Lecture 11: Detectors and Descriptors

Professor Fei-Fei Li
Stanford Vision Lab
What we will learn today?

• Local invariant features
  – Motivation
  – Requirements, invariances
• Keypoint localization
  – Harris corner detector
• Scale invariant region selection
  – Automatic scale selection
  – Laplacian-of-Gaussian detector
  – Difference-of-Gaussian detector (Problem Set 3 (Q2))
  – Combinations
• Local descriptors
  – An intro
What we will learn today?

• Local invariant features
  – Motivation
  – Requirements, invariances
• Keypoint localization
  – Harris corner detector
• Scale invariant region selection
  – Automatic scale selection
  – Laplacian-of-Gaussian detector
  – Difference-of-Gaussian detector
  – Combinations
• Local descriptors
  – An intro
Motivation

• Global representations have major limitations
• Instead, describe and match only local regions
• Increased robustness to
  – Occlusions
  – Articulation
  – Intra-category variations
Application: Image Matching

by Diva Sian

by swashford

Slide credit: Steve Seitz
Harder Case

by Diva Sian

by scgbt

Slide credit: Steve Seitz
Harder Still?

NASA Mars Rover images

Slide credit: Steve Seitz
Answer Below  (Look for tiny colored squares)

NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)
Application: Image Stitching
Application: Image Stitching

• Procedure:
  – Detect feature points in both images
Application: Image Stitching

• Procedure:
  – Detect feature points in both images
  – Find corresponding pairs
Application: Image Stitching

• Procedure:
  – Detect feature points in both images
  – Find corresponding pairs
  – Use these pairs to align the images
General Approach

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

\[ d(f_A, f_B) < T \]
Common Requirements

• Problem 1:
  – Detect the same point *independently* in both images

No chance to match!

We need a repeatable detector!
Common Requirements

• Problem 1:
  – Detect the same point *independently* in both images

• Problem 2:
  – For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor!
Invariance: Geometric Transformations
Levels of Geometric Invariance
Invariance: Photometric Transformations

- Often modeled as a linear transformation:
  - Scaling + Offset
Requirements

• Region extraction needs to be **repeatable** and **accurate**
  – **Invariant** to translation, rotation, scale changes
  – **Robust or covariant** to out-of-plane (≈affine) transformations
  – **Robust** to lighting variations, noise, blur, quantization

• **Locality**: Features are local, therefore robust to occlusion and clutter.

• **Quantity**: We need a sufficient number of regions to cover the object.

• **Distinctiveness**: The regions should contain “interesting” structure.

• **Efficiency**: Close to real-time performance.
Many Existing Detectors Available

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Lindeberg ‘98], [Lowe ‘99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR [Tuytelaars & Van Gool ‘04]
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- Others…

- Those detectors have become a basic building block for many recent applications in Computer Vision.
Keypoint Localization

- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

⇒ Look for two-dimensional signal changes
Finding Corners

• Key property:
  – In the region around a corner, image gradient has two or more dominant directions

• Corners are repeatable and distinctive

Corners as Distinctive Interest Points

- **Design criteria**
  - We should easily recognize the point by looking through a small window (*locality*)
  - Shifting the window in *any direction* should give a *large change* in intensity (*good localization*)
Harris Detector Formulation

• Change of intensity for the shift \([u,v]\):

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

Window function

Shifted intensity

Intensity

Window function \(w(x,y) = \)

1 in window, 0 outside

or

Gaussian

Slide credit: Rick Szeliski
Harris Detector Formulation

- This measure of change can be approximated by:

\[ E(u, v) \approx [u \ v] \begin{bmatrix} u \\ v \end{bmatrix} \]

where \( M \) is a 2x2 matrix computed from image derivatives:

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

Gradient with respect to \( x \), times gradient with respect to \( y \)

Sum over image region – the area we are checking for corner

\[
M = \begin{bmatrix} \sum \frac{I_x I_x}{I_x I_y} & \sum \frac{I_x I_y}{I_y I_y} \\ \sum \frac{I_x I_y}{I_x I_y} & \sum \frac{I_y I_y}{I_y I_y} \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]
\]
Harris Detector Formulation

where \( M \) is a 2\( \times \)2 matrix computed from image derivatives:

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

Sum over image region – the area we are checking for corner

Gradient with respect to \( x \), times gradient with respect to \( y \)

\[
M = \left[ \frac{\sum I_x I_x}{\sum I_x I_y} \frac{\sum I_x I_y}{\sum I_y I_y} \right] = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]
\]
What Does This Matrix Reveal?

