Why is computer vision difficult?

Viewpoint variation

Illumination

Scale

Intra-class variation

Motion (Source: S. Lazebnik)

Background clutter

Occlusion
Challenges: local ambiguity

But there are lots of cues we can exploit...

Source: S. Lazebnik
Bottom line

- Perception is an inherently ambiguous problem
  - Many different 3D scenes could have given rise to a particular 2D picture
  - We often need to use prior knowledge about the structure of the world

Course overview (tentative)

1. Low-level vision
   - image processing, edge detection, feature detection, cameras, image formation

2. Geometry and algorithms
   - projective geometry, stereo, structure from motion, Markov random fields

3. Recognition
   - face detection / recognition, category recognition, segmentation

4. Light, color, and reflectance

5. Advanced topics
Projects (tentative)

- Roughly five projects
- First one will be done solo, others in groups
- You can discuss the projects on a whiteboard, but all code must be your (or your group’s) own

- First project to be released today or tomorrow

Project: Image Scissors
Project: Feature detection and matching

Project: Creating panoramas
Project: Recognition

- Location recognition
- Face recognition
- Object category recognition

Grading

- Occasional quizzes (at the beginning of class)
- One prelim, one final exam

- Rough grade breakdown:
  - Quizzes: 5%
  - Midterm: 15%
  - Programming projects: 60%
  - Final exam: 15%
Late policy

• Two “late days” will be available for the semester
• Late projects will be penalized by 25% for each day it is late, and no extra credit will be awarded.

Questions?
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
Reading

- Szeliski, Chapter 3.1-3.2

What is an image?
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
  
  $g(x,y) = f(x,y) + 20$
  
  $g(x,y) = f(-x,y)$

- We’ll talk about a special kind of operator, convolution (linear filtering)
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?

Image filtering

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

Source: S. Seitz

Source: L. Zhang
Linear filtering

- One simple version: 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”)

\[
\begin{array}{ccc}
10 & 5 & 3 \\
4 & 6 & 1 \\
1 & 1 & 8 \\
\end{array}
= \begin{array}{ccc}
0 & 0 & 0 \\
0 & 0.5 & 0 \\
0 & 1 & 0.5 \\
\end{array} \rightarrow \begin{array}{c}
8 \\
\end{array}
\]

Local image data \hspace{1cm} kernel \hspace{1cm} 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
Mean filtering

\[ \begin{array}{ccc}
0 & 0 & 0 \\
0 & 90 & 0 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 0 \\
0 & 0 & 0 \\
\end{array} \]

* 

\[ \begin{array}{ccc}
0 & 0 & 0 \\
0 & 90 & 0 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 90 \\
0 & 90 & 0 \\
0 & 0 & 0 \\
\end{array} \]

= 

\[ \begin{array}{ccc}
0 & 10 & 20 \\
0 & 40 & 60 \\
0 & 60 & 60 \\
0 & 60 & 60 \\
0 & 60 & 60 \\
0 & 60 & 60 \\
0 & 60 & 60 \\
0 & 60 & 60 \\
0 & 0 & 0 \\
\end{array} \]

Linear filters: examples

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

* 

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

= 

Original Identical image

Source: D. Lowe
Linear filters: examples

Original * \( \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \) = 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

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

Source: D. Lowe

Sharpening

before  

after

Source: D. Lowe
Smoothing with box filter revisited

Gaussian Kernel

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

Source: C. Rasmussen
Gaussian filters

Mean vs. Gaussian filtering
Gaussian filter

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

\[
\begin{array}{ccc}
\ast & = & \\
\end{array}
\]

- Convolving twice with Gaussian kernel of width \( \sigma \) is convolving once with kernel of width \( \sigma \sqrt{2} \)

Source: K. Grauman

Sharpening revisited

- What does blurring take away?

\[
\begin{array}{ccc}
\text{original} & - & \text{smoothed (5x5)} \\
\end{array}
\]

Let’s add it back:

\[
\begin{array}{ccc}
\text{original} & + \alpha & \text{sharpened} \\
\end{array}
\]

Source: S. Lazebnik
Sharpen filter

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

- **Image**
- **Blurred Image**
- **Unit Impulse (Identity)**

- **Scaled Impulse**
- **Gaussian**
- **Laplacian of Gaussian**

---

Sharpen filter

- **Unfiltered**
- **Filtered**
“Optical” Convolution

Camera shake

= *


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

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

Questions?

• For next time:
  – Read Szeliski, Chapter 3.1-3.2