CSCI 1290: Comp Photo

Fall 2018 @ Brown University
James Tompkin

Many slides thanks to James Hays’ old CS 129 course, along with all of its acknowledgements.
Feedback from Project 0

• MATLAB:
  • Live Scripts != Scripts
  • *.mlx vs. *.m
  • Live scripts like Jupyter notebook

• Live scripts seem to take a really long time to compute.
Project 1 code

• How are people getting along?

• What issues are you facing? (Tell me now.)
Problem: Dynamic Range

The real world is high dynamic range.
Long Exposure

Real world

High dynamic range

Picture

0 to 255
Short Exposure

Real world

10^{-6} --- 10^{6}

High dynamic range

Picture

10^{-6} --- 10^{6}

0 to 255

Real world

High dynamic range

Picture

0 to 255

Real world

High dynamic range

Picture

0 to 255
Image

pixel \((312, 284) = 42\)

42 photons?
Camera Calibration

• **Geometric**
  - How pixel *coordinates* relate to *directions* in the world

• **Photometric**
  - How pixel *values* relate to *radiance* amounts in the world
Camera Calibration

• **Geometric**
  – How pixel coordinates relate to directions in the world in other images.

• **Photometric**
  – How pixel values relate to radiance amounts in the world in other images.
The Image Acquisition Pipeline

Scene radiance (W/sr/m²)

Lens

Radiance: radiant flux emitted, reflected, transmitted or received by a given surface. This is a *directional* quantity. (Often called ‘intensity’ – confusing.)

Watts per steradian per square meter
Irradiance: radiant flux (power) received by a surface per unit area. Not directional (collected at surface). (Often called ‘intensity’ – great, also confusing.)
The Image Acquisition Pipeline

Scene radiance \((\text{W/sr/m}^2)\)

Sensor irradiance \((\text{W/m}^2)\)

Sensor exposure \(\Delta t\)

Lens

Shutter

CCD

ADC

Remapping

Analog voltages

Digital values

Pixel values

Raw Image

JPEG Image
Varying Exposure
Camera is not a photometer!

- **Limited dynamic range**
  - Perhaps use multiple exposures?

- **Unknown, nonlinear response**
  - Not possible to convert pixel values to radiance

- **Solution:**
  - Recover response curve from multiple exposures,
    then reconstruct the *radiance map*
Camera is not a photometer!

- Key observation:
  - Scene is static, and while we might not know the absolute value of exposure at each pixel, we know that the scene radiance remains constant across the image sequence.
Recovering High Dynamic Range Radiance Maps from Photographs

Paul Debevec
Jitendra Malik

Computer Science Division
University of California at Berkeley

August 1997
Ways to vary exposure

- Shutter Speed (*)
- F/stop (aperture, iris)
- Neutral Density (ND) Filters
Shutter Speed

• Ranges: Canon D30: 30 to 1/4,000 sec.
  Sony VX2000: ¼ to 1/10,000 sec.

• Pros:
  Directly varies the exposure
  Usually accurate and repeatable

• Issues:
  Noise in long exposures
The Algorithm

Image series

Pixel Value $Z = f(\text{Exposure})$

Exposure = Radiance $\times \Delta t$

$\log \text{Exposure} = \log \text{Radiance} + \log \Delta t$
Imaging system response function

\[ \log \text{Exposure} = \log (\text{Radiance} \times \Delta t) \]

(CCD photon count)
The Algorithm

Image series

Assuming unit radiance for each pixel

Pixel Value $Z = f(\text{Exposure})$

Exposure = Radiance $\times \Delta t$

$\log \text{Exposure} = \log \text{Radiance} + \log \Delta t$
Response Curve

Assuming unit radiance for each pixel

After adjusting radiances to obtain a smooth response curve

 Pixel value

ln Exposure

 Pixel value

ln Exposure
The Math

• Let \( g(z) \) be the \textit{discrete} inverse response function

• For each pixel site \( i \) in each image \( j \), want:

\[
\ln \text{Radiance}_i + \ln \Delta t_j = g(Z_{ij})
\]

