Lecture 1: Images and image filtering

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

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

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

• Szeliski, Chapter 3.1-3.3
Announcements

• Project 1 (Hybrid Images) will be released early next week
  – This project will be done solo
  – Other projects planned to be done in groups of 2

• Project is in Python – we will provide skeleton code and instructions for setting up a Python environment for the project
What is an image?
What is an image?

We’ll focus on these in this class

(More on this process later)
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?

- 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 pixel values are a combination of the original pixel values

- 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, HEIF, 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

Local image data

<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>

Some function

Modified image data

7

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”)

<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>

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

<table>
<thead>
<tr>
<th>8</th>
</tr>
</thead>
</table>

Local image data  kernel  Modified image data

Source: L. Zhang
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
Mean filtering

\[
\begin{align*}
H \ast F &= G
\end{align*}
\]
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

$$F[x, y]$$

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

\[ F[x, y] \]

\[ G[x, y] \]
Linear filters: examples

Original

Source: D. Lowe
What image operation does filtering with this kernel perform?

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]
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

Source: D. Lowe
What image operation does filtering with this kernel perform? ([0, 0, 0; 1, 0, 0; 0, 0, 0])
Linear filters: examples

Original \* \begin{array}{c} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} = \text{Shifted left by 1 pixel}

Source: D. Lowe
Linear filters: examples

Original $\ast \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = $ Blur (with a mean filter)

Source: D. Lowe
Linear filters: examples

Original

\[ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \ast \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \text{Sharpening filter (accentuates edges)} \]

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

\[ \sigma = 1 \text{ pixel} \quad \sigma = 5 \text{ pixels} \quad \sigma = 10 \text{ pixels} \quad \sigma = 30 \text{ pixels} \]
Mean vs. Gaussian filtering
Gaussian filter

- Removes “high-frequency” components from the image (low-pass filter)
- Convolution with self is another Gaussian

\[ \ast \quad = \]

- 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 + α detail = 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

\[ \approx \]

Sharpen filter

unit impulse (identity kernel with single 1 in center, zeros elsewhere)
Sharpen filter

unfiltered

filtered
“Optical” convolution

**Camera shake**


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

Source: [https://www.diyphotography.net/diy_create_your_own_bokeh/](https://www.diyphotography.net/diy_create_your_own_bokeh/)
Filters: Thresholding

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

• Can thresholding be implemented with a linear filter?
Can thresholding be implemented with a linear filter?
Questions?