Lecture 10: Digital Camera Image Signal Processing: Denoising, HDR Imaging, and Tone Mapping
Logistics

- Moved my office hours to Friday 3:30 - 4:30, effective this week.
- Project proposal deadline 10/15. Submit a one page write-up. Talk to me if you still haven’t!
- Grades for previous assignments are released.
- Solution for WA2 is also on BB.
- WA3 is out. Due 10/2, 11:30 AM.
Image Signal Processing Pipeline

Lens → Sensor → Demosaic → WB & Color Correction → Denoising → HDR & Tone Mapping → Compression → AWB, AE, AF

RAW Pixels → Auto Exposure, Auto Focus, Auto White Balance
Denoising

- Sensor/sensing introduces various sources of noise.
- Post-processing stages impact noise levels too, e.g., color correction.
  - Since they manipulate noisy signals.
- Denoise sooner rather than later, and can denoise multiples times.
- The easiest way to denoise an image is to **blur** the image.
- Always a trade-off between image details retained vs. noise removed.
- Spatial denoising vs. temporal denoising
  - Single-image denoising vs. video denoising
Image Filtering

A color image is a 3D array, each element contains the RGB values (in whatever color space it's defined in).

Image filtering changes the value of each pixel without changing the image dimension. Usually each output pixel is generated from a block of pixels in the input image.

Mean/Moving Average Filter

\[ Out_{11} = \frac{In_{00} + In_{01} + In_{02} + In_{10} + In_{11} + In_{12} + In_{20} + In_{21} + In_{22}}{9} \]

https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html
Mean/Moving Average Filter

\[
Out_{11} = \frac{In_{00} + In_{01} + In_{02} + In_{10} + In_{11} + In_{12} + In_{20} + In_{21} + In_{22}}{9}
\]

https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html
Mean/Moving Average Filter

Image convolution operation

\[
\begin{bmatrix}
I_{00}, I_{01}, I_{02} \\
I_{10}, I_{11}, I_{12} \\
I_{20}, I_{21}, I_{22}
\end{bmatrix}
\ast
\begin{bmatrix}
K_{00}, K_{01}, K_{12} \\
K_{10}, K_{11}, K_{22} \\
K_{20}, K_{21}, K_{22}
\end{bmatrix}
= \sum_{i=0, j=1}^{i=2, j=2} (I_{ij} \times K_{ij})
\]

Out_{11} = \frac{I_{00} + I_{01} + I_{02} + I_{10} + I_{11} + I_{12} + I_{20} + I_{21} + I_{22}}{9}

https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html
Mean/Moving Average Filter

**Image convolution operation**

\[
\begin{bmatrix}
I_{00}, I_{01}, I_{02} \\
I_{10}, I_{11}, I_{12} \\
I_{20}, I_{21}, I_{22}
\end{bmatrix}
\otimes
\begin{bmatrix}
K_{00}, K_{01}, K_{02} \\
K_{10}, K_{11}, K_{12} \\
K_{20}, K_{21}, K_{22}
\end{bmatrix}
= \sum_{i=0,j=1}^{i=2,j=2} (I_{ij} \times K_{ij})
\]

**Convolution Kernel/Filter**

\[
Out_{11} = \frac{I_{00} + I_{01} + I_{02} + I_{10} + I_{11} + I_{12} + I_{20} + I_{21} + I_{22}}{9}
\]

\[
Out_{11} = \begin{bmatrix}
I_{00}, I_{01}, I_{02} \\
I_{10}, I_{11}, I_{12} \\
I_{20}, I_{21}, I_{22}
\end{bmatrix}
\otimes
\begin{bmatrix}
\frac{1}{9}, \frac{1}{9}, \frac{1}{9} \\
\frac{1}{9}, \frac{1}{9}, \frac{1}{9} \\
\frac{1}{9}, \frac{1}{9}, \frac{1}{9}
\end{bmatrix}
\]

https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html
Mean/Moving Average Filter

