Visual Features: Descriptors (SIFT, BRIEF, and ORB)

Cyrill Stachniss

Most slides have been created by Cyrill Stachniss but for several slides courtesy by Gil Levi, A. Efros, J. Hayes, D. Lowe and S. Savarese
Motivation
Motivation
Visual Features: Keypoints and Descriptors

- **Keypoint** is a (locally) distinct location in an image
- The feature **descriptor** summarizes the local structure around the keypoint
Keypoint and Descriptor

Keypoint

descriptor at the keypoint

\[
f = \begin{bmatrix} 0.02 \\ 0.04 \\ 0.1 \\ 0.03 \\ 0 \\ \ldots \end{bmatrix}
\]
Today’s Topics

- **Keypoints: Finding distinct points**
  - Harris corners
  - Shi-Tomasi corner detector
  - Förstner operator
  - Difference of Gaussians

- **Features: Describing a keypoint**
  - SIFT – Scale Invariant Feature Transform
  - BRIEF – Binary Robust Independent Elementary Features
  - ORB – Oriented FAST Rotated BRIEF
Keypoints: Difference of Gaussians Over Scale-Space Pyramid (Recap)

**Procedure**
Over different image pyramid levels
- Step 1: Gaussian smoothing
- Step 2: Difference-of-Gaussians and find extrema
- Step 3: maxima suppression for edges
Illustration (Recap)

differently sized images

Scale (next octave)

Scale (first octave)

Gaussian

Difference of Gaussian (DOG)

Image courtesy: Lowe 8
Keypoint Done. What about a Descriptor?

keypoint

descriptor at the keypoint

\[ f = \begin{bmatrix} 0.02 \\ 0.04 \\ 0.1 \\ 0.03 \\ 0 \\ \ldots \end{bmatrix} \]
Can We Describe Keypoints to Enable Matching Across Images?
Is it All About the Vector $f$...

keypoint descriptor at the keypoint

\[ f = \begin{bmatrix} 0.02 \\ 0.04 \\ 0.1 \\ 0.03 \\ 0 \\ \ldots \end{bmatrix} \]
Popular Features Descriptors

- HOG: Histogram of Oriented Gradients
- SIFT: Scale Invariant Feature Transform
- SURF: Speeded-Up Robust Features
- GLOH: Gradient Location and Orientation Histogram
- BRIEF: Binary Robust Independent Elementary Features
- ORB: Oriented FAST and rotated BRIEF
- BRISK: Binary Robust Invariant Scalable Keypoints
- FREAK: Fast REtinA Keypoint
- ...
Popular Features Descriptors

- HOG: Histogram of Oriented Gradients
- **SIFT**: Scale Invariant Feature Transform
- SURF: Speeded-Up Robust Features
- GLOH: Gradient Location and Orientation Histogram
- **BRIEF**: Binary Robust Independent Elementary Features
- **ORB**: Oriented FAST and rotated BRIEF
- BRISK: Binary Robust Invariant Scalable Keypoints
- FREAK: Fast REtinA Keypoint
- ...

SIFT Descriptor

- Image content is transformed into features that are **invariant to**
  - image translation,
  - rotation, and
  - scale

- They are **partially invariant to**
  - illumination changes and
  - affine transformations and 3D projections

- Suitable for detecting visual landmarks
  - **from different angles and distances**
  - **with a different illumination**
SIFT Features

A SIFT feature is given by a vector computed at a local extreme point in the scale space

\[ \langle p, s, r, f \rangle \]

- **pixel location** of the keypoint in the image
- **scale** (extrema in scale space from DoG)
- **orientation**: compute image gradients in a local region, build a histogram and select peak as the keypoint orientation
- 128-dim. **descriptor** generated from local image gradients
SIFT Features

A SIFT feature is given by a vector computed at a local extreme point in the scale space.

\[ \langle p, s, r, f \rangle \]

- Location
- Scale
- Orientation

128-dim. descriptor

- View-point dependent
- Mainly independent
SIFT Considers the Distribution of Gradients Around Keypoints

Image courtesy: Vedaldi and Fulkerson
SIFT Descriptor in Sum

- Compute image gradients in local 16x16 area at the selected scale
- Create an array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions (yields best results)

Example using a 8x8 area:
SIFT Illustration (1)

For a given keypoint, warp the region around it to orientation and scale and resize the region to 16X16 pixels.
SIFT Illustration (2)

Compute the gradients for each pixels (orientation and magnitude) and divide the pixels into 16, 4X4 pixels squares
SIFT Illustration (3)

For each square, compute gradient direction histogram over 8 directions
SIFT Illustration (4)

Concatenate the histograms to obtain a 128 (16*8) dimensional feature vector:
SIFT Descriptor Illustration

Image gradients $\rightarrow$ Keypoint descriptor $\rightarrow$ Feature vector (128)
SIFT Approach Done!

