

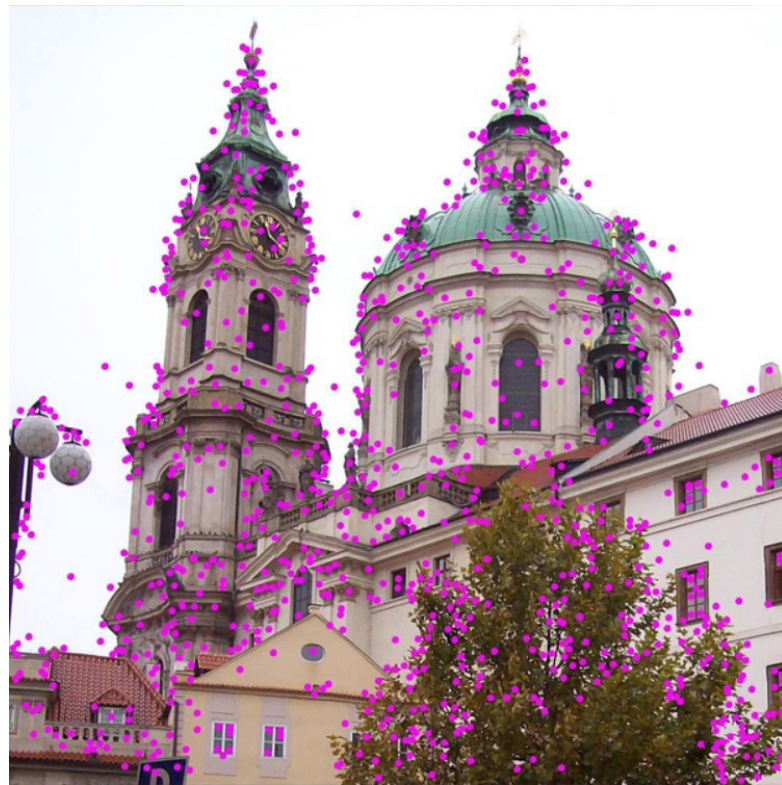
# Deep Learning in Feature Detection and Matching

Yubo Mao  
mao00071@morris.umn.edu

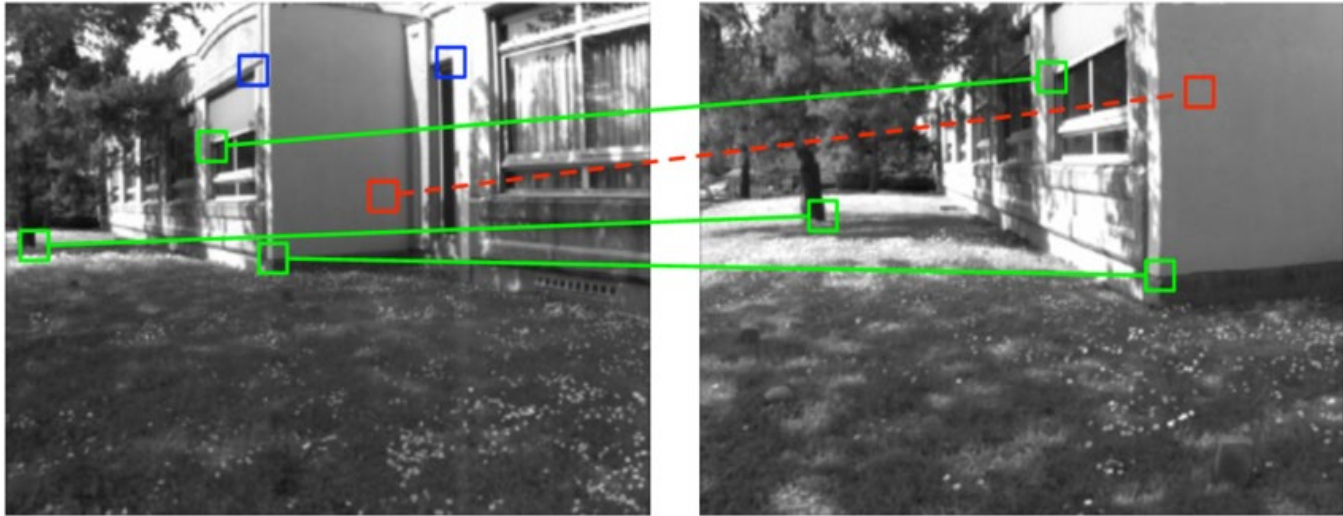


# Feature Detection

- **Feature:** keypoint + descriptor
- **Keypoint:** location of a feature in image
- **Descriptor:** description/appearance of a feature.