- First, let’s consider an axis-aligned corner:

\[
M = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix} = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}
\]
What Does This Matrix Reveal?

• First, let’s consider an axis-aligned corner:

\[
M = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix} = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}
\]

• This means:
  – Dominant gradient directions align with \(x\) or \(y\) axis
  – If either \(\lambda\) is close to 0, then this is not a corner, so look for locations where both are large.

• What if we have a corner that is not aligned with the image axes?
General Case

• Since $M$ is symmetric, we have

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

(Eigenvalue decomposition)

• We can visualize $M$ as an ellipse with axis lengths determined by the eigenvalues and orientation determined by $R$

![Diagram showing ellipse with axes lengths determined by eigenvalues and orientation determined by $R$.]
Interpreting the Eigenvalues

- Classification of image points using eigenvalues of $M$:
  - $\lambda_1$ and $\lambda_2$ are large; $\lambda_1 \sim \lambda_2$; $E$ increases in all directions
  - $\lambda_1 > \lambda_2$
  - $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions

Slide credit: Kristen Grauman
Corner Response Function

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

- Fast approximation
  - Avoid computing the eigenvalues
  - \( \alpha \): constant (0.04 to 0.06)
Window Function \( w(x, y) \)

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

- **Option 1: uniform window**
  - Sum over square window
  
  \[
  M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
  \]

  - Problem: not rotation invariant

- **Option 2: Smooth with Gaussian**
  - Gaussian already performs weighted sum

  \[
  M = g(\sigma) \ast \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
  \]

  - Result is rotation invariant
Summary: Harris Detector [Harris88]

- Compute second moment matrix (autocorrelation matrix)
  \[ M(\sigma_I, \sigma_D) = g(\sigma_I) \ast \begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma_I) \)

4. Cornerness function - two strong eigenvalues
  \[
  R = \det[M(\sigma_I, \sigma_D)] - \alpha[\text{trace}(M(\sigma_I, \sigma_D))]^2 \\
  = g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2
  \]

5. Perform non-maximum suppression
Harris Detector: Workflow
Harris Detector: Workflow
- computer corner responses R
Harris Detector: Workflow

- Take only the local maxima of R, where $R > \text{threshold}$
Harris Detector: Workflow
- Resulting Harris points
**Effect:** A very precise corner detector.

[Harris88]
Harris Detector – Responses [Harris88]
Harris Detector – Responses [Harris88]

• Results are well suited for finding stereo correspondences
Harris Detector: Properties

• Translation invariance?
Harris Detector: Properties

- Translation invariance
- Rotation invariance?

Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response $R$ is invariant to image rotation
Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?

Corner

All points will be classified as edges!

Not invariant to image scale!
What we will learn today?

• Local invariant features
  – Motivation
  – Requirements, invariances
• Keypoint localization
  – Harris corner detector
• Scale invariant region selection
  – Automatic scale selection
  – Laplacian-of-Gaussian detector
  – Difference-of-Gaussian detector (Problem Set 3 (Q2))
  – Combinations
• Local descriptors
  – An intro
From Points to Regions...

- The Harris and Hessian operators define interest points.
  - Precise localization
  - High repeatability

- In order to compare those points, we need to compute a descriptor over a region.
  - How can we define such a region in a scale invariant manner?

