Deep Visual SLAM Frontends:
SuperPoint, SuperGlue, and SuperMaps

Tomasz Malisiewicz
June 14, 2020

Joint Workshop on Long-Term Visual Localization, Visual Odometry and Geometric and Learning-based SLAM
@ CVPR 2020
Talk Outline

• **SuperPoint**: architectures and training paradigms you *need* to know to replace local features with Convolutional Neural Networks

• **SuperGlue**: how to utilize Graph Neural Networks and Attention to improve feature matching

• **SuperMaps**: moving beyond pairwise matching and a roadmap towards end-to-end Deep Visual SLAM
Part I: SuperPoint

The art and craft of designing ConvNets to replace SIFT.
Two parts of Visual SLAM

- **Frontend**: Image inputs
  - Deep Learning success: Images + ConvNets
- **Backend**: Optimization over pose and map quantities
  - Use Bundle Adjustment

Photo Credit: Cadena et al 2016
SuperPoint: A Deep SLAM Front-end

- Powerful fully convolutional design
- Points + descriptors computed jointly, **No Patches**
- Share VGG-like backbone
- Designed for real-time processing on a GPU
- Medium-sized backbone. Tasks share ~90% of compute

Keypoint / Interest Point Decoder

- No deconvolution layers
- Each output cell responsible for local 8x8 region
How To Train SuperPoint?

Image ➔ ConvNet ➔ Keypoint 2D Locations ➔ Keypoint Descriptors
Setting up the Training

- Siamese training with pairs of images
- Descriptor trained via metric learning (contrastive loss)
  - Straightforward given correspondence
- Keypoints trained via supervised keypoint labels
- Where do these come from?
How to get Keypoint Labels for Natural Images?

- Need large-scale dataset of annotated images
- Too hard for humans to label
Self-Supervised Training

Synthetic Shapes (has interest point labels)

First train on this

“Homographic Adaptation”

MS-COCO (no interest point labels)

Use resulting detector to label this
Synthetic Training

- Non-photorealistic shapes
- Heavy noise
- Effective and easy

Quads/Tris

Quads/Tris/Ellipses

Cubes

Checkerboards

Lines

Stars
Early Version of SuperPoint (MagicPoint)

Corner Detection Average Precision vs Degree of Noise
Synthetic Shapes, 160 x 120, (ε = 4)

Homographic Adaptation

- Simulate planar camera motion with homographies
- Self-labelling technique
  - Suppress spurious detections
  - Enhance repeatable points
SuperPoint Example #1

SuperPoint

LIFT

SIFT

ORB
SuperPoint Example #2
SuperPoint Example #3

SuperPoint

LIFT

SIFT

ORB
3D Generalizability of SuperPoint

- Trained+evaluated on planar, does it generalize to 3D?
- “Connect-the-dots” using nearest neighbor matches
- Works across many datasets / input modalities / resolutions!
Pre-trained SuperPoint Release

• Implemented in PyTorch
• Two files, minimal dependencies. Get up and running in 5 minutes or less!
• Released at 1st Deep Learning for Visual SLAM Workshop at CVPR 2018

[Image of pre-trained SuperPoint network output]

github.com/magicleap/SuperPointPretrainedNetwork
Can we apply SuperPoint to other tasks?

• What if we adapt the SuperPoint architecture to object instance detection?

CharucoNet can “see” in the dark

SuperPointVO

Can we improve SuperPoint with real data and a Visual Odometry backend?

VO Reconstruction on Freiburg-TUM RGBD
‘structure_texture_far’
VO Reconstruction on Freiburg-TUM RGBD
‘long_office_household’

Top-Down Trajectory
Benefits of VO-based SuperPoints

- Establish correspondence across time
- Learn which points are stable
Keypoint 2D Locations
Keypoint Stability
Keypoint 2D Locations
Keypoint Descriptors

ConvNet

Convolutional Frontend

Point Tracks

VO Backend

3D Points

stable
unstable
ignore

Labeled Point Tracks

Self-Supervision from VO

Supervision Signal

Input Monocular Sequence

ConvNet

6DOF Trajectory

#1
#3
#2
#1
#2
#2
#3
#4
#4
#4
#1
#2
#3

27
How to define Stability?

- For sufficiently long tracks, look at the reprojection error

$$X_{\text{stable}} = \begin{cases} \text{Stable} & \text{, if reprojection error is } < 1 \text{ pixel} \\ \text{Not Stable} & \text{, if reprojection error is } > 5 \text{ pixels} \\ \text{Ignore} & \text{, else} \end{cases}$$

- **Stable Points:** Positives
- **Not Stable Points:** Negatives
- **Other Points:** Ignore
VO Stability Labeling

t-junctions across depth aka “sliders”

lighting highlights

dynamic motion
Siamese Training on Sequences

Labeled Sequence

Randomly Select Pair

Random Homography

H₁

H₂

SuperPointVO

Keypoint Loss

Descriptor Loss

Keypoint Loss

Siamese Training on Sequences
SuperPointVO: Pose Estimation on ScanNet

• Small baseline of ~1 second: VO helps a tiny bit
SuperPointVO: Pose Estimation on ScanNet

