Locating and Describing Interest Points

Computer Vision
CS 543 / ECE 549
University of Illinois

Derek Hoiem

Acknowledgment: Many keypoint slides from Grauman&Leibe 2008 AAAI Tutorial
This section: correspondence and alignment

• Correspondence: matching points, patches, edges, or regions across images
This section: correspondence and alignment

- Alignment: solving the transformation that makes two things match better
Example: fitting an 2D shape template
Example: fitting a 3D object model
Example: estimating “fundamental matrix” that corresponds two views
Example: tracking points

frame 0

frame 22

frame 49

Your problem 1 for HW 2!
HW 2

• Interest point detection and tracking
  – Detect trackable points
  – Track them across 50 frames
  – In HW 3, you will use these tracked points for structure from motion

frame 0  frame 22  frame 49
HW 2

• Alignment of object edge images
  – Compute a transformation that aligns two edge maps
HW 2

- Initial steps of object alignment
  - Derive basic equations for interest-point based alignment
This class: interest points

- Note: “interest points” = “keypoints”, also sometimes called “features”

- Many applications
  - tracking: which points are good to track?
  - recognition: find patches likely to tell us something about object category
  - 3D reconstruction: find correspondences across different views
Human eye movements

Yarbus eye tracking
Human eye movements

Change blindness: http://www.simonslab.com/videos.html
This class: interest points

• Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
  – Which points would you choose?
Overview of Keypoint Matching

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

\[ d(f_A, f_B) < T \]
Goals for Keypoints

Detect points that are *repeatable* and *distinctive*
Key trade-offs

Detection

More Repeatable
- Robust detection
- Precise localization

More Points
- Robust to occlusion
- Works with less texture

Description

More Distinctive
- Minimize wrong matches

More Flexible
- Robust to expected variations
- Maximize correct matches
Choosing interest points

Where would you tell your friend to meet you?
Choosing interest points

Where would you tell your friend to meet you?
Many Existing Detectors Available

Hessian & Harris
Laplacian, DoG
Harris-/Hessian-Laplace
Harris-/Hessian-Affine
EBR and IBR
MSER
Salient Regions
Others...

[Beaudet ‘78], [Harris ‘88]
[Lindeberg ‘98], [Lowe 1999]
[Mikolajczyk & Schmid ‘01]
[Mikolajczyk & Schmid ‘04]
[Tuytelaars & Van Gool ‘04]
[Matas ‘02]
[Kadir & Brady ‘01]
Harris Detector [Harris88]

• Second moment matrix

\[ \mu(\sigma_1, \sigma_D) = g(\sigma_1)^* \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

*Intuition:* Search for local neighborhoods where the image content has two main directions (eigenvectors).
Harris Detector [Harris88]

- Second moment matrix

\[
\mu(\sigma_I, \sigma_D) = g(\sigma_I) \star \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}
\]

1. Image derivatives (optionally, blur first)

\[
\det M = \lambda_1 \lambda_2 \\
\text{trace } M = \lambda_1 + \lambda_2
\]

2. Square of derivatives

3. Gaussian filter \( g(\sigma_I) \)

4. Cornerness function – both eigenvalues are strong

\[
har = \det[\mu_I, \sigma_D] - \alpha[\text{trace}(\mu_I, \sigma_D)]^2 = g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2
\]

5. Non-maxima suppression
Harris Detector: Mathematics

\[ M = g(\sigma_t) \cdot \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Want large eigenvalues, and small ratio \( \frac{\lambda_1}{\lambda_2} < t \)

2. We know

\[
\det M = \lambda_1 \lambda_2 \\
\text{trace } M = \lambda_1 + \lambda_2
\]

3. Leads to

\[ \det M - k \cdot \text{trace}^2 (M) > t \]

\( (k: \text{empirical constant, } k = 0.04-0.06) \)

Nice brief derivation on wikipedia
Harris Detector – Responses [Harris88]

Effect: A very precise corner detector.
Harris Detector - Responses [Harris88]
Hessian Detector [Beaudet78]

- Hessian determinant

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

*Intuition:* Search for strong curvature in two orthogonal directions
Hessian Detector [Beaudet78]

- Hessian determinant

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

\[
\text{det } M = \lambda_1 \lambda_2 \\
\text{trace } M = \lambda_1 + \lambda_2
\]