- *I.e. how can we detect scale invariant interest regions?*
Naïve Approach: Exhaustive Search

- Multi-scale procedure
  - Compare descriptors while varying the patch size

\[
d(f_A, f_B) = \text{Similarity measure}
\]
Naïve Approach: Exhaustive Search

- Multi-scale procedure
  - Compare descriptors while varying the patch size

\[ d(f_A, f_B) \neq \]
Naïve Approach: Exhaustive Search

• Multi-scale procedure
  – Compare descriptors while varying the patch size

\[
\begin{align*}
  f_A & \quad \text{Similarity measure} \quad f_B \\
  d(f_A, f_B) & \neq
\end{align*}
\]
Naïve Approach: Exhaustive Search

• Comparing descriptors while varying the patch size
  – Computationally inefficient
  – Inefficient but possible for matching
  – Prohibitive for retrieval in large databases
  – Prohibitive for recognition

\[ d(f_A, f_B) \]
Automatic Scale Selection

• Solution:
  – Design a function on the region, which is “scale invariant”
    *(the same for corresponding regions, even if they are at different scales)*

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  – For a point in one image, we can consider it as a function of region size (patch width)

\[ f \]

Image 1  \hspace{2cm} Image 2

\[ f \]

Region size  \hspace{2cm} Region size

scale = \( \frac{1}{2} \)
Automatic Scale Selection

- Common approach:
  - Take a local maximum of this function.
  - Observation: region size for which the maximum is achieved should be *invariant* to image scale.

**Important: this scale invariant region size is found in each image independently!**
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{i\ldots i_m}(x, \sigma)) \]

\[ f(I_{i\ldots i_m}(x', \sigma)) \]

Slide credit: Krystian Mikolajczyk
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Normalize: Rescale to fixed size
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector

\[ \Delta f = \nabla^2 f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2} \]
Characteristic Scale

- We define the *characteristic scale* as the scale that produces peak of Laplacian response

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]
Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]
Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]
Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]

⇒ List of \((x, y, \sigma)\)

Slide adapted from Krystian Mikolajczyk
LoG Detector: Workflow
LoG Detector: Workflow

sigma = 11.9912
LoG Detector: Workflow
Technical Detail

- We can efficiently approximate the Laplacian with a difference of Gaussians:

\[
L = \sigma^2 \left( G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)
\]  

(Laplacian)

\[
DoG = G(x, y, k\sigma) - G(x, y, \sigma)
\]  

(Difference of Gaussians)
Difference-of-Gaussian (DoG)

• Difference of Gaussians as approximation of the LoG
  – This is used e.g. in Lowe’s SIFT pipeline for feature detection.

• Advantages
  – No need to compute 2\textsuperscript{nd} derivatives
  – Gaussians are computed anyway, e.g. in a Gaussian pyramid.
Key point localization with DoG

- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses

Candidate keypoints: list of \((x, y, \sigma)\)
DoG – Efficient Computation

• Computation in Gaussian scale pyramid
Results: Lowe’s DoG

Slide credit: Bastian Leibe
Example of Keypoint Detection

(a) 233x189 image
(b) 832 DoG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures (removing edge responses)
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection

Slide adapted from Krystian Mikolajczyk
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian
   (same procedure with Hessian ⇒ Hessian-Laplace)
Summary: Scale Invariant Detection

- **Given:** Two images of the same scene with a large *scale difference* between them.
- **Goal:** Find *the same* interest points *independently* in each image.
- **Solution:** Search for *maxima* of suitable functions in *scale* and in *space* (over the image).

- Two strategies
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation

  - *These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).*
What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector
- Scale invariant region selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussian detector
  - Combinations
- Local descriptors
  - An intro
Local Descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?

⇒ Next lecture...
Local Descriptors

• We know how to detect points
• Next question:

How to *describe* them for matching?

Point descriptor should be:
1. Invariant
2. Distinctive
What we have learned today?

• Local invariant features
  – Motivation
  – Requirements, invariances
• Keypoint localization
  – Harris corner detector
• Scale invariant region selection
  – Automatic scale selection
  – Laplacian-of-Gaussian detector
  – Difference-of-Gaussian detector (Problem Set 3 (Q2))
  – Combinations
• Local descriptors
  – An intro
Supplementary materials

• Hessian detector
**Hessian Detector** [Beaudet78]

- Hessian determinant

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

**Note:** these are 2\textsuperscript{nd} derivatives!

*Intuition:* Search for strong derivatives in two orthogonal directions
• Hessian determinant

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

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

In Matlab:

\[
I_{xx} \cdot I_{yy} - (I_{xy})^2
\]
Effect: Responses mainly on corners and strongly textured areas.
Hessian Detector – Responses [Beaudet78]