• Solve the overdetermined linear system:

\[
\sum_{i=1}^{N} \sum_{j=1}^{P} \left[ \ln \text{Radiance}_i + \ln \Delta t_j - g(Z_{ij}) \right]^2
\]

\( N = \# \text{ pixels} \)

\( P = \# \text{ images} \)
The Math

- Let $g(z)$ be the *discrete* inverse response function.
- For each pixel site $i$ in each image $j$, want:
  \[
  \ln \text{Radiance}_i + \ln \Delta t_j = g(Z_{ij})
  \]
- Solve the overdetermined linear system:
  \[
  \sum_{i=1}^{N} \sum_{j=1}^{P} \left[ \ln \text{Radiance}_i + \ln \Delta t_j - g(Z_{ij}) \right]^2 + \lambda \sum_{z=Z_{\text{min}}}^{Z_{\text{max}}} g''(z)^2
  \]

$N = \# \text{pixels}$
$P = \# \text{images}$

$g'' = \text{second derivative}$
$\lambda = \text{amount of smoothing}$
Results: Digital Camera

Kodak DCS460
1/30 to 30 sec

Recovered response curve

Pixel value vs. log Exposure
Reconstructed radiance map
Results: Color Film

• Kodak Gold ASA 100, PhotoCD
Recovered Response Curves

Red

Green

Blue

RGB
The Radiance Map
How do we store this?
Portable FloatMap (.pfm)

- 12 bytes per pixel, 4 for each channel

<table>
<thead>
<tr>
<th>sign</th>
<th>exponent</th>
<th>mantissa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Text header similar to Jeff Poskanzer’s .ppm image format:

```
PF
768 512
1
<binary image data>
```

Floating Point TIFF similar
Radiance Format
(.pic, .hdr)

32 bits / pixel

Red               Green               Blue             Exponent

(145, 215, 87, 149) = (145, 215, 87, 103) =
(145, 215, 87) * 2^(149-128) = (145, 215, 87) * 2^(103-128) =
(1190000, 1760000, 713000) (0.00000432, 0.00000641, 0.00000259)

**ILM’s OpenEXR (.exr)**

- 6 bytes per pixel, 2 for each channel, compressed

  - Sign
  - 5 bit exponent
  - 10 bit mantissa
  - Exponent
  - Mantissa

- Several lossless compression options, 2:1 typical
- Compatible with the “half” datatype in NVidia's Cg
- Supported natively on GeForce FX and Quadro FX

- Available at [http://www.openexr.net/](http://www.openexr.net/)
Now What?
Tone Mapping

- **How can we do this?**
  - Linear scaling?, thresholding? Suggestions?

![Tone Mapping Diagram]
The Radiance Map

Linearly scaled to display device
Simple Global Operator

• Compression curve needs to
  – Bring everything within range
  – Leave dark areas alone

• In other words
  – Asymptote at 255
  – Derivative of 1 at 0
Global Operator (Reinhart et al)

\[ L_{\text{display}} = \frac{L_{\text{world}}}{1 + L_{\text{world}}} \]
Global Operator Results
Reinhart Operator

Darkest 0.1% scaled to display device
What do we see?

Vs.
Issues with multi-exposure HDR

- Scene and camera need to be static
- Camera sensors are getting better and better
- Display devices are fairly limited anyway (although getting better).
High Dynamic Range as a look

High-dynamic range on smartphones

Apple, iPhone X/XS/XR
“We have a brand new feature we call smart HDR”

Always-on bracketed exposures:
- Even when you aren’t taking pictures, the camera saves to a buffer with varying exposures.
- Use that data to do HDR

Google Pixel phones

Burst of raw frames → Merged raw image → Final high-quality result
But often a global operator isn’t sufficient! I lose detail.

