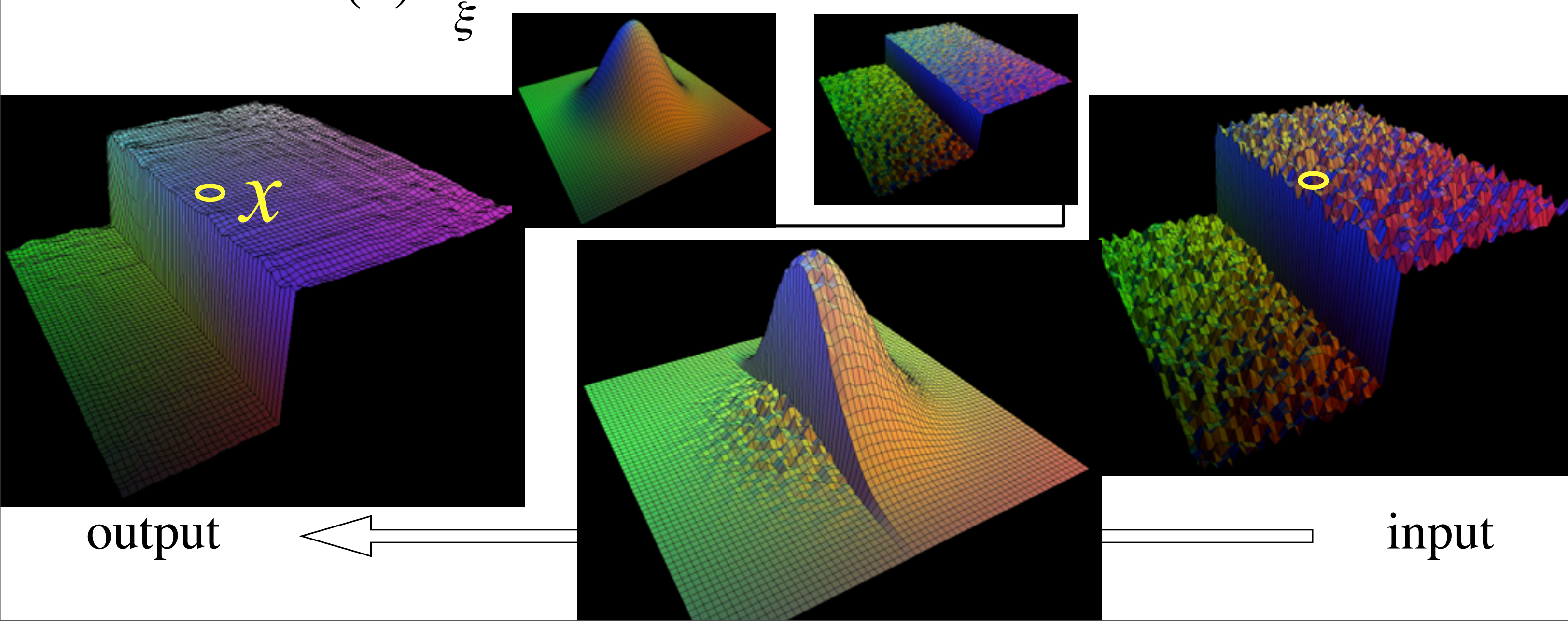


Normalization factor

[Tomasi and Manduchi 1998]

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

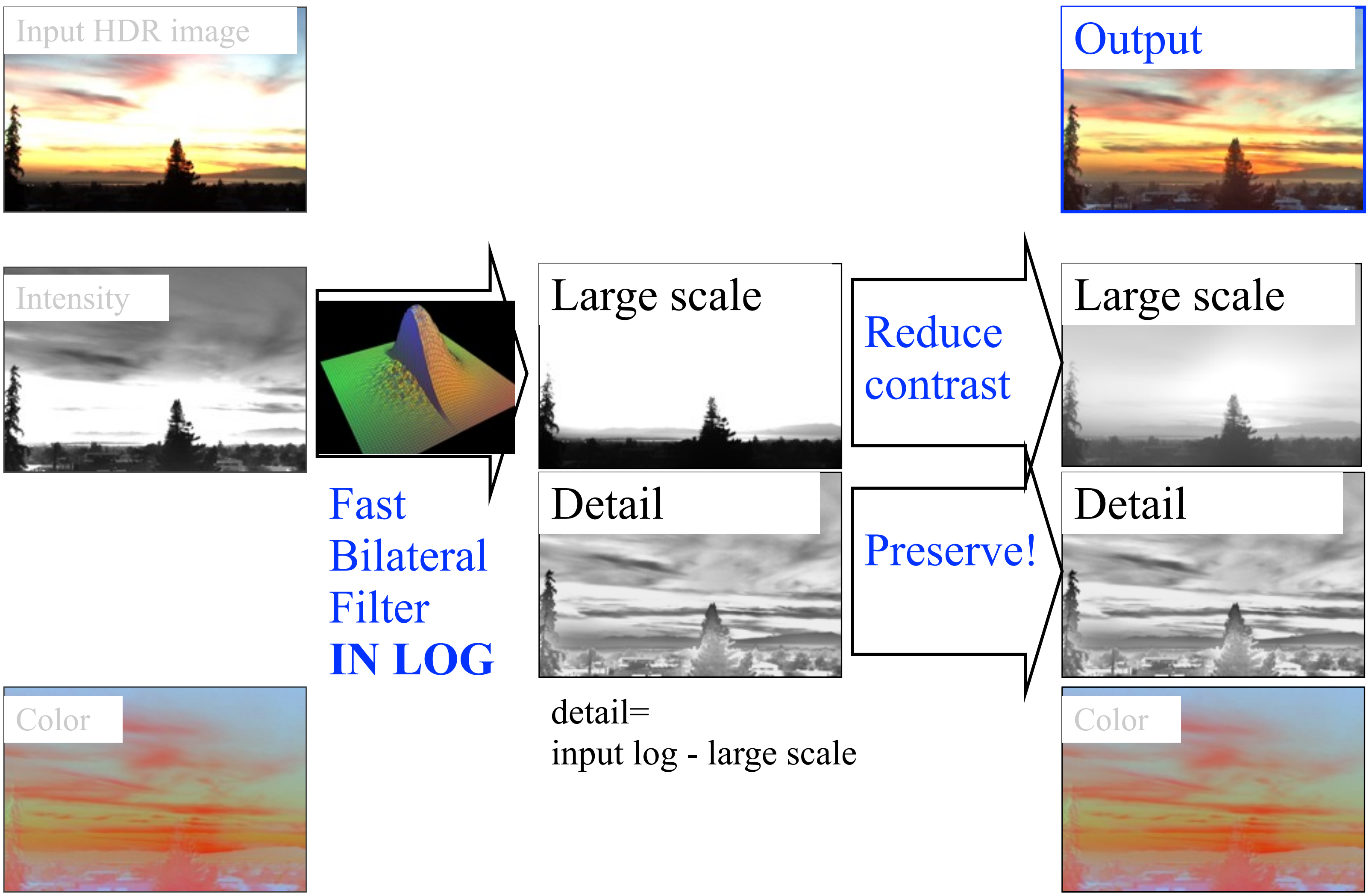
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



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



Input HDR image
Output
Intensity
Large scale
Reduce contrast
Large scale
Fast Bilateral Filter IN LOG
Detail
Preserve!
Detail
Color
detail=
input log - large scale
Color