

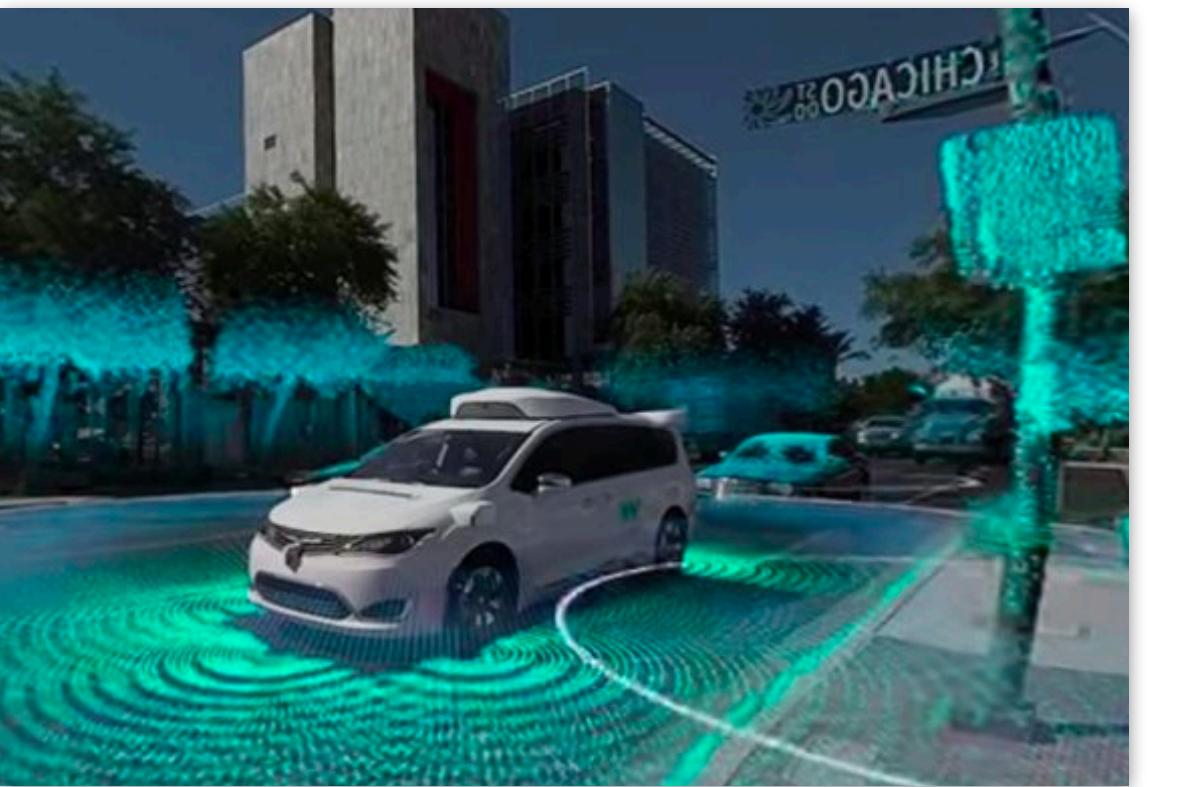
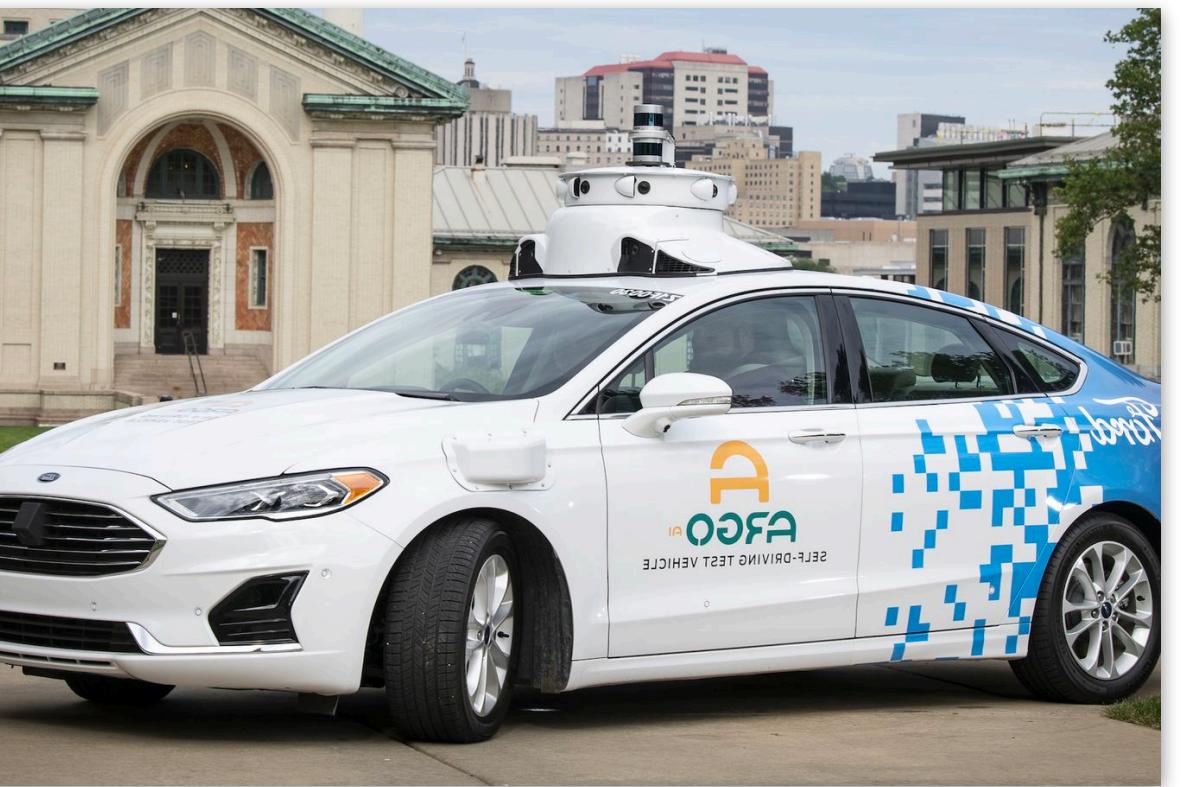
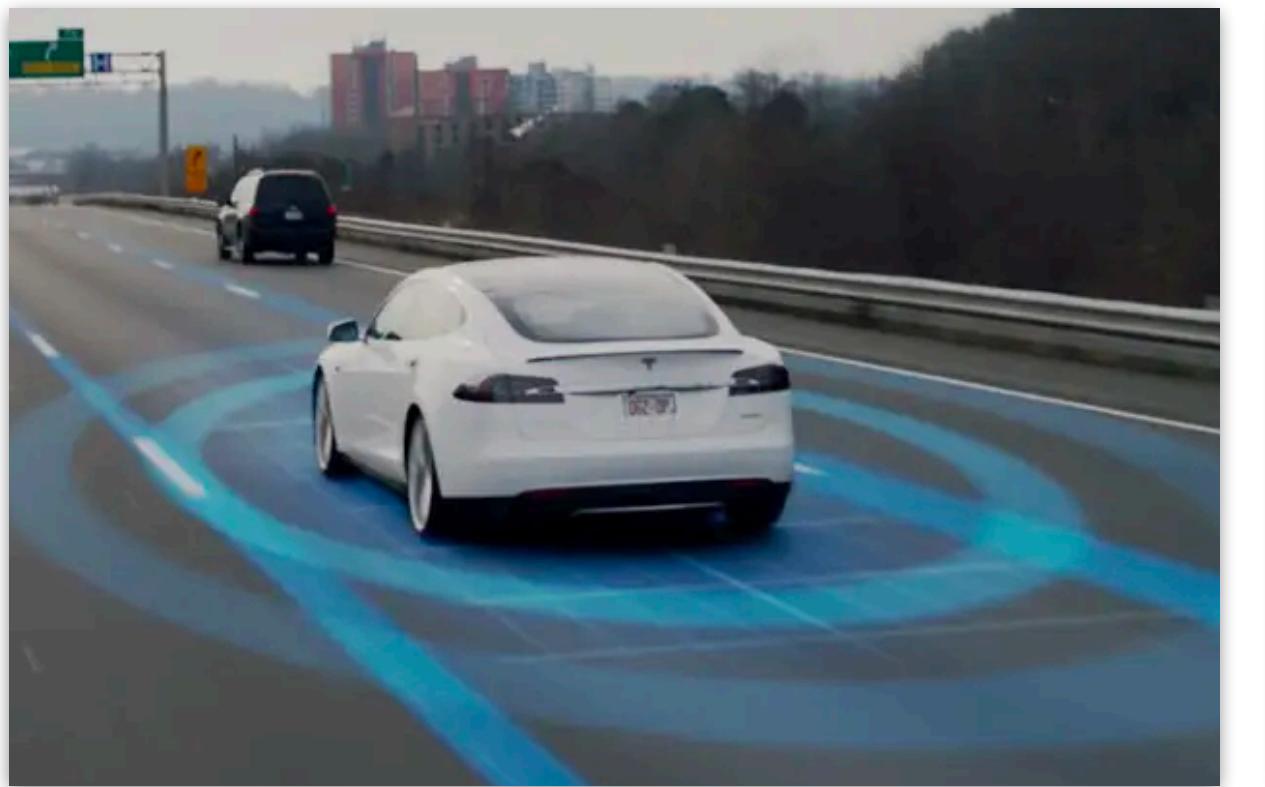
# **SuperPoint and SuperGlue: Lessons Learned**

Tomasz Malisiewicz (Meta Reality Labs)

June 20th, 2022

Image Matching: Local Features & Beyond  
CVPR 2022 Workshop





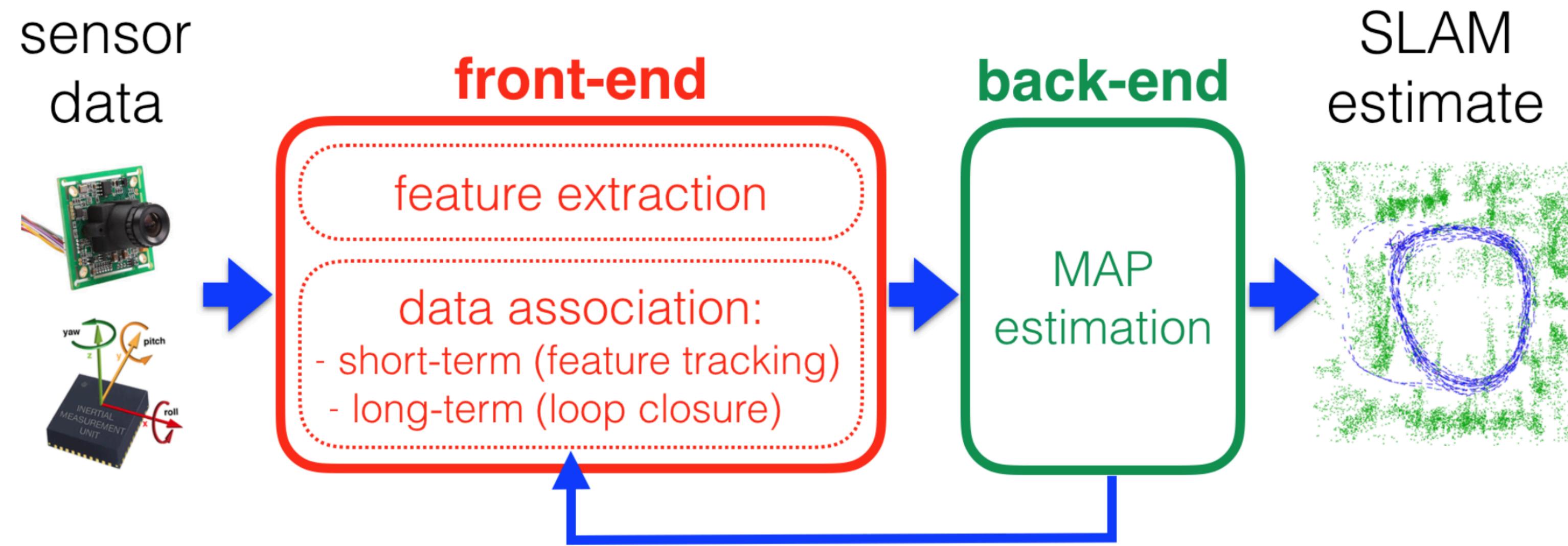
# Talk Outline

- **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
- **Lessons Learned:** What did I learn from these projects that I can teach you?

# Part I: SuperPoint

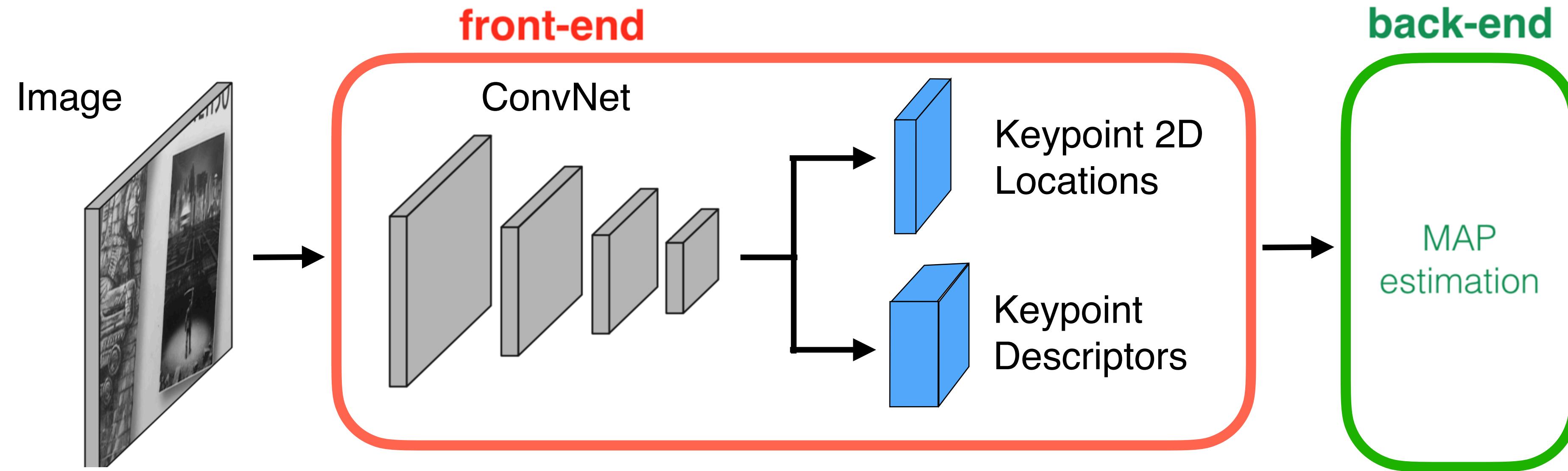
*The art and craft of designing  
ConvNets to replace SIFT.*

