CAP5415-Computer Vision
Lecture 4-Finding Features

Ulas Bagci
bagci@ucf.edu
Outline

- Sharpening
- Key-point Features
  - Harris corner detection
- Affine Invariance

- Read Szeliski, Chapter 4.
- Read Shah, Chapter 2.
- Read/Program CV with Python, Chapters 1 and 2.
- Next week, I will list potential projects.
Image Sharpening

Subtracting a blurred version of the image from the original image may lead to the sharpening
Image Sharpening

\[ I_{\text{sharp}} = I_{\text{original}} + \alpha I_{\text{edge}} \]
Motivation: **Matching**
Matching-Harder Case
Lecture 4 – Finding Features

- Multiscale low-level feature extraction
- Colours: Red, green, blue, yellow, etc.
- Intensity: On, off, etc.
- Orientations: 0°, 45°, 90°, 135°, etc.
- Other: Motion, junctions and terminators, stereo disparity, shape from shading, etc.

Input image

Attended location

Inhibition of return

Winner-take-all

Saliency map

Top-down attentional bias and training

Centre-surround differences and spatial competition

Feature maps

Feature combinations
Subsequent studies have shown that many complex actions can be recognized on the basis of such 'point-light displays', including

- facial expressions,
- Sign Language,
- arm movements,
- and various full-body actions.
Choosing interest points

Where would you tell your friend to meet you?

Slide Credit: James Hays
Features

• Automate object tracking
• Point matching for computing disparity
• Motion based segmentation
• Object Recognition
• 3D Object Reconstruction
• Robot Navigation
• Image Retrieval/Indexing
• ....
What is an interest point?

- **Expressive texture**
  - The point at which the direction of the boundary of object changes abruptly
  - Intersection point between two or more edge segments
What is an interest point?
Properties of Interest Points

- Detect all (or most) true interest points
- No false interest points
- Well localized
- Robust with respect to noise
- Efficient detection
Possible Approaches for Corner Detection

- Based on **brightness** of images
  - Usually image derivatives
- Based on **boundary** extraction
  - First step edge detection
  - Curvature analysis of edges
Correspondence Across Views

- Correspondence: matching points, patches, edges, or regions across images
Goals for Keypoints

Detect points that are *repeatable* and *distinctive*
Overview of Keypoint Matching

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors
Key Trade-offs

Detection of interest points

More Repeatable
- Robust detection
- Precise localization

More Interest Points
- Robust to occlusion
- Works with less texture

Description of patches

More Distinctive
- Minimize wrong matches

More Flexible
Local Features: Main Components

1) Detection: Identify the interest points

2) Description: Extract feature vector descriptor surrounding each interest point.

\[ \mathbf{x}_1 = \left[ x_1^{(1)}, \ldots, x_d^{(1)} \right] \]

\[ \mathbf{x}_2 = \left[ x_1^{(2)}, \ldots, x_d^{(2)} \right] \]

3) Matching: Determine correspondence between descriptors in two views

Kristen Grauman
Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.

No chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.
Goal: descriptor distinctiveness

- We want to be able to reliably determine which point goes with which.

- Must provide some invariance to geometric and photometric differences between the two views.
Some patches can be localized or matched with higher accuracy than others.
• Edge detectors often fail in corners. Why?
• Corner point can be recognized in a window
• Shifting a window in any direction should give a large change in intensity
• LOCALIZING and UNDERSTANDING shapes...
Basic Idea in Corner Detection

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions
Template Matching

\[
\begin{bmatrix}
-4 & 5 & 5 \\
-4 & 5 & 5 \\
-4 & -4 & -4
\end{bmatrix}
\quad \begin{bmatrix}
5 & 5 & 5 \\
-4 & 5 & -4 \\
-4 & -4 & -4
\end{bmatrix}
\]
Template Matching

Complete set of eight templates can be generated by successive 90 degree of rotations.
Template Matching

$$\begin{bmatrix}
-4 & 5 & 5 \\
-4 & 5 & 5 \\
-4 & -4 & -4 \\
\end{bmatrix}$$

$$\begin{bmatrix}
5 & 5 & 5 \\
-4 & 5 & -4 \\
-4 & -4 & -4 \\
\end{bmatrix}$$

Complete set of eight templates can be generated by successive 90 degree of rotations.

*Why the summation of filter is 0?*
Complete set of eight templates can be generated by successive 90 degree of rotations.

Why the summation of filter is 0?

Insensitive to absolute change in intensity!
Corner Detection using the Hessian Matrix

- Corners are characterized by high-curvature of intensity values.

\[
H(p) = \begin{bmatrix}
I_{xx}(p) & I_{xy}(p) \\
I_{xy}(p) & I_{yy}(p)
\end{bmatrix}
\]
Corner Detection using the **Hessian Matrix**

- Corners are characterized by high-curvature of intensity values.

\[
H(p) = \begin{bmatrix}
I_{xx}(p) & I_{xy}(p) \\
I_{xy}(p) & I_{yy}(p)
\end{bmatrix}
\]
Corner Detection using the Hessian Matrix

- Corners are characterized by high-curvature of intensity values.

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

Hessian Matrix at voxel \( p \).
Corner Detection using the Hessian Matrix

• Corners are characterized by high-curvature of intensity values.

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

Hessian Matrix at voxel p.  
Eigenvalues of the Hessian indicate corner features if both eigenvalues are large!
Corner Detection using the Hessian Matrix

- Corners are characterized by high-curvature of intensity values.

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

Hessian Matrix at voxel p.

Eigenvalues of the Hessian indicate corner features if both eigenvalues are large!