# Feature Matching

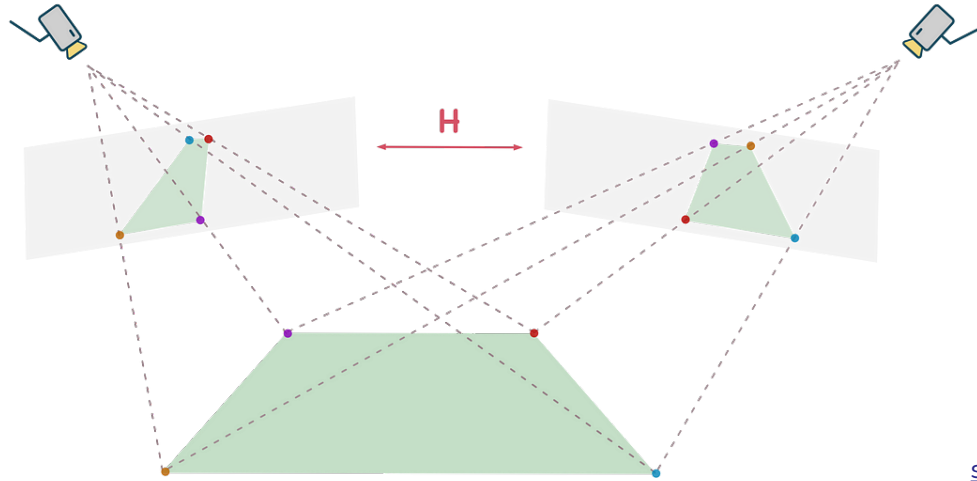


## Feature detection and matching

The connecting line between two images represents the matching relationship between features.

**Green:** correct match. **Red:** incorrect match. **Blue:** no corresponding feature to match.

# Homography Estimation



[source](#)

A homography describes the transformation of a plane to another plane when viewed from different positions.

# Homography Estimation

## Estimator:

- RANSAC(Random Sample Consensus) [robustness]
- DLT(Direct Linear Transform) [no robustness]

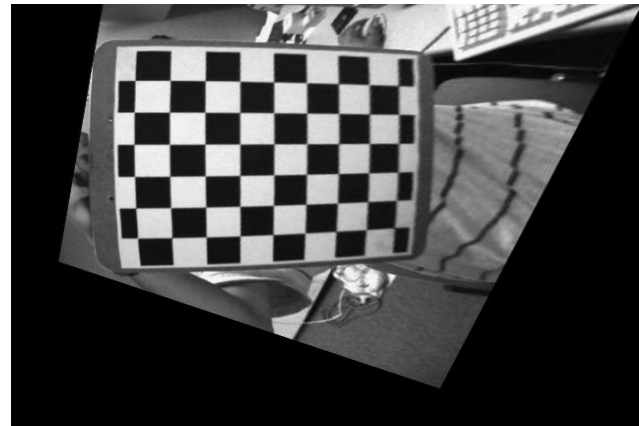
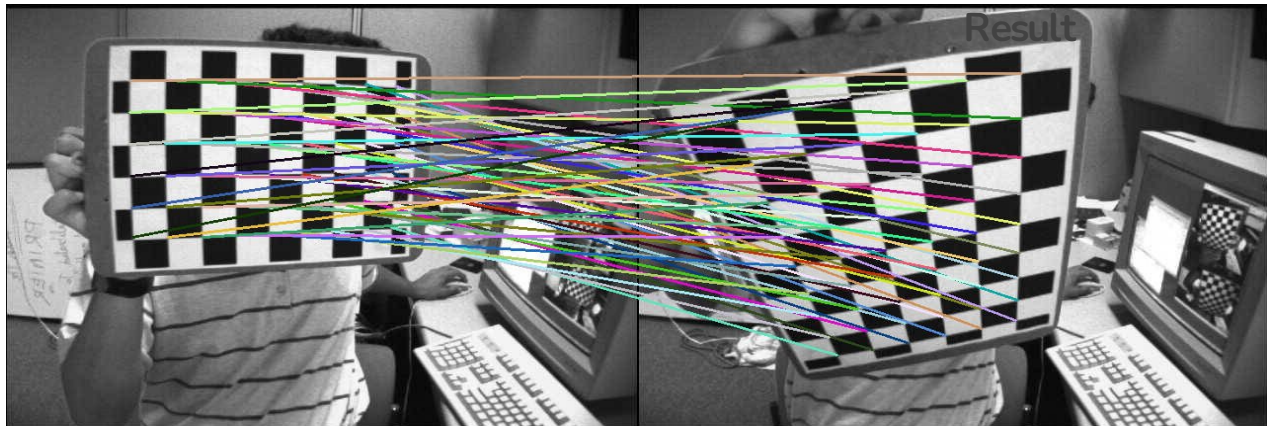
## Application

