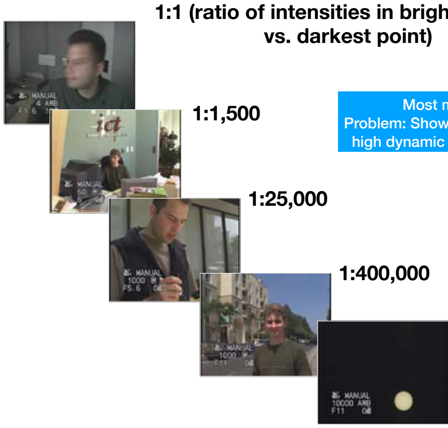
Tone Reproduction
Tone corrections
HDR and other applications of convolutions

Dynamic range of Eye vs Dynamic Range of Monitor

- Brightness (informally Luminosity) how much energy (in watts)
- This is different than colors
- Croud approximation R+G+B

The World is a High Dynamic Range (HDR)

1:1 (ratio of intensities in brightest point vs. darkest point)



1:1,500

1:25,000

1:400,000

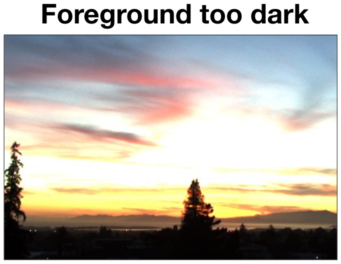
1:2,000,000,000

Most monitors: 1:255
Problem: Show images that has a very high dynamic range on our monitors

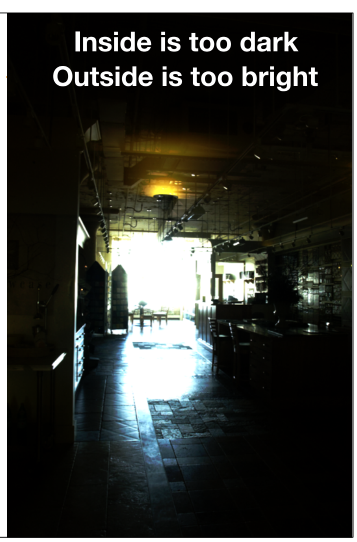
With HDR + Tone Mapping

Examples

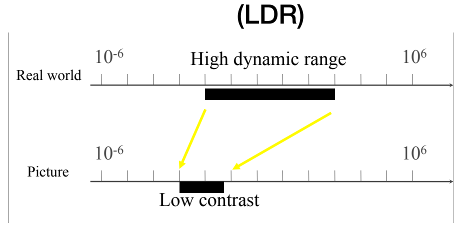
Sun overexposed
Foreground too dark



Inside is too dark
Outside is too bright



Somehow needs to map images with HDR into monitors (LDR)



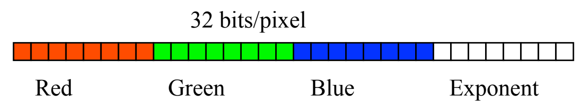
First Attempt: Linear transformation:
Need to map something to something - but does it looks anywhere realistic ?
Could easily change to color of the pixel - need to be careful !

1. Multiple problems:
2. Format of files of images with HDR
3. Changing intensities of images without changing colors: RGB vs HSV
4. Correct each pixel individually: (locall operations:) α, β, γ corrections (gain, bias)
5. Big topic: Global operations: Convolution, high-pass low-pass filters

Briefly: How to store HDR images

- Options: PPM could use 2 bytes 2^{16} bytes for R,G,B for each pixel
- Other options: Use a long float (8 bytes) for each value of R,G,B -overkill
- Common formats - to be discussed later

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

How not to change the color:

- We will modify the intensity $I = (R + G + B)/3$.
- For each pixel, we will map its original intensity I to a target intensity I' (devil in details)
- . Then will apply: $NewRGB = OldRGB \cdot (\frac{I}{I'})$
- The perception of the color should stay the same.
- Terms that we will use interchangeably (though they are not identical: Intensity I, lightness L , Brighntess B.

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

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

- α is a real-valued scaling factor known as **gain**
- β is a real-valued scaling factor known as **bias**

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

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'.