# Two parts of Visual SLAM



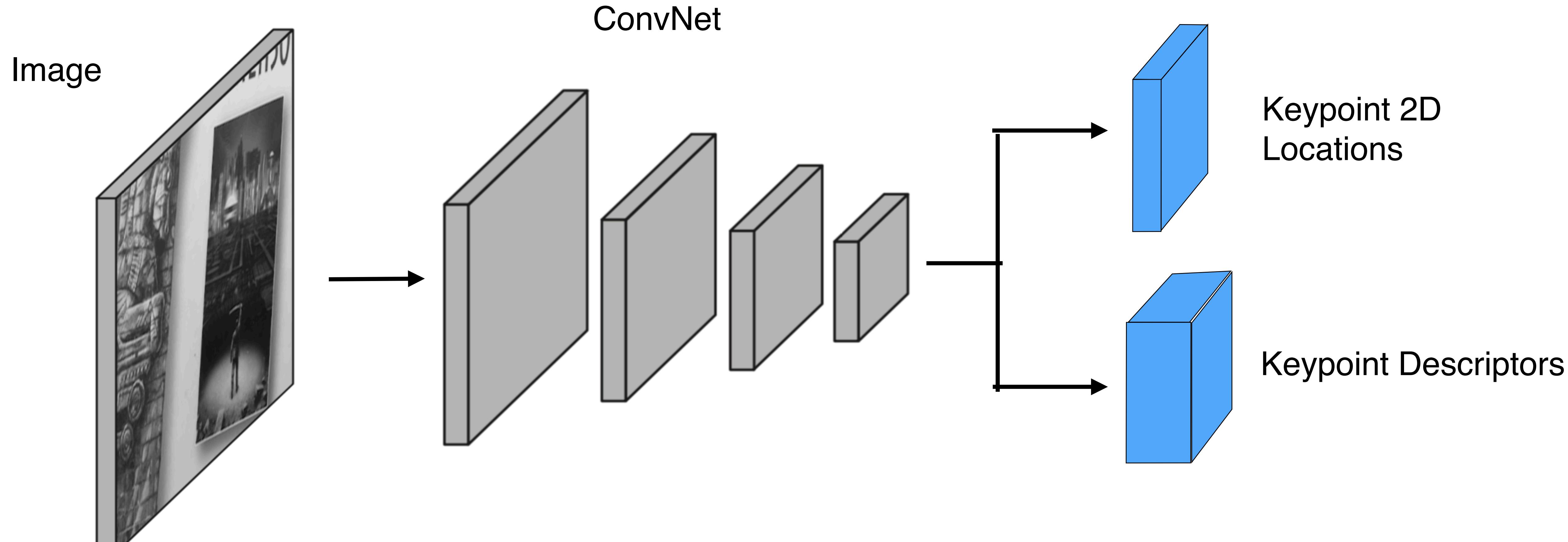
- **Frontend:** Image inputs
  - Deep Learning success: Images + ConvNets
- **Backend:** Optimization over pose and map quantities
  - Use Bundle Adjustment

# SuperPoint: A Deep SLAM Front-end

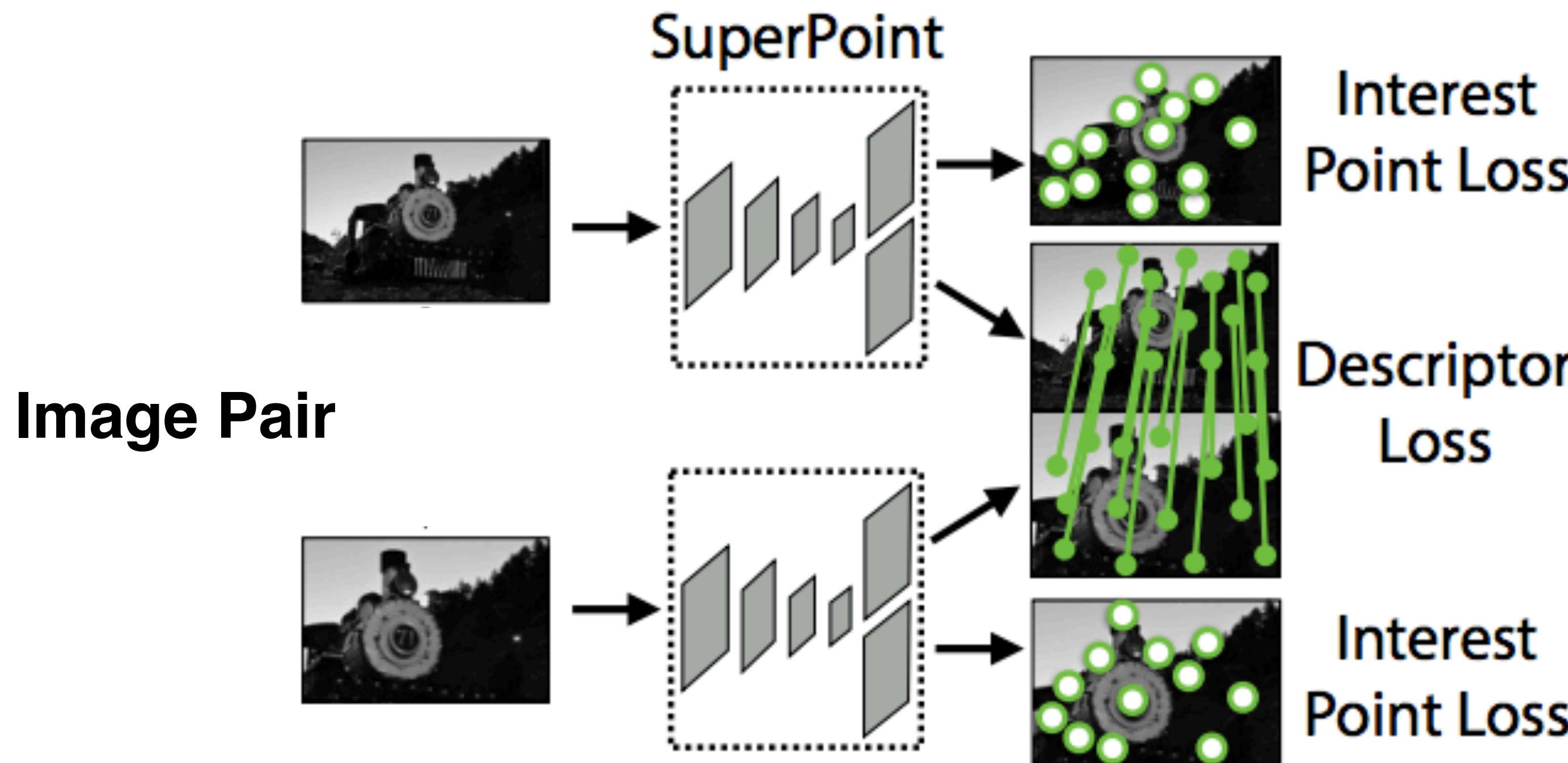


- Powerful fully convolutional design
  - Points + descriptors computed jointly, **No Patches**
  - Share VGG-like backbone
- Designed for real-time processing on a GPU
  - Medium-sized backbone. Tasks share ~90% of compute

# How To Train SuperPoint?



# 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?

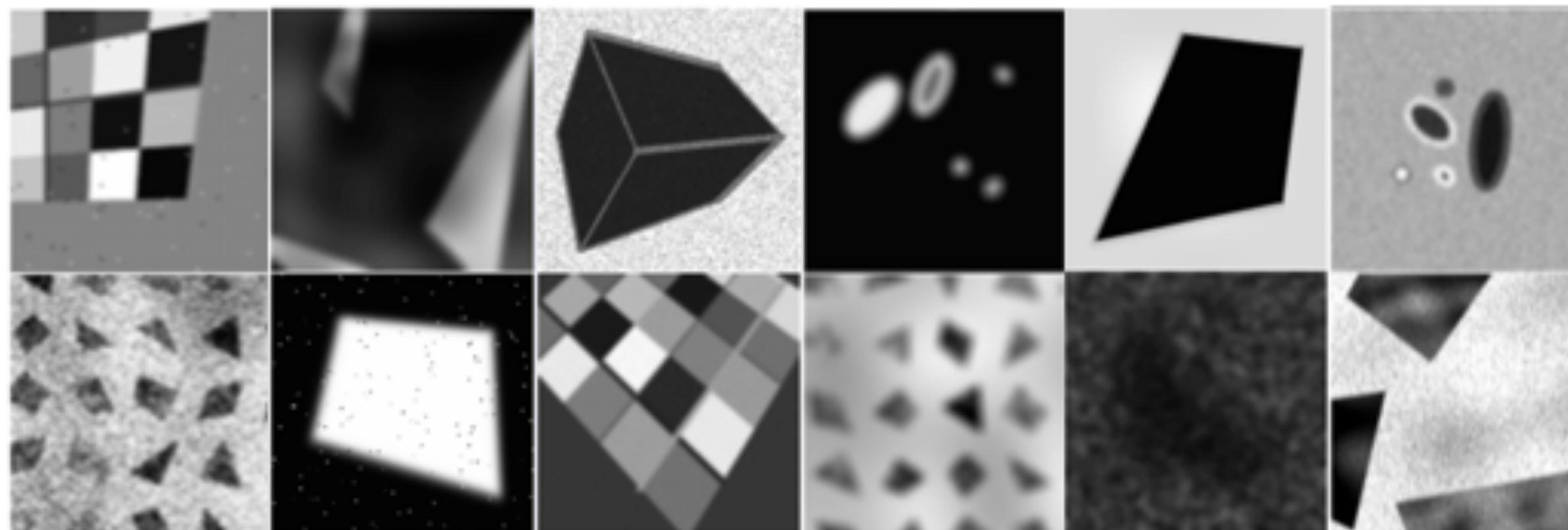
# How to get Keypoint Labels for Natural Images?



- 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

MS-COCO (no interest point labels)

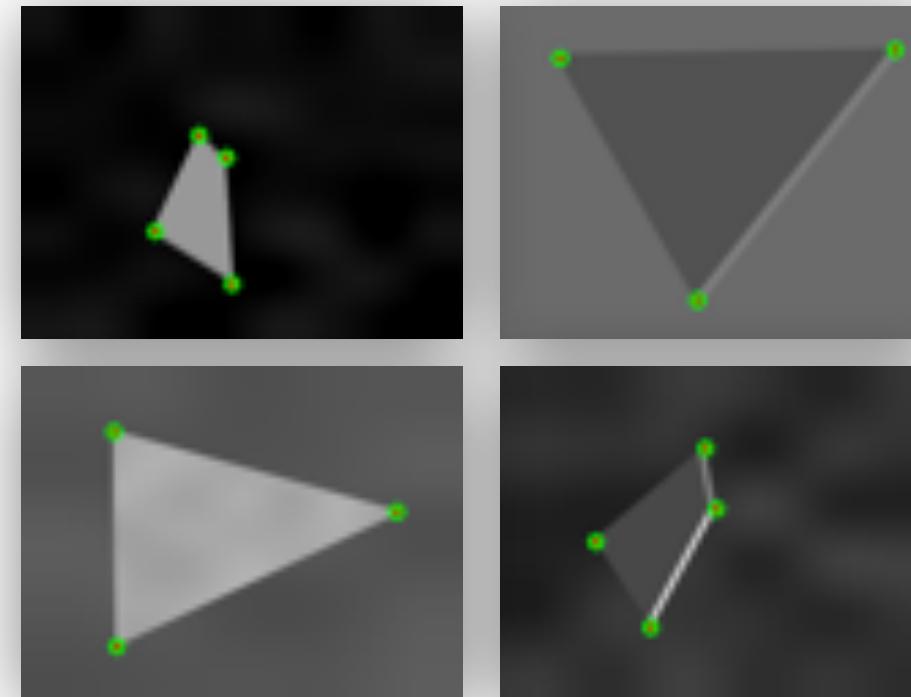


“Homographic  
Adaptation”

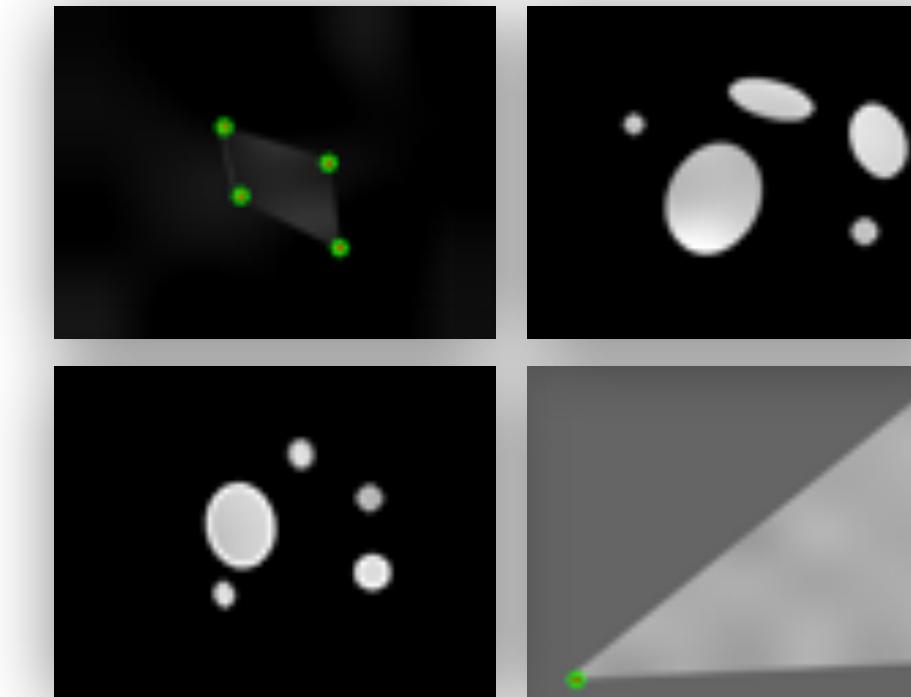
Use resulting  
detector to  
label this

# Synthetic Training

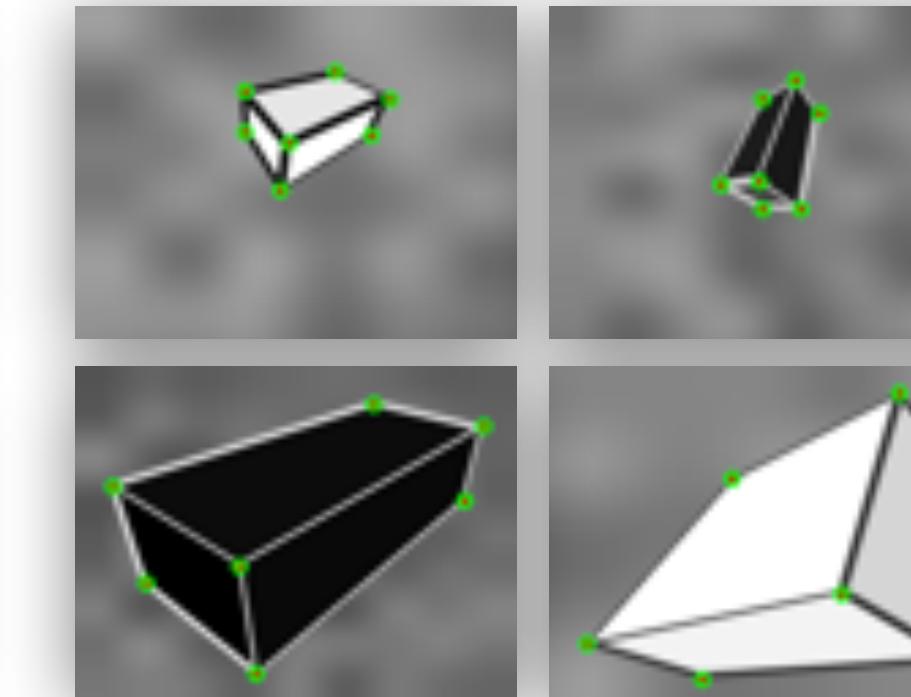
- Non-photorealistic shapes
- Heavy noise
- Effective and easy



