Cengiz Günay, Emory Univ.

Spring 2013
Done with games, except homework :)
Done with games, except homework :) 

Vision is one of our main perceptions

Computer vision is what robots use to understand their surrounding
Done with games, except homework :) 

- **Vision** is one of our main perceptions
- **Computer vision** is what robots use to understand their surrounding

3 lectures:
1. Object recognition (today)
2. 3D reconstruction
3. Motion analysis
Exit survey: Advanced Planning

- Why isn’t classical planning schema adequate for resource planning?
- What is the advantage gained in abstract plans by having surely-reachable versus potentially-reachable states?

Entry survey: Computer Vision I – Image Processing (0.25 points)

- List three specific tasks where computer vision would be desirable.
- What do you think are the major hurdles in computer vision?
A charge-coupled device (CCD) photo sensor array:
Focal Optics for Determining Distance and Size

See the videos, I’ll summarize:

\[ \frac{X}{Z} = \frac{x}{f} \]

What can we figure out from this?

Object’s distance (\(Z\)) & height (\(X\)) based on projection height (\(x\)) and focal distance (\(f\))
Focal Optics for Determining Distance and Size

See the videos, I’ll summarize:

\[ \frac{X}{Z} = \frac{x}{f} \]

What can we figure out from this?
Focal Optics for Determining Distance and Size

See the videos, I’ll summarize:

\[ \frac{X}{Z} = \frac{x}{f} \]

What can we can figure out from this?

- Object’s distance \((Z)\) & height \((X)\) based on projection height \((x)\) and focal distance \((f)\)
We All See a Perspective Projection

Vanishing points from parallel lines:
We All See a Perspective Projection

Vanishing points from parallel lines:
We All See a Perspective Projection

Vanishing points from parallel lines:
We All See a Perspective Projection

Vanishing points from parallel lines:

- Giant panda, or just close?
Object Recognition: How Hard Can It Be?

Problems:
- Rotation
- Scale
- Illumination
- Occlusion
- Viewpoint
- Deformation
Object Recognition: How Hard Can It Be?

Problems?
Object Recognition: How Hard Can It Be?

Problems?

- Rotation, scale, illumination, occlusion, viewpoint, deformation
Not Hard for Us

How does our brain do it?

Will have examples later.
Not Hard for Us
Not Hard for Us

Specularities
Cast shadow
Diffuse reflection, bright
Diffuse reflection, dark

Specularities
Diffuse reflection, dark
Cast shadow

How does our brain do it?
Will have examples later.
Not Hard for Us

Diffuse reflection, bright
Specularities
Diffuse reflection, dark
Cast shadow

How does our brain do it?
Not Hard for Us

Diffuse reflection, bright
Diffuse reflection, dark
Specularities
Cast shadow

How does our brain do it? Will have examples later.
Invariance is Crucial for Computer Vision

Must recognize objects **invariant** of their:
- Rotation, scale, illumination, occlusion, viewpoint, deformation
Must recognize objects invariant of their:
- Rotation, scale, illumination, occlusion, viewpoint, deformation

Let’s start by simplifying:
1. Greyscale (monochrome) images
2. Pixels can have values: 0...255
Even Terminator Has Monochrome Vision

**ANALYSIS: SERIES 1000 TERMINATOR PROTOTYPE**

- **T-1000 STATUS**: 83%
- **RATIO OF ATTACK**: D0933
- **A1**: F8367
- **A2**: G0894
- **DISTANCE 4FT**: A5
- **A6**: J0948
- **A7**: K8364
- **A8**: L3748
- **A9**: Z3864

**DEFENSE MODE LEVEL 69825**

**TARGET ACQUIRED**

**POSSIBILITY OF T-1000 TERMINATION:** 52%

**CAUTION: T-1000 CAPABLE OF KNIVES AND STABBING WEAPONS EQUIPED WITH HANDGUN VULNERABLE TO MOLTEN STEEL AND LIQUID NITROGEN**

**PRIMARY MISSION: ENSURE THE SURVIVAL OF JOHN CONNOR**
Extracting Features: Edge Detection

1. Spatial derivative?

2. Filter with mask:

\[
\begin{pmatrix}
1 \\
-1
\end{pmatrix}
\]
Extracting Features: Edge Detection

How to detect the vertical edge?

1. Spatial derivative?
2. Filter with mask:

\begin{bmatrix}
+1 \\
-1
\end{bmatrix}

Günay
Ch. 24, Computer Vision I – Object Reco
Spring 2013
Extracting Features: Edge Detection

How to detect the vertical edge?
How to detect the vertical edge?

