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

With HDR & tone mapping

Tuesday, March 6, 12

Examples

- Inside is too dark
- Outside is too bright
- Sun overexposed
- Foreground too dark

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

First Attempt: Linear transformation:
Need to map something to something - but does it look 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: (local operations:) \( \alpha, \beta, \gamma \) 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)**

<table>
<thead>
<tr>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>(145, 215, 87, 149)</td>
<td>(145, 215, 87, 103)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(145, 215, 87) * $2^{(149-128)}$</td>
<td>(145, 215, 87) * $2^{(103-128)}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1190000</td>
<td>1760000</td>
<td>713000</td>
<td>0.00000432</td>
</tr>
</tbody>
</table>


---

**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: $NewRGB = OldRGB \cdot \left( \frac{I'}{I} \right)$
- The perception of the color should stay the same.
- Terms that we will use interchangeably (though they are not identical): Intensity I, lightness L, brightness 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

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

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

---

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

- The relative change in contrast can be simplified as

$$\frac{\Delta S'}{\Delta S} = \frac{|(\alpha S_1 + \beta) - (\alpha S_2 + \beta)|}{|S_1 - S_2|}$$

$$= \frac{|(\alpha S_1 - \beta) - (\alpha S_2 - \beta)|}{|S_1 - S_2|}$$

$$= |\alpha|.$$ 

- 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.
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.

\[ S'_1 = \alpha S_1 + \beta, \quad S'_2 = \alpha S_2 + \beta. \]

\[ \Delta S' = |S'_1 - S'_2|, \quad \Delta S = |S_1 - S_2|. \]

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

Another idea: $\gamma$-correction

- Instead of

  - $\text{New}_\text{Intensity}(p) = \alpha \cdot \text{Old}_\text{Intensity} + \beta$

  - Do $\text{New}_\text{Intensity}(p) = \left( \alpha \cdot \text{Old}_\text{Intensity} + \beta \right)^\gamma$

  Here $0 \leq \gamma \leq 1$

  This is equivalent to

  - Calculate $\log_2(\text{Intensity})$
  - Multiply by $\gamma$
  - Calculate $2^\text{this value}$

Sidebar: Relating Contrast Sensitivities to Signal Processing

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

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

Contrast Sensitivity Function

Campbell-Robson Chart

Where the bands can be distinguished depends on both the person and distance.

Special frequency: Number of alternation black/white in a unit length (meter)

\[ 1 \text{ alternation} = 1 \text{ alternation} \]

\[ 2 \text{ alternations} \]

\[ 4 \text{ alternations} \]

\[ x = 2^n \]
Contrast Sensitivities Vary by Channel

Contrast Sensitivity Function (CSF)

Rescaling Examples

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:
  
  $$\frac{I_2}{I_1} = \frac{I_3}{I_2}$$

- 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

Putting it all together:

Gain, Bias, and Gamma

Not these operations still work on each pixel individually

- $C_{out} = (\alpha C_{in} + \beta)^\gamma$

- $\alpha$ is known as gain (exposure)

- $\beta$ is known as bias (offset)

- $\gamma$ maps to a non-linear curve (gamma correction)
**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

**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

**Approach: Visual Matching**

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

**Without HDR + Tone Mapping**

- 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 “high” or “low” regions are short in details.
- Examples: Sky, windows during day light

**What if rescaling and gamma-correction are not sufficient**

- A collection of algorithms that is useful in many scenarios uses the idea that in regions where there are slow changes from bright to dark, large regions can be identified as either “high” or “low” brightness.

- Or the way chatGPT rephrased it:
  - “A collection of algorithms that is useful in many scenarios uses the idea that in regions where there are slow changes 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 details. Examples of this include the sky and windows during daylight hours.”

---

**Headlights are ON in both photos**
No single global exposure can preserve both the colors of the sky and the details of the landscape, as shown on the rightmost images.

**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 treated differently than others

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

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

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

**Recap**

- Colors ok, but details in intensity are blurry

**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
\[ C_{\text{out}}(p) = \alpha C_{\text{in}}(p) + \beta \]

The discretized image (only 6 levels)

Oppenheim 1968, Chiu et al. 1993
• Reduce contrast of low-frequencies
• Keep mid and high frequencies

A better idea
• Compute a low pass filter:
  - For example, compute
  - better idea: Use convolution with a tent or a gaussian
  - in \( x \), details that change frequently (high special frequencies) are averaged.
  - computer \( \hat{h}(x) = f(x) - s(x) \) 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 \( h(x) \)
  - Rescale \( h(x) \) by using the the gain and bias \( \alpha, \beta \) such that it fits the monitor dynamic range.
  - \( s(x) \rightarrow \alpha \cdot s(x) + \beta \) Usually \( \alpha \leq 1 \).

Place back the missing details. The output is
\[ f_{\text{out}}(x) = \alpha h(x) + \beta + s(x) \]

Recap
Input HDR image
Sothing (using a Gaussian, box filter, or other)
\[ I(x,y) = \log_{10}(I_{\text{in}}(x,y)) \]
Actually
\[ I(x,y) = \log_{10}(I_{\text{in}}(x,y)) \]

Output
Low frequencies
\[ g(x,y) = I(x,y) \]

Large scale
\[ f(x,y) = \frac{g(x,y)}{3} \]

Reduce contrast
\[ f(x,y) = \alpha f(x,y) + \beta \]

High frequencies
The halo nightmare

- For strong edges
- Because they contain high frequency

The problem of edges

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

Gaussian filter as weighted average

- Weight of \( \xi \) depends on distance to \( x \)

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

Start with Gaussian filtering

- Here, input is a step function + noise

\[
J = \frac{1}{\text{distances}, \xi} f \otimes I
\]

Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- Penalty \( g \) on the intensity difference

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

Remember that the sum of weights \( \sum f(x,\xi) \) must be 1.

What to do if we skip some terms? (that is \( k(x) \)) We will divide the total sum by \( k(x) \) - see next slide.
Bilateral filtering

- 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

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

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

Bilateral filtering is non-linear

- The weights are different for each output pixel

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