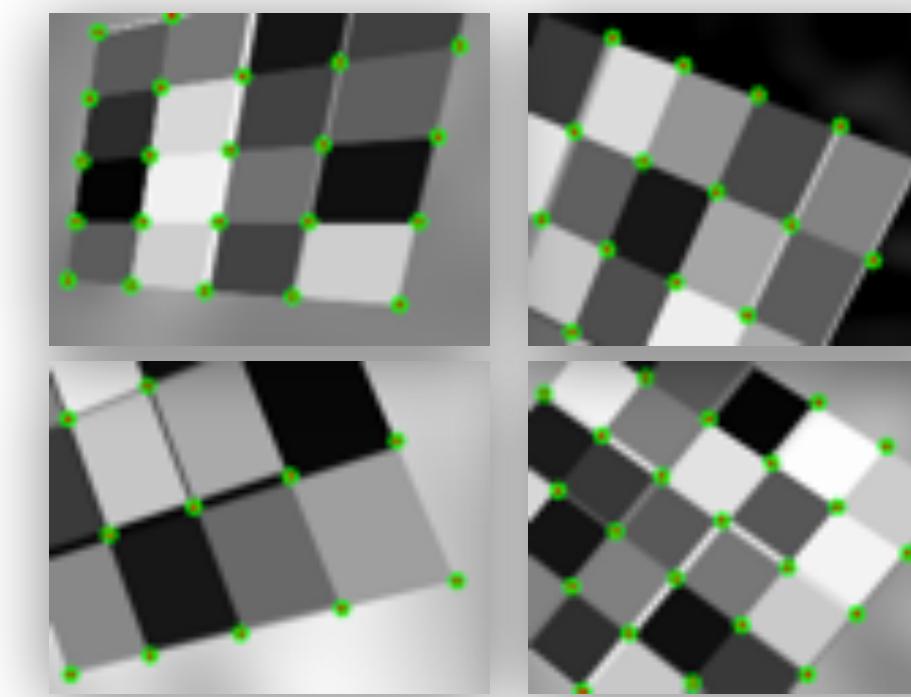
Quads/Tris



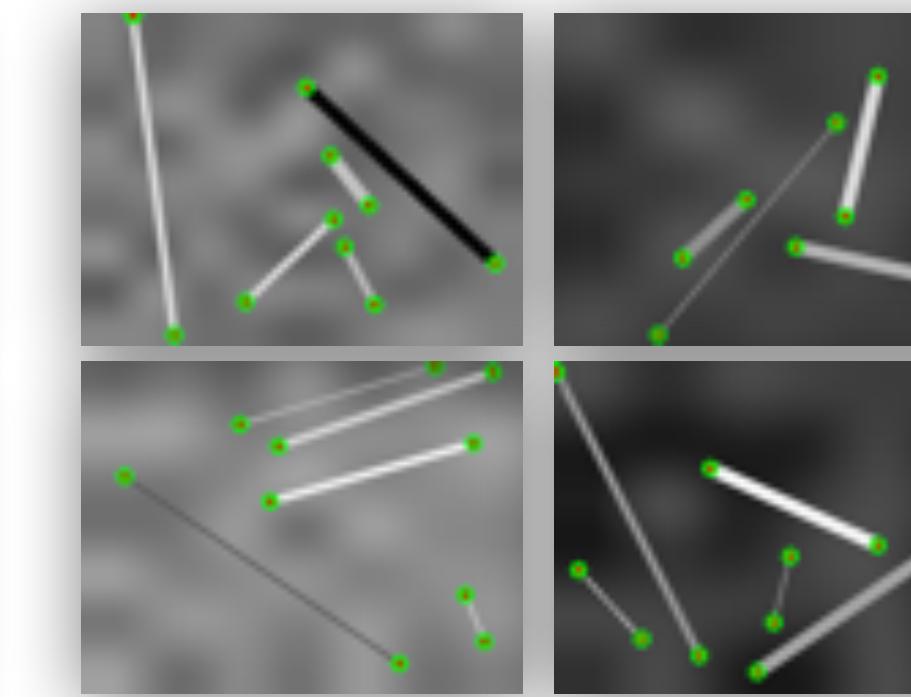
Quads/Tris/Ellipses



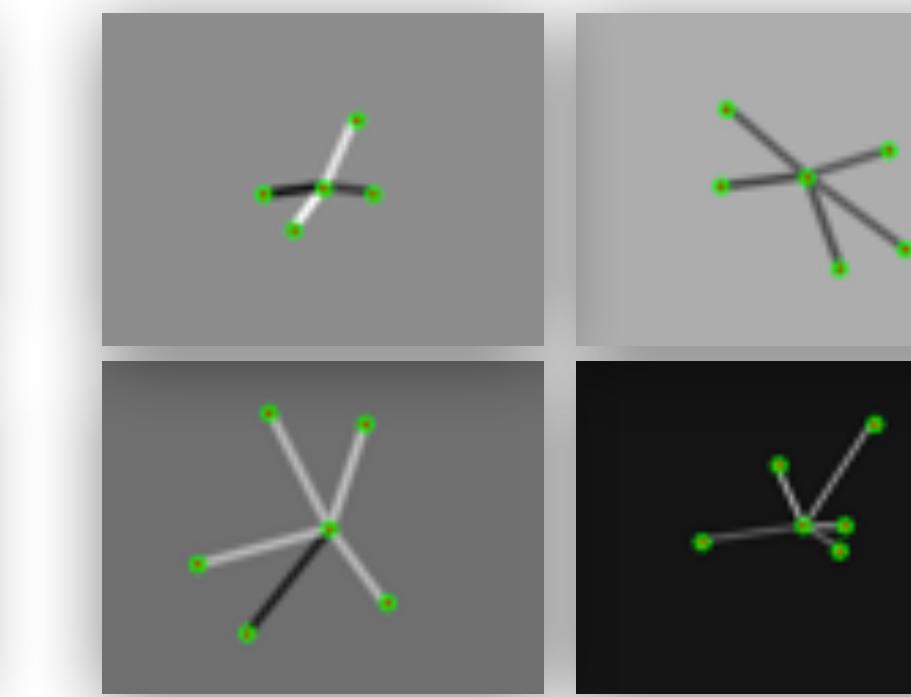
Cubes



Checkerboards

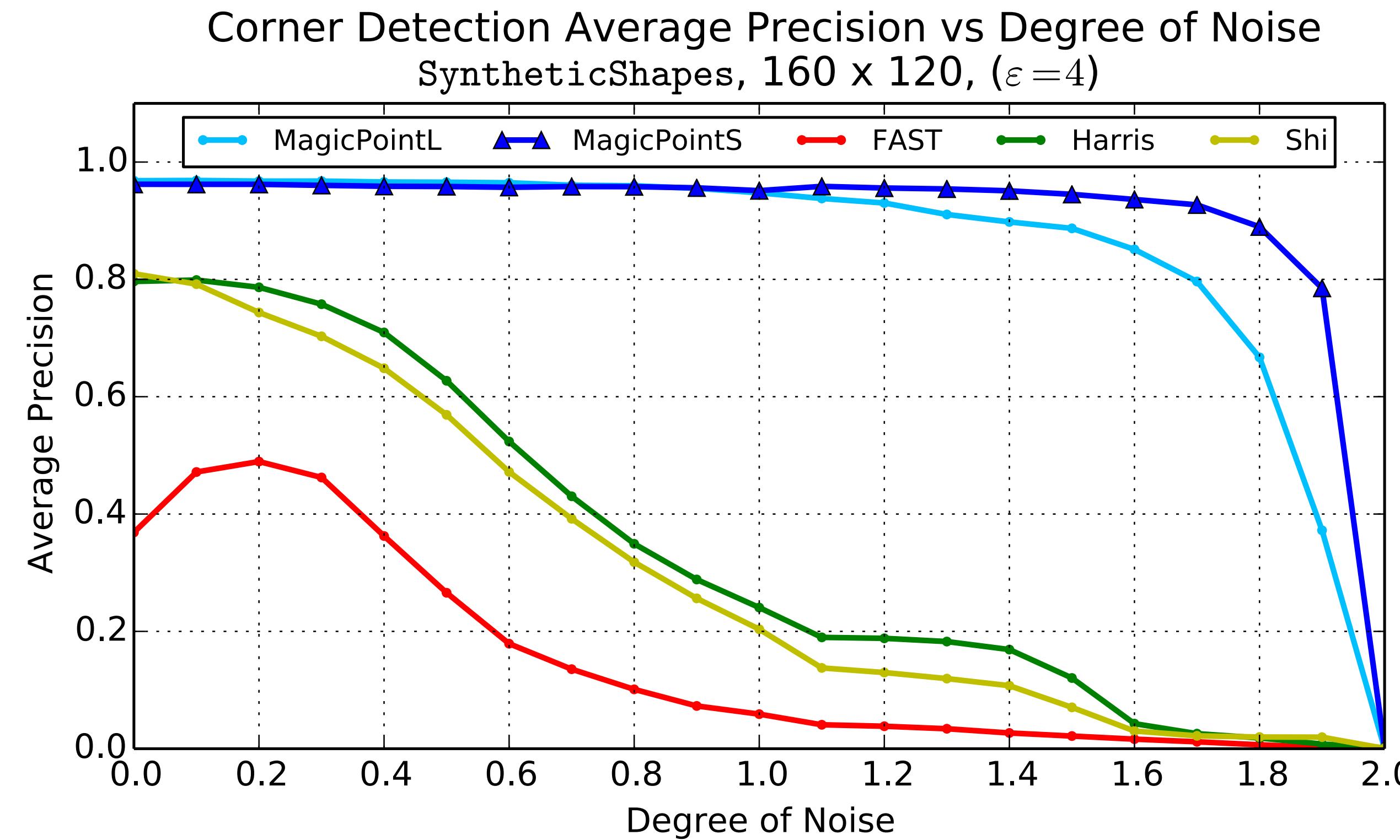


Lines



Stars

# Early Version of SuperPoint (MagicPoint)



Noise Legend

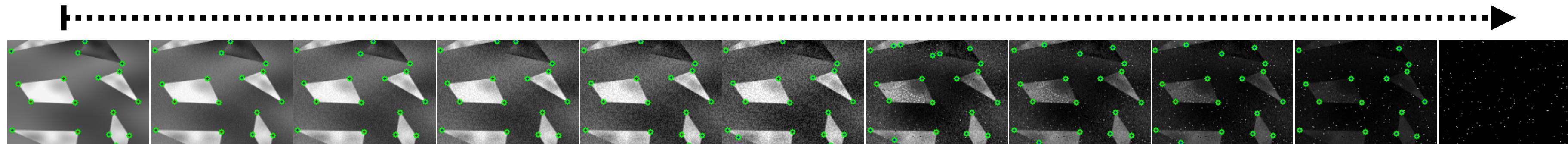


Image  
s=0

Linear Interpolation

Image+Noise<sub>1</sub>  
s=1

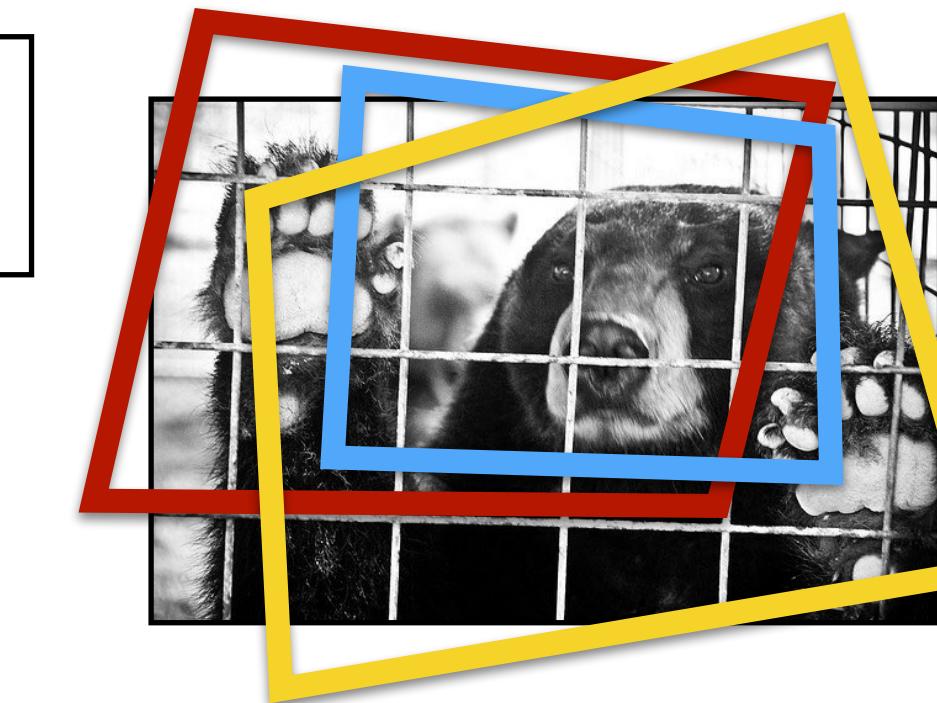
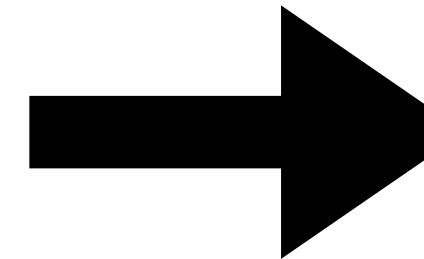
Linear Interpolation

