Learned Visual Features for Feature Matching

IN2107 Visual Feature Learning in Autonomous Driving

Students: Farid Yagubbayli (03708842) & Kuanhsun Wu (03758530)
Advisor: Burcu Karadeniz
Organizers: Burcu Karadeniz, Emec Ercelik, Sina Shafaei

Chair of Robotics, Artificial Intelligence and Real-Time Systems
TUM Department of Informatics
Garching bei München, 12 July 2019
Outline

- **Introduction**
- **Classical Models**
  - Detectors
  - Descriptors
  - Matching & Outlier Handling
  - Comparison
- **Deep Learning Methods**
  - Detectors
  - Descriptors
  - Matching
  - End-to-End learning
  - Comparison
- **Conclusions**
Motivation

3D Reconstruction

Object Recognition

Face Detection

Localization & Mapping

[1] Thomas Schöps, ETH Zurich Computer Vision Group
[2] Shutterstock
[3] OpenFace
[4] PiRobot.org

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
What is Feature?

Blob features
“Pre-defined Shapes”

Point features
“Interesting points”
Outline

- Introduction
- Classical Models
  - Detectors
  - Descriptors
  - Matching & Outlier Handling
  - Comparison
- Deep Learning Methods
  - Detectors
  - Descriptors
  - Matching
  - End-to-End learning
  - Comparison
- Conclusions
Feature Detectors

Find “interesting” points:
* Unique in neighborhood
* Robust to geometric changes
* Robust to photometric changes
* Has small spatial size
* Easy to detect

Image Gradients help us a lot…

[1] by Mehmet Soydaş
Feature Detectors

**SIFT**

- Scale (first octave)
- Difference of Gaussian (DOG)
- blur

**FAST**

- OpenCV Documentation

* Many variations exist to make detectors robust to Rotation, Illumination, Affine Transformation and etc.

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)

[2] OpenCV Documentation
Feature Descriptors

Feature vectors should be
* Unique
* Robust to geometric changes
* Robust to photometric changes
* Easy to compute / compare

Descriptor - Does many tricks

Feature Vector

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
Feature Descriptors / Real

SIFT Descriptor as an example.
> “How much does it change?” & “In which direction?”
> Divide neighborhood to sub-regions
> Count for each direction

[1] yg.aliyun.com, How to understand HoG
Feature Descriptors / Real

Many variations of SIFT - RoboSIFT, **DSP-SIFT** …

Additional **Domain-Size Pooling** for gradients

Feature Descriptors / Binarized

What is problem?

• Real descriptor vectors use **floating points**
• Floating point representation uses 4 bytes
• In case of SIFT, $4 \times 128 = \text{512 bytes for each descriptor}$

Solution?
- Convert floating point vectors to **binary strings**

How?
- May not need all dimensions, compress / summarize

**Methods**
- **PCA-DSC**: Principal Component Analysis **[✓ Less memory usage]**
- **LDAHash**: Linear Discriminant Analysis **[✓ Hamming distance function]**
- **LSH**: Locality Sensitive Hashing
Feature Descriptors / True Binary

Why not Binarized?

- We still need to find full real vectors before hashing => Memory

BRIEF

128, 256 or 512 comparisons
16, 32 or 64 bytes in memory

ORB

Oriented FAST + Rotated BRIEF

BRISK

Describe with Gaussian Blurred Concentric Circles

Orientation from long-distance point comparisons

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)

[1] by Philippe Fillatreau
[2] OpenCV Documentation
Comparison / Descriptor

Comparative Evaluation of Binary Features

<table>
<thead>
<tr>
<th>Detector/Descriptor</th>
<th>BRIEF</th>
<th>ORB</th>
<th>BRISK</th>
<th>SURF</th>
<th>SIFT</th>
<th>Harris</th>
<th>MSER</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg # Features</td>
<td>n/a</td>
<td>1347</td>
<td>7771</td>
<td>3766</td>
<td>4788</td>
<td>2543</td>
<td>693</td>
<td>8166</td>
</tr>
<tr>
<td>Detector Entropy</td>
<td>n/a</td>
<td>12.10</td>
<td>12.33</td>
<td>12.26</td>
<td>12.34</td>
<td>11.84</td>
<td>10.74</td>
<td>12.52</td>
</tr>
<tr>
<td>Detector ms/image</td>
<td>n/a</td>
<td>17</td>
<td>43</td>
<td>377(19)</td>
<td>572(25)</td>
<td>78(4.7)</td>
<td>117</td>
<td>2.7</td>
</tr>
<tr>
<td>Descriptor μ/feature</td>
<td>4.4(0.4)</td>
<td>4.8</td>
<td>12.9</td>
<td>143(6.6)</td>
<td>314(19)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Storage bytes/feature</td>
<td>16,32,64</td>
<td>32</td>
<td>64</td>
<td>64(256)</td>
<td>128(512)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Detector / Descriptor timings,
Comparative Evaluation of Binary Features

