



Robot Perception

Robust Estimation

Dr. Chen Feng

cfeng@nyu.edu

ROB-GY 6203, Fall 2023



Overview

+ Feature detection

Harris/FAST/DoG

+ Feature description & matching

SIFT/SURF

++ Linear & total least square

* RANSAC

Intuitions behind RANSAC

How RANSAC works

Why minimal solution is important

More example problems

Variations

*: know how to code

++: know how to derive

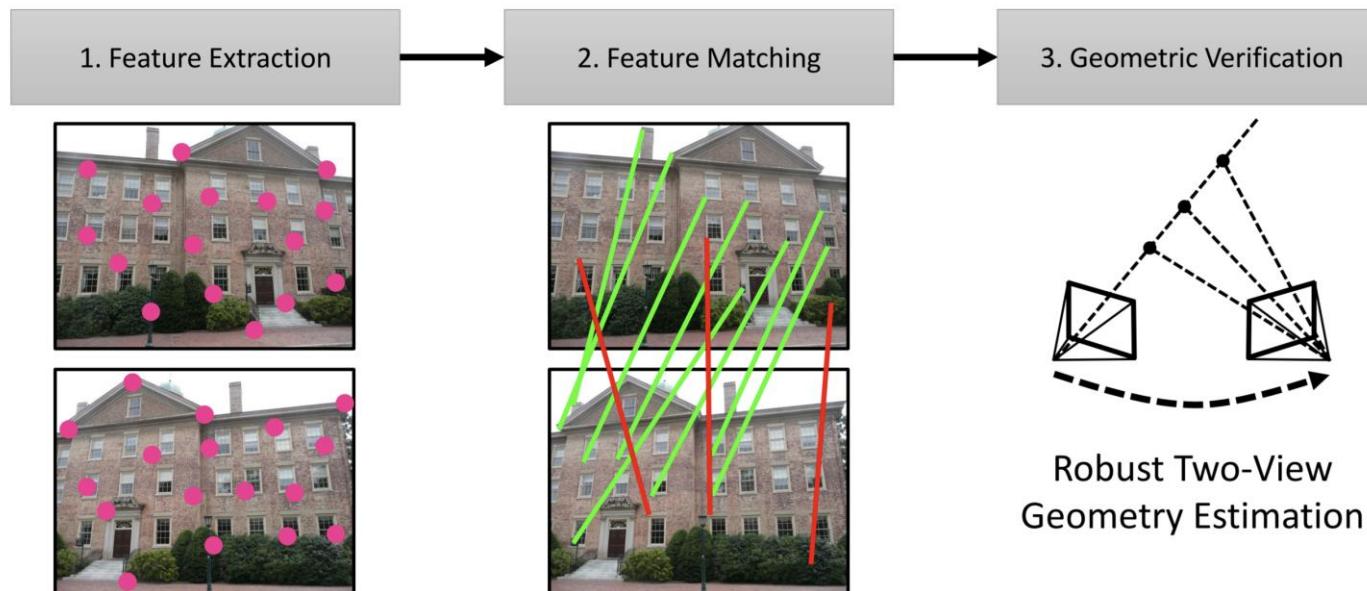
+: know the concept

References

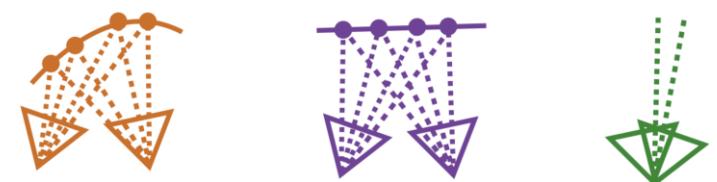
- HZ2003:
 - Section 4.7, 4.8, 11.6
- Corke 2011:
 - Section 14.2.3
- Sz2022:
 - Section 7.1, 7.2, 8.1.4
- DeTone, D., Malisiewicz, T. and Rabinovich, A., 2018. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops* (pp. 224-236).
- Sarlin, P.E., DeTone, D., Malisiewicz, T. and Rabinovich, A., 2020. Superglue: Learning feature matching with graph neural networks. In *CVPR* (pp. 4938-4947).