- Align images
- Correct for perspective distortions
- Perform image stitching

Desired view

Source

[source](#)



# Scale Invariant Feature Transform(SIFT)

- Developed by David Lowe in 1999
- Traditional, hand-crafted feature extraction algorithm.
- **Keypoint:** Difference of Gaussian(DOG)
- **Descriptor:** gradient information in keypoint's surrounding pixel

Potential keypoints

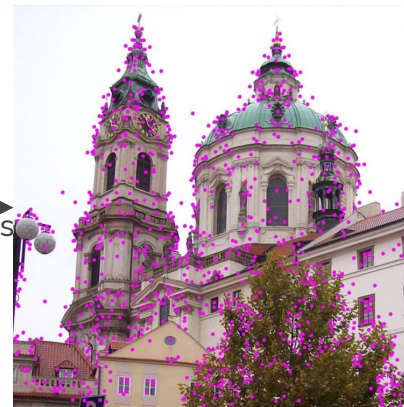


discards  
low-contrast

Established keypoints

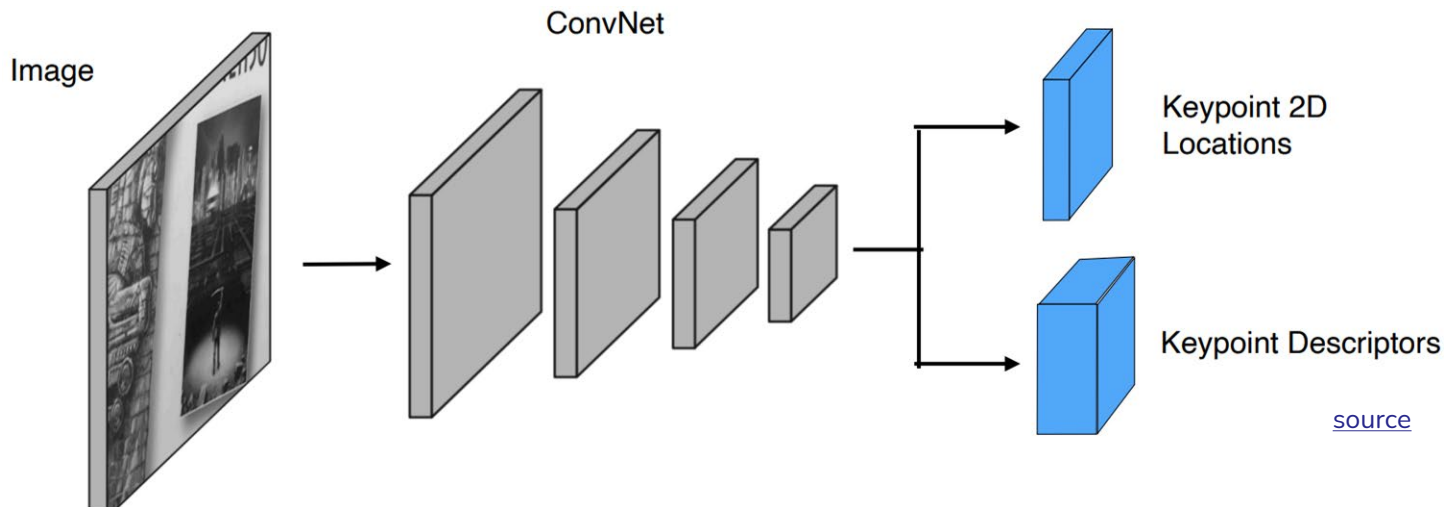


Eliminate  
edge responses