Noise<sub>2</sub>  
s=2

**Unlabeled  
Input  
Image**



**Synthetic Warp +  
Run Detector**

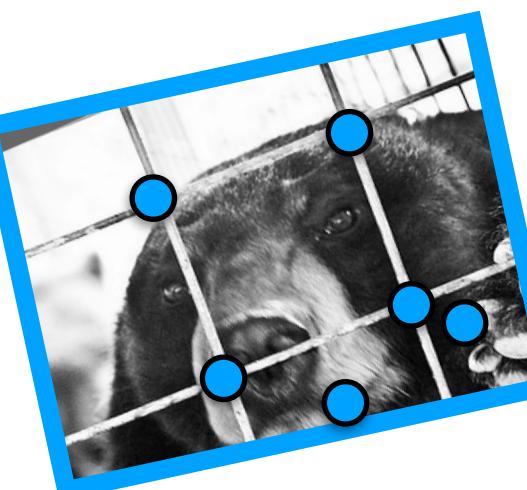


## Homographic Adaptation

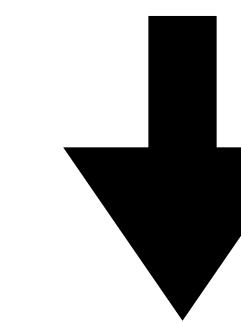
**Point Set #1**



**Point Set #2**

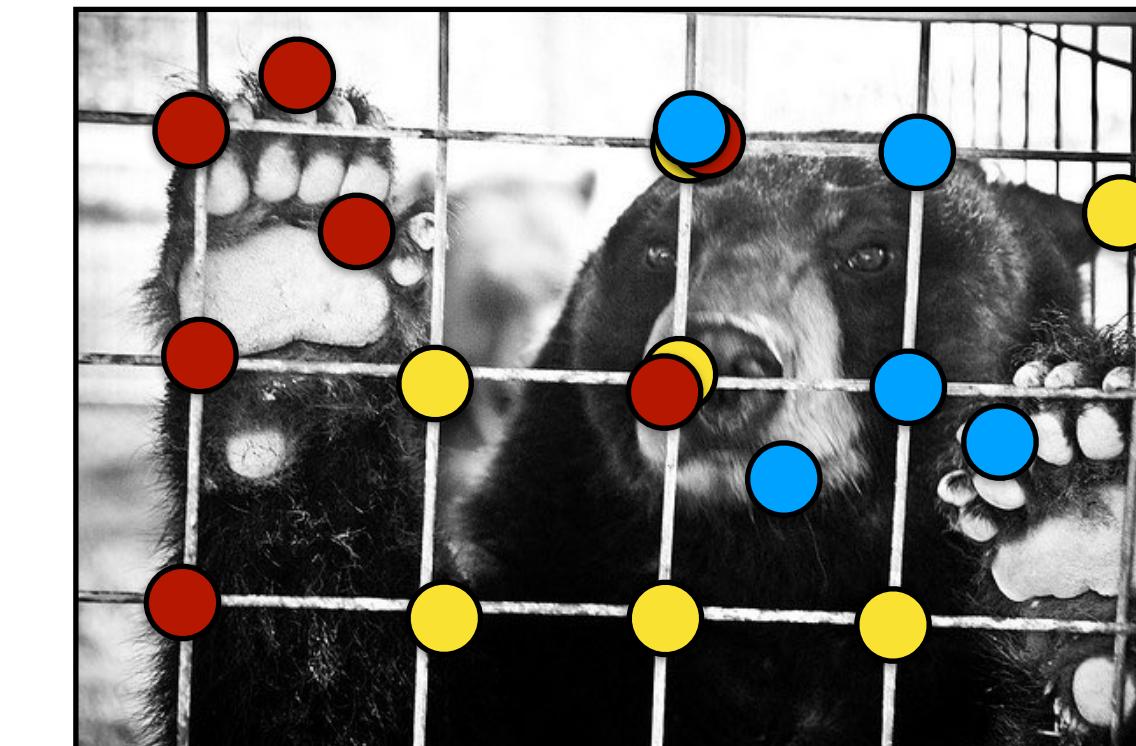


**Point Set #3**



**Point  
Aggregation**

**Detected Point Superset**

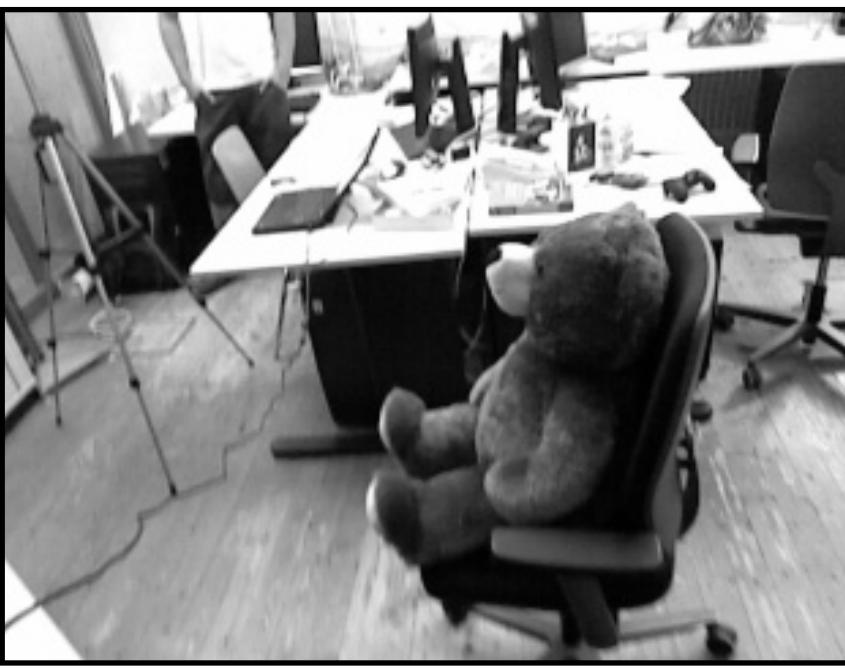


- Simulate planar camera motion with homographies
- Self-labelling technique
  - Suppress spurious detections
  - Enhance repeatable points

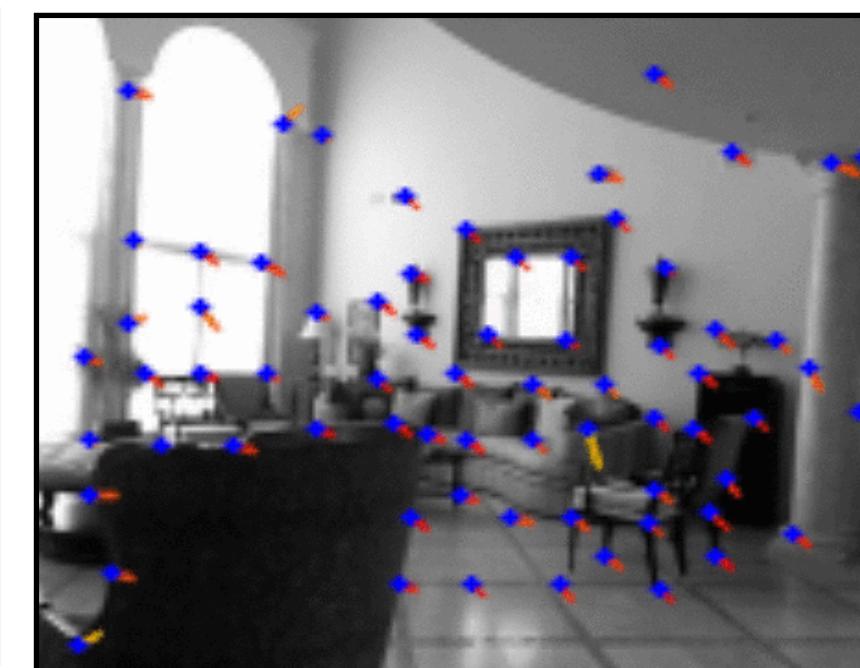
# 3D Generalizability of SuperPoint

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

*Freiburg (Kinect)*



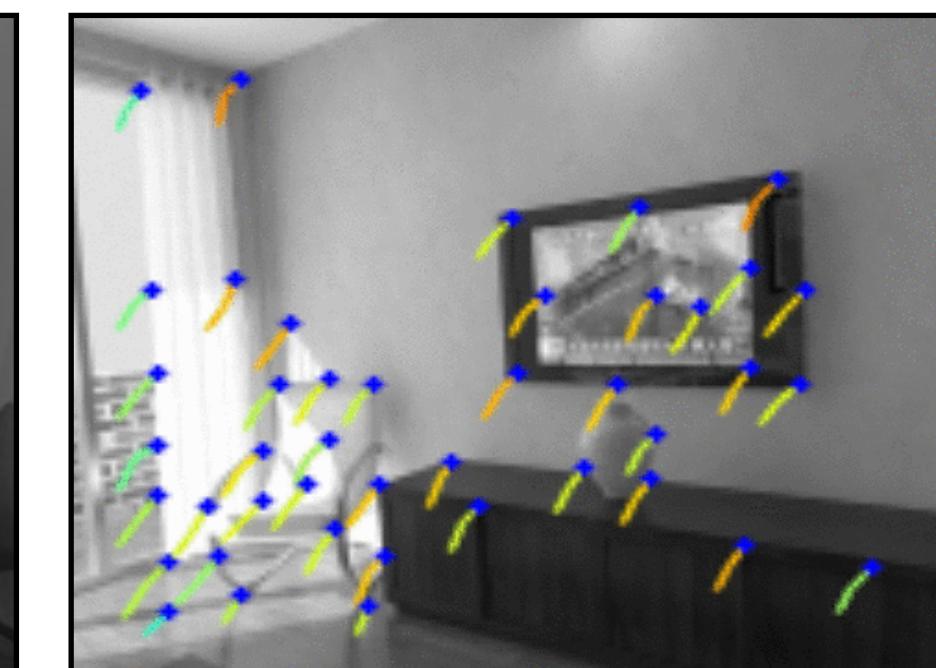
*NYU (Kinect)*



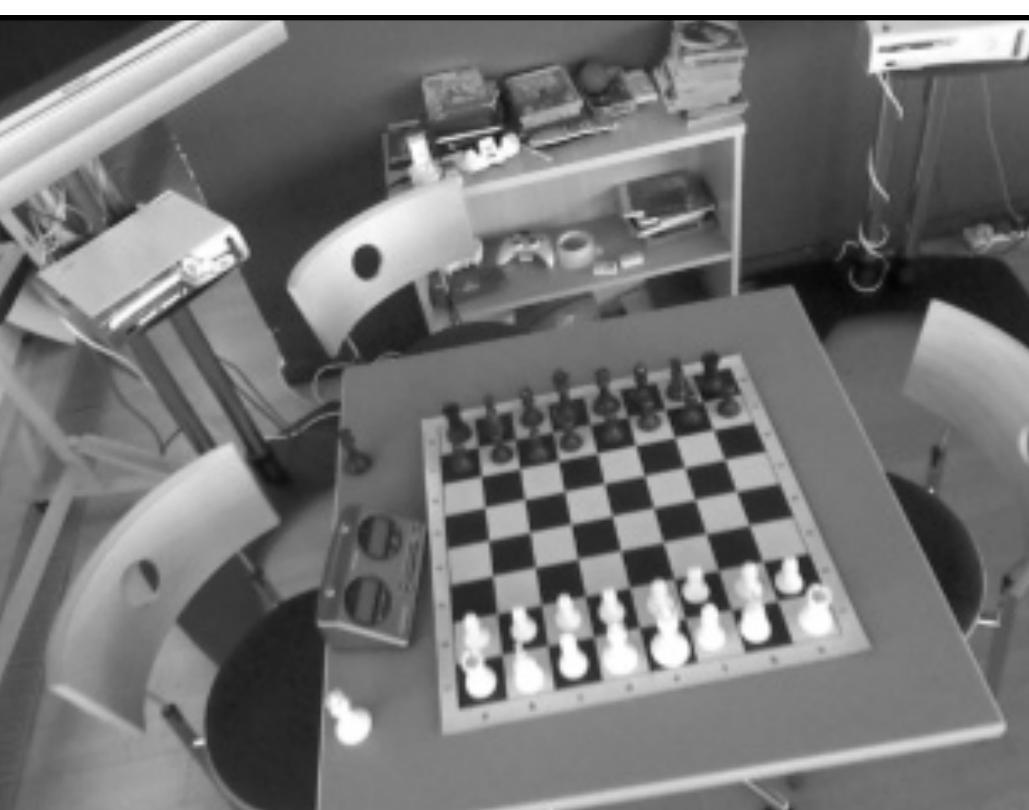
*MonoVO (fisheye)*



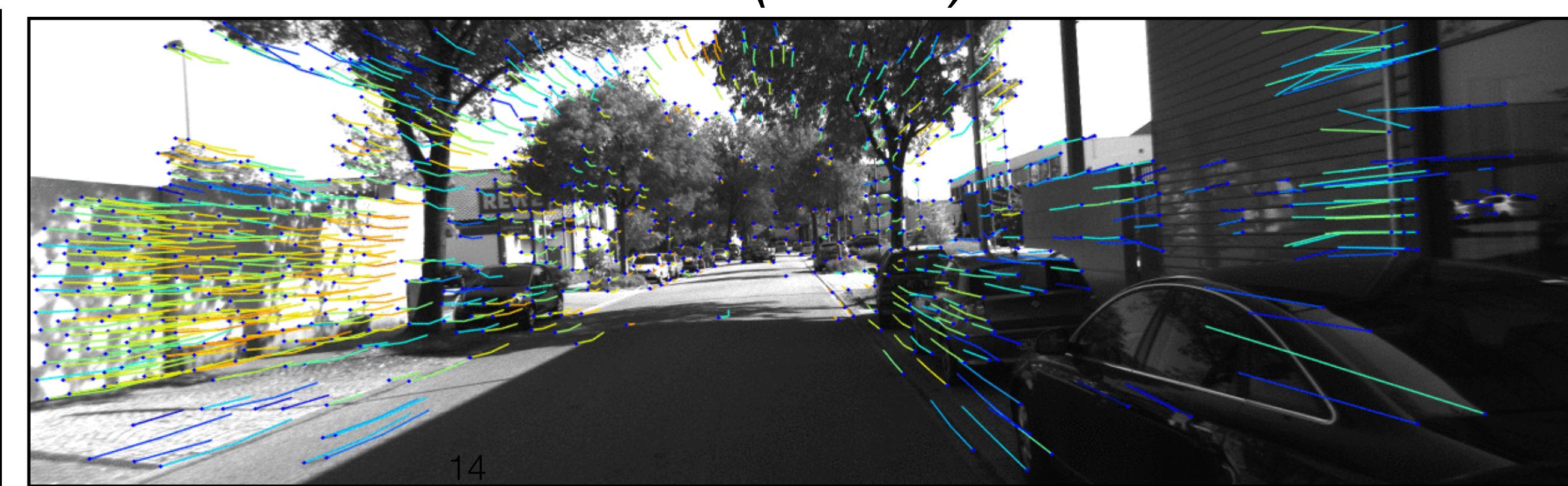
*ICL-NUIM (synth)*



*MS7 (Kinect)*

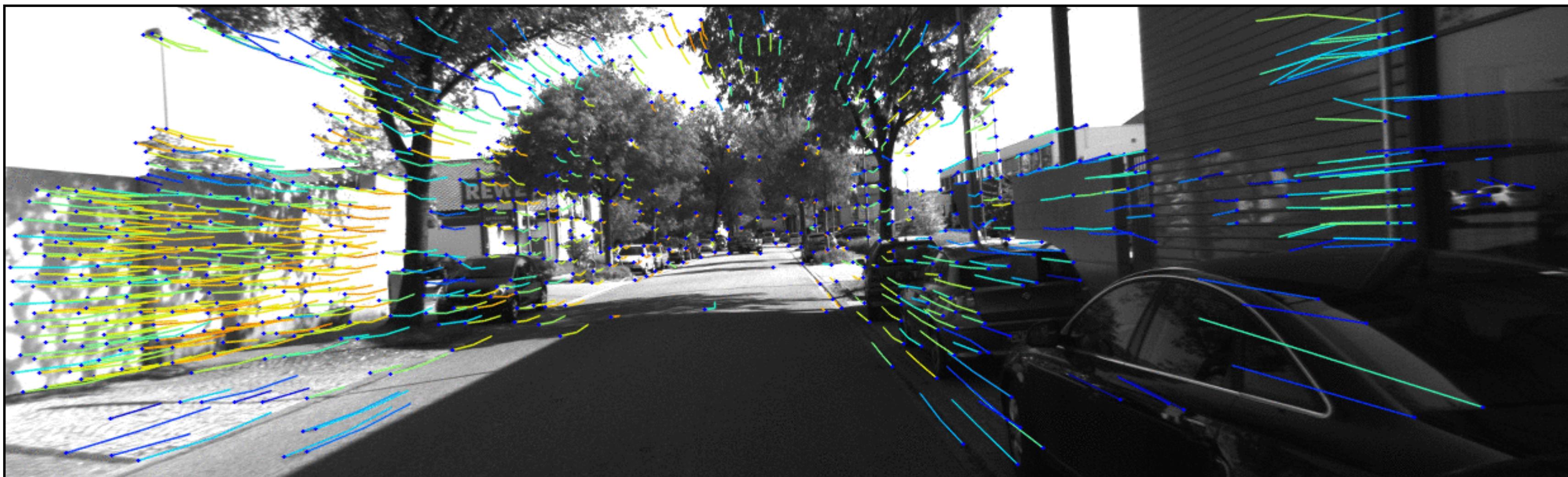


*KITTI (stereo)*



# 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



[github.com/magicleap/SuperPointPretrainedNetwork](https://github.com/magicleap/SuperPointPretrainedNetwork)

# 1. SuperPoint Lessons Learned: what did not work

- Before starting out with SuperPoint, we tried directly estimating relative poses using ConvNets
- That did not work for us!