WHAT ABOUT LOCAL TONE MAPPING?
Local tone mapping / bilateral filter

Flickr user mapgoblin
Contributions

• Contrast reduction for HDR images
  – Local tone mapping
  – Preserves details
  – Fewer halos than naïve filtering
  – Fast with optimized implementation of bilateral filter (otherwise sort of slow)
High-dynamic-range (HDR) images

- CG Images

- Multiple exposure photo [Debevec & Malik 1997]

- HDR sensors
Contrast reduction

- Match limited contrast of the medium
- Preserve details
A typical photo

- Sun is overexposed
- Foreground is underexposed
Gamma ($\gamma$) compression

- $X \rightarrow X^\gamma$
- Colors are washed-out
Gamma compression on intensity

- Colors are OK, but details (intensity high-frequency) are blurred
Quick and dirty intensity / color separation

• Compute the intensity (I) by averaging the color channels.

• Compute the chrominance channels: (R/I, G/I, B/I)
Gamma compression on intensity

- Colors are OK, but details (intensity high-frequency) are blurred
Chiu et al. 1993

- Reduce contrast of low-frequencies
- Keep high frequencies
Quick low/high frequency computation

- Blur intensity to get low frequencies
- Subtract original from low frequency
The halo nightmare

• For strong edges
• Because they contain high frequency

Reduce low frequency
Our approach

• Do not blur across edges
• Non-linear filtering
Start with Gaussian filtering

- Here, input is a step function + noise

\[ J = f \ast I \]
Start with Gaussian filtering

- Spatial Gaussian filter

\[ J = f \otimes I \]
Start with Gaussian filtering

- Output is blurred

\[ J = f \otimes I \]
Gaussian filter as weighted average

- Weight of $\xi$ depends on distance to $x$
The problem of edges

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

$$J(x) = \sum_\xi f(x, \xi)$$
**Principle of Bilateral filtering**

[Tomasi and Manduchi 1998]

- Penalty $g$ on the intensity difference

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

[Tomasi and Manduchi 1998]

- Spatial Gaussian f

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

[Tomasi and Manduchi 1998]

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

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

![Diagram showing bilateral filtering process](image)
Normalization factor

[Tomasi and Manduchi 1998]

- $k(x) = \sum_\xi \left[ f(x, \xi) \right] \left[ g(I(\xi) - I(x)) \right]$
Bilateral filtering is non-linear

[Tomasi and Manduchi 1998]

• The weights are different for each output pixel

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

- Sometimes, not enough “similar” pixels
- Happens for specular highlights
- Can be detected using normalization $k(x)$
- Simple fix (average with output of neighbors)

Weights with high uncertainty

Uncertainty
Contrast reduction

Input HDR image

Contrast too high!
Contrast reduction

- Input HDR image
- Intensity
- Color
Contrast reduction

Input HDR image

Intensity

Fast
Bilateral
Filter

Color

Large scale
Contrast reduction

Input HDR image

Intensity

Fast Bilateral Filter

Large scale

Detail

Color
Contrast reduction

Input HDR image

Intensity

Fast
Bilateral
Filter

Large scale

Detail

Reduce contrast

Large scale

Color
Contrast reduction

Input HDR image

Intensity

Fast Bilateral Filter

Large scale

Reduce contrast

Preserve!

Detail

Large scale

Detail

Color
Contrast reduction

Input HDR image

Intensity

Color

Fast Bilateral Filter

Large scale

Detail

Reduce contrast

Preserve!

Output

Large scale

Detail

Color
Informal comparison

Gradient-space [Fattal et al.]

Bilateral [Durand et al.]

Photographic [Reinhard et al.]
Informal comparison

Gradient-space
[Fattal et al.]

Bilateral
[Durand et al.]

Photographic
[Reinhard et al.]
Project 2

• Recover high dynamic range radiance map from multiple uncalibrated images with known exposure time.

• Compress high dynamic range image into low dynamic range visualization using bilateral filter based decomposition.