**keypoint** (via DoG)

**descriptor** (via gradient histogram)

\[
f = \begin{bmatrix}
0.02 \\
0.04 \\
0.1 \\
0.03 \\
0 \\
\ldots
\end{bmatrix}
\]
How To Match Them?
Based on Descriptor Difference
Lowe’s Ratio Test

- 3 Step test to eliminate ambiguous matches for a query feature $q$
- 1. Step: Find closest two descriptors to $q$, called $p_1$ and $p_2$ based on the Euclidian distance $d$
- 2. Step: Test if distance to best match is smaller than a threshold: $d(q, p_1) < T$
- 3. Step: Accept match only if the best match is substantially better than second:

$$\frac{d(q, p_1)}{d(q, p_2)} < \frac{1}{2}$$
Based on Ratio Test
Outliers

- Lowe’s Ratio test works well
- There will still remain few outliers
- Outliers require extra treatment
Binary Descriptors
or
“computing descriptor fast”
Why Binary Descriptors?

- Complex features such as SIFT work well and are a gold standard
- SIFT is expansive to compute
- SIFT has patenting issues
- Binary descriptors aim at generating small binary strings that are easy to compute and compare
Key Idea of Binary Descriptors

Fairly simple strategy

- Select a patch around a keypoint
- Select a set of pixel pairs in that patch
- For each pair, compare the intensities

\[ b = \begin{cases} 1 & \text{if } I(s_1) < I(s_2) \\ 0 & \text{otherwise} \end{cases} \]

- Concatenate all \( b \)'s to a bit string
Example

image area

index (for pairs)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

pairs: \{(5, 1), (5, 9), (4, 6), (8, 2), (3, 7)\}

tests: \[ b = 0 \quad b = 0 \quad b = 0 \quad b = 1 \quad b = 1 \]

result: \[ B = 00011 \]
Key Advantages of Binary Descriptors

- **Compact descriptor**
  The number of pairs gives the length in bits

- **Fast to compute**
  Simply intensity value comparisons

- **Trivial and fast to compare**
  Hamming distance

\[ d_{\text{Hamming}}(B_1, B_2) = \sum(\text{xor}(B_1, B_2)) \]
Different Binary Descriptors Differ Mainly by the Strategy of Selecting the Pairs
Important Remark – Pairs

In order to compare descriptors among images, we must:

- Use the same pairs
- Maintain the same order in which the pairs are tested

Different descriptors once determine the way the pairs are chosen and fix it!
BRIEF:
Binary robust independent elementary features

- First binary image descriptor
- Proposed in 2010
- 256 bit descriptor
- Provides five different geometries as sampling strategies
- Noise: operations performed on a smoothed image to deal with noise
BRIEF Sampling Pairs
BRIEF Sampling Pairs

- G I: Uniform random sampling
- G II: Gaussian sampling
- G III: $s_1$ Gaussian; $s_2$ Gaussian centered around $s_1$
- G IV: Discrete location from a coarse polar grid
- G V: $s_1=(0,0)$; $s_2$ are all location from a coarse polar grid
Performance: G I – G IV are all good, G V less useful
ORB:
Oriented FAST Rotated BRIEF

An extension to BRIEF that
- Adds rotation compensation
- Learns the optimal sampling pairs
ORB: Rotation Compensation

- Estimates the center of mass and the main orientation of the area/patch
- Image moment
  \[ m_{pq} = \sum_{x,y} x^p y^q I(x, y) \]
- Center of mass
  \[ C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \]
- Orientation
  \[ \theta = \text{atan2}(m_{01}, m_{10}) \]
ORB: Rotation Compensation

- Given CoM and orientation $C, \theta$, we can rotate the coordinates of all pairs by $\theta$ around $C'$:

$$s' = T(C, \theta) s$$

- Use the transformed pixel coordinates for performing the test
- Invariance to rotation in the plane
ORB: Learning Sampling Pairs

Pairs should be / have

- **uncorrelated** – so that each new pair adds new information to the descriptor
- **high variance** – it makes a feature more discriminative

- ORB defines a strategy for selection 256 pairs optimizing for both properties using a training database
ORB vs. SIFT

- ORB is 100x faster than SIFT
- ORB: 256 bit vs. SIFT: 4096 bit
- ORB is not scale invariant (achievable via an image pyramid)
- ORB mainly in-plane rotation invariant
- ORB has a similar matching performance as SIFT (w/o scale)
- Several modern online systems (e.g. SLAM) use binary features
Summary

- Keypoints and descriptor together define common visual features
- Keypoint defines the location
- Descriptor describes the appearance
- Several descriptors operating on gradient histograms (SIFT, SURF, ...)
- Binary descriptors for efficiency (BRIEF, ORB, ...)

Slide Information

- These slides have been created by Cyrill Stachniss as part of the Photogrammetry II course taught in 2014 and 2019
- The slides *heavily* reply on material by Gil Levi, Alexei Efros, James Hayes, David Lowe, and Silvio Savarese
- I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, 2014
cyrill.stachniss@igg.uni-bonn.de