## **2. SuperPoint Lessons Learned: shifting towards object-detection like philosophy**

- Utilizing all of my experience with object detection (during my PhD) help make a better SuperPoint

### **3. SuperPoint Lessons Learned: using MS-COCO for training**

- Why not use in-house datasets?
- Benefits of using public data?

## 4. SuperPoint Lessons Learned: SyntheticShapes got us off the ground!

- On-the-fly training data generation using simple OpenCV renderer in python
- Help us tame the training recipe

# 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 Sarlin<sup>1</sup>

Tomasz Malisiewicz<sup>2</sup>

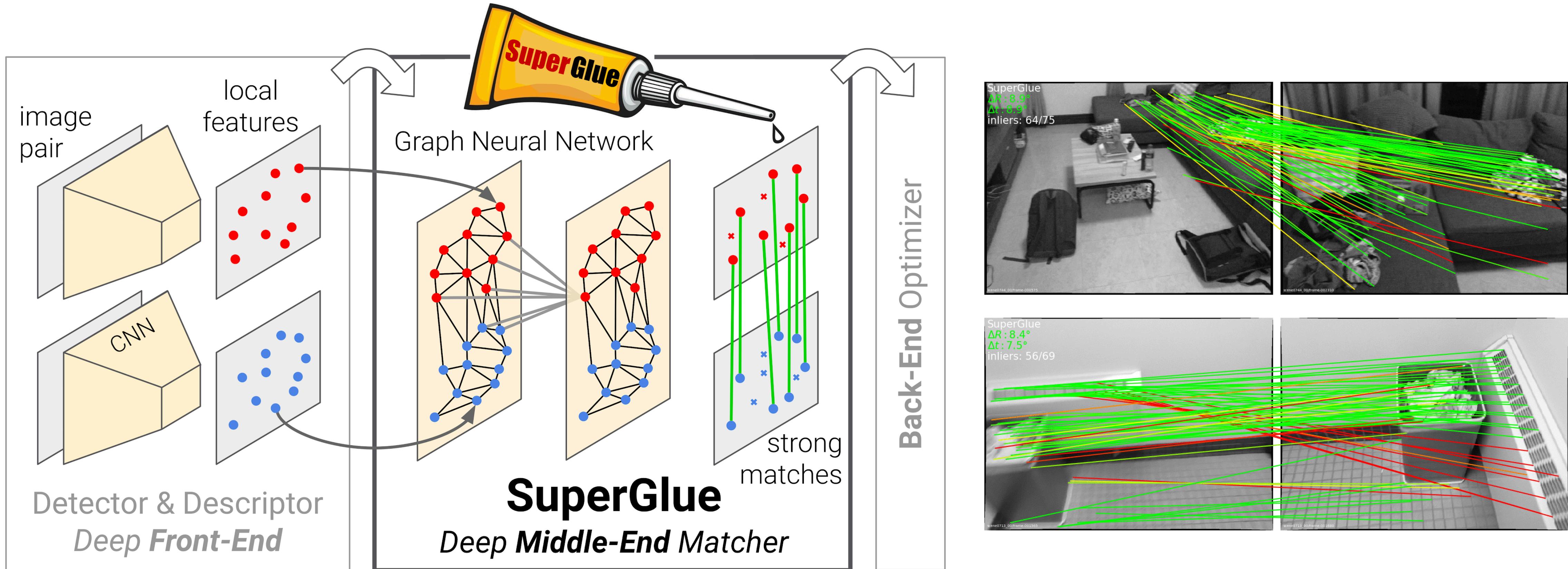
Daniel DeTone<sup>2</sup>

Andrew Rabinovich<sup>2</sup>

**ETH** zürich

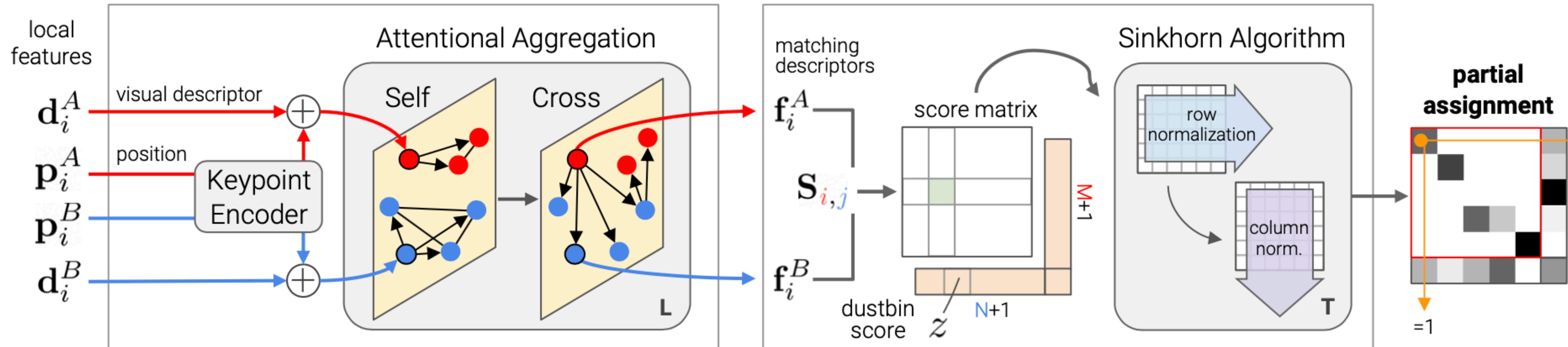


# SuperGlue = Graph Neural Nets + Optimal Transport



- 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!**



## A Graph Neural Network with attention

Encodes **contextual cues** & priors

**Reasons** about the 3D scene

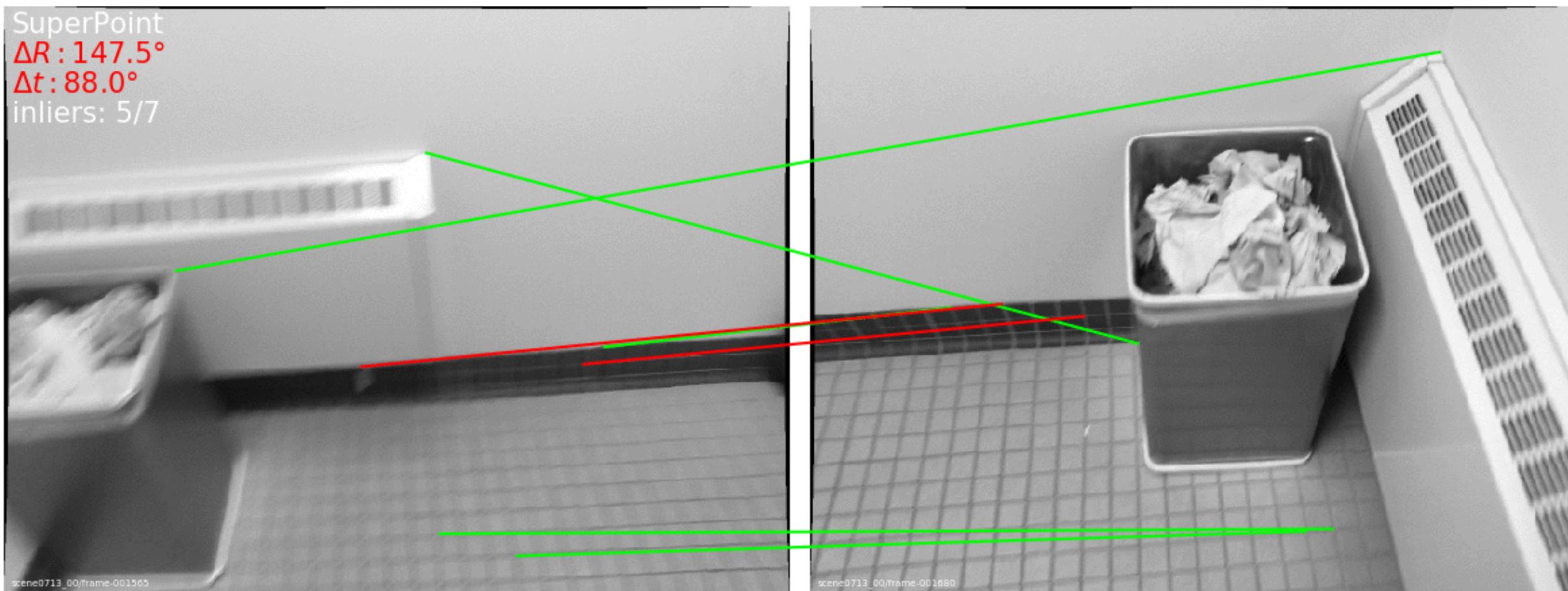
## Solving a partial assignment problem

Differentiable **solver**

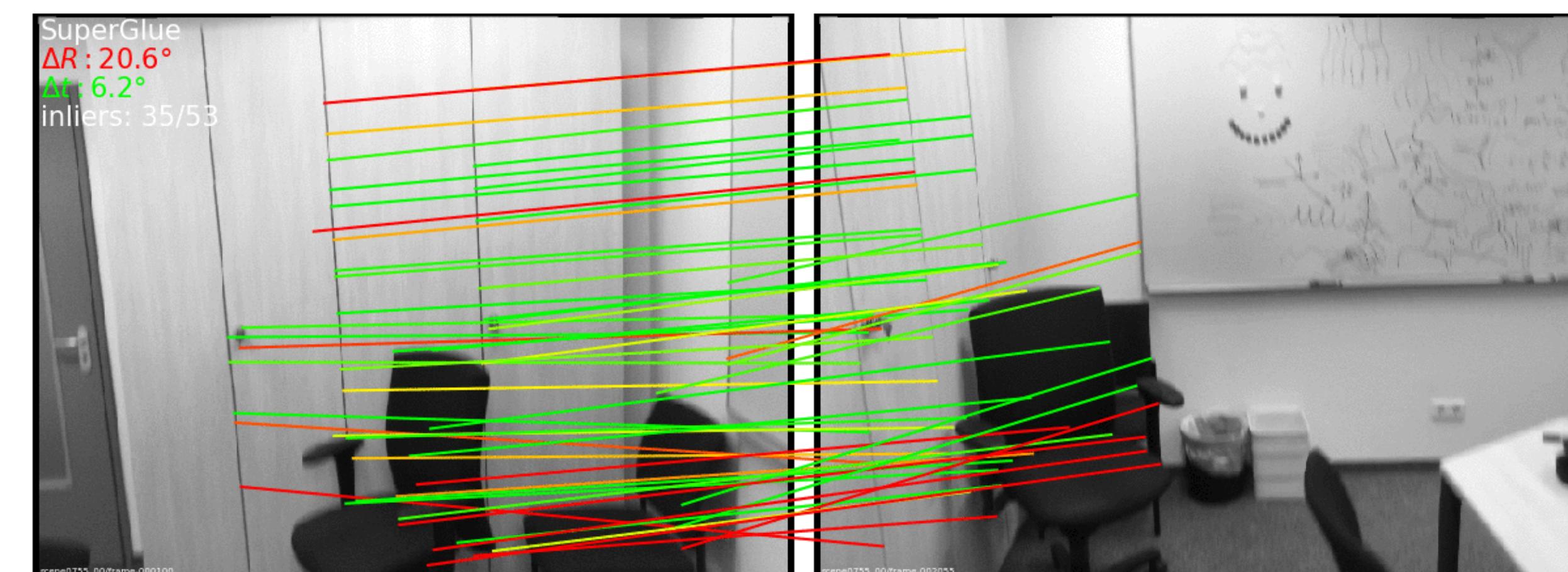
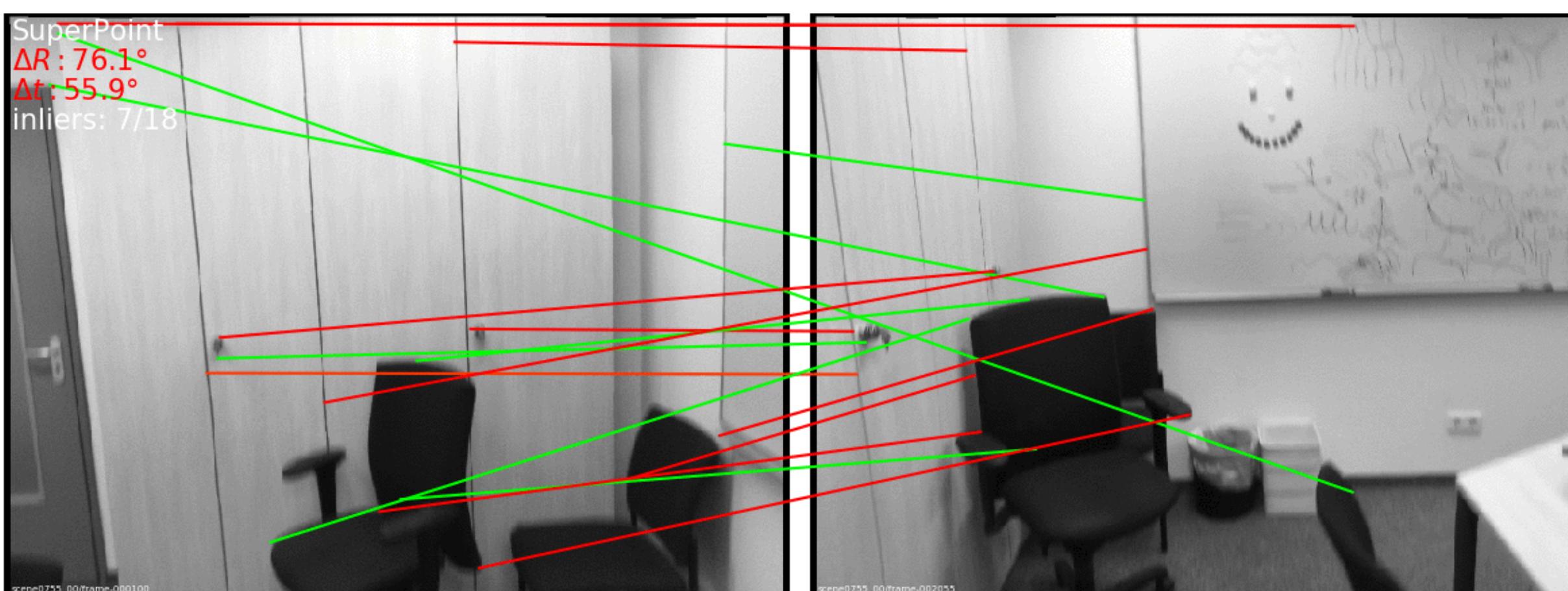
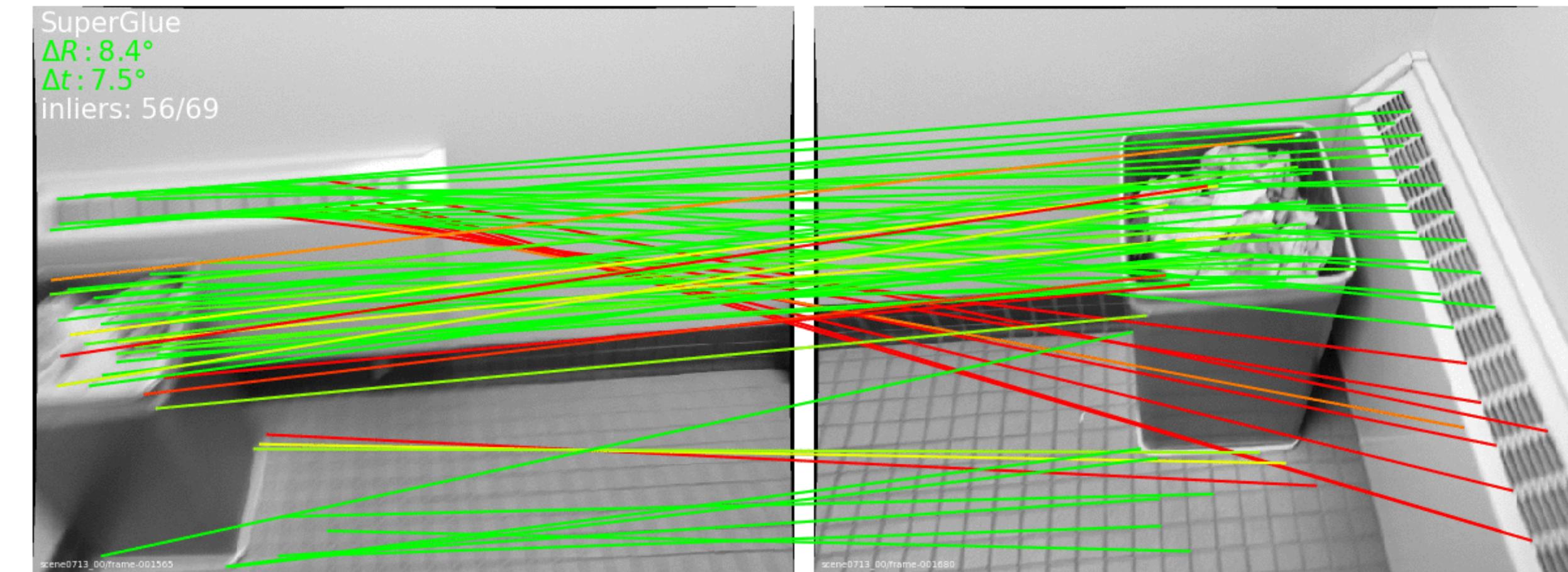
Enforces the assignment constraints  
= **domain knowledge**