# SuperPoint

- A deep learning-based feature extraction method. (deep learning version of SIFT)
- Fully Convolutional Neural Networks(FCN)
- Self-supervised training mechanism.



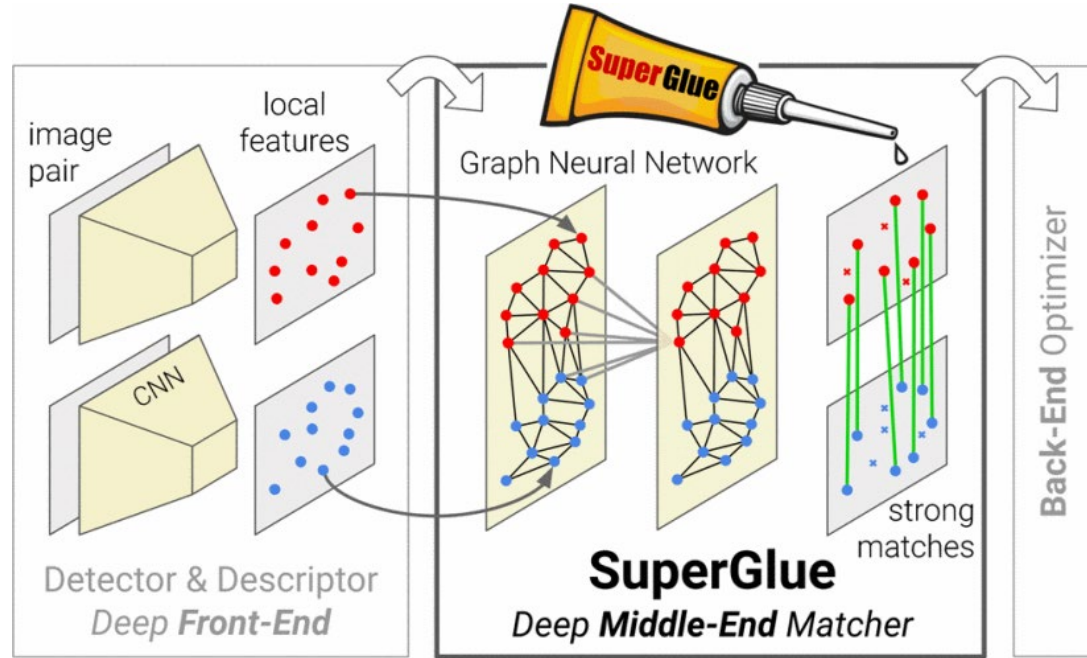
# SuperGlue

## Superglue

- A neural network that matches two sets of local features (SIFT, Superpoint)
- Graph Neural Networks (GNN)
- Using both keypoint and descriptor to match

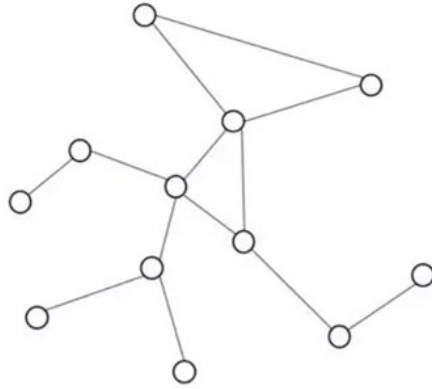
### NN(nearest neighbor)

- Traditional handcrafted matching method
- only use descriptor to match.