Data Association: Finding Correspondences Automatically

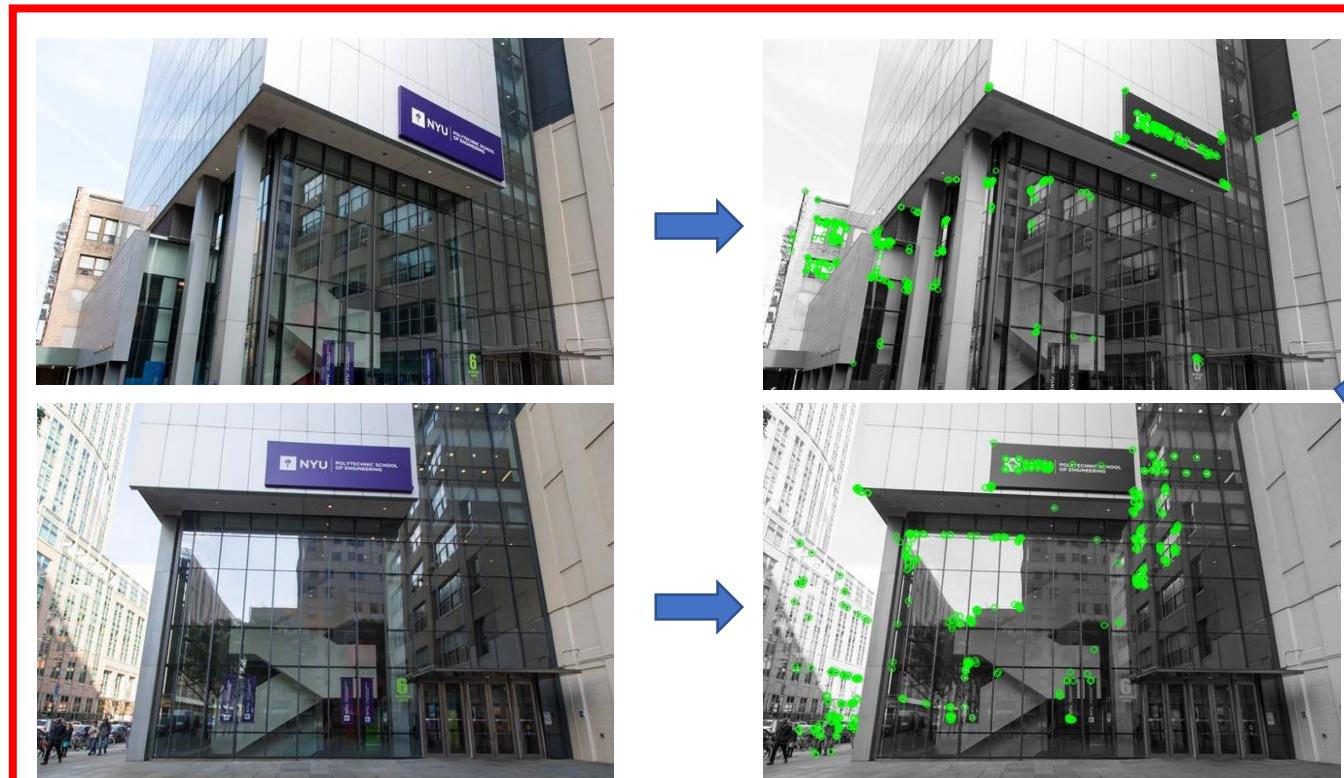


General	Planar	Panoramic
<ul style="list-style-type: none">• Fundamental matrix F (<i>uncalibrated</i>)• Essential matrix E (<i>calibrated</i>)• 7 correspondences• 5 correspondences	<ul style="list-style-type: none">• Homography H	<ul style="list-style-type: none">• Homography H