- One large, one small eigenvalues -> edge regions
- Two small eigenvalues indicate -> flat regions
Corner Detection using the **Hessian Matrix**

- Corners are characterized by high-curvature of intensity values.

\[
H(p) = \begin{bmatrix}
I_{xx}(p) & I_{xy}(p) \\
I_{xy}(p) & I_{yy}(p)
\end{bmatrix}
\]

- Eigenvalues of the Hessian indicate corner features if both eigenvalues are large!
  - One large, one small eigenvalues -> edge regions
  - Two small eigenvalues indicate -> flat regions

**Hessian matrix summarizes Distribution of gradients.**
Reminder: Eigenvalues/Eigenvectors

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

\[ \text{det}(H) = I_{xx}I_{yy} - I_{xy}^2 \]

\[ \text{Tr}(H) = I_{xx} + I_{yy} \]
Reminder: Eigenvalues/Eigenvectors

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

\[ \det(H) = I_{xx}I_{yy} - I_{xy}^2 \]

\[ \text{Tr}(H) = I_{xx} + I_{yy} \]

The eigenvalues of the matrix H are solutions of its characteristics polynomial

\[ \det(H - \lambda I_2) = 0 \]
Reminder: Eigenvalues/Eigenvectors

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

\[ \det(H) = I_{xx}I_{yy} - I_{xy}^2 \]
\[ \text{Tr}(H) = I_{xx} + I_{yy} \]

The eigenvalues of the matrix \( H \) are solutions of its characteristics polynomial

\[ \det(H - \lambda I_2) = 0 \]

\[ \det( \begin{bmatrix} I_{xx}(p) - \lambda & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) - \lambda \end{bmatrix} ) = 0 \]
Reminder: Eigenvalues/Eigenvectors

\[ H(p) = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \]

\[ \det(H) = I_{xx}I_{yy} - I_{xy}^2 \]

\[ \text{Tr}(H) = I_{xx} + I_{yy} \]

The eigenvalues of the matrix \( H \) are solutions of its characteristics polynomial

\[ \det(H - \lambda \mathbf{1}_2) = 0 \]

\[ \det\left( \begin{bmatrix} I_{xx}(p) - \lambda & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) - \lambda \end{bmatrix} \right) = 0 \]

\[ (I_{xx}(p) - \lambda)(I_{yy}(p) - \lambda) - I_{xy}(p)^2 = 0 \]
Harris and Stephens Corner Detector

• Instead of using *Hessian* of image $I$, use first derivative of smoothed $I$ (i.e., $L$)

$$G(p, \sigma) = \begin{bmatrix}
L_x^2(p, \sigma) & L_x(p, \sigma)L_y(p, \sigma) \\
L_x(p, \sigma)L_y(p, \sigma) & L_y^2(p, \sigma)
\end{bmatrix}$$
Harris and Stephens Corner Detector

• Instead of using Hessian of image I, use first derivative of smoothed I (i.e., L)

\[
G(p, \sigma) = \begin{bmatrix}
L_x^2(p, \sigma) & L_x(p, \sigma)L_y(p, \sigma) \\
L_x(p, \sigma)L_y(p, \sigma) & L_y^2(p, \sigma)
\end{bmatrix}
\]

• Instead of calculating eigenvalues, we will consider **cornerness** feature:
Harris and Stephens Corner Detector

• Instead of using *Hessian* of image I, use first derivative of smoothed I (i.e., L)

\[
G(p, \sigma) = \begin{bmatrix}
L_x^2(p, \sigma) & L_x(p, \sigma)L_y(p, \sigma) \\
L_x(p, \sigma)L_y(p, \sigma) & L_y^2(p, \sigma)
\end{bmatrix}
\]

• Instead of calculating eigenvalues, we will consider *cornerness* feature:

\[
\mathcal{H}(p, \sigma, \alpha) = \det(G) - \alpha \cdot \text{Tr}(G)
\]

*for small alpha (=1/25)*
Harris and Stephens Corner Detector

\[ \mathcal{H}(p, \sigma, \alpha) = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2) \]
Harris and Stephens Corner Detector

\[ \mathcal{H}(p, \sigma, \alpha) = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2) \]

- The same behavior as Hessian based detector, but now we directly use simple determinant and trace functions!

\[ \mathcal{H}(p, \sigma, \alpha) = \text{det}(G) - \alpha \text{Tr}(G) \]
Harris and Stephens Corner Detector

\[ \mathcal{H}(p, \sigma, \alpha) = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2) \]

- The same behavior as Hessian based detector, but now we directly use simple determinant and trace functions!

\[ \mathcal{H}(p, \sigma, \alpha) = \det(G) - \alpha \cdot \text{Tr}(G) \]

Other \textit{cornerness} measures:

- \( \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \) or \( \lambda_1 - \alpha \lambda_2 \)

Harmonic mean \quad \text{or} \quad \text{Triggs}
Lecture 4 – Finding Features
Lecture 4 – Finding Features
I_x

I_y

Square of derivatives

Gaussian Smoothing
Feature Extraction: **Corners**

9300 Harris Corners Pkwy, Charlotte, NC
Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.
Invariance and Covariance

• We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations
  – **Invariance**: image is transformed and corner locations do not change
  – **Covariance**: if we have two transformed versions of the same image, features should be detected in corresponding locations
Affine Transformation

translation
Affine Transformation

- Translation
- Rotation
Affine Transformation

- Translation
- Rotation
- Scaling
Affine Transformation

Corners are invariant to:
- Translation: ✔
- Rotation: ✔
- Scaling: ❌

E.g., a triangle remains a triangle after translation and rotation, but not after scaling.
Reminders

• **Deadline**: September 14, 2015  ←  PA1
• **Deadline**: September 24, 2015  ←  PA2