**SuperGlue requires both sets of local features:  
a paradigm shift in matching!**

## SuperPoint + NN + heuristics



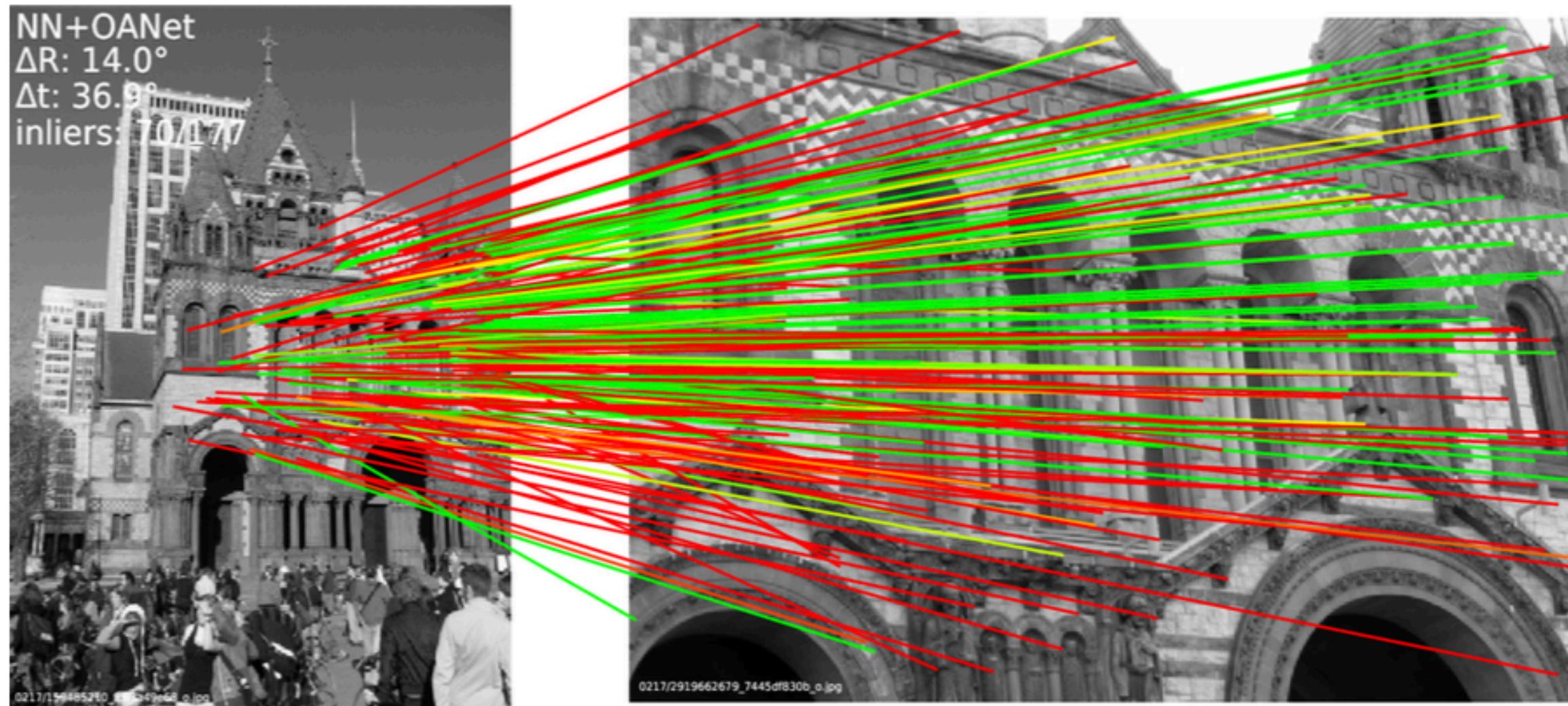
## SuperPoint + SuperGlue



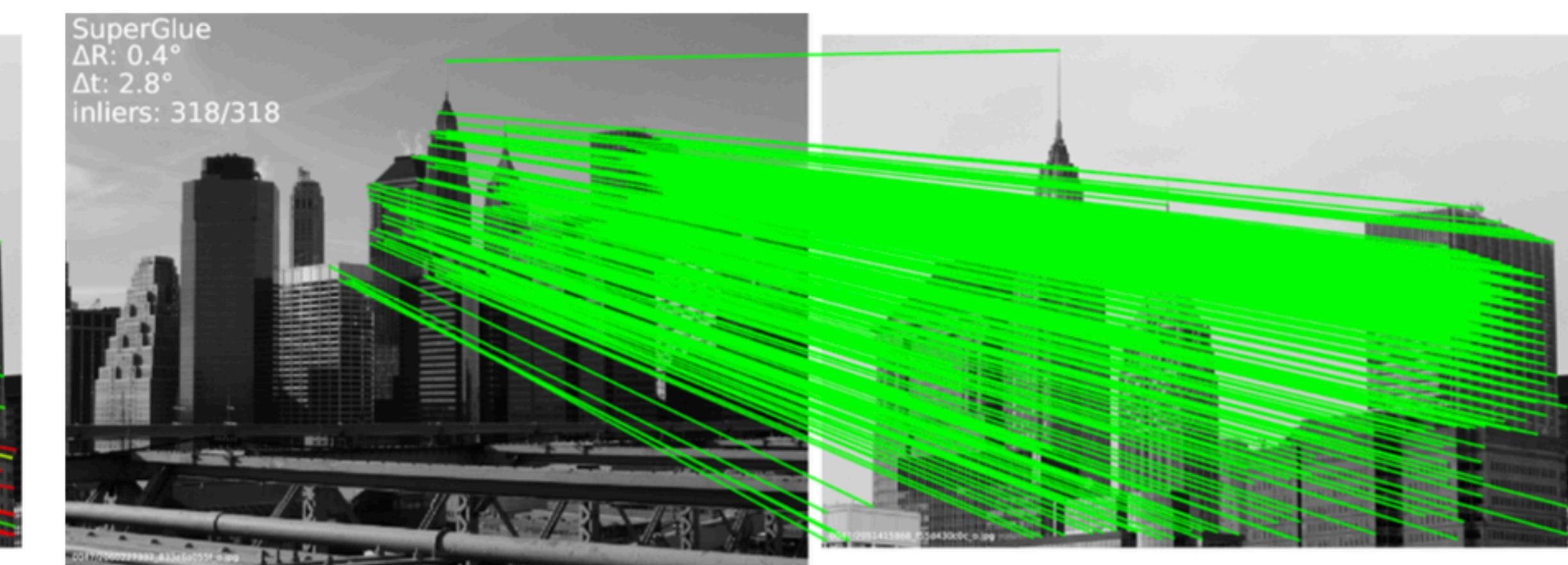
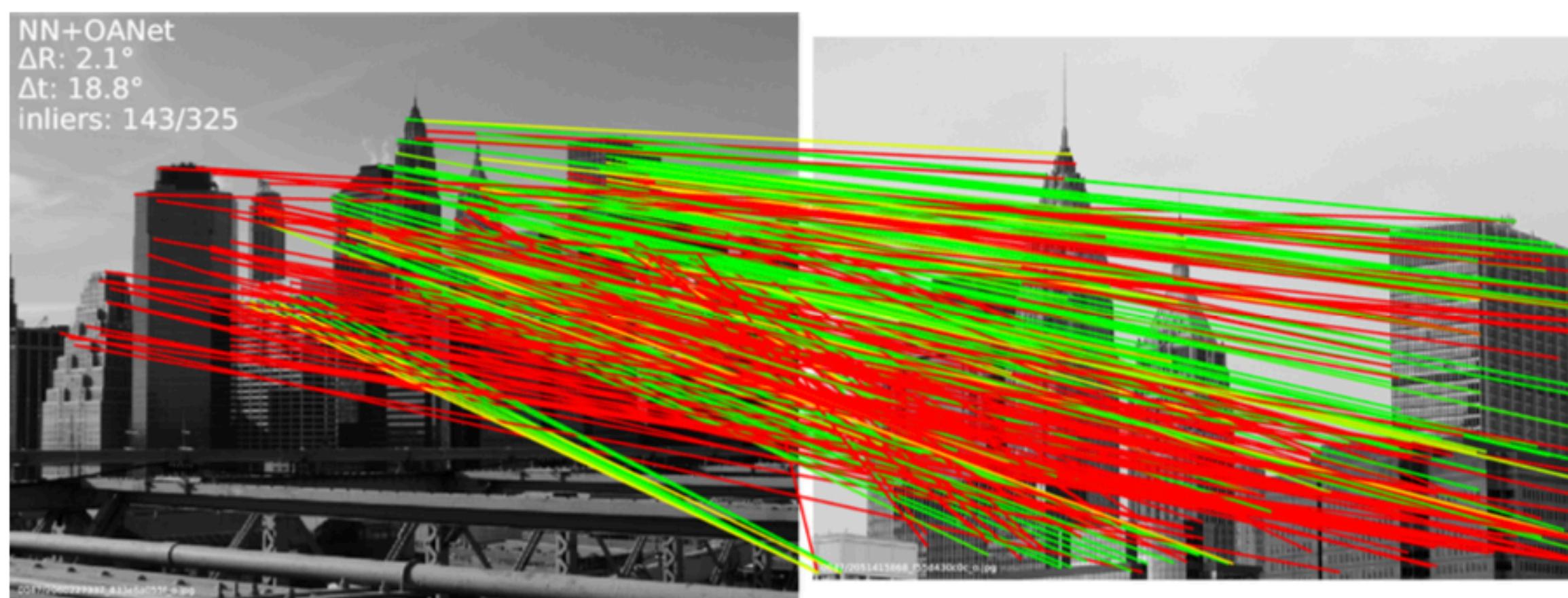
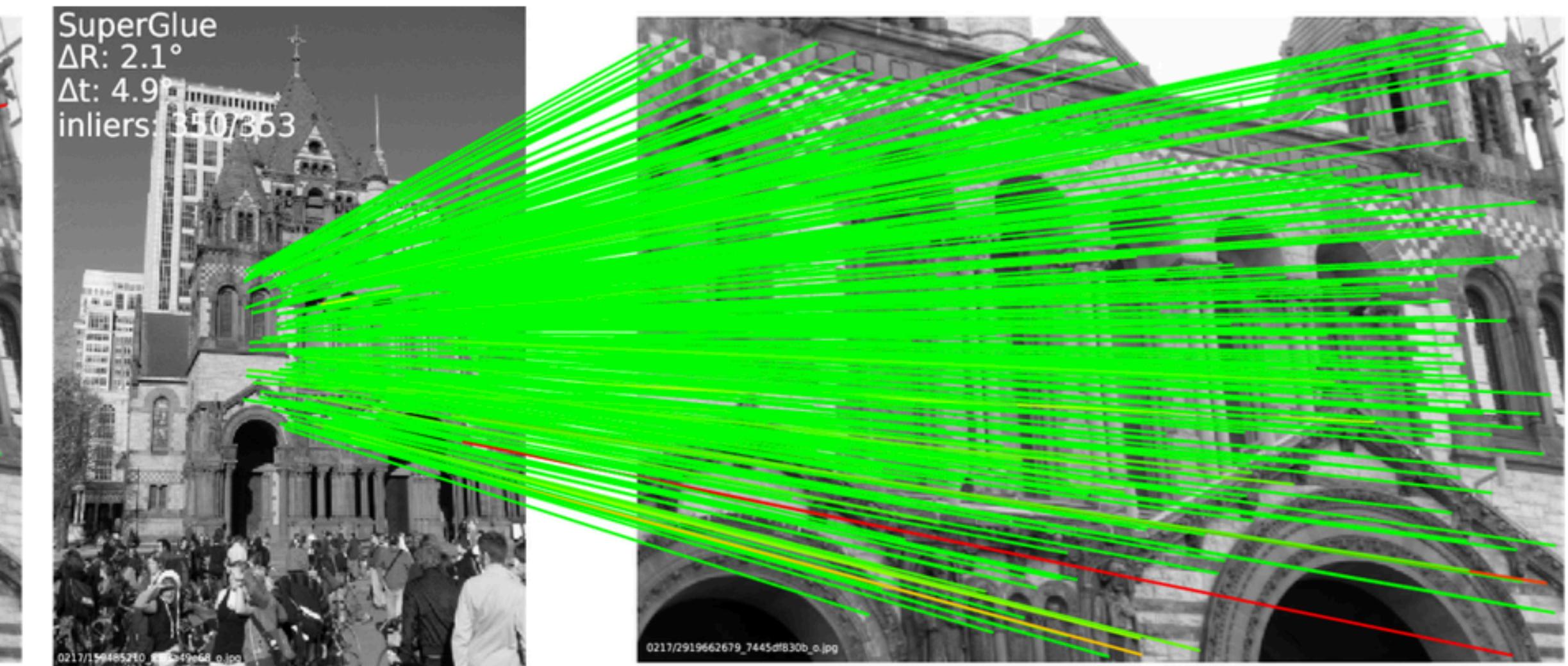
SuperGlue: more **correct matches** and fewer **mismatches**

# Results: outdoor - SfM

SuperPoint + NN + OA-Net (inlier classifier)

