Detecting corners
Overview of today’s lecture

• Why you want to find features.
• Harris corner detector.
• Multi-scale detection.
• Multi-scale blob detection.
Planar object instance recognition

Database of planar objects

Instance recognition
3D object recognition

Database of 3D objects

3D objects recognition
Recognition under occlusion
Location Recognition
Robot Localization
Image matching
Where are the corresponding points?
Challenges: Invariance

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …
Two Problems for Features

Feature detection
Two Problems for Features

Feature detection
Feature descriptor
Two Problems for Features

Feature detection
Feature descriptor
What makes a good feature?

Zoom-in demo
Want uniqueness

Look for unusual image regions
  • Lead to unambiguous matches in other images

How to define “unusual”? 
Local measures of uniqueness

Consider a small window of pixels

• Where are features good and bad?
Local measures of uniqueness

Consider a small window of pixels

• Where are features good and bad?

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.
Feature detection

Uniqueness =
How does it change when shifted by a *small amount*?

“flat” region:
no change in all directions

“edge”:
no change along the edge direction

“corner”:
significant change in all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.
Feature detection

Define

\[ E(u,v) = \text{amount of change when you shift the window by (u,v)} \]

\[ E(u,v) \text{ is small for all shifts} \]

\[ E(u,v) \text{ is small for some shifts} \]

\[ E(u,v) \text{ is small for no shifts} \]

We want \( \min_{(u,v)} E(u,v) \) to be ______
Feature detection: the math

Consider shifting the window $W$ by $(u,v)$

- how do the pixels in $W$ change?
- compare each pixel before and after by Sum of the Squared Differences (SSD)
- this defines an SSD “error” $E(u,v)$:

$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$
Small motion assumption

Taylor Series expansion of I:

\[ I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms} \]

If the motion \((u,v)\) is small, then first order approx is good

\[ I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \]

\[ \approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \]

shorthand: \( I_x = \frac{\partial I}{\partial x} \)

Plugging this into the formula on the previous slide…
Feature detection: the math

Consider shifting the window $W$ by $(u,v)$

- how do the pixels in $W$ change?
- compare each pixel before and after by summing up the squared differences
- this defines an “error” of $E(u,v)$:

$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$

$$\simeq \sum_{(x,y) \in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2$$

$$\simeq \sum_{(x,y) \in W} \left[ [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$
Feature detection: the math

This can be rewritten:

\[ E(u, v) = [u \ v] \left( \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \]
Feature detection: the math

This can be rewritten:

\[
E(u, v) = [u \ v] \left( \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}
\]

Which \([u \ v]\) maximizes \(E(u,v)\)?

Which \([u \ v]\) minimizes \(E(u,v)\)?
Feature detection: the math

This can be rewritten:

\[ E(u, v) = [u \ v] \left( \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \]

Which \([u \ v]\) maximizes \(E(u,v)\)?

Which \([u \ v]\) minimizes \(E(u,v)\)?
Feature detection: the math

This can be rewritten:

\[ E(u, v) = \begin{bmatrix} u \\ v \end{bmatrix} \left( \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \]

Eigenvector \( x_+ \) with the largest eigen value?

Eigenvector \( x_- \) with the smallest eigen value?
Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix \( A \) are the vectors \( x \) that satisfy:

\[
Ax = \lambda x
\]

The scalar \( \lambda \) is the **eigenvalue** corresponding to \( x \)

- The eigenvalues are found by solving:

\[
\text{det}(A - \lambda I) = 0
\]

- In our case, \( A = H \) is a 2x2 matrix, so we have

\[
\text{det} \begin{bmatrix}
h_{11} - \lambda & h_{12} \\
h_{21} & h_{22} - \lambda
\end{bmatrix} = 0
\]

- The solution:

\[
\lambda_{\pm} = \frac{1}{2} \left[ (h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2} \right]
\]
Feature detection

Local measure of feature uniqueness

- \( E(u,v) = \) amount of change when you shift the window by \((u,v)\)

We want \( \min_{(u,v)} E(u,v) \) to be large

\[
\min_{(u,v)} E(u,v) = \sqrt{u^2 + v^2} \cdot \lambda_-
\]
Eigenvalues of $\mathbf{H}$

$I$

$\lambda_+$

$\lambda_-$
Eigenvalues of $\mathbf{H}$

$I$   $\lambda_+$   $\lambda_-$
Feature detection summary

Here’s what you do

- Compute the gradient at each point in the image
- Create the $H$ matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ($\lambda_+ >$ threshold)
- Choose those points where $\lambda_-$ is a local maximum as features
Feature detection summary

Here’s what you do

• Compute the gradient at each point in the image
• Create the $H$ matrix from the entries in the gradient
• Compute the eigenvalues.
• Find points with large response ($\lambda_\text{max} >$ threshold)
• Choose those points where $\lambda_\text{max}$ is a local maximum as features

