Local features and image matching

Prof. Xin Yang
HUST
Last time

• RANSAC for robust geometric transformation estimation
  – Translation, Affine, Homography

• Image warping
  – Given a 2D transformation $T$ and a source image, compute a transformed image of source based on $T$

• Image mosaics
  – Given two images, compute the transformation between them and blend them based on image warping
Today

How to detect which features to match?

Computing local invariant features

- Detection of interest points
  - Harris corner detection
  - Scale invariant blob detection: LoG
- Description of local patches
Local features: main components

1) **Detection**: Identify the interest points

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

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

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

3) **Matching**: Determine correspondence between descriptors in two views
Local features: desired properties

• Repeatability
  – The same feature can be found in several images despite geometric and photometric transformations

• Saliency
  – Each feature has a distinctive description

• Compactness and efficiency
  – Many fewer features than image pixels

• Locality
  – A feature occupies a relatively small area of the image; robust to clutter and occlusion
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
Local features: main components

1) **Detection**: Identify the interest points

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

3) **Matching**: Determine correspondence between descriptors in two views
• What points would you choose?
**Corners** as distinctive interest points

- 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: Alyosha Efros, Darya Frolova, Denis Simakov
Corners as distinctive interest points

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

2 x 2 matrix of image derivatives (averaged in neighborhood of a point)

Notation:
\[ I_x \leftrightarrow \frac{\partial I}{\partial x} \quad I_y \leftrightarrow \frac{\partial I}{\partial y} \quad I_x I_y \leftrightarrow \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \]
What does this matrix reveal?

First, consider an axis-aligned corner:
What does this matrix reveal?

First, consider an axis-aligned corner:

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

- This means dominant gradient directions align with \( x \) or \( y \) axis
- Look for locations where both \( \lambda \)'s are large
- If either \( \lambda \) is close to 0, then this is not corner-like
- What if we have a corner that is not aligned with the image axes?
What does this matrix reveal?

Since $M$ is symmetric, we have

$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$$

$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of $M$ reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.
Corner response function

“edge”: 
\[ \lambda_1 \gg \lambda_2 \]
\[ \lambda_2 \gg \lambda_1 \]

“corner”: 
\[ \lambda_1 \text{ and } \lambda_2 \text{ are large, } \lambda_1 \sim \lambda_2; \]

“flat” region 
\[ \lambda_1 \text{ and } \lambda_2 \text{ are small;} \]

\[ f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \]
Harris corner detector

1) Compute $M$ matrix for each image window to get their *cornerness* scores.
2) Find points whose surrounding window gave large corner response ($f > \text{threshold}$)
3) Take the points of local maxima, i.e., perform non-maximum suppression
Example of Harris application
Example of Harris application

Compute corner response at every pixel.
Example of Harris application
Harris Detector: Steps
Harris Detector: Steps

Compute corner response $f$
Harris Detector: Steps

Find points with large corner response: \( f > \text{threshold} \)
Harris Detector: Steps

Take only the points of local maxima of $f$
Harris Detector: Steps
Properties of the Harris corner detector

Rotation invariant? Yes

\[
M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T
\]

Scale invariant?
Properties of the Harris corner detector

Rotation invariant? Yes

Scale invariant? No

All points will be classified as edges

Corner!
Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?
Automatic scale selection

Intuition:
• Find scale that gives local maxima of some function $f$ in both position and scale
• What can be the “signature” function?
Recall: Edge detection

\[ f \]

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

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

Edge = maximum of derivative

Source: S. Seitz
Recall: Edge detection

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.
Blob detection in 2D

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D: scale selection

- Laplacian-of-Gaussian = “blob” detector

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D

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

Original image at $\frac{3}{4}$ the size

Kristen Grauman
Original image at \( \frac{3}{4} \) the size

Kristen Grauman
Scale invariant interest points

Interest points are local maxima in both position and scale.

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow (x, y, \sigma) \]

\[ \Rightarrow \text{List of } (x, y, \sigma) \]
Scale-space blob detector:
Example

Image credit: Lana Lazebnik
We can approximate the Laplacian with a difference of Gaussians; more efficient to implement

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

(Laplacian)

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

(Difference of Gaussians)
Local features: main components

1) Detection: Identify the interest points

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

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

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

3) Matching: Determine correspondence between descriptors in two views
Geometric transformations

e.g. scale, translation, rotation
Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial
Raw patches as local descriptors

The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.
SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.

Why subpatches? Why does SIFT have some illumination invariance?
Making descriptor rotation invariant

- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.
SIFT descriptor [Lowe 2004]

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
Example

NASA Mars Rover images
Example

NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely
SIFT properties

• Invariant to
  – Scale
  – Rotation

• Partially invariant to
  – Illumination changes
  – Camera viewpoint
  – Occlusion, clutter
Local features: main components

1) Detection: Identify the interest points

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

3) Matching: Determine correspondence between descriptors in two views
Matching local features
Matching local features

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Kristen Grauman
At what SSD value do we have a good match?
To add robustness to matching, can consider ratio:
\[
\text{distance to best match} / \text{distance to second best match}
\]
If low, first match looks good.
If high, could be ambiguous match.
Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2\textsuperscript{nd} nearest descriptor

\begin{center}
\includegraphics[width=0.7\textwidth]{sift_matching.png}
\end{center}

\textit{Lowe IJCV 2004}
Recap: robust feature-based alignment

Source: L. Lazebnik
Recap: robust feature-based alignment

- Extract features

Source: L. Lazebnik
Recap: robust feature-based alignment

• Extract features
• Compute *putative matches*

Source: L. Lazebnik
Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)

Source: L. Lazebnik
Recap: robust feature-based alignment

• Extract features
• Compute *putative matches*
• Loop:
  – *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  – *Verify* transformation (search for other matches consistent with $T$)

Source: L. Lazebnik
Recap: robust feature-based alignment

• Extract features
• Compute *putative matches*
• Loop:
  – *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  – *Verify* transformation (search for other matches consistent with $T$)
Recent Work for Local Features

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
- ...
Automatic mosaicing

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html
Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]
Recognition of specific objects, scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002
Summary

• Interest point detection
  – Harris corner detector
  – Laplacian of Gaussian, automatic scale selection

• Invariant descriptors
  – Rotation according to dominant gradient direction
  – Histograms for robustness to small shifts and translations (SIFT descriptor)