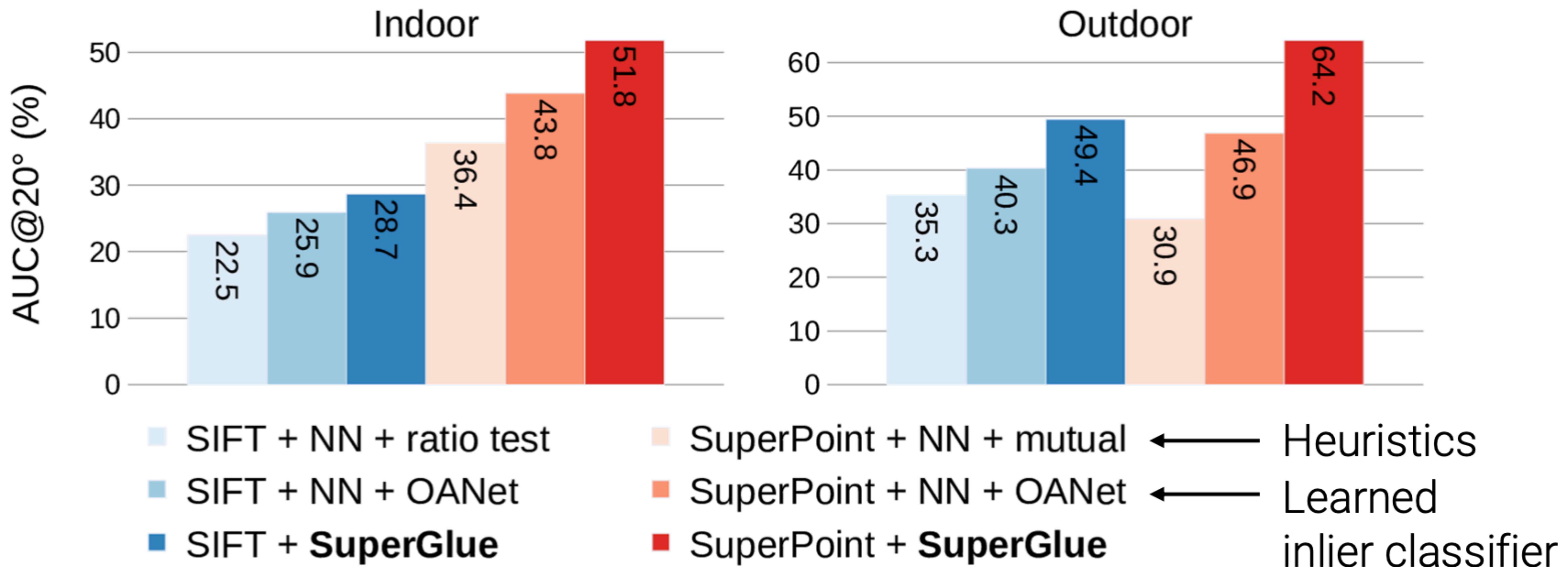


SuperPoint + **SuperGlue**



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

# Evaluation

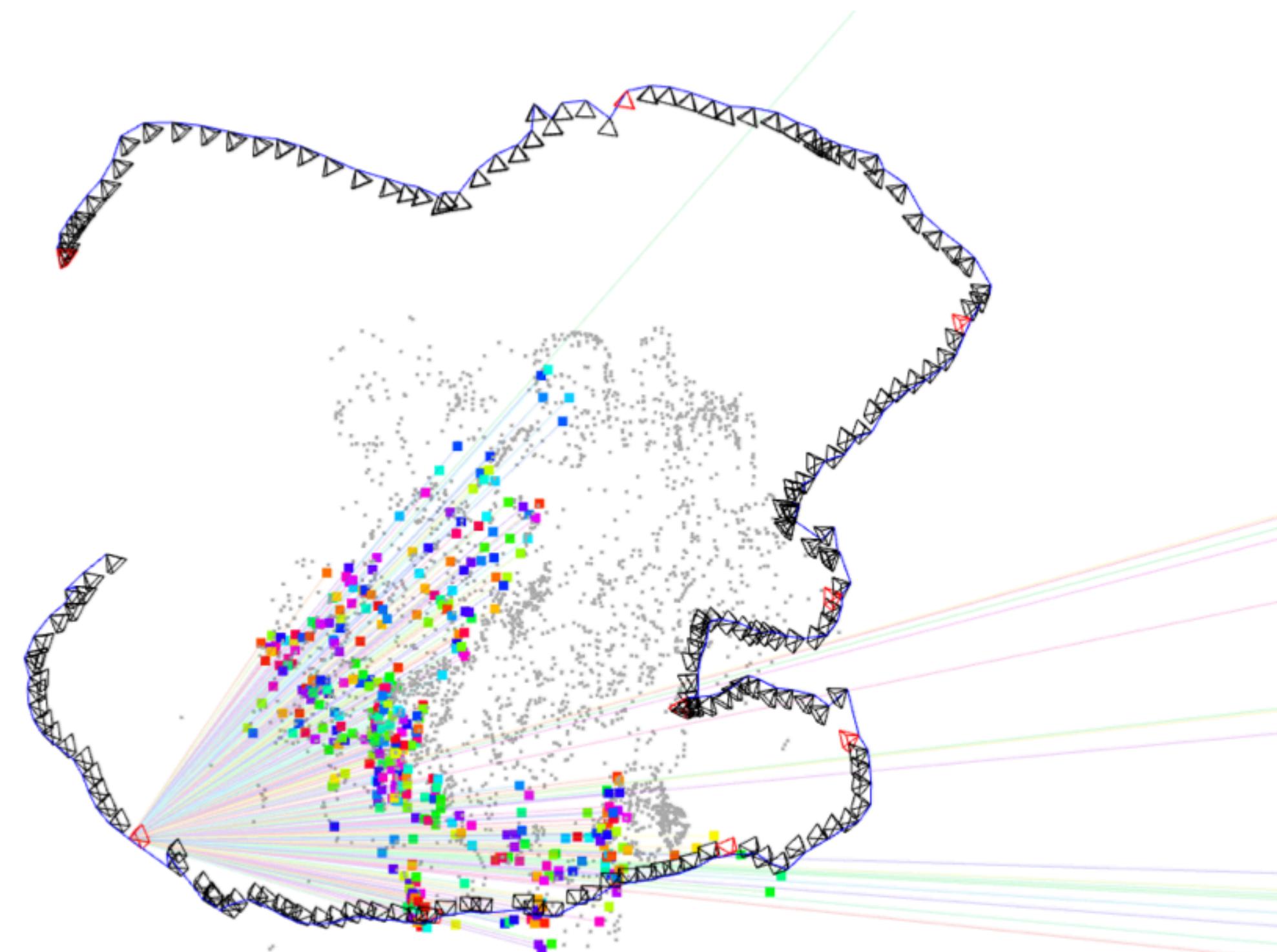
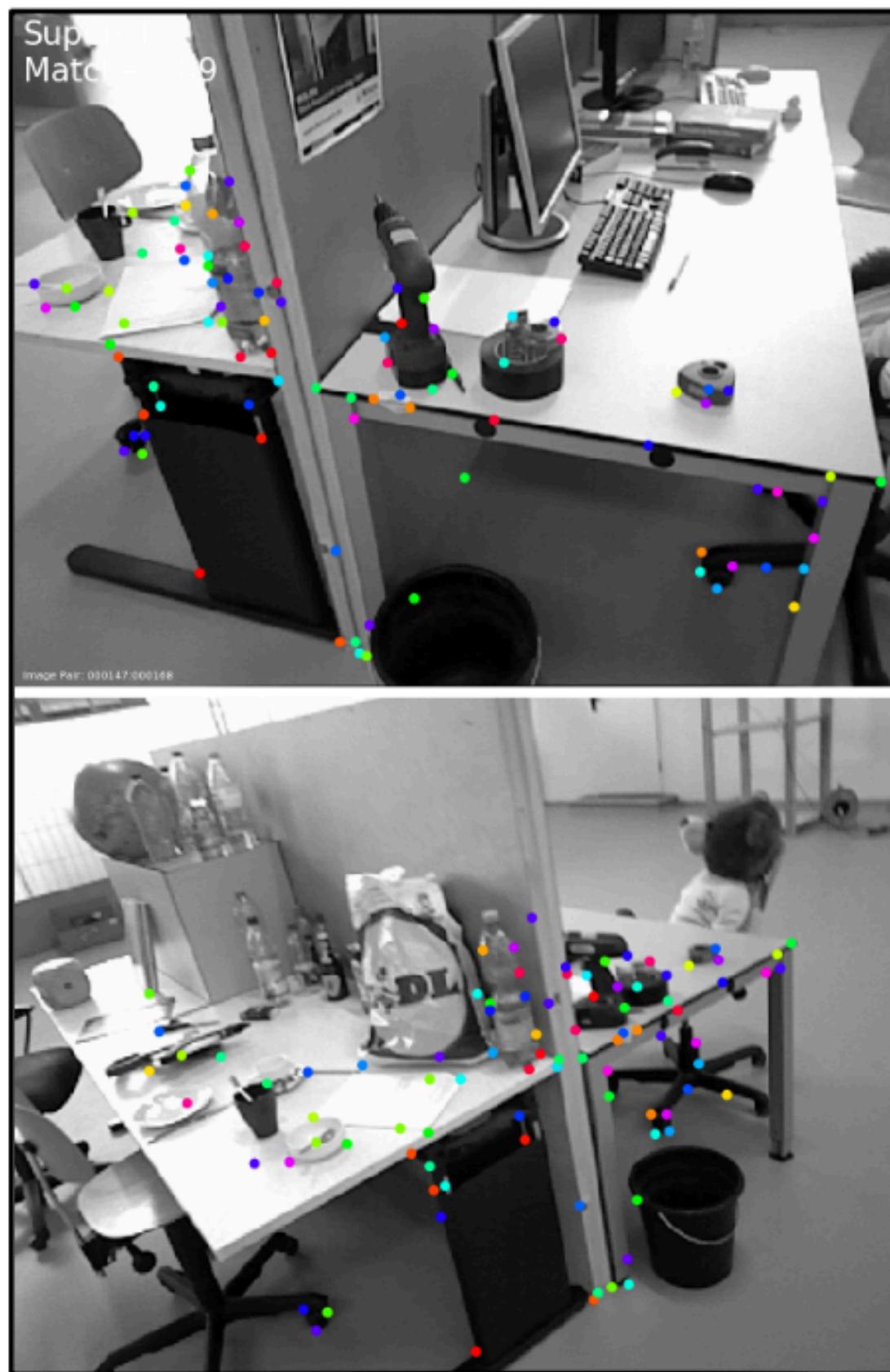


SuperGlue yields **large improvements** in all cases



Demo: 15 FPS for 512 keypoints on GPU

[psarlin.com/superglue](http://psarlin.com/superglue)

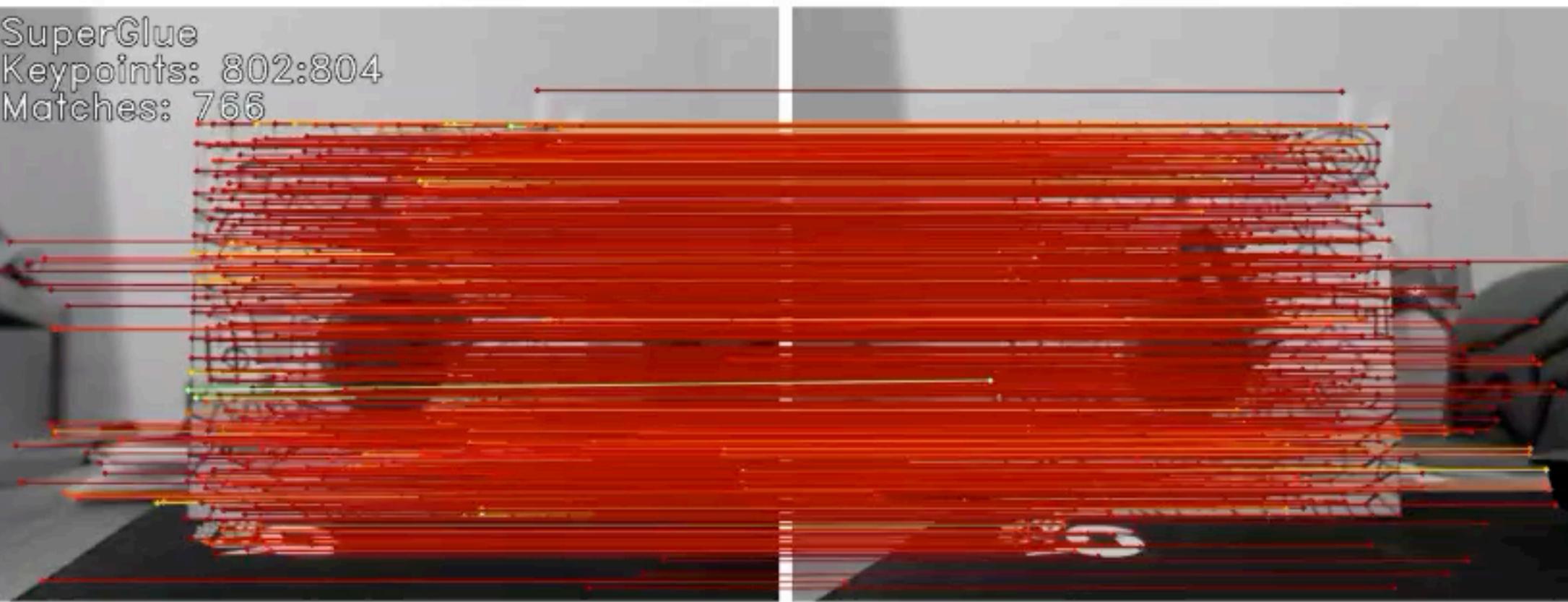


[github.com/magicleap/SuperGluePretrainedNetwork](https://github.com/magicleap/SuperGluePretrainedNetwork)

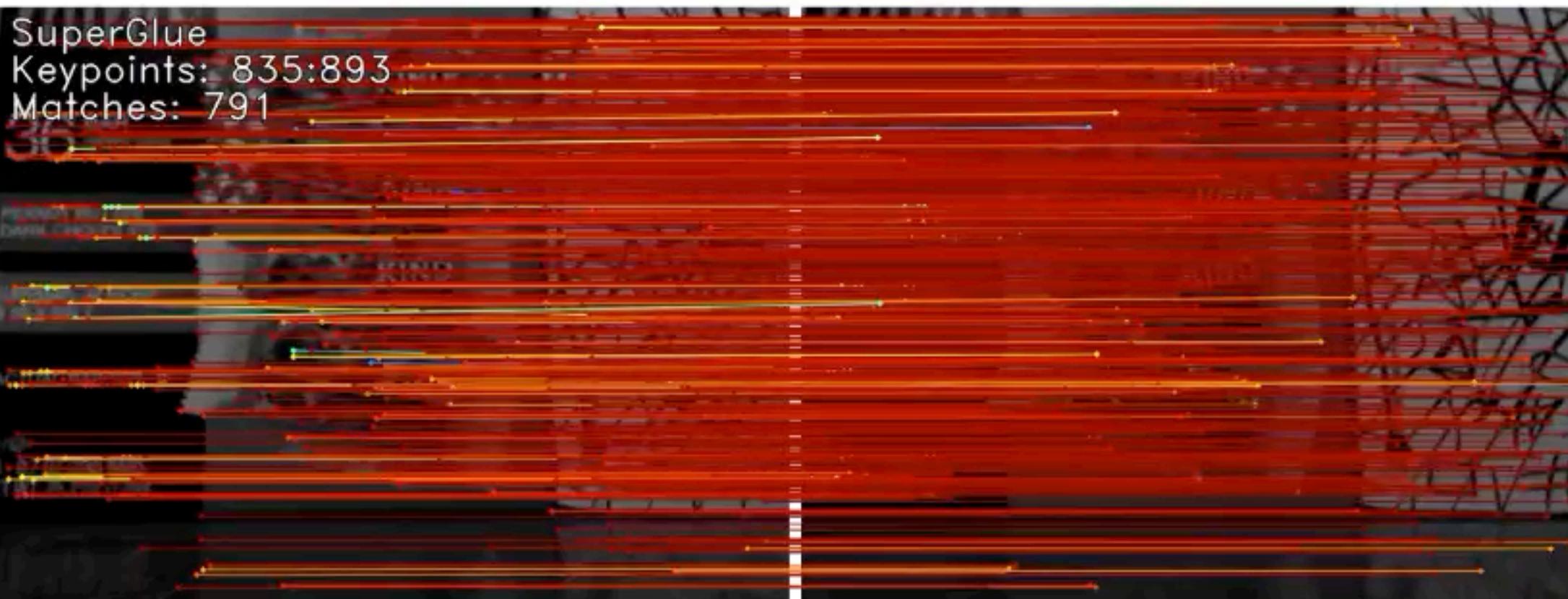
SuperGlue  
Keypoints: 975:969  
Matches: 934