Classification of Feature Descriptors

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
Comparison / Descriptor

Adding Cues to Binary Feature Descriptors for Visual Place Recognition

Results for Keypoint Coordinates with different Descriptors

Results for Semantic Labels with different Descriptors

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
Matching / Strategy

Two-Step

Find putative matches → Remove Outliers

Strategy

Correspondence Matrix

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)

[1] Matlab Documentation
[2] youtube.com, Iterative Closes Point (ICP) algorithm
Matching / Similarity

- Kappa-Squared
- Earth Movers (EMD)
- Euclidean
  - * Most used
- Hamming
  - * Binary vectors
- Hellinger
  - * RootSIFT
- Signature Quadratic Form
- Mahalanobis

Decision should depend on used **Descriptor type** & particular **use-case**!
Matching / Candidate Set Generation

Naïve / Exhaustive

Approximate

Approximation speed-ups process but results won't be global anymore!

* Real-valued descriptors: k-D Tree
* Binary-valued descriptors: LSH
Comparison / Matching

Adding Cues to Binary Feature Descriptors for Visual Place Recognition

PR analysis on KITTI dataset with different Matching Strategies

Farid Yagoubayli (03708842) | Kuanhsun Wu (03758530)
Random Sample Consensus (RANSAC) is one of the popular algorithms iteratively try to find “good” pairs from putative matches by rejecting outliers.

Step 1. Random Sampling

Step 2. Estimate Homography

\[ H \]

Step 3. Evaluate Homography

\#inliers > thresh \implies \text{reject others}

else \implies \text{step 1}
RANSAC works but slow. Fortunately there’s room for improvements.

**PROSAC - Progressive Sample Consensus**

Improves Step 1 by weighting pairs instead of random selection.
Assumption: There’s reasonable distance metric.
Be careful: Degenerate configuration

**WaldSAC**

Improves Step 3 by probabilistic approach to homography quality
Speed-ups from 2 to 7 times

<table>
<thead>
<tr>
<th>N=1322</th>
<th>RANSAC</th>
<th>WaldSAC</th>
<th>PROSAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inliers</td>
<td>884</td>
<td>889</td>
<td>885</td>
</tr>
<tr>
<td># of models</td>
<td>112</td>
<td>148</td>
<td>5</td>
</tr>
<tr>
<td># of verifications</td>
<td>1322</td>
<td>594</td>
<td>1322</td>
</tr>
<tr>
<td>speed-up</td>
<td>1.0</td>
<td>2.2</td>
<td>12.9</td>
</tr>
</tbody>
</table>

A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus

And many more…
Matching / GMS

Grid Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence

Verify putative matches using neighborhood support!

[1] GMS CVPR 2017 Paper
Matching / GMS

Grid Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence

- Fast enough for real-time applications
- Assumes suitable spatial size for each neighborhood
- Each feature point treated individually, therefore no global structure

Comparison of GMS with other matchers in TUM Dataset (results are similar for Strecha Dataset).

Grid-based Motion Statistics

[1] youtube.com, GMS CVPR 2017 Demo
Matching / GMS

**GMS + PROSAC**
Solves GMS’s locality problem by introducing Epipolar Geometric Constrain (EGC)

---

**Left Picture**
- Detecting ORB Features
- Describing ORB Features

**Right Picture**
- Detecting ORB Features
- Describing Features

- Calculating Hamming Distance
- Brute-Force Match
- Eliminating Mismatches via GMS
- Calculating EGC Model
- Eliminating Mismatches via EGC PROSAC
- Excellent Matching Set

---

[1] by Panpan Zhao et al., GMS-PROSAC Paper

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
Comparison / Descriptor

Reliable Visual Localization for Autonomous Vehicles in Urban Outdoor Environments

Descriptor performances under various conditions in Autonomous Driving domain
Outline

• Introduction
• Classical Models
  • Detectors
  • Descriptors
  • Matching & Outlier Handling
• Deep Learning Methods
  • Detectors
  • Descriptors
  • Matching
  • End-to-End learning
  • Comparison
• Conclusions
TILDE (Temporally Invariant Learned Detector)