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

In Matlab:

\[
I_{xx} \ast I_{yy} - (I_{xy})^2
\]
Effect: Responses mainly on corners and strongly textured areas.
Hessian Detector – Responses [Beaudet78]
So far: can localize in x-y, but not scale
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)

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

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

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

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

• Difference-of-Gaussian = “blob” detector

K. Grauman, B. Leibe
Difference-of-Gaussian (DoG)

K. Grauman, B. Leibe
DoG – Efficient Computation

- Computation in Gaussian scale pyramid

Original image

Sampling with step $\sigma^4 = 2$

Scale (first octave)

Scale (next octave)

$\sigma = 2^4$

Gaussian

Difference of Gaussian (DOG)

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

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \Rightarrow (x, y, s) \]
Results: Difference-of-Gaussian

K. Grauman, B. Leibe
Orientation Normalization

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

[Lowe, SIFT, 1999]
Maximally Stable Extremal Regions [Matas ‘02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range
Example Results: MSER
Available at a web site near you...

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

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

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

K. Grauman, B. Leibe
Local Descriptors: SIFT Descriptor

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

[Lowe, ICCV 1999]
Details of Lowe’s SIFT algorithm

• Run DoG detector
  – Find maxima in location/scale space
  – Remove edge points

• Find all major orientations
  – Bin orientations into 36 bin histogram
    • Weight by gradient magnitude
    • Weight by distance to center (Gaussian-weighted mean)
  – Return orientations within 0.8 of peak
    • Use parabola for better orientation fit

• For each (x,y, scale, orientation), create descriptor:
  – Sample 16x16 gradient mag. and rel. orientation
  – Bin 4x4 samples into 4x4 histograms
  – Threshold values to max of 0.2, divide by L2 norm
  – Final descriptor: 4x4x8 normalized histograms

\[
H = \begin{bmatrix}
D_{xx} & D_{xy} \\
D_{xy} & D_{yy}
\end{bmatrix}
\]

\[
\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r + 1)^2}{r}
\]

Lowe IJCV 2004
Matching SIFT Descriptors

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

![Graph showing the repeatability of SIFT as a function of image noise. The graph compares the percentage of matched features under different criteria: matching location and scale, matching location, scale, and orientation, and the nearest descriptor in the database. The repeatability decreases with increasing image noise.](image-url)
SIFT Repeatability
SIFT Repeatability

Matching location, scale, and orientation
Nearest descriptor in database

Number of keypoints in database (log scale)
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

GPU implementation available

Feature extraction @ 200Hz
(detector + descriptor, 640×480 img)

http://www.vision.ee.ethz.ch/~surf

[Bay, ECCV’06], [Cornelis, CVGPU’08]
Local Descriptors: Shape Context

Count the number of points inside each bin, e.g.:

Count = 4
\vdots
Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Belongie & Malik, ICCV 2001
Local Descriptors: Geometric Blur

Compute edges at four orientations
Extract a patch in each channel

Apply spatially varying blur and sub-sample

(Idealized signal)

Example descriptor

Berg & Malik, CVPR 2001

K. Grauman, B. Leibe
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
## Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

<table>
<thead>
<tr>
<th>Feature Detector</th>
<th>Corner</th>
<th>Blob</th>
<th>Region</th>
<th>Rotation invariant</th>
<th>Scale invariant</th>
<th>Affine invariant</th>
<th>Repeatability</th>
<th>Localization accuracy</th>
<th>Robustness</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Hessian</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>SUSAN</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Harris-Laplace</td>
<td>✓ (✓)</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Hessian-Laplace</td>
<td>(✓)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>DoG</td>
<td>✓ (✓)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>SURF</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>✓ (✓)</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Hessian-Affine</td>
<td>✓ (✓)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Salient Regions</td>
<td>✓ (✓)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Edge-based</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MSER</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Intensity-based</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Superpixels</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Tuytelaars Mikolajczyk 2008
Choosing a descriptor

• Again, need not stick to one

• For object instance recognition or stitching, SIFT or variant is a good choice
Things to remember

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

• Descriptors: robust and selective
  – spatial histograms of orientation
  – SIFT
Next time

- Feature tracking