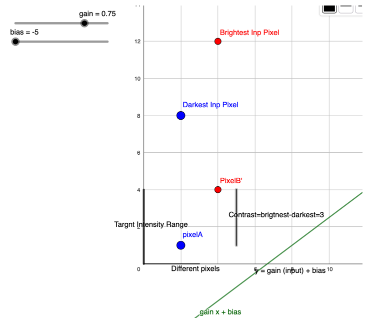
$$\begin{aligned} \triangle S' &= |S_1' - S_2'|, \\ \triangle S &= |S_1 - S_2|. \end{aligned}$$

- 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 gain and bias

https://www.geogebra.org/m/fskd5rpb



Want to maximize contrast = brightness point -darkest non-zero one

Another idea: γ-correction

- Instead of
  - $New\_Intensity(p) = \alpha \cdot Old\_Intensity + \beta$
  - Do

$$New\_Intensity(p) = \left(\alpha \cdot Old\_Intensity + \beta\right)^{\gamma}$$

Here $0 \leq \gamma \leq 1$

This is equivalent to

Calculate $\log_2(Intensity)$

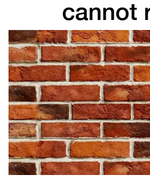
Multiply by γ

Calculate $2^{this\ value}$

Sidebar: Relating Contrast Sensitivities to Signal Processing

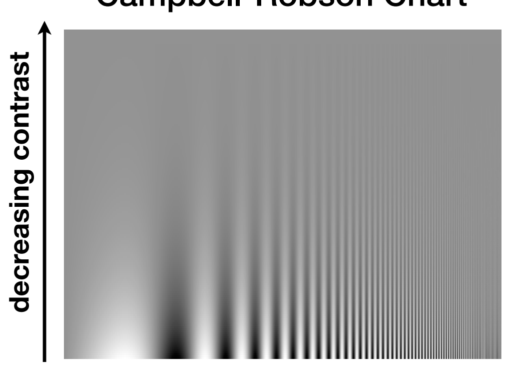
Spatial Frequency - how many pixels do we need to move (geographically) until the intensity changes signifcantly.

If the spatial frequency is high, but the display cannot reflect it, the image looks blurry.



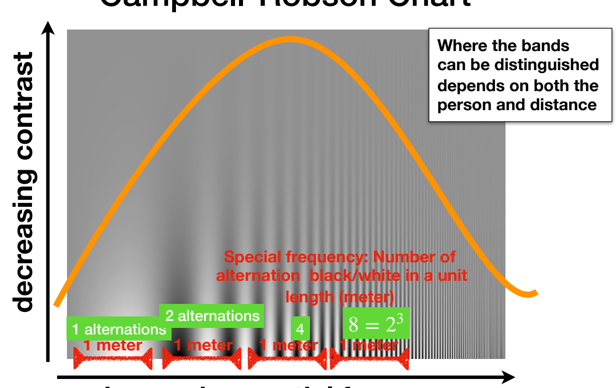
Contrast Sensitivity Function

Campbell-Robson Chart



Contrast Sensitivity Function

Campbell-Robson Chart



Contrast Sensitivities Vary by Channel

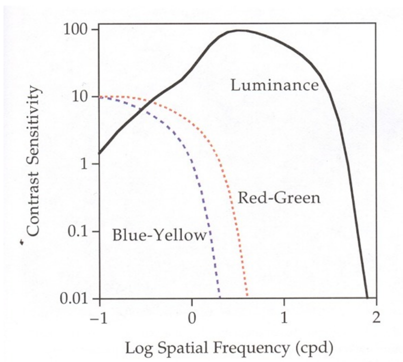


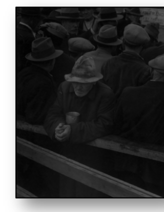
Figure 1-18. Spatial contrast sensitivity functions for luminance and chromatic contrast.

