Lecture 4

Feature Detectors and Descriptors: Corners, Blobs and SIFT

COS 429: Computer Vision

Slides adapted from: Szymon Rusinkiewicz, Jia Deng, Svetlana Lazebnik, Steve Seitz
Last time: edge detection

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, 0 \end{bmatrix} \]

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \end{bmatrix} \]
This time: keypoints

- Corners

- Blobs
Why Extract Keypoints?

- **Motivation: panorama stitching**
  - We have two images – how do we combine them?
Why Extract Keypoints?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract keypoints
Step 2: match keypoint features
Why Extract Keypoints?

• Motivation: panorama stitching
  – We have two images – how do we combine them?

Step 1: extract keypoints
Step 2: match keypoint features
Step 3: align images
Applications

- Keypoints are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition
Characteristics of Good Keypoints

- **Repeatability**
  - Can be found despite geometric and photometric transformations

- **Salience**
  - Each keypoint is distinctive

- **Compactness and efficiency**
  - Many fewer keypoints than image pixels

- **Locality**
  - Occupies small area of the image; robust to clutter and occlusion
Corners
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

Slide credit: S. Lazebnik
Corner Detection: Mathematics

Change in appearance of window $W$ for the shift $[u,v]$:

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

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

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

Change in appearance of window $W$ for the shift $[u, v]$:

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

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

$E(u, v)$
Corner Detection: Mathematics

- First-order Taylor approximation for small motions \([u, v]\):
  \[
  I(x + u, y + v) \approx I(x, y) + I_x u + I_y v
  \]
- Let's plug this into \(E(u,v)\):
  \[
  E(u, v) = \sum_{(x,y)\in W} [I(x + u, y + v) - I(x, y)]^2
  \]
  \[
  \approx \sum_{(x,y)\in W} [I(x, y) + I_x u + I_y v - I(x, y)]^2
  \]
  \[
  = \sum_{(x,y)\in W} [I_x u + I_y v]^2 = \sum_{(x,y)\in W} I_x^2 u^2 + 2I_x I_y uv + I_y^2 v^2
  \]
The quadratic approximation can be written as

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

where \( M \) is a \textit{second moment matrix} computed from image derivatives:

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

(the sums are over all the pixels in the window \( W \))
The surface $E(u,v)$ is locally approximated by a quadratic form. Let’s try to understand its shape.

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

$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix}$$

Specifically, in which directions does it have the smallest/greatest change?
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.
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.

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 $R$.

Slide credit: S. Lazebnik
Recap so far

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

- **$I(x, y)$**
- **$E(u, v)$**
- Direction of the fastest change
- Direction of the slowest change

$$\lambda_{\text{max}}^{-1/2} \quad \lambda_{\text{min}}^{-1/2}$$

$$E(3,2)$$
At an edge

\[
I(x, y)
\]

\[
E(u, v)
\]

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

- The direction along the edge results in no change
- \( \lambda_{\text{min}} \) is very small
At a corner

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

- All directions result in high change
- $\lambda_{\text{min}}$ is large
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”**: $\lambda_1 \gg \lambda_2$.
- **“Flat” region**: $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions.

$\lambda_1$ and $\lambda_2$ are large,
$\lambda_1 \sim \lambda_2$;
$E$ increases in all directions
Corner response function

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

\(\alpha\): constant

- **“Corner”**
  - \(R > 0\)

- **“Edge”**
  - \(R < 0\)

- **“Flat” region**
  - \(|R|\) small
The Harris corner detector

1. Compute partial derivatives at each pixel
2. Compute second moment matrix $M$ in a Gaussian window around each pixel:

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

The Harris corner detector

1. Compute partial derivatives at each pixel
2. Compute second moment matrix $M$ in a Gaussian window around each pixel
3. Compute corner response function $R$

Harris Detector: Steps

Two images of the same object
Harris Detector: Steps

Compute corner response $R$
The Harris corner detector

1. Compute partial derivatives at each pixel
2. Compute second moment matrix $M$ in a Gaussian window around each pixel
3. Compute corner response function $R$
4. Threshold $R$
5. Find local maxima of response function (nonmaximum suppression)

Harris Detector: Steps

Find points with large corner response: $R > \text{threshold}$

Slide credit: S. Lazebnik
Harris Detector: Steps

Take only the points of local maxima of $R$
Harris Detector: Steps

Slide credit: S. Lazebnik
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

Slide credit: S. Lazebnik
Affine intensity change

• Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

• Intensity scaling: $I \rightarrow aI$

Partially invariant to affine intensity change
• Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation
Image rotation

Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation
Corner location is not covariant to scaling!