Pose Accuracy (frame difference = 60)

- Medium baseline of ~2 seconds: VO starts helping
SuperPointVO: Pose Estimation on ScanNet

- Widest baseline of ~3 seconds, biggest performance gap
Part II: SuperGlue

Deep Matching with SuperPoint: Can we learn to solve the correspondence problem?
SuperGlue: Learning Feature Matching with Graph Neural Networks

Paul-Edouard Sarlin¹
Tomasz Malisiewicz²
Daniel DeTone²
Andrew Rabinovich²

ETH zürich

SuperGlue = Graph Neural Nets + Optimal Transport

- Extreme **wide-baseline** image pairs in **real-time** on GPU
- State-of-the-art indoor+outdoor matching with SIFT & SuperPoint

SuperGlue’s goal is to be better than motion-guided matching without any motion model!
SuperGlue requires both sets of local features: a paradigm shift in matching!

A Graph Neural Network with attention

Encodes contextual cues & priors
Reasons about the 3D scene

Solving a partial assignment problem

Differentiable solver
Enforces the assignment constraints = domain knowledge

SuperGlue requires both sets of local features: a paradigm shift in matching!
SuperPoint + NN + heuristics

SuperPoint + SuperGlue

SuperGlue: more **correct matches** and fewer **mismatches**
Results: outdoor - SfM

SuperPoint + NN + OA-Net (inlier classifier)

SuperGlue

SuperGlue: more **correct matches** and fewer **mismatches**
Evaluation

SuperGlue yields **large improvements** in all cases.
Demo: 15 FPS for 512 keypoints on GPU

psarlin.com/superglue

github.com/magicleap/SuperGluePretrainedNetwork
Part III: SuperMaps

What comes after SuperPoint + SuperGlue?
<table>
<thead>
<tr>
<th><strong>SuperPoint+SuperGlue</strong></th>
<th><strong>SuperMaps</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Works with a <strong>pair</strong> of images</td>
<td>Works with a <strong>set</strong> of images</td>
</tr>
<tr>
<td>Uses <strong>classical</strong> pose estimation system</td>
<td>Estimates pose <strong>inside</strong> the network</td>
</tr>
<tr>
<td><strong>No loop closure</strong> mechanism</td>
<td><strong>Keyframe embeddings</strong> to close loops</td>
</tr>
<tr>
<td>Modules trained <strong>independently</strong></td>
<td><strong>Joint end-to-end training</strong></td>
</tr>
<tr>
<td>Has <strong>multiple</strong> notions of receptive field</td>
<td>A <strong>unified</strong> notion of receptive field</td>
</tr>
</tbody>
</table>
Quō vādis Visual SLAM?

(some open problems at the intersection of DL and SLAM that will drive innovation)

1. Multi-user SLAM: Creating representations/maps that work across a large number of agents

2. Integrating object recognition capabilities into SLAM frontends

3. Enabling life-long learning: letting the system automatically improve over time
Summary

• **SuperPoint**: A Convolutional Neural Network Architecture for Visual SLAM frontends
  
  • *Self-Supervised Learning via*:
    
    • Homographies
    
    • Visual Odometry Backend
    
    • CharucoNet: Pattern-specific SuperPoints: can “see” in the dark

• **SuperGlue**: Amazing success in applying Graph Neural Networks and Attention to wide baseline image matching problems

• **SuperMaps**: Ideas for going beyond pairwise matching and end-to-end SLAM
SuperGlue
Learning Feature Matching with Graph Neural Networks
CVPR 2020 Oral
1st place
in 2 visual localization challenges
Joint Workshop on Long-Term Visual Localization, Visual Odometry and Geometric and Learning-based SLAM

Winning entry:
restricted keypoints (2k) / standard descriptors (512 bytes)
SuperGlue Presentations @ CVPR 2020

Local Feature Challenge
Monday, June 15th: 9:10am PT

Handheld Devices Challenge
Monday, June 15th: 9:35am PT

3D Scene Understanding for Vision, Graphics, and Robotics Workshop
Monday, June 15th: 10:25 am PT

CVPR 2020 Oral Presentation
Wednesday, June 17th: 10:40 am PT & 10:40 pm PT

Image Matching: Local Features & Beyond Workshop
Friday, June 19th: 11:45 am PT

Paul-Edouard Sarlin
ETHZ Ph.D. Student
Thank you

Tomasz Malisiewicz
https://tom.ai/

Daniel DeTone
https://danieldetone.com/

Paul-Edouard Sarlin
https://psarlin.com/

Follow us on Twitter:

@quantombone
@ddetone
@pesarlin

Research Questions: 49