Rescaling Examples

gain = 1, bias = 55

gain = 1, bias = -55

gain = 2, bias=0



gain = .5, bias=0

Human Perception

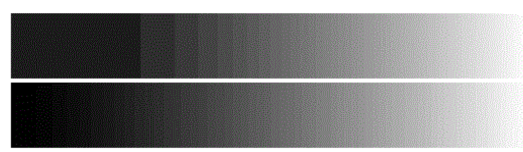
- Eye distinguishes color intensities as a function of the ratio between intensities.
- Consider $I_1 < I_2 < I_3$, for the step between I_1 and I_2 to look like the step from I_2 to I_3, it must be that:

  $$I_2 / I_1 = I_3 / I_2$$

- 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 $I_p = (I_a)^\gamma$



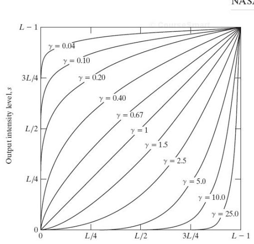
Linear

Gamma2.2

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

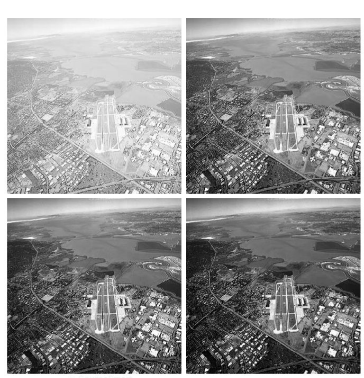
Example: Gamma Correction

$$s = r^\gamma$$



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
Not these operations still work on each pixel indevidually

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

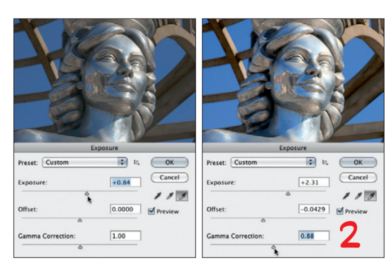
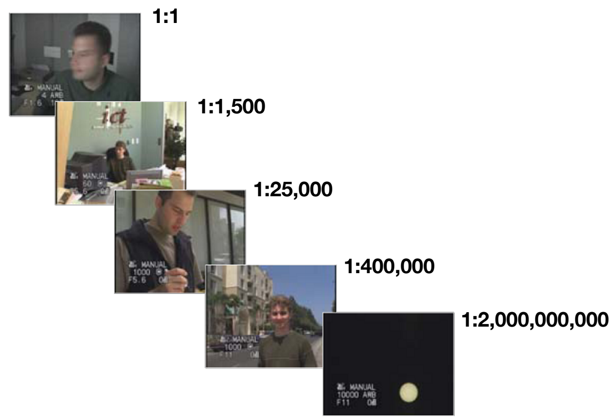


Image of Photoshop from Christian Bloch - The HDRI Handbook 2.0

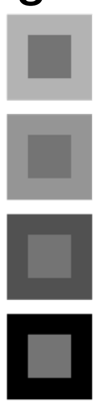
Dynamic Range

The World is a High Dynamic Range (HDR)



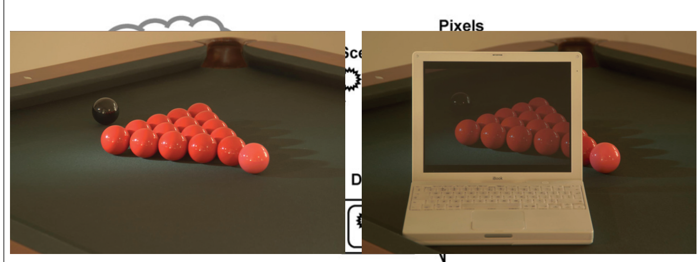
Eyes and Dynamic Range

- 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

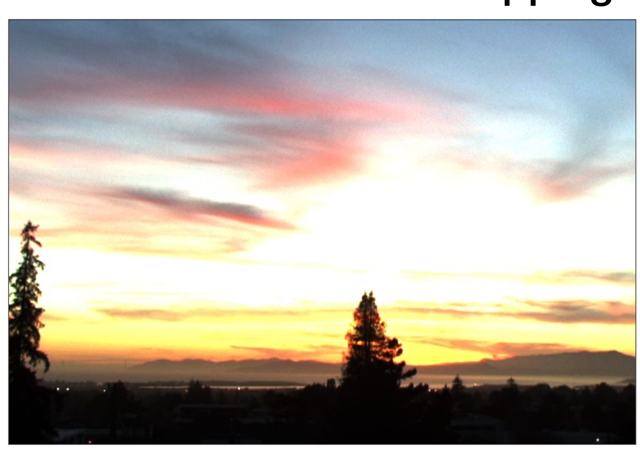


- We do not need to reproduce the true radiance as long as it gives us a visual match.