A mean filter averages neighboring pixels, making pixels more similar to each other — blurring! A mean filter is a low-pass filter in signal processing parlance.

\[
Out_{11} = \begin{bmatrix}
I_{00}, I_{01}, I_{02} \\
I_{10}, I_{11}, I_{12} \\
I_{20}, I_{21}, I_{22}
\end{bmatrix} \ast \frac{1}{9} \begin{bmatrix}
1,1,1 \\
1,1,1 \\
1,1,1
\end{bmatrix}
\]
Gaussian Filter

A Gaussian filter also averages neighboring pixels, but gives more weight to closer neighbors. It’s still a low-pass filter.

https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm
https://slideplayer.com/slide/2489074/
Bilateral Filter: Edge-Preserving Denoising

- **Issue**: denoising using blurring works only when the scene is smooth (low spatial frequency). When scenes are not smooth (high frequency), i.e., with edges, blurring will destroy edges and the image looks less sharp.

  - Very similar to naive demosaic artifacts.

- **Solution**: don’t blur across the edge!

- What’s an edge? Simple but effective measure: **color difference**!
**Bilateral Filter: Edge-Preserving Denoising**

- **Idea**: Use the product of two kernels as the filtering kernel
  - Weights in the first kernel are dictated by pixel **distance**. Closer pixels have higher weights.
  - Weights in the second kernel are dictated by pixel **color**. More similar pixels have higher weights.
- Color difference usually calculated as pixel value difference. The original bilateral filtering paper [ICCV 98] uses perceptual difference (i.e., in the CIELAB space).
Bilateral Filter: Edge-Preserving Denoising

- Noisy image
- Gaussian filtered image
- Bilateral filtered image
Luminance vs. Chroma Denoising

- Will have rigorous definition of luminance vs. chroma later.
- Informally, luminance means brightness and chroma means hue and saturation. Both can be derived from a RGB space (non-linear transformation).
- Human eyes are more sensitive to noise in chroma.
- Chroma noises tend to have low frequency, and luminance noises tend to have high frequency. Low-pass filters are less thus effective to chroma noises.
- Could have separate luminance and chroma denoising in the pipeline.

https://www.cambridgeincolour.com/tutorials/image-noise-2.htm
Image Signal Processing Pipeline

Lens ➔ Sensor ➔ Demosaic ➔ WB & Color Correction ➔ Denoising ➔ HDR & Tone Mapping ➔ Compression ➔ AWB, AE, AF

- RAW Pixels
- Auto White Balance
- Auto Exposure
- Auto Focus
Natural Scenes Have High Dynamic Range

Scene Dynamic Range

10 cd/m²
30,000 cd/m²
300 cd/m²
10,000 cd/m²
2,000 cd/m²
60 cd/m²

Range of Visible Luminance in Candellas per Meter Squared (cd/m²) Log Scale

Rendering the Print: the Art of Photography
The Problem

Scene \( \rightarrow \) Camera \( \rightarrow \) Display \( \rightarrow \) Print \( \rightarrow \) Eyes

Rendering the Print: the Art of Photography

Original Scene Dynamic Range

Camera Sensor Dynamic Range

Quality In-Size Pigment Print

LCD display DR is not much better
Two Tasks

1. HDR Imaging: how to capture an HDR scene with a lower DR capturing device. Limited by the camera.

2. Tone Mapping: how to display HDR images on lower DR display devices. Limited by the display.

**Scene**  High DR (natural) scenes

**Camera**  Lower DR capturing device (e.g., camera)

**Display**  Even lower DR output device (e.g., display, paper)

**Print**
Goal of HDR Imaging

Goal: Pixel values represent the *full range* of scene luminance while maintaining the *relative* ratios.

