Local Feature Extraction and Learning for Computer Vision

Bin Fan, Chinese Academy of Sciences, China
Jiwen Lu, Tsinghua University, China
Pascal Fua, EPFL, Switzerland
Local Image Descriptors: A Tool for Matching “Things”

Which pixel goes where?
Local Image Descriptors: A Tool for Matching “Things”

Which region goes where?
By matching “things” we can…

Dense city 3D reconstruction/ Structure from motion

Content-based web image search
By matching “things” we can...

... track objects in real-time even when there are occlusions and motion blur.
By matching “things” we can…

Mobile augmented reality

Real-time pedestrian detection
By matching “things” we can...

... detect objects in crowded scenes.
By matching “things” we can...

... mosaic images into panoramas.
Local Image Descriptors: A Tool for Matching “Things”

Which region goes where?
Local Image Descriptors

Distinctiveness

Robustness

Distinctiveness
Local Descriptor Trends

- Early methods
- SIFT and its variants
- Binary descriptor
- Learning based methods
- CNN based methods

Timeline:
- 04
- 07
- 10
- 15
Deep Learning Revolution

ALL HAIL

OUR NEW ROBOT-OVERLORDS
A Deep Casualty?

- The SIFT paper is the most cited computer vision paper ever.
- But it’s not as dominant as it once was.

→ Will it endure? Yes!
Keypoints Remain Relevant

- When accurate geometric recovery matters, they remain unequaled.
- They are efficient for real-time applications.
- They provide an effective way
  - to compress the information present in large images,
  - to recognize specific locations.
- The algorithms do not need to be retrained for each new application.
- Some or all elements of the pipeline can deeply reformulated.

→ Future algorithms will combine Deep Learning and keypoint matching.
Outline of the Tutorial

• Classic Local Features

• Towards High Performance Descriptors (Floating Point)
  • Handcrafted Descriptors
  • Learned Descriptors

• Towards Efficient Descriptors (Binary)
  • Handcrafted Descriptors
  • Learned Descriptors

• Applications
Outline of the Tutorial

• **Classic Local Features**

• **Towards High Performance Descriptors (Floating Point)**
  • Handcrafted Descriptors
  • Learned Descriptors

• **Towards Efficient Descriptors (Binary)**
  • Handcrafted Descriptors
  • Learned Descriptors

• Applications
Classic Local Features

• SIFT: Scale Invariant Feature Transform

• SURF: Speeded Up Robust Features

• Daisy
SIFT Pipeline

1. Image
2. Local Feature Detector
3. Orientation Estimation
4. Feature Descriptor
Scale space detection

**GSS:** Gaussian Scale Space is produced by iteratively convolving the last layer image with a Gaussian kernel.

**DoGSS:** DoG Scale Space is produced by subtracting neighboring GSS layers.

*By Cmglee - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=42549151*
Non-max suppression in Scale Space

Search extrema in DoGSS to locate initial keypoints.
Keypoint Refinement

- **Refine:**
  - Fit a 3D \((x,y,\text{scale})\) curve to the initial keypoint, and find the peak in the curve as the refined keypoint.

- **Elimination:**
  - Discard keypoints with low refined DoG response.
  - Discard keypoints with high edge response.
Dominant Orientation Estimation

Gradient and angle:

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

Orientation selection
Descriptor construction

Image gradients ➔ Keypoint descriptor
Descriptor construction

1. Find the blurred image of the closest scale in scale space
2. Sample the points around the keypoint
3. Rotate the gradients and coordinates by dominant orientation
4. Separate the region into subregions
5. Create histogram for each sub region with 8 bins
6. Normalization
Speeded Up Robust Features

• Aim: faster than SIFT, while still being robust.
  • 3-7 times faster than SIFT, with similar matching performance.

• Key Idea: Haar filters and Integral Image

• Well received!
  • More than 8000 citations.
  • CVIU Most Cited Paper
  • Koenderink Prize of ECCV’16
    • fundamental contributions in computer vision that stood the test of time
Keypoint Detection

• Uses determinant of Hessian matrix

• Approximate 2nd derivatives in Hessian matrix with box filters

\[
H(x, \sigma) = \begin{bmatrix}
L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\
L_{xy}(x, \sigma) & L_{yy}(x, \sigma)
\end{bmatrix}
\]

\[
\hat{H}(x, \sigma) = \begin{bmatrix}
D_{xx}(x, \sigma) & D_{xy}(x, \sigma) \\
D_{xy}(x, \sigma) & D_{yy}(x, \sigma)
\end{bmatrix}
\]

\[
R(x, \sigma) = L_{xx}L_{yy} - L_{xy}^2 \approx D_{xx}D_{yy} - (0.9D_{xy}^2)
\]
Keypoint Detection

$L_{xx}$  $L_{yy}$  $L_{xy}$

$D_{xx}$  $D_{yy}$  $D_{xy}$
SURF vs SIFT: Scale Space

- Increase Filter Size
- Fix Filter Size
- Fix Image Size
- Decrease Image Size
• The Haar wavelet responses (x and y) are represented as vectors.

• Sum all responses within a sliding orientation window covering an angle of 60 degree.

• The longest vector is the dominant orientation
Descriptor Extraction

SURF [Bay '04]
Descriptor Extraction

1. Split the interest region (20s x 20s) into 4 x 4 square sub-regions.
2. Calculate Haar wavelet responses dx and dy, and weight the responses with a Gaussian kernel.
3. Sum the response over each sub-region for dx and dy, then sum the absolute value of response.
4. Concatenate summation results in all sub-regions, forming a 64D SURF descriptor.
DAISY Descriptor

- Log-polar grid arrangement
- Gaussian pooling of histograms of gradient orientations
- Efficient for dense computation, but not for sparse keypoints!
Efficient Dense Computation of Features

- The computation mostly involves 1D convolutions.
- Rotating the descriptor only involves reordering the histograms.