Lecture 9: Feature Descriptors
Announcements

• PA 1 grading (almost done)
  – Finalized grades next week
Choosing a detector

• What do you want it for?
  – Precise localization in x-y: Harris
  – Good localization in scale: Difference of Gaussian
  – Flexible region shape: MSER

• Best choice often application dependent
  – Harris-/Hessian-Laplace/DoG work well for many natural categories
  – MSER works well for buildings and printed things

• Why choose?
  – Get more points with more detectors

• There have been extensive evaluations/comparisons
  – [Mikolajczyk et al., IJCV’05, PAMI’05]
  – All detectors/descriptors shown here work well
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian

Harris points

Harris-Laplace points
K. Grauman, B. Leibe
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris detection
Maximally **Stable Extremal Regions**


• Maximally Stable Extremal Regions
  – *Threshold* image intensities: $I > \text{thresh}$ for several increasing values of thresh
  – Extract *connected components* (“Extremal Regions”)
  – Find a threshold when region is “Maximally Stable”, i.e. *local minimum* of the relative growth
  – Approximate each region with an *ellipse*
Example Results: MSER
Choosing a detector

• What do you want it for?
  – Precise localization in x-y: Harris
  – Good localization in scale: Difference of Gaussian
  – Flexible region shape: MSER

• Best choice often application dependent
  – Harris-/Hessian-Laplace/DoG work well for many natural categories
  – MSER works well for buildings and printed things

• Why choose?
  – Get more points with more detectors

• There have been extensive evaluations/comparisons
  – [Mikolajczyk et al., IJCV’05, PAMI’05]
  – All detectors/descriptors shown here work well
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.
Image representations

• Templates
  – Intensity, gradients, etc.

• Histograms
  – Color, texture, SIFT descriptors, etc.
Local Descriptors

• Most features can be thought of as templates, histograms (counts), or combinations

• The ideal descriptor should be
  – Robust
  – Distinctive
  – Compact
  – Efficient

• Most available descriptors focus on edge/gradient information
  – Capture texture information

K. Grauman, B. Leibe
Image Representations: Histograms

Global histogram

- Represent distribution of features
  - Color, texture, depth, ...

Images from Dave Kauchak
What kind of things do we compute histograms of?

- Color
  - L*a*b* color space
  - HSV color space
- Texture (filter banks or HOG over regions)
  - HOG: Histogram of Oriented Gradients
Orientation Normalization

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

[Lowe, SIFT, 1999]
Rotation invariance for feature descriptors

- Find dominant orientation of the image patch
  - This is given by $\mathbf{x}_{\text{max}}$, the eigenvector of $\mathbf{M}$ corresponding to $\lambda_{\text{max}}$ (the larger eigenvalue)
  - Rotate the patch according to this angle

Figure by Matthew Brown
Multiscale Oriented PatcheS descriptor

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
What kind of things do we compute histograms of?

SIFT – Lowe IJCV 2004
**Scale Invariant Feature Transform**

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

![Image gradients](image1.png)

Adapted from slide by David Lowe
**SIFT vector formation**

- Computed on rotated and scaled version of window according to computed orientation & scale
  - resample the window
- Based on gradients weighted by a Gaussian of variance 1.5 times the window (for smooth falloff)
SIFT vector formation

- 4x4 array of gradient orientation histogram weighted by magnitude
- 8 orientations x 4x4 array = 128 dimensions
- Motivation: some sensitivity to spatial layout, but not too much

Image gradients

Keypoint descriptor

showing only 2x2 here but is 4x4
Ensure smoothness

- Gaussian weight
- Trilinear interpolation
  - a given gradient contributes to 8 bins:
    4 in space times 2 in orientation
Reduce effect of illumination

- 128-dim vector normalized to 1
- Threshold gradient magnitudes to avoid excessive influence of high gradients
  - after normalization, clamp gradients $>0.2$
  - renormalize
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:
Local Descriptors: SURF

Fast approximation of SIFT idea
Efficient computation (Haar wavelets)
⇒ 6 times faster than SIFT
Equivalent quality for object identification

GPU implementation available
Feature extraction @ 200Hz
(detector + descriptor, 640×480 img)
http://www.vision.ee.ethz.ch/~surf

[Bay, ECCV’06], [Cornelis, CVGPU’08]

K. Grauman, B. Leibe
Other descriptors

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

- FREAK: Fast Retina Keypoint
  - Perceptually motivated
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