Range of Visible Luminance in Candellas per Meter Squared (cd/m²) Log Scale

- 10 cd/m²
- 30,000 cd/m²
- 300 cd/m²
- 2,000 cd/m²
- 60 cd/m²
- 10,000 cd/m²
Recall: Radiance and Luminance

- What really matters is **relative** luminance. Displays are not able to reproduce the actual natural luminance. Eyes are sensitive to relative luminance anyways.
  
  ▶ My LG display's max luminance is 300 cd/m². Natural scenes can be ~100,000 cd/m².
Recall: Radiance and Luminance

Radiance $L_e(\lambda)$: power per unit direction per unit area

\[ I_v = 683 \frac{lm}{W} \int I_e(\lambda)V(\lambda)d(\lambda) \]

Luminance $I_e(\lambda)$: “perceived power” per unit direction (solid angle) per unit area

- What really matters is relative luminance. Displays are not able to reproduce the actual natural luminance. Eyes are sensitive to relative luminance anyways.
- My LG display's max luminance is 300 cd/m². Natural scenes can be ~100,000 cd/m².
Recall: Radiance and Luminance

Radiance $L_\varepsilon(\lambda)$: power per unit direction per unit area

\[
I_v = 683 \frac{lm}{W} \int I_\varepsilon(\lambda)V(\lambda)d(\lambda)
\]

Luminance $I_\varepsilon(\lambda)$: “perceived power” per unit direction (solid angle) per unit area

- What really matters is relative luminance. Displays are not able to reproduce the actual natural luminance. Eyes are sensitive to relative luminance anyways.
  - My LG display’s max luminance is 300 cd/m$^2$. Natural scenes can be $\sim$100,000 cd/m$^2$.
- Luminance and radiance are related by a known luminosity function. So recovering luminance ratio is equivalent to recovering radiance ratio.
How to Recover Scene Radiance?

Radiance (L): power/direction/area

P: # of photons hit this pixel during exposure

How to Recover Scene Radiance?

Scene radiance is proportional to the radiance on sensor

\[
\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}
\]

Radiance (L): power/direction/area

P: # of photons hit this pixel during exposure

https://medium.com/hipster-color-science/the-units-of-spectral-radiance-bb97c3b27cd
How to Recover Scene Radiance?

Scene radiance is proportional to the radiance on sensor

\[
\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}
\]

\[
L_2 \approx \frac{P \times E}{\theta \times A \times t}
\]

P: # of photons hit this pixel during exposure

https://medium.com/hipster-color-science/the-units-of-spectral-radiance-bb97c3b27c6d
How to Recover Scene Radiance?

Scene radiance is proportional to the radiance on sensor

\[
\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}
\]

\[
L_2 \approx \frac{P \times E}{\theta \times A \times t}
\]

- **Average energy per photon**
- **Exposure time**
- **Pixel area**

https://medium.com/hipster-color-science/the-units-of-spectral-radiance-bb97c3b27c6d
How to Recover Scene Radiance?

Scene radiance is proportional to the radiance on sensor

$$\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}$$

Approximate. Remember Vignetting?

$$L_2 \approx \frac{P \times E}{\theta \times A \times t}$$

Radiance (L): power/direction/area

P: # of photons hit this pixel during exposure

Average energy per photon

Exposure time

Pixel area

https://medium.com/hipster-color-science/the-units-of-spectral-radiance-bb97c3b27c6d
How to Recover Scene Radiance?

What camera provides us: raw pixel value

Scene radiance is proportional to the radiance on sensor

$$\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}$$

Approximate. Remember Vignetting?

$$L_2 \approx \frac{P \times E}{\theta \times A \times t}$$

Average energy per photon

Exposure time

Pixel area

P: # of photons hit this pixel during exposure

P: # of photons hit this pixel during exposure
How to Recover Scene Radiance?

What camera provides us: raw pixel value

Scene radiance is proportional to the radiance on sensor

$L_1 \propto L_2$, which is proportional to $P$! But how do we go from raw pixel value to photon quantity $P$???

