Lecture 10: Harris Corner Detector
Announcements

• HW 1 will be out early next week

• Quiz 1 graded
Finding Corners

• Key property: in the region around a corner, image gradient has two or more dominant directions
• Corners are repeatable and distinctive

Feature extraction: Corners

9300 Harris Corners Pkwy, Charlotte, NC

Slides from Rick Szeliski, Svetlana Lazebnik, and Kristin Grauman
Local measures of uniqueness

Suppose we only consider a small window of pixels

– What defines whether a feature is a good or bad candidate?
Corner Detection: Basic Idea

- We should easily recognize the point by looking through a small window.
- Shifting a window in any direction should give a large change in intensity.

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

Source: A. Efros
Harris corner 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 (SSD)
- this defines an SSD “error” $E(u,v)$:

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

Change in appearance of window for the shift \([u, v]\):

\[
E(u, v) = \sum_{x,y} w(x, y) \left[ I(x+u, y+v) - I(x, y) \right]^2
\]

We want to find out how this function behaves for small shifts

\[ E(u, v) \]
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 approximation 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...
Corner detection: the math

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

- define an SSD “error” $E(u, v)$:

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

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

$$\approx \sum_{(x, y) \in W} [I_x u + I_y v]^2$$
Corner detection: the math

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

- define an SSD “error” $E(u,v)$:

$$E(u, v) \approx \sum_{(x,y) \in W} [I_x u + I_y v]^2$$

$$\approx Au^2 + 2Buv + Cv^2$$

$$A = \sum_{(x,y) \in W} I_x^2$$
$$B = \sum_{(x,y) \in W} I_x I_y$$
$$C = \sum_{(x,y) \in W} I_y^2$$

- Thus, $E(u,v)$ is locally approximated as a quadratic error function
Interpreting the second moment matrix

The surface $E(u,v)$ is locally approximated by a quadratic form. Let’s try to understand its shape.

$$E(u,v) \approx [u \ v] \ M \ [u \ v]$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$
The second moment matrix

The surface $E(u,v)$ is locally approximated by a quadratic form.

\[ E(u, v) \approx Au^2 + 2Buv + Cv^2 \]

\[ \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ A = \sum_{(x,y) \in W} I_x^2 \]

\[ B = \sum_{(x,y) \in W} I_x I_y \]

\[ C = \sum_{(x,y) \in W} I_y^2 \]

Let’s try to understand its shape.
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[
A = \sum_{(x,y) \in W} I_x^2
\]

\[
B = \sum_{(x,y) \in W} I_x I_y
\]

\[
C = \sum_{(x,y) \in W} I_y^2
\]

Horizontal edge: \( I_x = 0 \)

\[
M = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix}
\]
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ A = \sum_{(x, y) \in W} I_x^2 \]

\[ B = \sum_{(x, y) \in W} I_x I_y \]

\[ C' = \sum_{(x, y) \in W} I_y^2 \]

Vertical edge: \( I_y = 0 \)

\[ M = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \]
Interpreting the second moment matrix

Consider a horizontal “slice” of $E(u, v)$: $\begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.
Quick eigenvalue/eigenvector review

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

\[
A \mathbf{x} = \lambda \mathbf{x}
\]

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

- The eigenvalues are found by solving:

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

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

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

- The solution:

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

Once you know \( \lambda \), you find \( \mathbf{x} \) by solving

\[
\begin{bmatrix}
  h_{11} - \lambda & h_{12} \\
  h_{21} & h_{22} - \lambda
\end{bmatrix} \begin{bmatrix}
  x \\
  y
\end{bmatrix} = 0
\]
Consider a horizontal “slice” of $E(u, v)$: $[u \ v] \ M \ \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

Diagonalization of $M$:

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by $R$. 

- Direction of the fastest change: $(\lambda_{\text{max}})^{-1/2}$
- Direction of the slowest change: $(\lambda_{\text{min}})^{-1/2}$
Corner detection: the math

\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

Eigenvalues and eigenvectors of \( M \)

- Define shift directions with the smallest and largest change in error
- \( x_{\text{max}} \) = direction of largest increase in \( E \)
- \( \lambda_{\text{max}} \) = amount of increase in direction \( x_{\text{max}} \)
- \( x_{\text{min}} \) = direction of smallest increase in \( E \)
- \( \lambda_{\text{min}} \) = amount of increase in direction \( x_{\text{min}} \)

\[ M x_{\text{max}} = \lambda_{\text{max}} x_{\text{max}} \]
\[ M \cdot x_{\text{min}} = \lambda_{\text{min}} x_{\text{min}} \]
Interpreting the eigenvalues

Classification of image points using eigenvalues of $M$:

- **Corner**: $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions.
- **Edge**: $\lambda_2 >> \lambda_1$.
- **Flat** region: $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions.

$\lambda_1$ and $\lambda_2$ are eigenvalues of the matrix $M$. The classification is based on the values of these eigenvalues.
Corner detection: the math

How are $\lambda_{\text{max}}$, $x_{\text{max}}$, $\lambda_{\text{min}}$, and $x_{\text{min}}$ relevant for feature detection?

• Need a feature scoring function
Corner detection: the math

How are $\lambda_{\text{max}}$, $x_{\text{max}}$, $\lambda_{\text{min}}$, and $x_{\min}$ relevant for feature detection?

- Need a feature scoring function

Want $E(u,v)$ to be large for small shifts in all directions

- the minimum of $E(u,v)$ should be large, over all unit vectors $[u \ v]$
- this minimum is given by the smaller eigenvalue ($\lambda_{\min}$) of $M$
Corner detection summary

Here’s what you do

- Compute the gradient at each point in the image
- Create the $M$ matrix from the entries in the gradient
- Compute the eigenvalues
- Find points with large response ($\lambda_{\text{min}} > \text{threshold}$)
- Choose those points where $\lambda_{\text{min}}$ is a local maximum as features
Corner 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 ($\lambda_{\text{min}} > \text{threshold}$)
• Choose those points where $\lambda_{\text{min}}$ is a local maximum as features
The Harris operator

$\lambda_{\text{min}}$ is a variant of the “Harris operator” for feature detection

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\text{determinant}(H)}{\text{trace}(H)}$$

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

\[ R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \]

\[ \alpha: \text{constant (0.04 to 0.15)} \]
The Harris operator

Harris operator

$\lambda_{\min}$
Harris Detector [Harris88]

- **Second moment matrix**

\[
\mu(\sigma_I, \sigma_D) = g(\sigma_I) \* \begin{bmatrix}
  I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\
  I_xI_y(\sigma_D) & I_y^2(\sigma_D)
\end{bmatrix}
\]

1. Image derivatives (optionally, blur first)

\[
\det M = \lambda_1 \lambda_2 \\
\text{trace } M = \lambda_1 + \lambda_2
\]

2. Square of derivatives

3. Gaussian filter \( g(\sigma_I) \)

4. Cornerness function – both eigenvalues are strong

\[
har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))^2] = g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2
\]

5. Non-maxima suppression
Weighting the derivatives

• In practice, using a simple window $W$ doesn’t work too well

\[ H = \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \]

• Instead, we’ll weight each derivative value based on its distance from the center pixel

\[ H = \sum_{(x,y) \in W} w_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \]
Harris detector example
f value (red high, blue low)
Threshold ($f > \text{value}$)
Find local maxima of $f$
Harris features (in red)