Local Feature Detection (cont’d) + Description

Prof. Adriana Kovashka
University of Pittsburgh
September 21, 2016
Announcements

• HW3W graded
• HW3P due tonight
  – Please read the entire assignment before you start!
  – Let’s make sure we’re on the same page for Part I...
  – Problems? Concerns?
• HW4W, HW4P out
Plan for today

• Feature detection / keypoint extraction
  – Corner detection (recap + properties)
  – Blob detection

• Feature description (of detected features)
Local features: desired properties

• Locality
  – A feature occupies a relatively small area of the image
  – Robust to clutter and occlusion

• Repeatability and flexibility
  – The same feature can be found in several images despite geometric, photometric transformations
  – Robustness to expected variations
  – Maximize correct matches

• Distinctiveness
  – Each feature has a distinctive description
  – Minimize wrong matches

• Compactness and efficiency
  – Many fewer features than image pixels

Adapted from K. Grauman and D. Hoiem
Corners as distinctive interest points

• We should easily recognize the keypoint 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

A. Efros, D. Frolova, D. Simakov
Harris Detector: Algorithm

• Compute image gradients $I_x$ and $I_y$ for all pixels

• For each pixel
  – Compute
    $M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$
  
  by looping over neighbors $x, y$

  – compute
    $R = \det M - k (\text{trace } M)^2$

    ($k$ : empirical constant, $k = 0.04-0.06$)

• Find points with large corner response function $R$ ($R > \text{threshold}$)

• Non-max suppression: Take the points of locally maximum $R$ as the detected feature points (i.e., pixels where $R$ is bigger than for all the 4 or 8 neighbors)
Example of Harris application
Example of Harris application

- Corner response at every pixel
More Harris responses

*Effect:* A very precise corner detector.
More Harris responses
Invariance vs covariance

“A function is *invariant* under a certain family of transformations if its value does not change when a transformation from this family is applied to its argument. A function is *covariant* when it commutes with the transformation, i.e., applying the transformation to the argument of the function has the same effect as applying the transformation to the output of the function. […] [For example,] the area of a 2D surface is invariant under 2D rotations, since rotating a 2D surface does not make it any smaller or bigger. But the orientation of the major axis of inertia of the surface is covariant under the same family of transformations, since rotating a 2D surface will affect the orientation of its major axis in exactly the same way.”


http://homes.esat.kuleuven.be/%7Etuytelaar/FT_survey_interestpoints08.pdf
What happens if: 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
What happens if: Image translation

- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation
What happens if: Image rotation

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

Corner location is covariant w.r.t. rotation
What happens if: Scaling

Invariant to image scale?
What happens if: Scaling

Corner location is not covariant to scaling!

All points will be classified as edges
Problem:

- How do we choose corresponding circles *independently* in each image?
- Do objects in the image have a characteristic scale that we can identify?
Scale Invariant Detection

- Solution:
  - Design a function on the region which is “scale invariant” (has the same shape even if the image is resized)
  - Take a local maximum of this function

Adapted from A. Torralba

![Graphs showing scale invariance](image)
Scale Invariant Detection

• A “good” function for scale detection: has one stable sharp peak

[Graphs showing different functions for region size vs. f]

• For usual images: a good function would be a one which responds to contrast (sharp local intensity change)
Automatic Scale Selection

How to find corresponding patch sizes?

\[ f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma')) \]
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{1...m}(x, \sigma)) \]

\[ f(I_{1...m}(x', \sigma)) \]
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)

\[ f(I_{i_1...i_m}(x, \sigma)) \]

\[ f(I_{i_1...i_m}(x', \sigma)) \]
Automatic Scale Selection

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

- Function responses for increasing scale (scale signature)
What Is A Useful Signature Function?