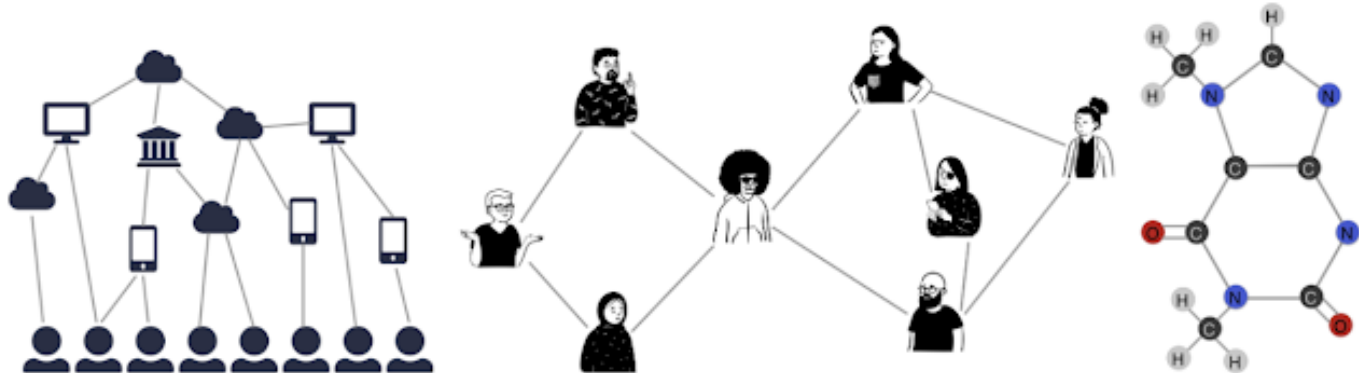




# Graph Neural Network (GNN)



- A graph is composed with entities (nodes) and their interactions (edges).
- Each node has at least one connection with any other nodes.



# Graph Neural Network (GNN)

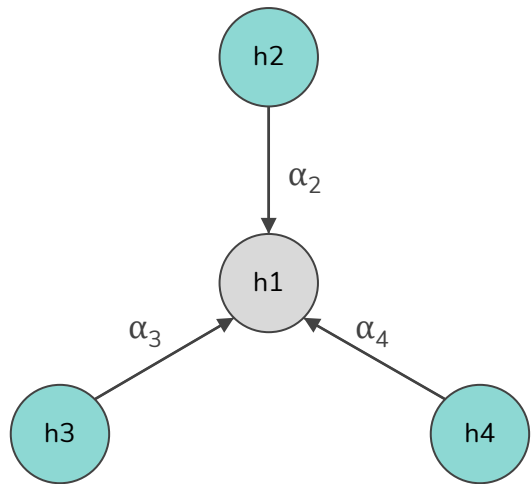
## Message Passing:

1. Sending message ( $h_2, h_3, h_4 \rightarrow h_1$ )
2. Message aggregation ( $m = h_2 + h_3 + h_4$ )
3. Update self information ( $h_1 = h_1 + m$ )

## Attention Mechanism:

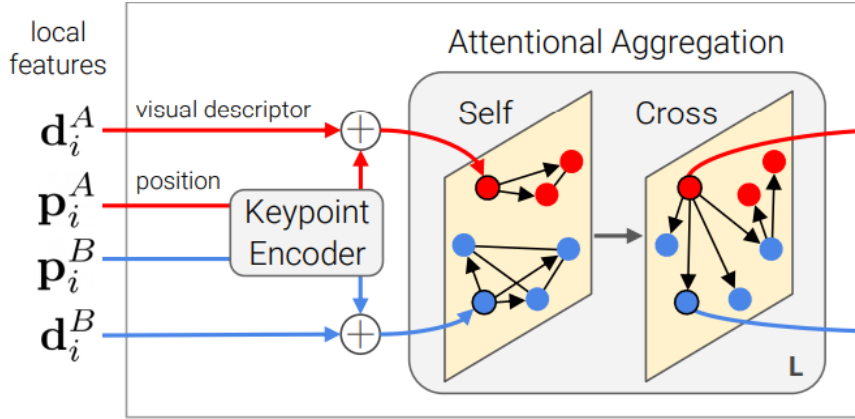
Weight:  $\alpha$

Message aggregation ( $m = \alpha_2 h_2 + \alpha_3 h_3 + \alpha_4 h_4$ )