- Regression-based approach to extract feature points under drastic illumination changes
- Base on full image
- Not robust to viewpoint change and scale
CovDet (Covariant Feature Detectors)

- Detection as a regression problem
- Patch-based
- Not unique solution

\[ \ell_{\text{covariant}} = \sum_{i=1}^{n} \| \phi(g_i \ast x_i) - g_i \circ \phi(x_i) \|^2_F \]
TCovDet (Transformed Covariant Feature Detectors)

- Embed “standard patch” which is from TILDE
- Extract pre-determined features obtained by hand-crafted detectors
The additional geometric constraints enforces the network to choose points which are stable.
## Descriptor

<table>
<thead>
<tr>
<th></th>
<th>DeepDesc</th>
<th>PN-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer of network</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Time for one descriptor</td>
<td>0.76ms</td>
<td>10μS</td>
</tr>
<tr>
<td>Loss</td>
<td>Hinge</td>
<td>Triplet</td>
</tr>
</tbody>
</table>

- **Hard negative mining**: more negative pairs
- **find and correct high-rank error**

**Hinge Margin** [20]

\[
L^- = \max(0, \mu - \Delta)
\]

**Our SoftPN**

\[
l(\tau) = \left( \frac{e^{\Delta(p_1,p_2)}}{e^{\Delta(p_1,p_2)} + e^{\min(\Delta(p_1,n),\Delta(p_2,n))}} + e^{\min(\Delta(p_1,n),\Delta(p_2,n))} \right)^2
\]

\[
l(\mathcal{P}) = \begin{cases} 
\Delta = \|D(p_L) - D(p_R)\|_2 & \text{if } \mathcal{L} = 1 \\
\max(0, \mu - \|D(p_L) - D(p_R)\|_2) & \text{if } \mathcal{L} = -1 
\end{cases}
\]

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
DOAP (Descriptors Optimized for Average Precision)

- Directly optimize a ranking-based retrieval performance metric, Average Precision
- Free of Hard negative mining
MatchNet

- Metric can significantly improve performance: metric Learning
- Output from the two towers are concatenated as the metric network’s input
LIFT (Learned Invariant Feature Transform)

- Combined TILDE and DeepDesc by adding Spatial Transformers layers as Orientation Estimator
- Impossible to train the whole architecture, so train back from descriptor
SuperPoint (Self-Supervised Interest Point Detection and Description)

- Shared encoder with CNNs incorporate both detector and descriptor dense information tensor
- To keep the model fast and easy to train, both decoders use non-learned upsampling
Homographic Adaptation

- The process can be repeated iteratively to continually self-supervise
- pseudo-ground truth point, ground truth correspondence from H as ground true
Comparison / Detector

- TCDET-S (TCovDet) generally outperform others
- FAST-T is the fastest
**Comparison / Descriptor**

![Comparison Table]

- **Superpoint**
  - Dimension: 256
  - GPU: 13(ms), 70FPS

---

Farid Yagubbayli (03708842) | Kuanhsun Wu (03758530)
Autonomous Driving Application

• Stanford’s entry in the 2007 DARPA Urban Challenge (Histogram, Haar Filter)
• Vision-based Offline-Online Perception Paradigm for Autonomous Driving
  FAST Detector
• Are we ready for Autonomous Driving?
  The KITTI Vision Benchmark Suite
  HOG Detector
Outline

- Introduction
- Classical Models
  - Detectors
  - Descriptors
  - Matching & Outlier Handling
  - Comparison
- Deep Learning Methods
  - Detectors
  - Descriptors
  - Matching
  - End-to-End learning
  - Comparison
- Conclusions
Conclusion

• Except some recent techniques (DOAP & SuperPoint), handcrafted methods outperform deep learning based techniques in feature description.

• Application of recent Deep Learning based techniques into Autonomous Driving domain is an open research area.

• Combined approach of learned and handcrafted methods can be useful to take advantages of both methods.

• Usage of learned methods is constrained by dataset.

• Binary descriptors in both parts show competitive results and common in practical applications. They can be used whenever applications provides robustness against outliers.

• SuperPoint is the most suitable method for autonomous driving, combine detector and descriptor in a single network with high mAP and real time computational cost.

• DOAP has the highest mean average value of all tasks.

• Handcrafted detectors, descriptors have strength in terms of computational time.
Thank you. Questions?
References

References


References


References

References

• In Winter Conference on Applications of Computer Vision (WACV), 2015.