$L_2 \approx \frac{P \times E}{\theta \times A \times t}$

Approximate. Remember Vignetting?

$L_1 = \frac{n_1^2}{n_2^2}$
Recovering Photon Quantity from Pixel Value

- Barring noise and quantization error, the **RAW pixel value** \( n \) and the incident **photon quantity** \( P \) are close to **linear** between the noise floor and saturation.

\[
Q = P \times SSF \times QE \\
V = \frac{Q}{C}g \\
n = \left\lfloor \frac{V - V_{\text{min}} + s/2}{s} \right\rfloor
\]

**Linear is good** because it means the photon quantity is **proportional** to raw pixel value. Remember: **relative** radiance is all we care about.

We could theoretically calculate the slope, which let us recover the absolute scene luminance from the pixel value.
Dynamic Range Issue

Pixel value

Noise floor

Photons

Under-exposed

Over-exposed

Need an HDR sensor or HDR algorithm.

https://iphonephotographyschool.com/hdr-iphone/
Dynamic Range Issue

Under-exposed

Over-exposed

Need an HDR sensor or HDR algorithm.

Pixel value

Noise floor

Under exposed

Photons

https://iphonephotographyschool.com/hdr-iphone/
Dynamic Range Issue

- Under-exposed
- Over-exposed

Need an HDR sensor or HDR algorithm.

https://iphonephotographyschool.com/hdr-iphone/
Dynamic Range Issue

Pixel value vs. Photons

Noise floor

Under-exposed

Over-exposed

Need an HDR sensor or HDR algorithm.

Noise floor

OK now!

https://iphonephotographyschool.com/hdr-iphone/
Dynamic Range Issue

Problem: full luminance range isn’t accurately captured.

- When pixels are under-exposed, we in theory could recover the luminance, but the SNR would be very low, so the luminance values are noisy.
  - Simply amplifying all the pixel values will not work because the noise gets amplified too. The SNR is the same (noisy).
  - Denoising has the effect of enhancing DR.
- When pixels are over-exposed, pixel values don’t reflect scene luminance.
Dynamic Range Issue

Problem: full luminance range isn’t accurately captured.

- When pixels are under-exposed, we in theory could recover the luminance, but the SNR would be very low, so the luminance values are noisy.
  - Simply amplifying all the pixel values will not work because the noise gets amplified too. The SNR is the same (noisy).
  - Denoising has the effect of enhancing DR.
- When pixels are over-exposed, pixel values don’t reflect scene luminance.
Dynamic Range Issue

Problem: full luminance range isn’t accurately captured.

- When pixels are under-exposed, we in theory could recover the luminance, but the SNR would be very low, so the luminance values are noisy.
  - Simply amplifying all the pixel values will not work because the noise gets amplified too. The SNR is the same (noisy).
  - Denoising has the effect of enhancing DR.
- When pixels are over-exposed, pixel values don’t reflect scene luminance.
Dynamic Range Issue

Problem: full luminance range isn’t accurately captured.

- When pixels are under-exposed, we in theory could recover the luminance, but the SNR would be very low, so the luminance values are noisy.
  - Simply amplifying all the pixel values will not work because the noise gets amplified too. The SNR is the same (noisy).
  - Denoising has the effect of enhancing DR.
- When pixels are over-exposed, pixel values don’t reflect scene luminance.
HDR Imaging Through Neutral Density (ND) Filter

Original Scene

Graduated ND Filter

Filter more photons
Gradual blend
Filter less photons

Output Image

https://www.cambridgeincolour.com/tutorials/graduated-neutral-density-filters.htm
Tiffen 77CGND6 77mm Graduated 0.6 ND Filter. 4X exposure difference. High-end GND filters provide ~ 4 orders of magnitude of difference.
What’s Happening When Using a GDN Filter?