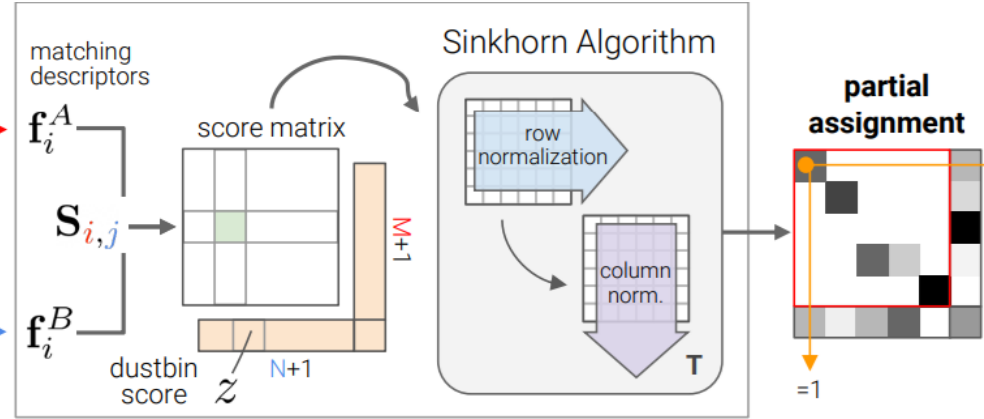


# SuperGlue

## Attentional Graph Neural Network



## Optimal Matching Layer

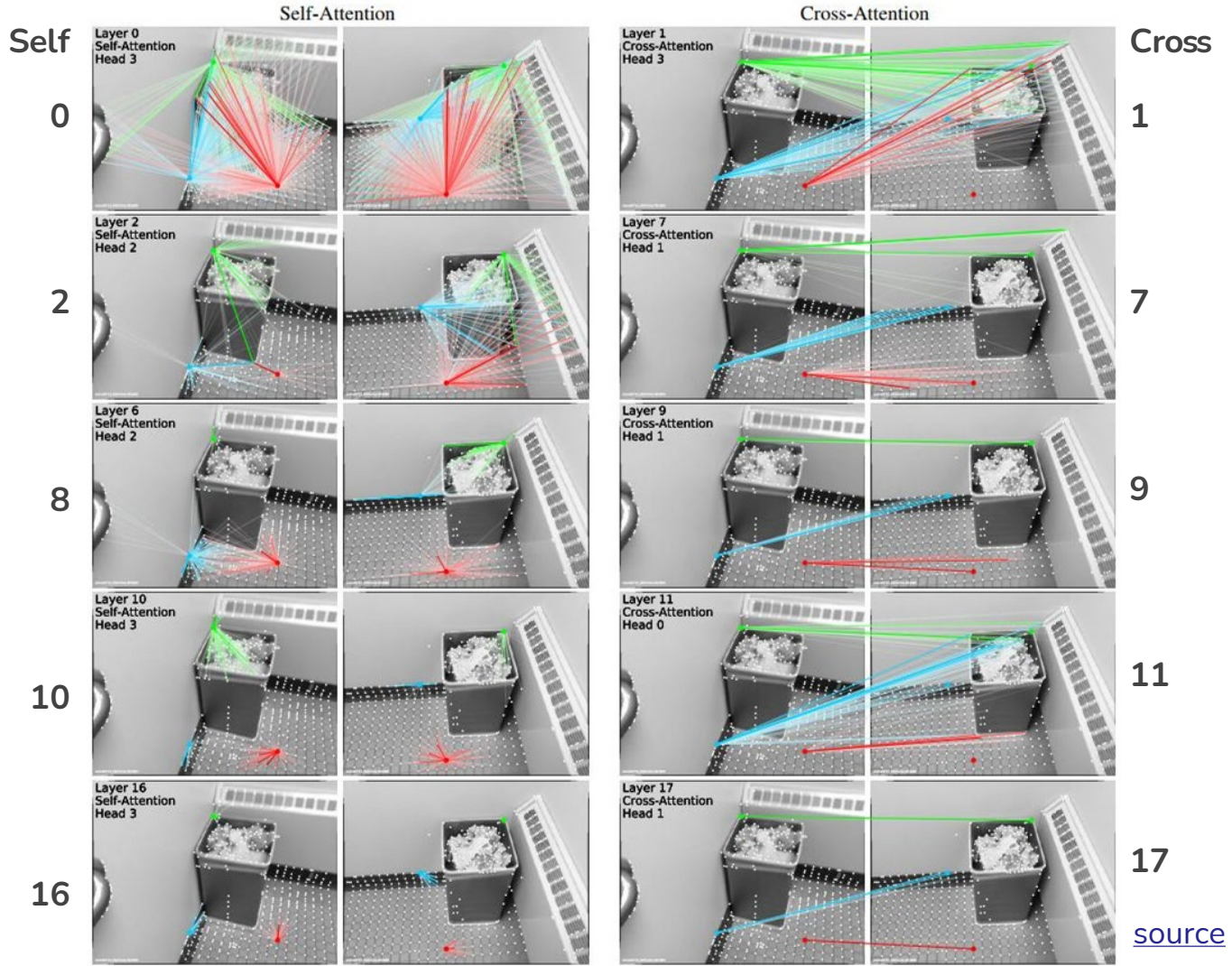


source

$$\mathbf{x}_i = \mathbf{d}_i + \text{MLP}_{\text{enc}}(\mathbf{p}_i)$$

**d**: descriptor  
**p**: keypoint position  
**x**: feature point(d+p)

$${}^{(\iota+1)}\mathbf{x}_i = {}^{(\iota)}\mathbf{x}_i + \text{MLP}\left(\left[{}^{(\iota)}\mathbf{x}_i \parallel m_{\varepsilon \rightarrow i}\right]\right)$$



