Lecture 16: Computer Vision Fundamentals
Topics

1. What is Computer Vision?
2. Applications of CV
3. Images as 2D arrays
4. Basic Image Processing
5. Image Filtering

• Many slides borrowed from James Hays, Irfan Essa, and others.

• Intro CV course: CS 4476
  • This spring: Judy Hoffmann
  • Coming Fall: Frank Dellaert
Motivation

• Robots need to act in the world
• One of the cheapest and richest sensors is a camera
• Unfortunately, understanding camera images is not easy
• Since the sixties, researchers have tried to tackle this problem
• Since 2012, deep learning has led to incredible progress
• Perception for robotics is following closely behind
1. What is Computer Vision?

Computer Graphics: Models to Images
Comp. Photography: Images to Images

Computer Vision: Images to Models
Computer Vision

Make computers understand images and video or any visual data.

What kind of scene?
Where are the cars?
How far is the building?
…
Vision is really hard

- Vision is an amazing feat of natural intelligence
  - Visual cortex occupies about 50% of Macaque brain
  - One third of human brain devoted to vision (more than anything else)

Is that a queen or a bishop?
Why computer vision matters

Safety
Health
Security
Comfort
Fun
Robotics
Ridiculously brief history of computer vision

• 1966: Minsky assigns computer vision as an undergrad summer project
• 1960’s: interpretation of synthetic worlds
• 1970’s: some progress on interpreting selected images
• 1980’s: ANNs come and go; shift toward geometry and increased mathematical rigor
• 1990’s: face recognition; statistical analysis in vogue
• 2000’s: broader recognition; large annotated datasets available; video processing starts
• 2010’s: Deep learning with ConvNets
• 2020’s: Widespread autonomous vehicles?
• 2030’s: robot uprising?
2. Applications of Computer Vision

- Examples of real-world applications
Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software
Object recognition (in mobile phones)

E.g. Google Lens
Face detection

- Digital cameras (you know these as “phones”) detect faces
Login without a password...

Fingerprint scanners on many new laptops, other devices

Face recognition systems now widely in use on smartphones
Sports

Sportvision first down line
Nice explanation on www.howstuffworks.com

http://www.sportvision.com/video.html
Special effects: motion capture

*Pirates of the Carribean*, Industrial Light and Magic
Augmented Reality and Virtual Reality

Magic Leap, Oculus, Hololens, etc.
Medical imaging

3D imaging
MRI, CT

Image guided surgery
Grimson et al., MIT
Smart cars

- **Mobileye**
  - Market Capitalization: 11 Billion dollars
  - Bought by Intel for 15 Billion dollars
Computer Vision in space

Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read “Computer Vision on Mars” by Matthies et al.

NASA's Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.
3. Images as 2D Arrays
Image Acquisition Pipeline

* Analog (incoming light) to digital (pixels)
A Digital Image (W X H)

width = 512 pixels
height = 512 pixels
512 X 512 pixels = 262,144 pixels = 0.26 MP image
A Digital Image!

- Numeric representation in 2-D (x and y)
- Referred to as $l(x,y)$ in continuous function form, $l(i,j)$ in discrete
- **Image Resolution**: expressed in terms of Width and Height of the image
A “picture element” that contains the light intensity at some location \((i,j)\) in the image.

\[ I(i,j) = \text{Some Numeric Value} \]
Characteristics of a Digital Image

- A two-dimensional array of pixels and respective intensities
- Image can be represented as a Matrix
- Intensity Values range from 0 = Black to 255 = White
Common data types

Data types used to store pixel values:
- unsigned char
- uint8
- unsigned char 8bit
- $2^n \ (2^1, 2^2, 2^4, 2^8, \text{ etc.})$
Digital Image Formats

Images can also be 16, 24, 32 bits-per-pixel:
  • 24 bits per pixel usually means 8 bits per color
  • At the two highest levels, the pixels themselves can carry up to 16,777,216 different colors

Common raster image formats:
  • GIF, JPG, PPM, TIF, BMP, etc.
Digital Image is a Function

100 120 121 122 30 40
120 120 121 122 70 40
60 50 40 41 7 8
100 120 121 122 1 0
200 120 200 122 12 14
200 220 225 250 30 40

Continuous Signal

Discrete Signal

Slide adapted from Steve Seitz and Aaron Bobick
Digital Image is a Function

\[ I(x, y) \]

<table>
<thead>
<tr>
<th>x or i</th>
<th>y or j</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 120 121 122 30 40</td>
<td></td>
</tr>
<tr>
<td>120 120 121 122 70 40</td>
<td></td>
</tr>
<tr>
<td>60 50 40 41 7 8</td>
<td></td>
</tr>
<tr>
<td>100 120 121 122 1 0</td>
<td></td>
</tr>
<tr>
<td>200 120 200 122 12 14</td>
<td></td>
</tr>
<tr>
<td>200 220 225 250 30 40</td>
<td></td>
</tr>
</tbody>
</table>

Slide adapted from Steve Seitz and Aaron Bobick
Digital Image is a Function

- Typically, the functional operation requires discrete values
  - Sample the two-dimensional (2D) space on a regular grid
  - Quantize each sample (rounded to “nearest integer”)
- Matrix of integer values (Range: 0-255)
Digital Image Statistics

- Image statistics - average, median, mode
  - Scope - entire image or smaller windows/regions
- Histogram - distribution of pixel intensities in the image
  - Can be separate for each channel, or region-based too
Color Digital Image: An Example