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.

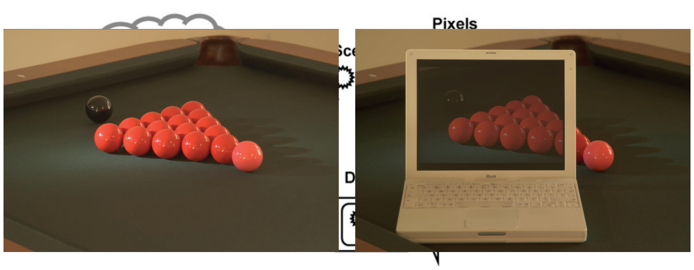
Without HDR + Tone Mapping



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

Approach: Visual Matching



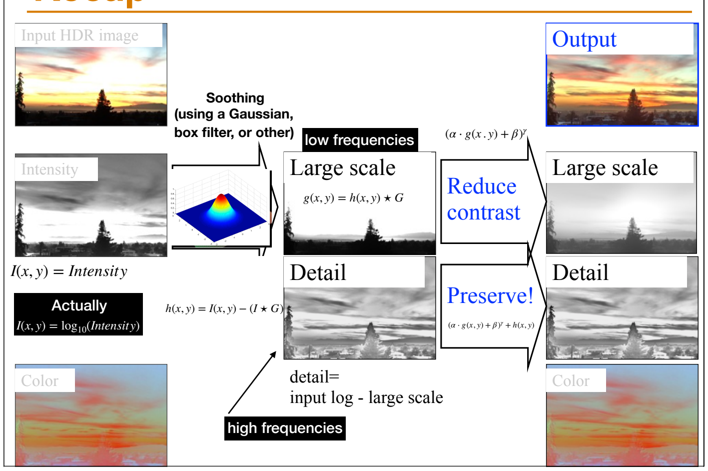
- We do not need to reproduce the true radiance as long as it gives us a visual match.
- Much more important to keep contrast between each pixel and its neighborhood pixels
- Means: Some neighborhoods must be treed different than others

Consider only the radiance map (the intensity at each pixel)

- Will produce two maps of intensities:
  - $I_{slow}(x, y)$ - changes slow, but large amplitude and large contrast
  - $I_{fast}(x, y)$ - changes rapidly, but small amplitude.
  - (α, β, γ)-correction will be applied only to I_{slow}
  - Put them back together