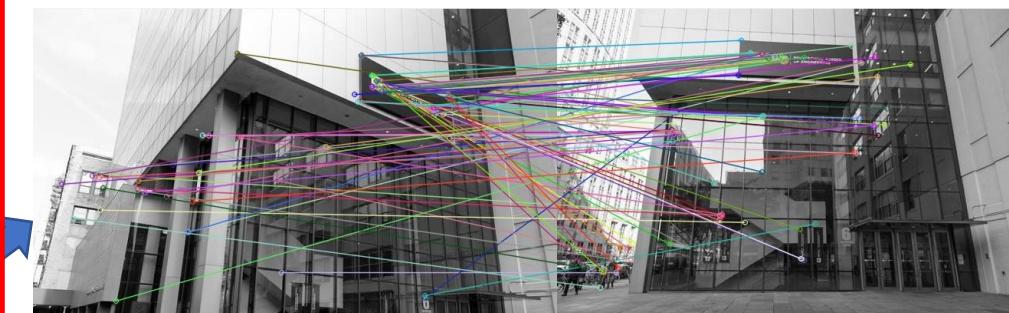


Data Association: Finding Correspondences Automatically



Feature Detection
(Harris/FAST/SIFT/SURF/ORB/LSD)

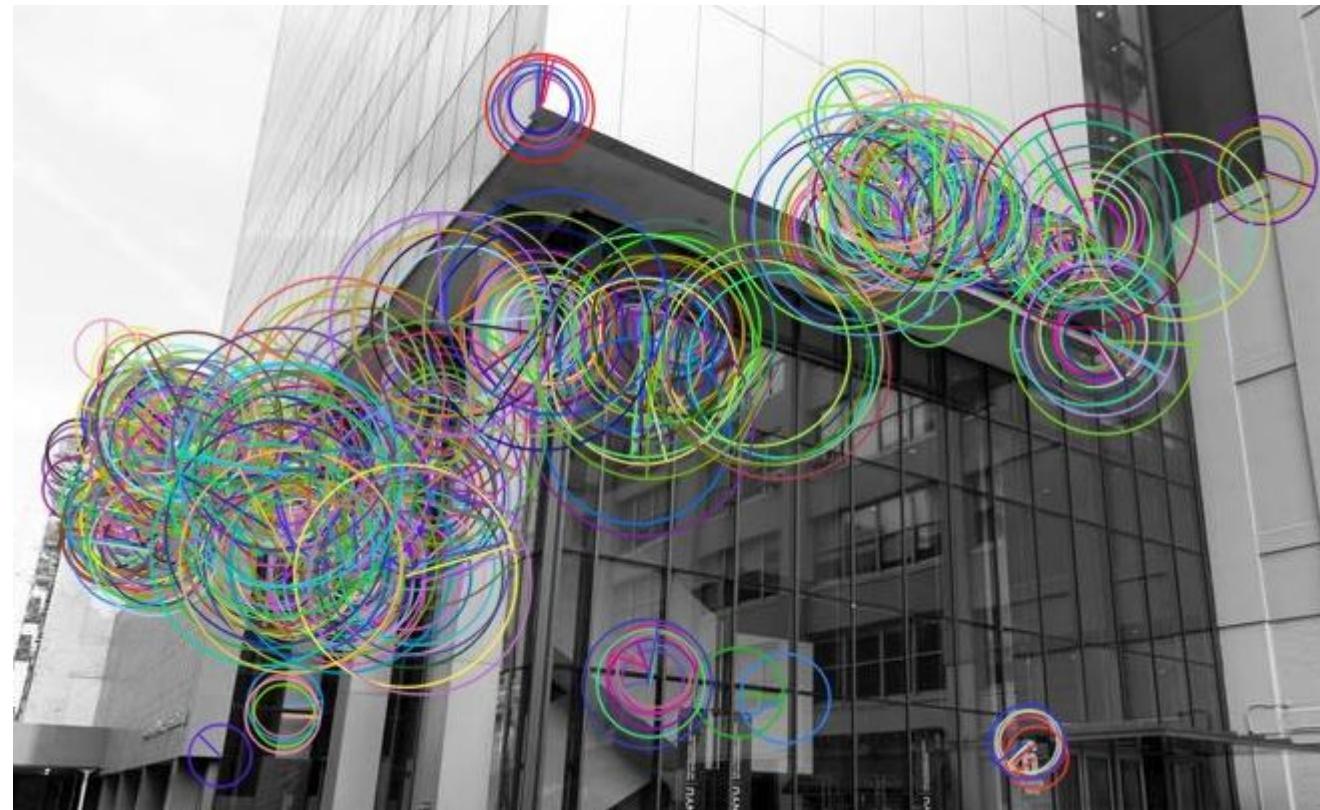
Feature Description/Matching
(SIFT/SURF/ORB)