• Solving linear systems!
  • You might need to brush up.
What about avoiding radiance map reconstruction entirely? *Exposure Fusion* [Mertens, 2007]
Exposure Fusion

Tom Mertens\textsuperscript{1} \hspace{1cm} Jan Kautz\textsuperscript{2} \hspace{1cm} Frank Van Reeth\textsuperscript{1}

\textsuperscript{1} EDM - Hasselt University \hspace{1cm} \textsuperscript{2} University College London, UK

https://dl.acm.org/citation.cfm?id=1338586 / https://mericam.github.io/
underexposed  overexposed  both
High Dynamic Range photography

Multi-exposure sequence
HDR Photography

- **HDR assembly**: LDRs $\rightarrow$ HDR
  - [Mann and Picard, IS&T ‘95]
  - [Debevec and Malik, SIGGRAPH ‘97]

- **Tone mapping**: HDR $\rightarrow$ LDR
  - [Durand and Dorsey, SIGGRAPH ‘02]
  - [Reinhard et al., SIGGRAPH ‘02]
  - [Fattal et al., SIGGRAPH ‘02]
  - [Li et al., SIGGRAPH ‘05]
  - [Lischinski et al., SIGGRAPH ‘06]

- **Exposure Fusion**: LDRs $\rightarrow$ LDR
Multi-exposure Assembly Tone mapping display

Response curve calibration bilateral gradient ...

High Dynamic Range photography
Idea: fuse multi-exposure sequence
Image Fusion

• Applications
  – Extended depth-of-field
    [Ogden et al. ‘85]
    [Agarwala et al., SIGGRAPH ‘04]
  – Multi-sensor photography
    [Burt et al. ‘93]
  – Non-photorealistic video
    [Raskar et al., NPAR ‘04]

• Exposure Fusion
  – Manual (Photoshop)
  – Image pyramid [Burt et al. ‘93]
Pyramid Fusion

[Ogden et al. '85, Burt et al. '93]
Pyramid Fusion

Pyramid decomposition

"multi-scale edge filter"

abs
max

reconstruct

[Ogden et al. ’85, Burt et al. ‘93]
Pyramid Fusion

Exposure fusion
Multi-exposure

Weight

Challenges:
1. Good weights
2. Good blending
weight = f(intensity)

Contrast:

weight = | h * image |,

$$\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0 
\end{bmatrix}$$
Multi-resolution blending

Naïve blending

Weights

Multi-resolution blending
Multi-resolution Blending

[Burt and Adelson, ACM TOG ‘83]
Multi-Resolution Blending
Multi-Resolution Blending

Laplacian pyramid

Gaussian pyramid

[Burt and Adelson, ACM TOG ‘83]
Results
Exposure fusion

Mean image

Exposure fusion
Well-exposedness

contrast + saturation
Pyramid fusion
[Ogden85; Burt93]

Exposure fusion

Single exposure
[Durand and Dorsey ‘02]

[Reinhard et al. ‘02]

[Li et al. ‘05]

Exposure Fusion
Flash/No Flash

Also see: [Eisemann and Durand ’04, Petschnigg et al. ’04, Agrawal et al. ’05]
Conclusion

• Simple, fast and flexible
• Pixel selection + blend
  – perceptual measures
  – multi-resolution blending
• Few parameters to tweak
• Not physically-based
• Future work
  – More measures
  – More applications
  – Real-time GPU implementation for video

- contrast
- saturation
- well-exposedness
Acknowledgments

• Photos:
  – Min Kim
  – Jesse Levinson
  – Jacques Joffre
  – Amit Agrawal

• European Regional Development Fund

• http://research.edm.uhasselt/~tmertens
  – Matlab code
http://research.edm.uhasselt.be/~tmertens
- Additional slides -
Extended DOF

Graphcuts
[Agrawala ’04]

Exposure Fusion
IR  visible  Exposure fusion
Maui Sunset

-2 stops

+2 stops

Mean image

Exposure fusion
Maui Sunset

Middle exposure

Exposure fusion
- End -