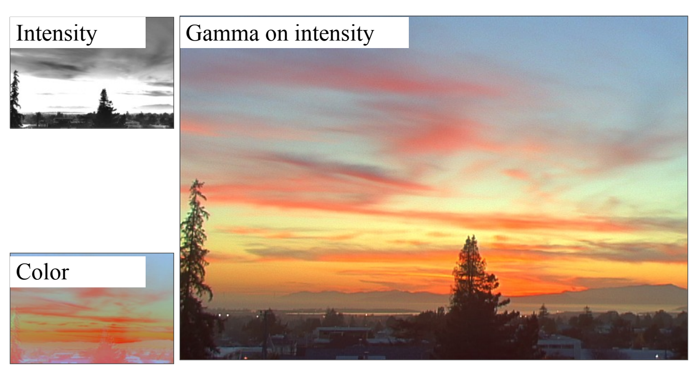
https://www.geogebra.org/m/w6qnvw9g

Recap



Gamma compression on Intensity

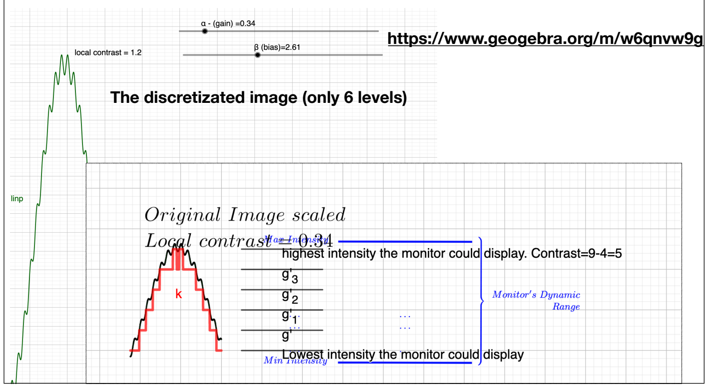
- Colors ok, but details in intensity are blurry



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 $C_{out}(p_i) = \alpha\ C_{in}(p_i)\ + \beta$

https://www.geogebra.org/m/w6qnvw9g

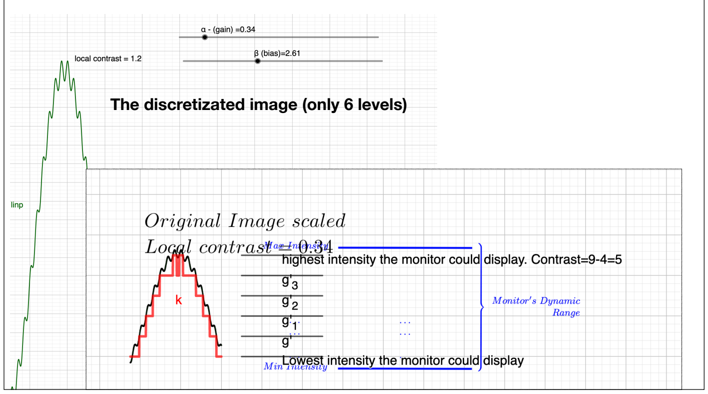


Oppenheim 1968, Chiu et al. 1993

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

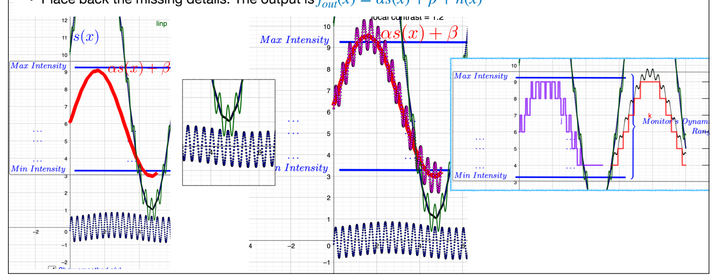
Given - an image with a large dynamic range (e.g. from image of outdoor scene
Need to compress it to much smaller dynamic range (monitor)
but avoid decreasing local contrast

In the GG demo, the input image represents the input image
Option 1 - compress $C_{out}(p_i) = \alpha\ C_{in}(p_i)\ + \beta$