[source](#)



# Homography Estimation Comparison

Match precision(P) and recall(R)

Local features	Matcher	Homography estimation AUC		P	R
		RANSAC	DLT		
	NN	39.47	0.00	21.7	65.4
	NN + mutual	42.45	0.24	43.8	56.5
SuperPoint	NN + PointCN	43.02	45.40	76.2	64.2
	NN + OANet	44.55	52.29	82.8	64.7
	<b>SuperGlue</b>	<b>53.67</b>	<b>65.85</b>	<b>90.7</b>	<b>98.3</b>

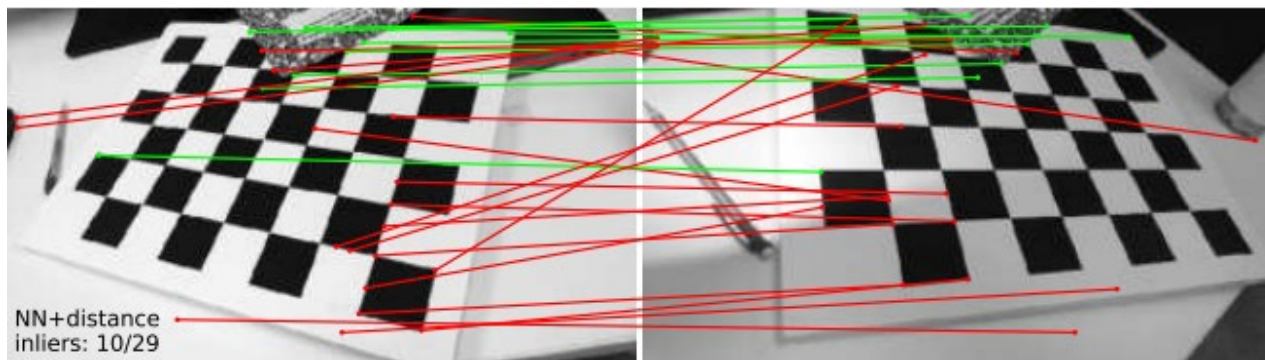
[source](#)