- Laplacian of Gaussian = “blob” detector

K. Grauman, B. Leibe
Blob detection in 2D

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Difference of Gaussian $\approx$ Laplacian

- 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)

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

(Difference of Gaussians)

\[
I(k\sigma) - I(\sigma) = \text{DoG}
\]
Laplacian pyramid example

- Allows detection of increasingly coarse detail
Difference of Gaussian: Efficient computation

- Computation in Gaussian scale pyramid

Original image

Sampling with step $\sigma^4 = 2$

Scale (first octave)

Scale (next octave)

$\sigma = \frac{1}{2}$

$\sigma = 2^4$

Gaussian

Difference of Gaussian (DOG)

K. Grauman, B. Leibe
Find **local maxima** in position-scale space of Difference-of-Gaussian

Adapted from K. Grauman, B. Leibe

Find places where $X$ greater than all of its neighbors (in green)

$\Rightarrow$ List of $(x, y, s)$
Results: Difference-of-Gaussian

K. Grauman, B. Leibe
Additional references

- **Survey paper on local features:**
  [http://homes.esat.kuleuven.be/%7Etuytelaas/FT_survey_interestpoints08.pdf](http://homes.esat.kuleuven.be/%7Etuytelaas/FT_survey_interestpoints08.pdf)

- **Making Harris detection scale-invariant and more on scale invariance:**
  “Indexing based on scale invariant interest points” by Krystian Mikolajczyk and Cordelia Schmid, in ICCV 2001
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.
Geometric transformations

e.g. scale, translation, rotation

K. Grauman
Photometric transformations
Scale-Invariant Feature Transform (SIFT) descriptor

Histogram of oriented gradients
- Captures important texture information
- Robust to small translations / affine deformations

[Lowe, ICCV 1999]
Computing gradients

\( L = \text{the image intensity} \)

\[
m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}
\]

\[
\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right)
\]

- \( \tan(\alpha) = \frac{\text{opposite side}}{\text{adjacent side}} \)
Gradients

\[ m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \]

\[ \theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y))) \]

\[ m(x, y) = \sqrt{1 + 0} = 1 \]

\[ \theta(x, y) = \tan^{-1}(0/1) = 0 \]
Gradients

\[ m(x, y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \]

\[ \theta(x, y) = \tan^{-1}\left(\frac{(L(x+1,y) - L(x,y-1))}{(L(x+1,y) - L(x-1,y))}\right) \]
Gradients

\[ m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \]

\[ \theta(x, y) = \arctan((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y))) \]

\[
\begin{align*}
m(x, y) &= \sqrt{1 + 1} = 1.41 \\
\theta(x, y) &= \arctan(1/1) = 45
\end{align*}
\]
Basic idea:

- Take 16x16 square window around detected feature
- Compute gradient orientation for each pixel
- Create histogram over edge orientations weighted by magnitude
- That's your feature descriptor!
Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
Quantize the gradient orientations i.e. snap each gradient to one of 8 angles
Each gradient contributes not just 1, but magnitude(gradient) to the histogram, i.e. stronger gradients contribute more
16 cells * 8 orientations = 128 dimensional descriptor for each detected feature

Adapted from L. Zitnick, D. Lowe
Scale Invariant Feature Transform

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Quantize the gradient orientations i.e. snap each gradient to one of 8 angles
- Each gradient contributes not just 1, but magnitude(gradient) to the histogram, i.e. stronger gradients contribute more
- 16 cells * 8 orientations = 128 dimensional descriptor for each detected feature
- Normalize + clip (threshold normalize to 0.2) + normalize the descriptor
- After normalizing, we have:

\[ \sum_i d_i = 1 \quad \text{such that: } d_i < 0.2 \]

Adapted from L. Zitnick, D. Lowe
Making descriptor rotation invariant

- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation
SIFT is robust

• 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

• Can be made to work without feature detection, resulting in “dense SIFT” (more points means robustness to occlusion)

• One commonly used implementation
  • http://www.vlfeat.org/overview/sift.html

Adapted from S. Seitz
Examples of using SIFT
Examples of using SIFT
Examples of using SIFT

Images from S. Seitz
Applications of local invariant features

- Object recognition
- Indexing and retrieval
- Robot navigation
- 3D reconstruction
- Wide baseline stereo
- Motion tracking
- Image alignment
- Panoramas and mosaics
- ...

Adapted from K. Grauman and L. Lazebnik

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html
Summary

• Keypoint detection: repeatable and distinctive
  – Corners, blobs, stable regions
  – Laplacian of Gaussian, automatic scale selection

• Descriptors: robust and selective
  – Histograms for robustness to small shifts and translations (SIFT descriptor)

Adapted from D. Hoiem, K. Grauman