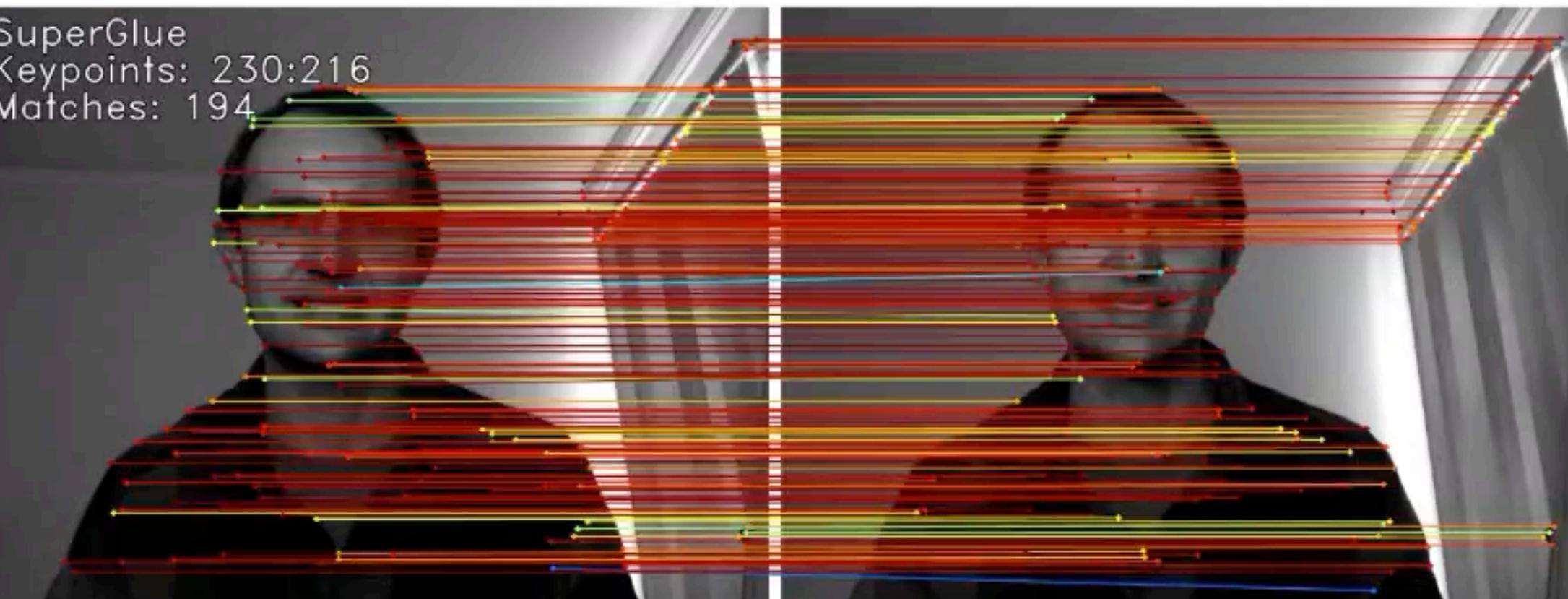
SuperGlue  
Keypoints: 802:804  
Matches: 766



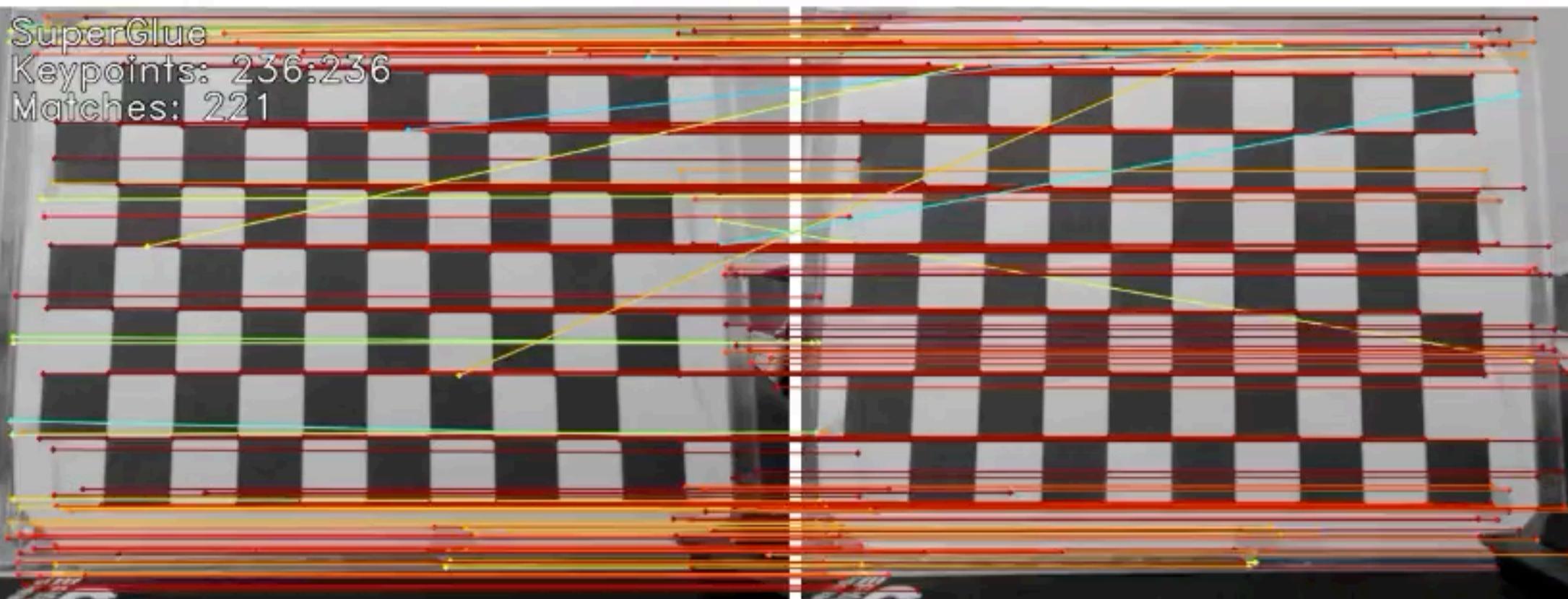
SuperGlue  
Keypoints: 835:893  
Matches: 791



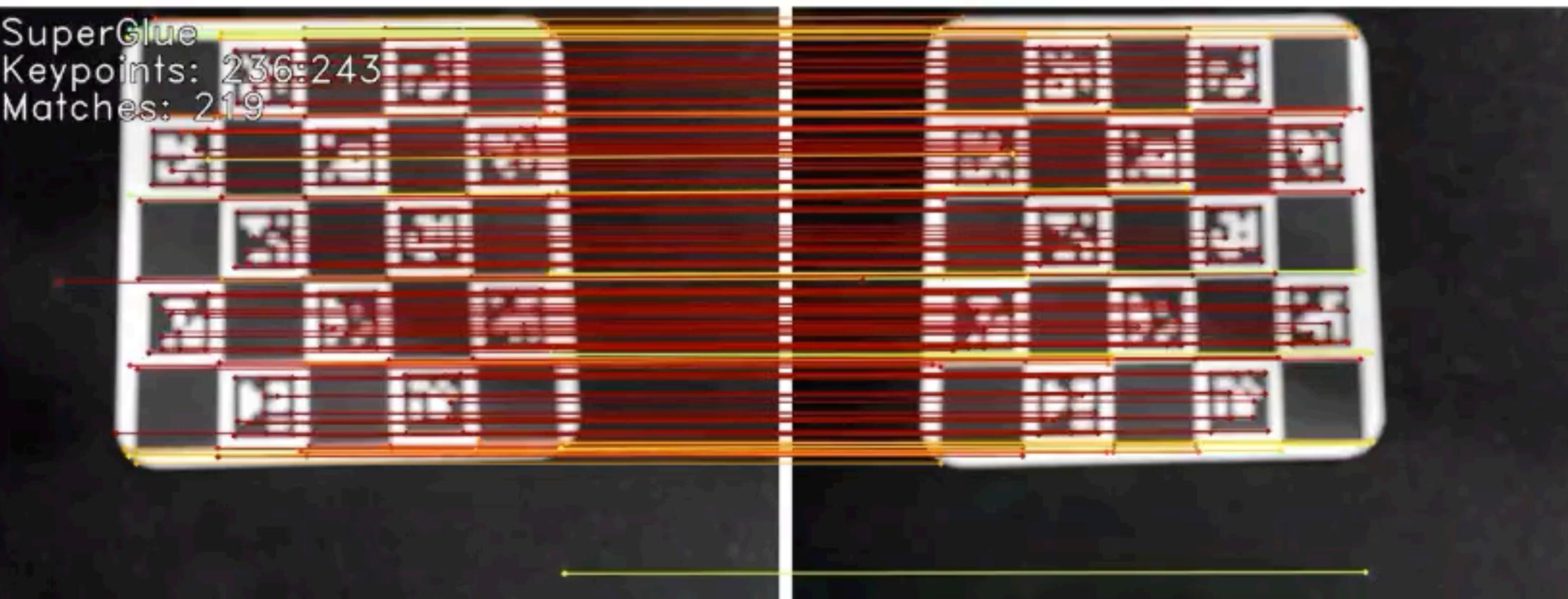
SuperGlue  
Keypoints: 230:216  
Matches: 194



SuperGlue  
Keypoints: 236:236  
Matches: 221



SuperGlue  
Keypoints: 236:243  
Matches: 219



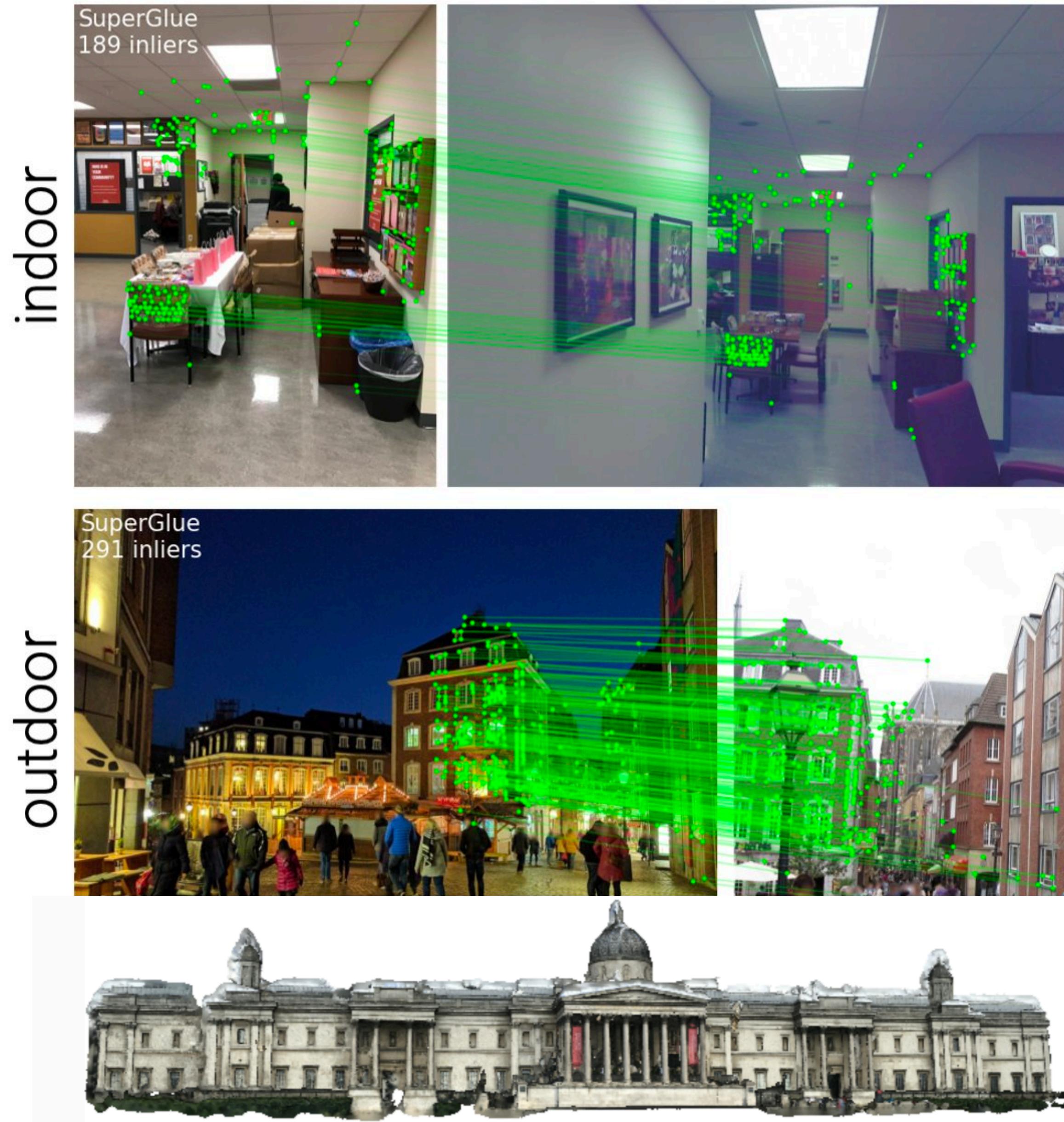


Image Matching: Local Features & Beyond

CVPR Workshop: Friday, June 19, 2020

# SuperGlue

## Learning Feature Matching with Graph Neural Networks

CVPR 2020 Oral

1<sup>st</sup> 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)

# 1. SuperGlue Lessons Learned: Experienced Candidate Key to Internship Success

- Paul-Edouard Sarlin had the key background before starting the 6+ month internship
- Internship had to get extended to get awesome paper out the door

## 2. SuperGlue Lessons Learned: Moving away from practical systems

- We decided to move away from the precomputed descriptors paradigm
- Input to network is 2 images — not ideal for real systems
- We pivoted towards working on great science

# Part III: Meta Lessons Learned

*What did we learn? What  
can I teach you?*

# 1. Re-invent yourself

- Every few years, you will have to re-invent yourself as a researcher, especially during the decade after your Ph.D.

## 2. Help create careers

- Your post-PhD impact will influence the young researchers you work alongside. Some will continue to pursue a Ph.D., and some will get high-tech jobs. You will feel proud of “your students” just like you were proud of your “first papers.”

### 3. The more you publish, the more people know of your work

- It is not always easy to publish papers while in industry, but it is a worthwhile endeavor.
- By giving talks, you will meet future collaborators, future employees, future employers, etc.
- Tip for Postdocs: **every talk you give is a job talk!**

# Thank you

**Tomasz Malisiewicz**  
<https://tom.ai/>



@quantombone

**Daniel DeTone**  
<https://danieldetone.com/>



@ddetone

**Paul-Edouard Sarlin**  
<https://psarlin.com/>



@pesarlin

Follow us on Twitter:



Research Questions:

