Recap: Local Feature Matching Outline

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Recap: Requirements for Local Features

- Problem 1: Detect the same point independently in both images
- Problem 2: For each point correctly recognize the corresponding one

Recap: Harris Detector [Harris88]

- Compute second moment matrix (autocorrelation matrix)
- Cornerness function - two strong eigenvalues

Recap: Harris Detector Responses [Harris88]

Effect: A very precise corner detector.
Recap: Hessian Detector [Beaudet78]

- Hessian determinant

\[
\text{Hessian}(I) = \begin{bmatrix}
I_{xx} & I_{xy} \\
I_{xy} & I_{yy}
\end{bmatrix}
\]

\[
det(\text{Hessian}(I)) = I_{xx}I_{yy} - I_{xy}^2
\]

In Matlab:

\[
I_{xx}, I_{xy}, I_{yy}\times (I_{xy} - (I_{xx}I_{yy})^2, I_{xy}^2, I_{xx}I_{yy})^2
\]

Recap: Hessian Detector Responses [Beaudet78]

Effect: Responses mainly on corners and strongly textured areas.

Topics of This Lecture

- Local Feature Extraction (cont’d)
  - Scale Invariant Region Selection
  - Orientation normalization
  - Affine Invariant Feature Extraction
- Local Descriptors
  - SIFT
- Applications

From Points to Regions...

- The Harris and Hessian operators define interest points.
  - Precise localization
  - High repeatability

- In order to compare those points, we need to compute a descriptor over a region.
- How can we define such a region in a scale invariant manner?
- I.e. how can we detect scale invariant interest regions?

Naïve Approach: Exhaustive Search

- Multi-scale procedure
  - Compare descriptors while varying the patch size

Similarity measure \( d(f_a, f_b) \)
Naïve Approach: Exhaustive Search

- Multi-scale procedure
  - Compare descriptors while varying the patch size

  \[ d(f_x, f_y) \]

  \[ d(f_x, f_y) \]

Automatic Scale Selection

- Solution:
  - Design a function on the region, which is “scale invariant” (the same for corresponding regions, even if they are at different scales)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

    \[ f = \text{average intensity} \]

- For a point in one image, we can consider it as a function of region size (patch width)

  \[ f \]
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit: Krystian Mikolajczyk
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = "blob" detector

Characteristic Scale

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


Slide credit: Svetlana Lazebnik

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

⇒ List of \((x, y, \sigma)\)
LoG Detector: Workflow

- We can efficiently approximate the Laplacian with a difference of Gaussians:

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

(Difference of Gaussians)

- Advantages?
  - No need to compute 2nd derivatives.
  - Gaussians are computed anyway, e.g. in a Gaussian pyramid.

Key point localization with DoG

- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses

Candidate keypoints: list of (x, y, θ)

DoG - Efficient Computation

- Computation in Gaussian scale pyramid
Results: Lowe’s DoG

Harris-Laplace [Mikolajczyk ’01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian (same procedure with Hessian ⇒ Hessian-Laplace)

Summary: Scale Invariant Detection
- Given: Two images of the same scene with a large scale difference between them.
- Goal: Find the same interest points independently in each image.
- Solution: Search for maxima of suitable functions in scale and in space (over the image).
- Two strategies
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation
  - These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).

Topics of This Lecture
- Local Feature Extraction (cont’d)
  - Scale Invariant Region Selection
  - Orientation normalization
  - Affine Invariant Feature Extraction
- Local Descriptors
  - SIFT
- Applications

Rotation Invariant Descriptors
- Find local orientation
  - Dominant direction of gradient for the image patch
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.
Orientation Normalization: Computation

- Compute orientation histogram [Lowe, SIFT, 1999]
- Select dominant orientation
- Normalize: rotate to fixed orientation

Topics of This Lecture

- Local Feature Extraction (cont’d)
  - Scale Invariant Region Selection
  - Orientation normalization
  - Affine Invariant Feature Extraction
- Local Descriptors
  - SIFT
- Applications