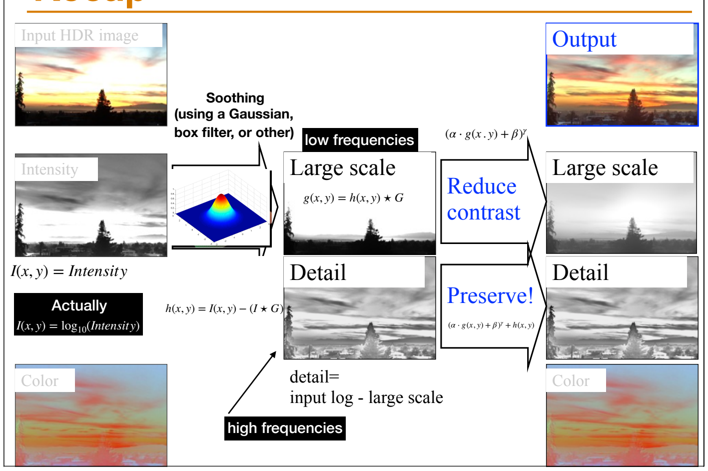


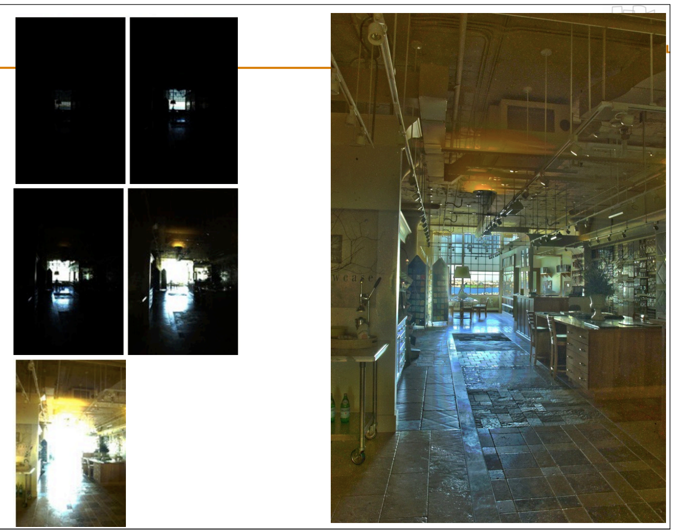
A better idea

- $f(x) =$ **Brightness level at pixel** x
- Compute $s(x)$ - a low pass filter:
  - For example, compute $s(x) = \big(f(x-1) + f(x) + f(x+1)\big)/3$ (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 $f_{out}(x) = \alpha s(x) + \beta + h(x)$



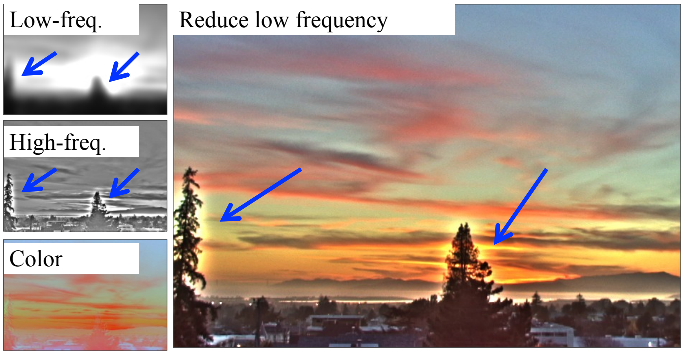
Recap





The halo nightmare

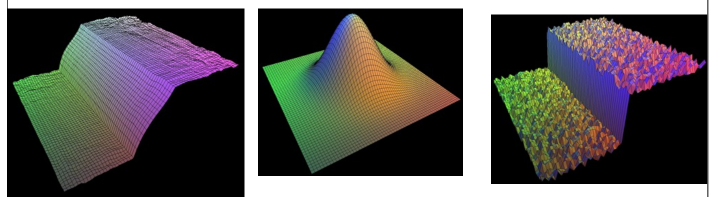
- For strong edges
- Because they contain high frequency



Start with Gaussian filtering

- Here, input is a step function + noise

$$J \quad = \quad \underset{\text{Weight (eg } \frac{1}{distance(x,\xi)}\text{)}}{f} \otimes \underset{\text{Intensity at } \xi}{I}$$

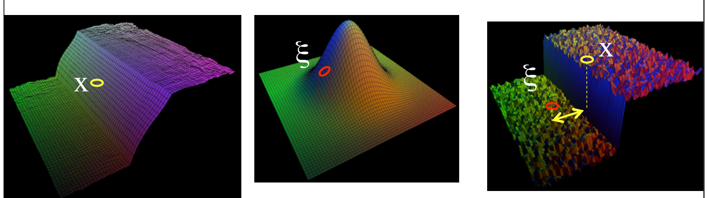


output ⟵ input

Gaussian filter as weighted average

- Weight of ξ depends on distance to x

$$J(x) = \sum_{\xi} \underset{\text{Weight (eg } 1/distance(x,\xi)\text{)}}{f(x,\xi)} \quad \underset{\text{Intensity at } \xi}{I(\xi)}$$