Pixel value vs. Photons
What’s Happening When Using a GDN Filter?
What’s Happening When Using a GDN Filter?

Pixel value

Photons
What’s Happening When Using a GDN Filter?

Pixel value

Photons
What’s Happening When Using a GDN Filter?
What’s Happening When Using a GDN Filter?
What’s Happening When Using a GDN Filter?

A bit cheating since we effectively change the scene luminance, reducing its DR.
What’s Happening When Using a GDN Filter?

A bit cheating since we effectively change the scene luminance, reducing its DR.

Main downsides: not very precise; could also shift the chromaticity of the color.
HDR Imaging Through Multiple Exposures

- Basic idea: take multiple images of the scene with varying exposure times, align them, and combine them together to generate the final image.
- Each exposure provides accurate luminance for some, not all, scene points.
- Each scene point will be mapped to a pixel in each exposure. The most accurate luminance information of the point is from a pixel that’s well exposed but not over-exposed.
- Better yet, we can average across images to reduce noise.
https://en.wikipedia.org/wiki/High-dynamic-range_imaging
HDR Imaging Through Multiple Exposures

- New pixel value = 80, ignore 255 from the second image since it’s saturated and can’t be used to derive the luminance information.
- New pixel Value = \((38 \times 4 + 162)/8 = 39.25\). Pixel values from both images can be used to derive the scene luminance; averaging them reduces noise.
- For each pixel, if not black nor saturated, convert to photon/second, and then average across frames. In the end, we can relative radiance for all pixels.

Since radiance and photon quantity is linear, can think of 80 as 80 photons.

- Photon/second: \(160/2 = 80\)
- Photon/second: \((38\times 4 + 162)/8 = 39.25\)
HDR Imaging Through Multiple Exposures

- New pixel value = 80, ignore 255 from the second image since it’s saturated and can’t be used to derive the luminance information.
- New pixel Value = (38*4+162)/8 = 39.25. Pixel values from both images can be used to derive the scene luminance; averaging them reduces noise.
- For each pixel, if not black nor saturated, convert to photon/second, and then average across frames. In the end, we can relative radiance for all pixels.

Can conceptually extend to multiple exposures, and apply the same process to all three color channels.
New pixel value = 80, ignore 255 from the second image since it’s saturated and can’t be used to derive the luminance information.

New pixel Value = \((38 \times 4 + 162)/8 = 39.25\). Pixel values from both images can be used to derive the scene luminance; averaging them reduces noise.

For each pixel, if not black nor saturated, convert to photon/second, and then average across frames. In the end, we can relative radiance for all pixels.
HDR Imaging Through Multiple Exposures

When working with sRGB images the radiance to pixel value curve is non-linear. Fit a curve to find the relationship (Recovering High Dynamic Range Radiance Maps from Photographs, SIGGRAPH 1997).

Other issues:
- Need to align images.
- Subjective to motion blur.

- Exposure is unknown, fit to find a smooth curve

Assuming unit radiance for each pixel

After adjusting radiances to obtain a smooth response curve
Figure 8:
(a) An actual photograph, taken with conventional print film at two seconds and scanned to PhotoCD.
(b) The high dynamic range radiance map, displayed by linearly mapping its entire dynamic range into the dynamic range of the display device.
(c) The radiance map, displayed by linearly mapping the lower of its dynamic range to the display device.
(d) A false-color image showing relative radiance values for a grayscale version of the radiance map, indicating that the map contains over five orders of magnitude of useful dynamic range.
(e) Rendering of the radiance map using adaptive histogram compression.
(f) Rendering of the radiance map using histogram compression and also simulating various properties of the human visual system, such as glare, contrast sensitivity, and scotopic retinal response. Images (e) and (f) were generated by a method described in [23]. Images (d-f) courtesy of Gregory Ward Larson.