1. Spatial derivative?

```
| 255 | 251 | 7  | 1  | 3  |
| 211 | 241 | 5  | 9  | 0  |
| 218 | 249 | 8  | 2  | 2  |
| 212 | 241 | 5  | 4  | 0  |
```

```
0...255
↑  ↑
black  white
```
Extracting Features: Edge Detection

How to detect the vertical edge?

1. Spatial derivative?
2. Filter with mask: \[ +1 \quad -1 \]
How to detect the vertical edge?

1. Spatial derivative?
2. Filter with mask: \[
\begin{pmatrix}
+1 & -1
\end{pmatrix}
\]
Extracting Features: Edge Detection

How to detect the vertical edge?

1. Spatial derivative?

Günay
Ch. 24, Computer Vision I – Object Reco
Spring 2013 11 / 27
How to detect the vertical edge?

1. **Spatial derivative?**
Edge Detection: Linear Filter

What we did is called \textit{convolution}:

\[
I \otimes g = I'
\]

For each pixel, we multiply by \textit{mask} and sum:

\[
I'(x, y) = \sum_{u, v} I(x-u, y-v) g(u, v)
\]

Does that equation look familiar? Perceptron? What are the weights? The mask, \(g\). What's the advantage? Works in parallel!
Edge Detection: Linear Filter

What we did is called **convolution**:

\[
I \otimes g = I'
\]

For each pixel, we multiply by **mask** and sum:

\[
l'(x, y) = \sum_{u,v} l(x - u, y - v) g(u, v)
\]
What we did is called \textit{convolution}:

\[ I \otimes g = I' \]

For each pixel, we multiply by \textit{mask} and sum:

\[ I'(x, y) = \sum_{u,v} I(x - u, y - v) g(u, v) \]

Does that equation look familiar?
Edge Detection: Linear Filter

What we did is called convolution:

\[ I \otimes g = I' \]

For each pixel, we multiply by mask and sum:

\[ I'(x, y) = \sum_{u,v} I(x - u, y - v) g(u, v) \]

Does that equation look familiar? Perceptron?
What we did is called \textit{convolution}:

\[
I \otimes g = I'
\]

For each pixel, we multiply by \textit{mask} and sum:

\[
l'(x, y) = \sum_{u,v} I(x-u, y-v) g(u, v)
\]

Does that equation look familiar? Perceptron?

- What are the weights?
Edge Detection: Linear Filter

What we did is called **convolution**: \[ I \otimes g = I' \]

For each pixel, we multiply by **mask** and sum:

\[ I'(x, y) = \sum_{u,v} I(x - u, y - v) g(u, v) \]

Does that equation look familiar? Perceptron?

- What are the weights? The mask, \( g \).
Edge Detection: Linear Filter

What we did is called convolution:

\[ I \otimes g = I' \]

For each pixel, we multiply by mask and sum:

\[ I'(x, y) = \sum_{u,v} I(x - u, y - v) g(u, v) \]

Does that equation look familiar? Perceptron?

- What are the weights?
  - The mask, \( g \).
- What’s the advantage?

Works in parallel!
Edge Detection: Linear Filter

What we did is called **convolution**:

$$ \mathbf{I} \otimes \mathbf{g} = \mathbf{I'} $$

For each pixel, we multiply by **mask** and sum:

$$ \mathbf{I'}(x, y) = \sum_{u,v} \mathbf{I}(x-u, y-v) \mathbf{g}(u,v) $$

Does that equation look familiar? Perceptron?

- What are the weights? The mask, \( \mathbf{g} \).
- What’s the advantage? Works in parallel!
Neurons Can Do It Faster?
Detect Only Vertical Edges?

What type of filter finds horizontal edges?

Quiz
Detect Only Vertical Edges?

What type of filter finds horizontal edges?
Horizontal and Vertical Gradients

Original:
Horizontal and Vertical Gradients

Vertical gradient:
Horizontal and Vertical Gradients

Horizontal gradient:
Combining Gradients

Horizontal mask gives vertical gradient ($l_x$) and vice versa:

$$l_x = l \otimes \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix}$$

$$l_y = l \otimes \begin{bmatrix} -1 \\ +1 \end{bmatrix}$$
Combining Gradients

Horizontal mask gives vertical gradient \((l_x)\) and vice versa:

\[
\begin{align*}
l_x &= l \otimes \begin{bmatrix} -1 & +1 \end{bmatrix} \\
l_y &= l \otimes \begin{bmatrix} -1 \\ +1 \end{bmatrix}
\end{align*}
\]

How to combine them?
Combining Gradients

Horizontal mask gives vertical gradient ($I_x$) and vice versa:

$$I_x = I \otimes \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix}$$

$$I_y = I \otimes \begin{bmatrix} -1 \\ +1 \end{bmatrix}$$

How to combine them?

$$E = \sqrt{I_x^2 + I_y^2}$$
Horizontal and Vertical Gradients

Combined gradients:
Horizontal and Vertical Gradients

Original:
Horizontal and Vertical Gradients

Horizontal gradient:
Horizontal and Vertical Gradients

Vertical gradient:
Canny Edge Detector is Uncanny!