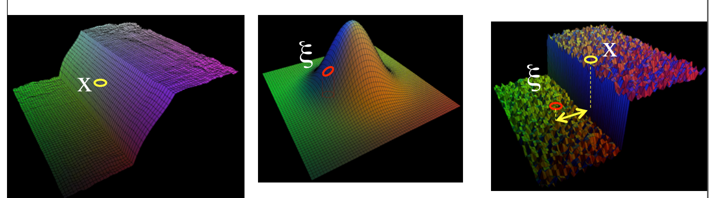
output ⟵ input

Gaussian filter as weighted average

- Weight of ξ depends on distance to x

Non Bilateral

$$\underset{\text{Averaged (smoothed) value at pixel x}}{J(x)} = \sum_{\xi} \underset{\text{Weight - e.g. } 1/distance(x,\xi)}{f(x,\xi)} \quad \underset{\text{Intensity at } \xi}{I(\xi)}$$

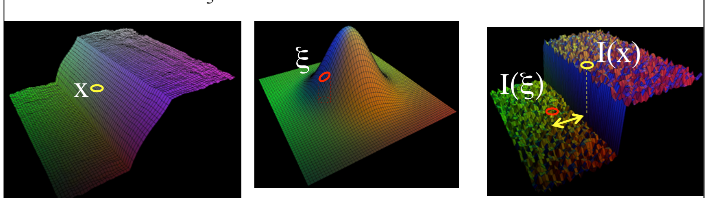


output ⟵ input

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]

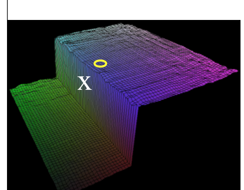
- Penalty g on the intensity difference

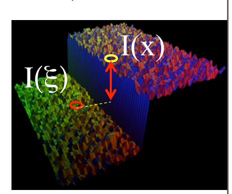
If $I(\xi) - I(x)$ is large, then $g(I(\xi) - I(x)) \approx 0$

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

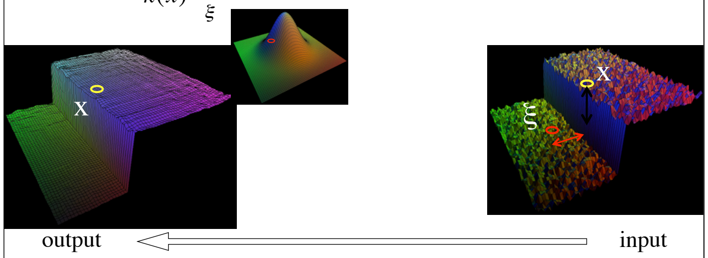
Bilateral filtering

[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)

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

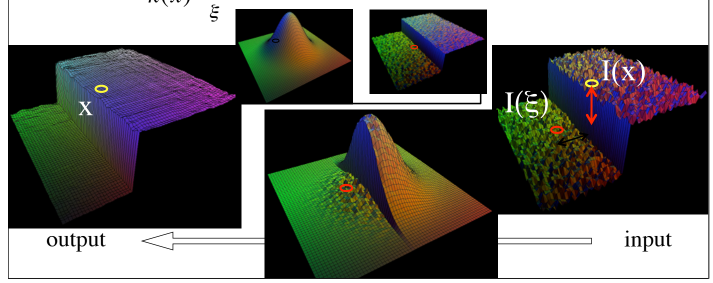


Bilateral filtering

[Tomasi and Manduchi 1998]

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

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

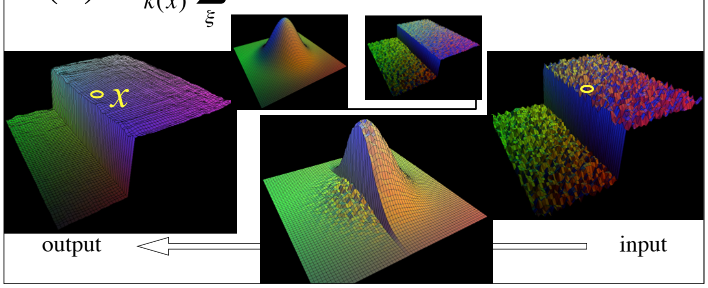


Normalization factor

[Tomasi and Manduchi 1998]

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

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



Bilateral filtering is non-linear

[Tomasi and Manduchi 1998]

- **The weights are different for each output pixel**

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