The Need for Invariance

- Up to now, we had invariance to
  - Translation
  - Scale
  - Rotation
- Not sufficient to match regions under viewpoint changes
  - For this, we need also affine adaptation

Affine Adaptation

- Problem:
  - Determine the characteristic shape of the region.
  - Assumption: shape can be described by “local affine frame”.
- Solution: iterative approach
  - Use a circular window to compute second moment matrix.
  - Compute eigenvectors to adapt the circle to an ellipse.
  - Recompute second moment matrix using new window and iterate…

Iterative Affine Adaptation

1. Detect keypoints, e.g. multi-scale Harris
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

Affine Normalization/Deskewing

- Steps
  - Rotate the ellipse’s main axis to horizontal
  - Scale the x axis, such that it forms a circle
Affine Adaptation Example

Scale-invariant regions (blobs)

Affine adaptation example

Affine-adapted blobs

Summary: Affine-Inv. Feature Extraction

Extract affine regions
Normalize regions
Eliminate rotational ambiguity
Compare descriptors

Invariance vs. Covariance

• Invariance:
  features(transform(image)) = features(image)

• Covariance:
  features(transform(image)) = transform(features(image))

Covariant detection \implies invariant description

Topics of This Lecture

• Local Feature Extraction (cont’d)
  - Orientation normalization
  - Affine Invariant Feature Extraction

• Local Descriptors
  - SIFT
  - Applications

• Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation

Local Descriptors

• We know how to detect points
• Next question:

How to describe them for matching?

Point descriptor should be:
1. Invariant
2. Distinctive
Local Descriptors
- Simplest descriptor: list of intensities within a patch.
- What is this going to be invariant to?

Feature Descriptors
- Disadvantage of patches as descriptors:
  - Small shifts can affect matching score a lot

Feature Descriptors: SIFT
- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

Overview: SIFT
- Extraordinarily robust matching technique
  - Can handle changes in viewpoint up to ~60 deg. 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

Working with SIFT Descriptors
- One image yields:
  - n 2D points giving positions of the patches [n x 2 matrix]
  - n scale parameters specifying the size of each patch [n x 1 vector]
  - n orientation parameters specifying the angle of the patch [n x 1 vector]
  - n 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch [n x 128 matrix]

Local Descriptors: SURF
- Fast approximation of SIFT idea
  - Efficient computation by 2D box filters & integral images
  - 6 times faster than SIFT
  - Equivalent quality for object identification
    - http://www.vision.ee.ethz.ch/~surf

- GPU implementation available
  - Feature extraction @ 100Hz (detector + descriptor, 640x480 img)
You Can Try It At Home...

- For most local feature detectors, executables are available online:
  - http://robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.vision.ee.ethz.ch/~surf

Topics of This Lecture

- Local Feature Extraction (cont’d)
  - Orientation normalization
  - Affine Invariant Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation

Applications of Local Invariant Features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
  - Specific objects
  - Textures
  - Categories
  - ...
Panorama Stitching

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

Recognition of Specific Objects, Scenes

B. Leibe

Recognition of Categories

Constellation model

Bags of words

Value of Local Features

- Advantages
  - Critical to find distinctive and repeatable local regions for multi-view matching.
  - Complexity reduction via selection of distinctive points.
  - Describe images, objects, parts without requiring segmentation; robustness to clutter & occlusion.
  - Robustness: similar descriptors in spite of moderate view changes, noise, blur, etc.

- How can we use local features for such applications?
  - Next week: matching and recognition

References and Further Reading

- More details on homography estimation can be found in Chapter 4.7 of
  - R. Hartley, A. Zisserman
    Multiple View Geometry in Computer Vision
    2nd Ed., Cambridge Univ. Press, 2004

- Details about the DoG detector and the SIFT descriptor can be found in
  - D. Lowe, Distinctive image features from scale-invariant keypoints,
    IJCV 60(2), pp. 91-110, 2004

- Try the available local feature detectors and descriptors
  - http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries