Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm
Lecture 1: Images and image filtering

Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm
Lecture 1: Images and image filtering

Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm
CS5670: Intro to Computer Vision
Noah Snavely

Lecture 1: Images and image filtering

Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm
Reading

- Szeliski, Chapter 3.1-3.2
Announcements

• You should have been invited to Piazza
• We will add students to CMS this week
Announcements

• Project 1 (Hybrid Images) will be released tomorrow or Wednesday
  – This project will be done solo
  – Other projects planned to be done in groups of 2
Announcements

• We provide a walkthrough for setting up a python environment for the project
• As a backup, we also have a course virtual machine (VM) for you to run the assignments
• The assignment also works on lab machines
What is an image?
What is an image?

We’ll focus on these in this class

(source: A. Efros)
What is an image?

- A grid (matrix) of intensity values

(common to use one byte per value: 0 = black, 255 = white)
What is an image?

• We can think of a (grayscale) image as a function, \( f \), from \( \mathbb{R}^2 \) to \( \mathbb{R} \):
  
  – \( f(x,y) \) gives the **intensity** at position \( (x,y) \)

  – A **digital** image is a discrete (**sampled, quantized**) version of this function
Image transformations

• As with any function, we can apply operators to an image

\[ g(x,y) = f(x,y) + 20 \]

\[ g(x,y) = f(-x,y) \]

• Today we’ll talk about a special kind of operator, *convolution* (linear filtering)
Filters

• Filtering
  – Form a new image whose pixels are a combination of the original pixels

• Why?
  – To get useful information from images
    • E.g., extract edges or contours (to understand shape)
  – To enhance the image
    • E.g., to remove noise
    • E.g., to sharpen and “enhance image” a la CSI
  – A key operator in Convolutional Neural Networks
Canonical Image Processing problems

• Image Restoration
  – denoising
  – deblurring

• Image Compression
  – JPEG, JPEG2000, MPEG

• Computing Field Properties
  – optical flow
  – disparity

• Locating Structural Features
  – corners
  – edges
Question: Noise reduction

• Given a camera and a still scene, how can you reduce noise?

Take lots of images and average them!

What’s the next best thing?

Source: S. Seitz
Image filtering

- Modify the pixels in an image based on some function of a local neighborhood of each pixel

<table>
<thead>
<tr>
<th>10</th>
<th>5</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

Local image data

Some function

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

Modified image data

Source: L. Zhang
Linear filtering

• One simple version of filtering: linear filtering (cross-correlation, convolution)
  – Replace each pixel by a linear combination (a weighted sum) of its neighbors

• The prescription for the linear combination is called the “kernel” (or “mask”, “filter”)

Local image data

<table>
<thead>
<tr>
<th>10</th>
<th>5</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

kernel

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Modified image data

<table>
<thead>
<tr>
<th>8</th>
</tr>
</thead>
</table>
Cross-correlation

Let $F$ be the image, $H$ be the kernel (of size $2k+1 \times 2k+1$), and $G$ be the output image.

$$G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i + u, j + v]$$

This is called a **cross-correlation** operation:

$$G = H \otimes F$$

- Can think of as a “dot product” between local neighborhood and kernel for each pixel.
Convolution

• Same as cross-correlation, except that the kernel is “flipped” (horizontally and vertically)

\[ G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i - u, j - v] \]

This is called a convolution operation:

\[ G = H \ast F \]

• Convolution is commutative and associative
Convolution

Adapted from F. Durand
Mean filtering

\[
H \ast F = G
\]
Mean filtering/Moving average
Mean filtering/Moving average

\[ F[x, y] \]

\[ G[x, y] \]
Mean filtering/Moving average

\[ F[x, y] \]  

\[ G[x, y] \]
Mean filtering/Moving average

\[ F[x, y] \]

\[ G[x, y] \]
Mean filtering/Moving average

\[ F[x, y] \]

\[ G[x, y] \]
Mean filtering/Moving average
Linear filters: examples

Original \* \[
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\end{array}
\] = Identical image

Source: D. Lowe
Linear filters: examples

Original

\[ \ast \]

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
\end{array}
\]

Shifted left
By 1 pixel

Source: D. Lowe
Linear filters: examples

Original  *  \[\frac{1}{9}\]  \[\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}\]  =  Blur (with a mean filter)

Source: D. Lowe
Linear filters: examples

Original

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

Source: D. Lowe
Sharpening

before

after

Source: D. Lowe
Smoothing with box filter revisited

Source: D. Forsyth
Gaussian Kernel

\[ G_\sigma = \frac{1}{2\pi \sigma^2} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \]

Source: C. Rasmussen
Gaussian filters

$\sigma = 1$ pixel  $\sigma = 5$ pixels  $\sigma = 10$ pixels  $\sigma = 30$ pixels
Mean vs. Gaussian filtering
Gaussian filter

• Removes “high-frequency” components from the image (low-pass filter)

• Convolution with self is another Gaussian

\[ \text{Convolving twice with Gaussian kernel of width } \sigma \]

\[ = \text{convolving once with kernel of width } \sigma \sqrt{2} \]

Source: K. Grauman
Sharpening revisited

• What does blurring take away?

original

-  

smoothed (5x5)

=  

detail

(Let’s add it back:)

original

+ \alpha

=  

sharpened

(This “detail extraction” operation is also called a high-pass filter)

Photo credit: https://www.flickr.com/photos/geezaweezer/16089096376/
Sharpen filter

\[ F = \alpha (F - F \ast H) = \]

- **Image**
- **Blurred image**

- **Scaled impulse**
- **Gaussian**
- **Laplacian of Gaussian**

Unit impulse (identity kernel with single 1 in center, zeros elsewhere)
“Optical” Convolution

Camera shake

= *


Bokeh: Blur in out-of-focus regions of an image.

Source: http://lullaby.homepage.dk/diy-camera/bokeh.html
Filters: Thresholding

\[ g(m, n) = \begin{cases} 
255, & f(m, n) > A \\ 
0, & \text{otherwise} 
\end{cases} \]
Linear filters

• Is thresholding a linear filter?
Questions?