All points will be classified as edges
Blobs
Feature detection with scale selection

- We want to extract features with characteristic scale that is covariant with the image transformation
Blob detection: Basic idea

- To detect blobs, convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting scale space
Blob detection: Basic idea

Find maxima and minima of blob filter response in space and scale.

Source: N. Snavely
Blob filter

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Recall: Edge detection

\[ f \]

\[ \frac{d}{dx} g \]

\[ f \ast \frac{d}{dx} g \]

Edge = maximum of derivative

Source: S. Seitz
Edge detection, Take 2

Edge = zero crossing of second derivative

Source: S. Seitz
From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

**Spatial selection**: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob.

Slide: S. Lazebnik
Scale selection

- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response.

- However, Laplacian response decays as scale increases:
The response of a derivative of Gaussian filter to a perfect step edge decreases as $\sigma$ increases.

\[
\frac{1}{\sigma \sqrt{2\pi}}
\]
Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as $\sigma$ increases.
- To keep response the same (scale-invariant), must multiply Gaussian derivative by $\sigma$.
- Laplacian is the second Gaussian derivative, so it must be multiplied by $\sigma^2$. 
Effect of scale normalization

Original signal

Unnormalized Laplacian response

Scale-normalized Laplacian response

maximum

Slide: S. Lazebnik
Blob detection in 2D

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

Scale-normalized: \[ \nabla^2_{\text{norm}} g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \]
Scale selection

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius \( r \)?
At what scale does the Laplacian achieve a maximum response to a binary circle of radius $r$?

For maximum response: align the zeros of the Laplacian with the circle.

The Laplacian in 2-D is given by (up to scale):

$$(x^2 + y^2 - 2\sigma^2)e^{-(x^2+y^2)/(2\sigma^2)}$$

Therefore, the maximum response occurs at $\sigma = r/\sqrt{2}$. 
We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center.

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
Scale-space blob detector
Scale-space blob detector

sigma = 11.9912
1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space
Scale-space blob detector: Example
Efficient implementation

- Laplacian of Gaussian can be approximated by Difference of Gaussians
- Assignment 1, question 3
Efficient implementation

From feature detection to feature description

- Scaled and rotated versions of the same neighborhood will give rise to blobs that are related by the same transformation.

- What to do if we want to compare the appearance of these image regions?
  - **Normalization**: transform these regions into same-size circles
  - Problem: rotational ambiguity
SIFT descriptors
After blob detection and scale normalization
Eliminating rotation ambiguity

• To assign a unique orientation to circular image windows:
  • Create histogram of local gradient directions in the patch
  • Assign canonical orientation at peak of smoothed histogram
SIFT detected features

- Detected features with characteristic scales and orientations:

From feature detection to feature description
Properties of Feature Descriptors

- Easily compared (compact, fixed-dimensional)
- Easily computed
- **Invariant**
  - Translation
  - Rotation
  - Scale
  - Change in image brightness
  - Change in perspective?
SIFT Descriptor

- Divide $16 \times 16$ window into $4 \times 4$ grid of cells
- Compute an orientation histogram for each cell
  - $16$ cells $\times 8$ orientations $= 128$-dimensional descriptor

Properties of SIFT

Extraordinarily robust detection and description technique

- Handles changes in viewpoint (~ 60 degree out-of-plane rotation)
- Handles significant changes in illumination (sometimes even day vs night)
- Fast and efficient—can run in real time
- Lots of code available
A hard feature matching problem

NASA Mars Rover images

Slide credit: S. Lazebnik
NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely

Slide credit: S. Lazebnik
Going deeper

Scale-invariant regions (blobs)

Going deeper

Affine-adapted blobs


Slide: S. Lazebnik
Next class: fitting, Hough transforms, RANSAC