Called “non-local max suppression”
The Harris operator

\( \lambda_- \) is a variant of the “Harris operator” for feature detection

\[
f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}
= \frac{\text{determinant}(H)}{\text{trace}(H)}
= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>Flat</td>
</tr>
</tbody>
</table>
The Harris operator

\( \lambda_- \) is a variant of the “Harris operator” for feature detection

\[
    f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+} = \frac{\text{determinant}(H)}{\text{trace}(H)} = \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
</tr>
</tbody>
</table>

Flat
The Harris operator

\( \lambda_- \) is a variant of the “Harris operator” for feature detection

\[
f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}
= \frac{\text{determinant}(H)}{\text{trace}(H)}
= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>
The Harris operator

λ_ is a variant of the “Harris operator” for feature detection

\[
f = \frac{\lambda_-\lambda_+}{\lambda_- + \lambda_+} = \frac{\text{determinant}(H)}{\text{trace}(H)} = \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>λ_+</th>
<th>λ_-</th>
<th>f</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
<td>Flat</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
<td>Edge</td>
</tr>
</tbody>
</table>
The Harris operator \( \lambda_- \) is a variant of the “Harris operator” for feature detection.

\[
f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+} = \frac{\text{determinant}(H)}{\text{trace}(H)} = \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>2.5</td>
<td>3</td>
<td>1.36</td>
</tr>
</tbody>
</table>

Flat

Edge

?
The Harris operator

$\lambda_-$ is a variant of the “Harris operator” for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{\text{determinant}(H)}{\text{trace}(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

<table>
<thead>
<tr>
<th>$\lambda_+$</th>
<th>$\lambda_-$</th>
<th>$f$</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
<td>Flat</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
<td>Edge</td>
</tr>
<tr>
<td>2.5</td>
<td>3</td>
<td>1.36</td>
<td>Corner</td>
</tr>
</tbody>
</table>
The Harris operator

\( \lambda_\cdot \) is a variant of the “Harris operator” for feature detection

\[
f = \frac{\lambda_\cdot \lambda_+}{\lambda_- + \lambda_+} = \frac{\text{determinant}(H)}{\text{trace}(H)} = \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>2.5</td>
<td>3</td>
<td>1.36</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2.73</td>
</tr>
</tbody>
</table>

Flat
Edge
Corner
?
The Harris operator

\( \lambda_+ \) is a variant of the “Harris operator” for feature detection

\[
\begin{align*}
f &= \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+} \\
    &= \frac{\text{determinant}(H)}{\text{trace}(H)} \\
    &= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\end{align*}
\]

<table>
<thead>
<tr>
<th>( \lambda_+ )</th>
<th>( \lambda_- )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.02</td>
<td>0.012</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>2.5</td>
<td>3</td>
<td>1.36</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2.73</td>
</tr>
</tbody>
</table>
The Harris operator

\( \lambda_- \) is a variant of the “Harris operator” for feature detection

\[
f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+} = \frac{\text{determinant}(H)}{\text{trace}(H)} = \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}
\]

- The *trace* is the sum of the diagonals, i.e., \( \text{trace}(H) = h_{11} + h_{22} \)
- Very similar to \( \lambda_- \) but less expensive (no square root)
- Called the “Harris Corner Detector” or “Harris Operator”
- Lots of other detectors, this is one of the most popular
The Harris operator

Harris operator

$\lambda_-$
Harris detector example
f value (red high, blue low)
Threshold ($f > \text{value}$)
Find local maxima of f
Harris features (in red)
Harris corner response is invariant to rotation

Ellipse rotates but its shape (eigenvalues) remains the same

Corner response R is invariant to image rotation
Harris corner response is invariant to intensity changes

Partial invariance to *affine intensity* change

- Only derivatives are used $\Rightarrow$ invariance to intensity shift $I \rightarrow I + b$

- Intensity scale: $I \rightarrow a \cdot I$
The Harris detector is not invariant to changes in …
The Harris corner detector is not invariant to scale.
Multi-scale detection
Scale invariant detection

Suppose you’re looking for corners

Key idea: find scale that gives local maximum of $f$
  - $f$ is a local maximum in both position and scale
Automatic scale selection

Lindeberg et al., 1996

Slide from Tinne Tuytelaars
Automatic scale selection
Automatic scale selection

$f(I_{h\ldots m}(x,\sigma))$
Automatic scale selection
Automatic scale selection
Automatic scale selection

\[ f(I_{h\cdot h}(x,\sigma)) \]
Automatic scale selection

\[ f(I_{h...i_m}(x, \sigma)) \]  

\[ f(I_{h...i_m}(x', \sigma')) \]
References

Basic reading:
• Szeliski textbook, Sections 4.1.