Recovering High Dynamic Range Radiance Maps from Photographs, SIGGRAPH 1997
https://www.learnopencv.com/exposure-fusion-using-opencv-cpp-python/
The radiance map. You need high-precision data representation, e.g., floating-point, to accurately store the data in the radiance map.
HDR Imaging Recap

- **Scene**: High DR (natural) scenes
- **Camera**: Lower DR capturing device (e.g., camera)
- **Display**: Even lower DR output device (e.g., display, paper)
- **Print**:

- Key goal is to represent the relative luminance of scene points accurately across the entire luminance range.
- In principle, it’s easy to combine multiple exposures when raw pixel value vs. radiance is linear. In reality, the key is to capture this non-linearity.
- Averaging across exposures also serves denoising.
- GND can be used when accurate relative luminance isn’t required.
HDR Imaging Recap

- **Scene**: High DR (natural) scenes
- **Camera**: Lower DR capturing device (e.g., camera)
- **Display**
  - Even lower DR output device (e.g., display, paper)
- **Print**

- Key goal is to represent the relative luminance of scene points accurately across the entire luminance range.
- In principle, it’s easy to combine multiple exposures when raw pixel value vs. radiance is linear. In reality, the key is to capture this non-linearity.
- Averaging across exposures also serves denoising.
- GND can be used when accurate relative luminance isn’t required.
Tone Mapping: Problem 1

- HDR imaging allows us to accurately capture the full dynamic range of the scene, which could be ~20 stops.

- Typical Display: ~10 stops (LG 27UK850): the ratio between the brightness pixel and the darkest pixel is about 1024 under the same display power. You can dim the display, but that weakens all pixels and the DR is still the same.
  - HDR Display: 14.2 stops (UHD Alliance Certified HDR LCD display)
  - Paper: 6-8 stops (typical value); related to pigment density

- So we need to compress the contrast of ~100,000:1 to ~1000:1.
Tone Mapping: Problem 2

- Natural scenes in theory contain **continuous** luminance levels, whereas digital images inherently quantize continuous luminance levels to **discrete** levels.

- As an image is encoded in sRGB, only 8 bits, i.e., 256 levels, are available to represent the luminance levels. As a result, subtle contrast (luminance difference) might end up being quantized to the same level, losing the details when being displayed.

- Note that encoding color using higher bit-depth, e.g., 10 bits, alleviates the second problem but doesn’t affect/solve the first problem at all.
Another Perspective

- Some information is guaranteed to be lost. The goal is to exploit human vision system and light physics to generate images that mimic human’s expected visual response.

How to represent the full range of these numbers using only 8 bits per color channel, as required by sRGB (and most displays), while being visually pleasing?
Linear Mapping

- Map the highest luminance pixel to 255 and lowest to 0. What do you think will happen for this image?
Linear Mapping

- Map the highest luminance pixel to 255 and lowest to 0. What do you think will happen?
Linear Mapping

- Map the highest value to 255 and lowest to 0. Why? Most of the luminance levels are clustered toward the low end.
Another Try of Linear Mapping

- Linearly map the darkest 0.1% luminance in the DR to the [0, 255] range, and blow out the rest (i.e., 255 for the rest). What do you think will happen for this image?
Another Try

- Linearly map the darkest 0.1% luminance in the DR to the [0, 255] range, and blow out the rest (i.e., 255 for the rest). What do you think will happen for this image?
Non-Linear Mapping

\[ L_{\text{display}} = \frac{L_{\text{world}}}{1 + L_{\text{world}}} \]

Then scale to pixel values.

\[ L_{\text{display}} = A L_{\text{world}}^\gamma \]

Gamma encoding. A is chosen to scale to desired pixel values.

Added after the lecture: the gamma encoding here is used purely as an engineering trick to boost the brightness. It’s technically not related to the gamma encoding we used before to reduce quantization error in color encoding.
Non-Linear Mapping

\[ L_{\text{display}} = \frac{L_{\text{world}}}{1 + L_{\text{world}}} \]

Then scale to pixel values.

\[ L_{\text{display}} = A L_{\text{world}}^\gamma \]

Gamma encoding. A is chosen to scale to desired pixel values.