Combined gradients:
Canny Edge Detector is Uncanny!

Canny edge detector (by John Canny):
Other Edge Detection Masks

Sobel masks

\[
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{bmatrix}
\quad
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
\]

Prewitt masks

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

Kirsh masks

\[
\begin{bmatrix}
5 & -3 & -3 \\
5 & 0 & -3 \\
5 & -3 & -3 \\
\end{bmatrix}
\quad
\begin{bmatrix}
5 & 5 & 5 \\
-3 & 0 & -3 \\
-3 & -3 & -3 \\
\end{bmatrix}
\]
A Gaussian Mask?

What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. Blur
5. Sharpen

What's the Point of Blurring Images?

1. Downsampling
2. Noise reduction
A Gaussian Mask?

What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. Blur
5. Sharpen
A Gaussian Mask?

What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. **Blur**
5. Sharpen

What’s the Point of Blurring Images?
A Gaussian Mask?

What will it do?
1. Edge filter
2. Dot filter
3. Corner
4. **Blur**
5. Sharpen

What’s the Point of Blurring Images?
1. Downsampling
A Gaussian Mask?

What will it do?
1. Edge filter
2. Dot filter
3. Corner
4. **Blur**
5. Sharpen

What’s the Point of Blurring Images?
1. Downsampling
2. Noise reduction
Gaussian Mask in Action

Canny filter:
Gaussian Mask in Action

Canny with Gaussian:
Tricks with Linear Filters

\[ l' = l \otimes f \otimes g \]

where

- \( f \) is Gaussian mask and
- \( g \) is gradient mask.
Tricks with Linear Filters

\[ l' = l \otimes f \otimes g \]

where

\( f \) is Gaussian mask and
\( g \) is gradient mask.

Does the order matter?

No. Linear operations are transitive.

\[ l' = l \otimes g \otimes f \]

Can we combine them?

Yes. We'll get a new linear mask/kernel.

\[ l' = l \otimes (f \otimes g) \]
Tricks with Linear Filters

\[ l' = l \otimes f \otimes g \]

where

- \( f \) is Gaussian mask and
- \( g \) is gradient mask.

Does the order matter? **No.** Linear operations are transitive.

\[ = l \otimes g \otimes f \]
Tricks with Linear Filters

\[ l' = l \otimes f \otimes g \]

where

- \( f \) is Gaussian mask and
- \( g \) is gradient mask.

Does the order matter? **No.** Linear operations are transitive.

\[ = l \otimes g \otimes f \]

Can we combine them?
Tricks with Linear Filters

\[ l' = l \otimes f \otimes g \]

where

- \( f \) is Gaussian mask and
- \( g \) is gradient mask.

Does the order matter? **No.** Linear operations are transitive.

\[ = l \otimes g \otimes f \]

Can we combine them? **Yes.** We’ll get a new linear mask/kernel.

\[ = l \otimes (f \otimes g) \]
Gaussian Mask Combined with Gradient
Neurons Are Doing Exactly That!

J Jin, Y Wang, HA Swadlow & JM Alonso (2011)

"Population receptive fields of ON and OFF thalamic inputs to an orientation column in visual cortex"

Corner Detection
Corner Detection

Harris Corner Detector

\[ \sum (I_x)^2 \rightarrow \text{LARGE} \]

\[ \sum (I_y)^2 \rightarrow \text{LARGE} \]
Corner Detection

\[
\begin{pmatrix}
\Sigma I_x^2 & \Sigma I_x I_y \\
\Sigma I_x I_y & \Sigma I_y^2
\end{pmatrix} \sim \text{EIGENVALUES}
\]
Modern Feature Detectors

They are:

1. Localizable
2. Unique signatures
Modern Feature Detectors

They are:

1. Localizable
2. Unique signatures

Two major algorithms:

1. HOG: Histogram of Oriented Gradients
2. SiFT: Scale-invariant Feature Transform