Lecture 5: Feature descriptors and matching
Reading

• Szeliski: 4.1
Announcements

• Project 1 Artifacts due tomorrow, Friday 2/17, at 11:59pm

• Project 2 will be released next week

• In-class quiz at the beginning of class Thursday
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

Kristen Grauman
Harris features (in red)
Image transformations

• Geometric
  
  Rotation

• Photometric
  
  Intensity change
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
Harris detector: Invariance properties
-- Image translation

• Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation
Harris detector: Invariance properties
-- Image rotation

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

Corner location is covariant w.r.t. rotation
Harris detector: Invariance properties – Affine intensity change

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scaling: $I \rightarrow a I$

*Partially invariant* to affine intensity change
Harris Detector: Invariance Properties

• Scaling

Corner

All points will be classified as edges

Not invariant to scaling
Suppose you’re looking for corners

Key idea: find scale that gives local maximum of $f$

- in both position and scale
- One definition of $f$: the Harris operator
Automatic scale selection

Lindeberg et al., 1996
Automatic scale selection
Automatic scale selection

Function responses in increasing scale
Scale trace (signature)
Automatic scale selection
Automatic scale selection

Function responses in increasing scale
Scale trace (signatures)

$f(I_{h_i m}(x, \sigma))$
Automatic scale selection

$\tilde{f}(I_t(x, \sigma))$
Automatic scale selection

Function responses in increasing scale
Scale trace (signature)

\[ f(I_{h...m}(x, \sigma)) \]

\[ f(I_{h...m}(x', \sigma')) \]
Automatic scale selection

Normalize: rescale to fixed size
Implementation

• Instead of computing $f$ for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid.

(sometimes need to create in-between levels, e.g. a $\frac{3}{4}$-size image)
Feature extraction: Corners and blobs
Another common definition of $f$

- The *Laplacian of Gaussian* (LoG)

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

(very similar to a *Difference of Gaussians* (DoG) – i.e. a Gaussian minus a slightly smaller Gaussian)
Laplacian of Gaussian

- “Blob” detector

Find maxima and minima of LoG operator in space and scale
• At what scale does the Laplacian achieve a maximum response for a binary circle of radius $r$?
Characteristic scale

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

Find local maxima in position-scale space

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

sigma = 11.9912
Scale-space blob detector: Example
Scale Invariant Detection

- Functions for determining scale $f = \text{Kernel} \ast \text{Image}$

Kernels:

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]  

(Laplacian)

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

(Difference of Gaussians)

where Gaussian

\[ G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

Note: both kernels are invariant to scale and rotation
Questions?
Feature descriptors

We know how to detect good points
Next question: **How to match them?**

**Answer:** Come up with a *descriptor* for each point, find similar descriptors between the two images
Feature descriptors

We know how to detect good points
Next question: **How to match them?**

Lots of possibilities (this is a popular research area)

- Simple option: match square windows around the point
- State of the art approach: SIFT
Invariance vs. discriminability

• Invariance:
  – Descriptor shouldn’t change even if image is transformed

• Discriminability:
  – Descriptor should be highly unique for each point
Image transformations

• Geometric
  
  Rotation

  Scale

• Photometric
  
  Intensity change
Invariance

• Most feature descriptors are designed to be invariant to
  – Translation, 2D rotation, scale

• They can usually also handle
  – Limited 3D rotations (SIFT works up to about 60 degrees)
  – Limited affine transformations (some are fully affine invariant)
  – Limited illumination/contrast changes
How to achieve invariance

Need both of the following:

1. Make sure your detector is invariant
2. Design an invariant feature descriptor
   – Simplest descriptor: a single 0
     • What’s this invariant to?
   – Next simplest descriptor: a square window of pixels
     • What’s this invariant to?
   – Let’s look at some better approaches...
Rotation invariance for feature descriptors

- Find dominant orientation of the image patch
  - This is given by $x_{\text{max}}$, the eigenvector of $H$ corresponding to $\lambda_{\text{max}}$ (the larger eigenvalue)
  - Rotate the patch according to this angle
Take 40x40 square window around detected feature

- Scale to 1/5 size (using prefiltering)
- Rotate to horizontal
- Sample 8x8 square window centered at feature
- Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window

Adapted from slide by Matthew Brown
Detections at multiple scales

Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.
Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

Adapted from slide by David Lowe
SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

Adapted from slide by David Lowe
Properties of SIFT

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
Other descriptors

• HOG: Histogram of Gradients (HOG)
  – Dalal/Triggs
  – Sliding window, pedestrian detection

• FREAK: Fast Retina Keypoint
  – Perceptually motivated

• LIFT: Learned Invariant Feature Transform
  – Learned via deep learning

https://arxiv.org/abs/1603.09114
Summary

• Keypoint detection: repeatable and distinctive
  – Corners, blobs, stable regions
  – Harris, DoG

• Descriptors: robust and selective
  – spatial histograms of orientation
  – SIFT and variants are typically good for stitching and recognition
  – But, need not stick to one
Questions?