- Color image = 3 color channels (images, with their own intensities) blended together
- Makes 3D data structure of size: Width X Height X Channels
- Each pixel has therefore 3 intensities: Red (R), Green (G), Blue (B)

<table>
<thead>
<tr>
<th>Color</th>
<th>Red Channel</th>
<th>Green Channel</th>
<th>Blue Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Basic Image Processing

- Contrast
- Brightness
- Gamma
- Histogram equalization
- Arithmetic
- Compositing
Contrast

\[ g(x) = a \cdot f(x), \quad a=1.1 \]
Brightness

\[ g(x) = f(x) + b, \quad b=16 \]
Gamma correction

\[ g(x) = (f(x))^{1/\gamma} \]

- \( \gamma = 1.2 \)
Histogram Equalization

- Non-linear transform to make histogram flat
- Still a per-pixel operation $g(x) = h(f(x))$
## Point-Process: Pixel/Point Arithmetic

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>122</td>
<td>140</td>
<td>142</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>120</td>
<td>141</td>
<td>144</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>121</td>
<td>144</td>
<td>146</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>121</td>
<td>144</td>
<td>145</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>121</td>
<td>145</td>
<td>147</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

\[ \begin{array}{c|c|c|c|c} 
120 & 122 & 140 & 142 & 143 \\
121 & 120 & 141 & 144 & 147 \\
122 & 121 & 144 & 146 & 11 \\
125 & 121 & 144 & 145 & 10 \\
126 & 121 & 145 & 147 & 13 \\
\end{array} \]

\[ + \]

\[ \begin{array}{c|c|c|c|c} 
120 & 122 & 140 & 142 & 143 \\
121 & 80 & 40 & 144 & 10 \\
122 & 81 & 40 & 0 & 151 \\
125 & 80 & 40 & 0 & 152 \\
126 & 70 & 40 & 0 & 153 \\
\end{array} \]

\[ = \]

\[ \begin{array}{c|c|c|c|c} 
240 & 244 & 280 & 284 & 286 \\
121 & 200 & 181 & 288 & 157 \\
122 & 202 & 184 & 146 & 162 \\
125 & 201 & 184 & 145 & 164 \\
126 & 191 & 185 & 147 & 166 \\
\end{array} \]
Pixel/Point Arithmetic: An Example

Image 1

- Image 2

Image 1 - Image 2

Binary(Image 1 - Image 2)
Matte: an alpha image
aF
\[(1-a)B\]
KeyMix: aF + (1-a)B
5. Image Filtering

Image filtering: compute function of local neighborhood at each position

• Very important!
  • Enhance images
    • Denoise, resize, increase contrast, etc.
  • Extract information from images
    • Texture, edges, distinctive points, etc.
  • Detect patterns
    • Template matching
  • Deep Convolutional Networks
Example: box filter

$$g[\cdot, \cdot]$$

$$\frac{1}{9}$$

$$\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}$$

Slide credit: David Lowe (UBC)
Image filtering

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ g[\cdot, \cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Image filtering

\[ g[\cdot, \cdot] = \frac{1}{9} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \]

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l] \]
Image filtering

\[ g[\cdot, \cdot] = \frac{1}{9} \]

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Image filtering

\[ f[\ldots] \]

\[ g[\cdot,\cdot] = \frac{1}{9} \]

\[ h[\ldots] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Image filtering

\[ f[\cdot,\cdot] \]

\[ h[\cdot,\cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]

Credit: S. Seitz
Image filtering

$$f[\ldots]$$

$$g[\cdot,\cdot] \frac{1}{9}$$

$$h[\ldots]$$

$$h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]$$

Credit: S. Seitz
Image filtering

\[ f[\cdot, \cdot] \]

\[ g[\cdot, \cdot] \]

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

\[
h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]
\]

Credit: S. Seitz
Image filtering

\[ f[\ldots] \quad g[\cdot, \cdot] \]

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

\[
\begin{array}{cccccccc}
0 & 10 & 20 & 30 & 30 & 30 & 20 & 10 \\
0 & 20 & 40 & 60 & 60 & 60 & 40 & 20 \\
0 & 30 & 60 & 90 & 90 & 90 & 60 & 30 \\
0 & 30 & 50 & 80 & 80 & 90 & 60 & 30 \\
0 & 30 & 50 & 80 & 80 & 90 & 60 & 30 \\
0 & 30 & 50 & 80 & 80 & 90 & 60 & 30 \\
0 & 20 & 30 & 50 & 50 & 60 & 40 & 20 \\
10 & 20 & 30 & 30 & 30 & 30 & 20 & 10 \\
10 & 10 & 10 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Box Filter

What does it do?

• Replaces each pixel with an average of its neighborhood

• Achieve smoothing effect (remove sharp features)

\[
g[\cdot, \cdot] = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

Slide credit: David Lowe (UBC)
Smoothing with box filter
Median filters

• A Median Filter operates over a window by selecting the median intensity in the window.

• What advantage does a median filter have over a mean filter?

• Is a median filter a kind of convolution?
Comparison: salt and pepper noise

Mean  Gaussian  Median

3x3

5x5

7x7

© 2006 Steve Marschner • 58  Slide by Steve Seitz
Summary

1. **Computer Vision** defined
2. **Applications** of CV are plentiful!
3. Images are **2D arrays** of pixel values
4. **Basic image processing**: contrast, intensity, histogram eq., arithmetic
5. **Image filtering**: convolution (linear) and non-linear (median)