Added after the lecture: the gamma encoding here is used purely as an engineering trick to boost the brightness. It’s technically not related to the gamma encoding we used before to reduce quantization error in color encoding.
Non-Linear Mapping

\[ L_{\text{display}} = \frac{L_{\text{world}}}{1 + L_{\text{world}}} \]

Then scale to pixel values.

\[ L_{\text{display}} = AL_{\text{world}}^\gamma \]

Gamma encoding. A is chosen to scale to desired pixel values.

Added after the lecture: the gamma encoding here is used purely as an engineering trick to boost the brightness. It’s technically not related to the gamma encoding we used before to reduce quantization error in color encoding.
Non-Linear

\[ L_{\text{display}} = \frac{L_{\text{world}}}{1 + L_{\text{world}}} \]

\[ L_{\text{display}} = AL_{\text{world}}^\gamma \]

Added after the lecture: the gamma encoding here is used purely as an engineering trick to boost the brightness. It’s technically not related to the gamma encoding we used before to reduce quantization error in color encoding.
Local Tone Mapping

- All the mapping schemes so far are *global* mapping, in that they apply the same operator to all the pixels. Different pixels, regardless of where they are in the scene, are mapped in the same way if they have the same radiance.

- Problem: since contrasts are globally reduced, local contrasts are diminished. Human eyes adapt to local field around the current fixation.

- What does it mean to preserve local contrast? Make sure in local regions brighter pixels look (much) brighter than dark pixels. Doesn’t mean the exactly same luminance difference or ratio needs to be maintained.

- Many local tone mapping schemes strive to preserve, sometimes even enhance, local contrast. We will give you some intuition.
Local Contrast is Important

- Intuitively, at an edge brighter color becomes brighter and dark color becomes darker. Our brain does this “contrast enhancement” so we can better discern details at edges.

Mach band illusion: the right edge of the same step appears to be darker than the left side in the same step.

https://www.cambridgeincolour.com/tutorials/unsharp-mask.htm
https://py3109.wordpress.com/whatsat/
Local Tone Mapping

- Advanced tone mapping schemes maintain local contrast (while reducing global contrast). Typically done for luminance only.
Local Tone Mapping

- Advanced tone mapping schemes maintain local contrast (while reducing global contrast). Typically done for luminance only.

Local Tone Mapping

- Advanced tone mapping schemes maintain local contrast (while reducing global contrast). Typically done for luminance only.
- Sometimes people take a standard DR image and enhance its local contrast, which gives an illusion of HDR as we feel that details are more pronounced. But the actual DR is unchanged.

Local Tone Mapping

- Advanced tone mapping schemes maintain local contrast (while reducing global contrast). Typically done for luminance only.
- Sometimes people take a standard DR image and enhance its local contrast, which gives an illusion of HDR as we feel that details are more pronounced. But the actual DR is unchanged.
- But be careful not to enhance local contrast too much. Otherwise we get the “HDR look”: images look very sharp and details are artificially exaggerated but overall the contrast feels “flat”.

“The HDR Look”?
“The HDR Look”? 
“The HDR Look”?
“The HDR Look”?
HDR and Tone Mapping Recap

- HDR imaging and tone mapping are related but distinct issues.
- HDR imaging concerns with reproducing the full range of the (relative) radiance in the scene; limited by cameras, which have much lower DR than scenes. HDR imaging generates HDR images, which carry full DR of the scene.
- Tone mapping concerns with how to present HDR images, either generated from HDR imaging or graphics renders, in LDR media (e.g., display/paper).
- Typical camera captures, even not through the HDR imaging process, have higher DR than your display; some amount of tone mapping is usually needed.
- Tone mapping predates HDR imaging, and was initially used in computer graphics. With global illumination techniques, graphics renders could generate HDR images earlier than HDR imaging algorithms were popularized.
Google Pixel 2’s HDR Pipeline (HDR+)