**Outlier Rejectors:** mutual NN constraint, PointCN, Order-Aware Network (OANet)

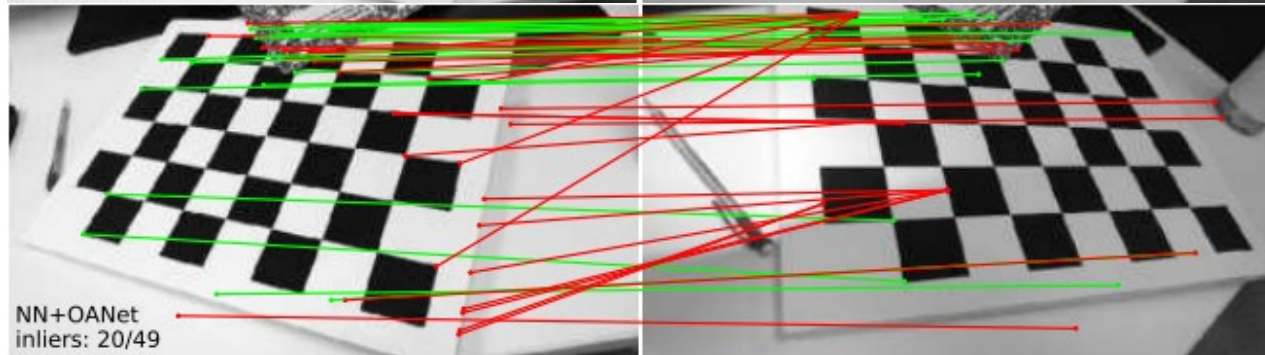
TP: True Positive  
FP: False Positive  
FN: False Negative

$$Prec = \frac{TP}{TP + FP}$$

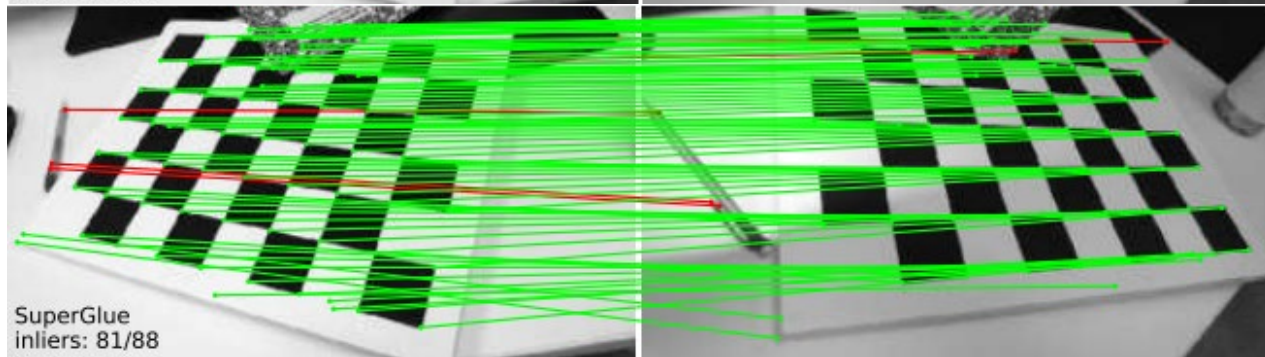
$$Recall = \frac{TP}{TP + FN}$$



NN+Distance



NN+OANet



SuperGlue

[source](#)



# Reference

<https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

<https://ieeexplore.ieee.org/document/8575521>

[https://openaccess.thecvf.com/content\\_CVPR\\_2020/supplemental/Sarlin\\_SuperGlue\\_Learning\\_Feature\\_CVPR\\_2020\\_supplemental.pdf](https://openaccess.thecvf.com/content_CVPR_2020/supplemental/Sarlin_SuperGlue_Learning_Feature_CVPR_2020_supplemental.pdf)