Local Invariant Features: Detection & Description

Presentation:
Frédéric Devernay, INRIA

With many slides from Tinne Tuytelaars and others
Motivation

- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions
  - Articulation
  - Intra-category variations
Approach

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

\[ d(f_A, f_B) < T \]
Requirements

• Region extraction needs to be repeatable and precise
  ➢ Translation, rotation, scale changes
  ➢ (Limited out-of-plane (~affine) transformations)
  ➢ Lighting variations

• We need a sufficient number of regions to cover the object

• The regions should contain “interesting” structure
Many Existing Detectors Available

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Crowley & Parker ‘84], [Lindeberg ‘93-‘98], [Lowe ‘99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR [Tuytelaars & Van Gool ‘04]
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- Others...
Keypoint Localization

• Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

⇒ *Look for two-dimensional signal changes*
Hessian Detector [Beaudet78]

- Hessian determinant

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

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

- Hessian determinant

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

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

In Matlab:

\[
I_{xx} \ast I_{yy} - (I_{xy})^2
\]
**Effect:** Responses mainly on corners and strongly textured areas.
Hessian Detector – Responses [Beaudet78]
Harris Detector [Harris88]

- Second moment matrix
  (autocorrelation matrix)

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

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

- Second moment matrix (autocorrelation matrix)

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

1. Image derivatives

2. Square of derivatives

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

4. Cornerness function - both eigenvalues are strong

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

5. Non-maxima suppression
Harris Detector – Responses [Harris88]

Effect: A very precise corner detector.
Harris Detector – Responses [Harris88]
Automatically Scale Selection

\[ f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I'_{i_1 \ldots i_m}(x', \sigma')) \]

Same operator responses if the patch contains the same image up to scale factor
How to find corresponding patch sizes?
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)
Automatic Scale Selection

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

- Laplacian-of-Gaussian = “blob” detector
Laplacian-of-Gaussian (LoG)

- Local maxima in scale space of Laplacian-of-Gaussian

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

- Difference of Gaussians as approximation of the Laplacian-of-Gaussian
DoG - Efficient Computation

- Computation in Gaussian scale pyramid
Results: Lowe’s DoG
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection
Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian (same procedure with Hessian ⇒ Hessian-Laplace)
Maximally Stable Extremal Regions [Matas ‘02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range
Example Results: MSER
You Can Try It At Home...

- For most local feature detectors, executables are available online:
- [http://robots.ox.ac.uk/~vgg/research/affine](http://robots.ox.ac.uk/~vgg/research/affine)
- [http://www.vision.ee.ethz.ch/~surf](http://www.vision.ee.ethz.ch/~surf)
Orientation Normalization

- Compute orientation histogram (gradient magnitude and orientation by finite differences)
- Select dominant orientation (gradient histogram weighted by magnitude and Gaussian window, $\sigma=1.5s$)
- Normalize: rotate to fixed orientation (interpolate with parabola, keep all peaks within 80% of max)

[Lowe, SIFT, 1999]
Local Descriptors

• The ideal descriptor should be
  - Repeatable
  - Distinctive
  - Compact
  - Efficient

• Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color still relatively seldomly used
    (more suitable for homogenous regions)
Local Descriptors: SIFT Descriptor

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

[Lowe, ICCV 1999] [Lowe, IJCV 2004]
SIFT Descriptor

- 4x4 Gradient window
- Histogram of 4x4 samples per window in 8 directions
- Gaussian weighting around center (σ is 0.5 times that of the scale of a keypoint)
- 4x4x8 = 128 dimensional feature vector
SIFT Descriptor - Lighting changes

- Gains do not affect gradients
- Normalization to unit length removes contrast
- Saturation affects magnitudes much more than orientation
- Threshold gradient magnitudes to 0.2 and renormalize
Local Descriptors: SURF

- Fast approximation of SIFT idea
  - Efficient computation by 2D box filters & integral images $\Rightarrow$ 6 times faster than SIFT
  - Equivalent quality for object identification

- GPU implementation available
  - Feature extraction @ 100Hz (detector + descriptor, 640×480 img)
  - [http://www.vision.ee.ethz.ch/~surf](http://www.vision.ee.ethz.ch/~surf)

[Bay, ECCV’06], [Cornelis, CVGPU’08]
Methodology

- Using integral images for major speed up
  - Integral Image (summed area tables) is an intermediate representation for the image and contains the sum of gray scale pixel values of image
  - Second order derivative and Haar-wavelet response
- In order to bring in information about the polarity of the intensity changes, extract the sum of absolute value of the responses → feature vector of length 64
- Normalize the vector into unit length

\[ I_f(x) = \sum \sum I(i, j) \]

\[ S = A - B - C + D \]
**SURF-64 descriptor**

**Figure 3.4:** To build the descriptor, a quadratic grid with $4 \times 4$ square sub-regions is laid over the interest point (left). For each sample, the wavelet responses are computed. For this figure $2 \times 2$ vectors per sub-region for reasons of illustration. For each sub-region (right), the sums of $dx$, $|dx|$, $dy$, and $|dy|$ relative to the orientation of the grid, are computed.
SURF descriptor examples
Local Descriptors: Shape Context

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

Count = 4
Count = 10

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

Belongie & Malik, ICCV 2001
Applications of local invariant features

- Panorama stitching
- High-resolution document scan (similar to panorama)
- 3D modelling
- Location recognition
Panorama stitching

Brown, ICCV 2003
3D modelling

(from Sudderth et al., 2006)
Location recognition

Figure 13. This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640 × 315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.
Simultaneous Localization and Mapping (SLAM)

Figure 5: Bird’s eye view of the SIFT landmarks (including ceiling features) in the database after 249 frames. The cross at (0,0) indicates the initial robot position and the dashed line indicates the robot path.
Recent SLAM improvements

- Real time (with a laptop)
- Robust to motion blur
- Source code available (PTAM)

Parallel Tracking and Mapping for Small AR Workspaces

Extra video results made for ISMAR 2007 conference

Georg Klein and David Murray
Active Vision Laboratory
University of Oxford

[Klein et al.]
So, What Local Features Should I Use?

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

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

• More features are better
  - Combining several detectors often helps