Summarizing, on a Qualcomm Snapdragon 810 not subject to thermal throttling the time required to produce an output photograph ranges from 2.5 to 4 seconds, depending on the number of frames in the burst. For a low light shot taking 4 seconds, this breaks down as 1 second to capture the frames, 500 ms for alignment, 1200 ms for merging, and 1600 ms for finishing. For a daylight shot taking 2.5 seconds, we measure 100 ms for capture, 250 ms for alignment, 580 ms for merging, and 1600 ms for finishing.

Viewfinder executed on hardware ISP. Real-time but no HDR. Main HDR pipeline written in Halide and executes in SIMD/multi-core CPUs. Viewfinder image different from the final output.
Tone Mapping in HDR+

Tone Mapping in HDR+

Global tone mapping operator. Output pixel value (y) depends only on the radiance (x).

Global tone mapping operator. Output pixel value \( y \) depends only on the radiance \( x \).

Local tone mapping. Pixels with same radiance could be mapped to different values. Heatmap shows the mapping distribution.

Live HDR+ on Pixel 4 and Pixel 4a

Goal: produce HDR image in live viewfinder. But full HDR+ algorithm is slow for live. Idea: use (fast) global tone mapping operator in individual small tiles. Per-tile curve is generated by a neural network.
Reference Material

- **Denoising**
  - Guided Image Filtering (ECCV 2020). Use an additional image to guide filtering.
  - Chapter 6.5, Image Processing for Embedded Devices: From CFA Data to Image-video Coding, discusses how various ISP stages affect noise. Chapter 6.9 talks about two kinds of denoising: sigma filtering (essentially Gaussian-like kernels) and bilateral filtering.

- **HDR digital camera systems**
  - Burst photography for high dynamic range and low-light imaging on mobile cameras (SIGGRAPH Asia 2006). In Google Pixel and Pixel 2.
  - Pixel Visual Core: Google's Fully Programmable Image, Vision, and AI Processor For Mobile Devices (HotChips’30)
  - Live HDR+ and Dual Exposure Controls on Pixel 4 and 4a, uses HDRNet (SIGGRAPH 2017).
Reference Material

- **HDR Imaging**
  - **GND filter**: it talks about different filter parameters.
  - *Recovering High Dynamic Range Radiance Maps from Photographs* (SIGGRAPH 1997). The original multi-exposure HDR paper; talks about how to infer luminance from pixel value when the relationship is non-linear. Worth reading very much.
  - Chapter 6.2.8, *Color Imaging: Fundamentals and Applications*, talks about how to derive the relationship between incident the refracted light luminance (equation we use in slide 23).
  - Chapter 6.4.7, *Physically-based Rendering*, talks about camera sensing from a radiometric perspective, deriving the energy incident on a pixel from incident luminance.
  - *High Dynamic Range Image Encodings*: talks about encoding HDR images. Note that this is **NOT** tone mapping. Encoding concerns with accurately recording the luminance information.
  - Satya Malik has a post on HDR imaging + tone mapping in OpenCV. High-level.
Reference Material

› **Tone Mapping**

› **Spatially Nonuniform Scaling Functions for High Contrast Images** (GI 1993). Early work on exploiting local contrast in tone mapping.


› **Fast Bilateral Filtering for the Display of High-Dynamic-Range Images** (SIGGRAPH 2002). The paper uses bilateral filtering in tone mapping.


› **Unsharp mask**. Could be used for sharpening and tone mapping. Good to understand the important of local contrast and how to enhance it.

› **Exposure Fusion**. Avoiding radiance map construction entirely; directly going from LDR photos to image encoding for LDR display.

› **Tone curves in Photoshop**. Helps you understand how contrast changes photos.