- Homography estimation
- F-matrix estimation
- PnP problem
- ...
- RANSAC to reject matching outliers



Corner/Blob Detection





Corner Detection Criteria Illustrations

Repeatability



Illumination
invariance



Scale
invariance

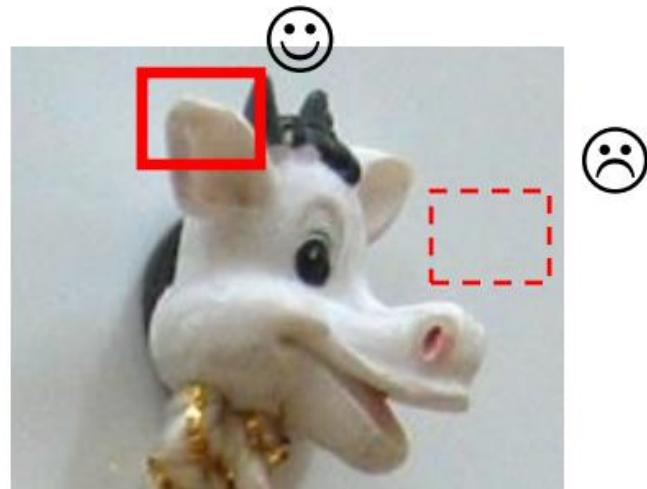


Pose invariance
•Rotation
•Affine

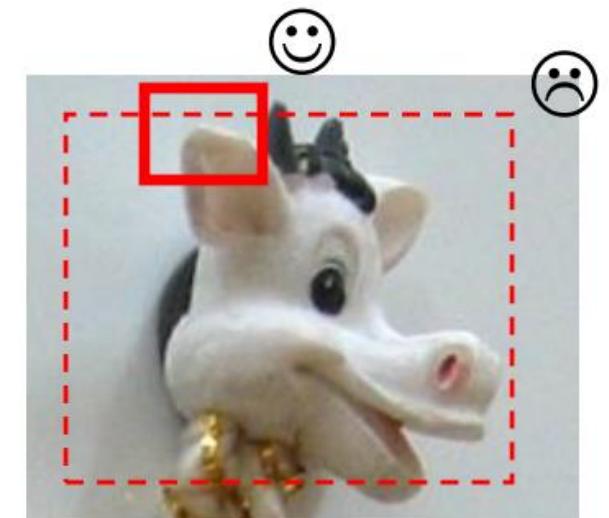


Corner Detection Criteria Illustrations

- Saliency



- Locality





Harris Corner

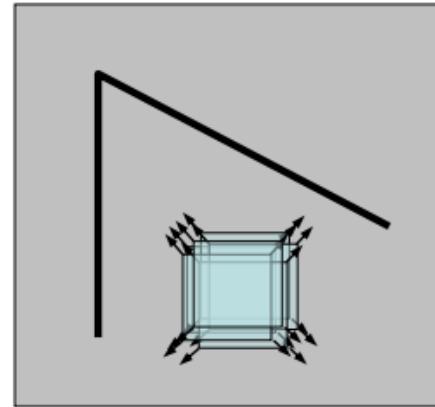
- C.Harris and M.Stephens. "A Combined Corner and Edge Detector." Proceedings of the 4th Alvey Vision Conference: pages 147--151.



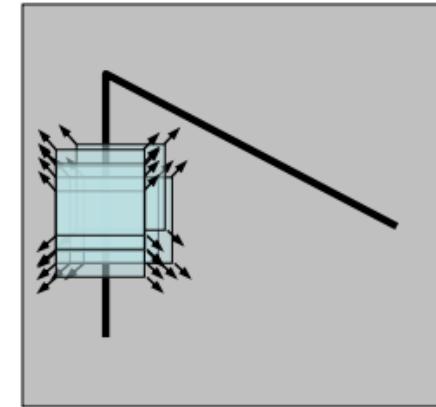


Harris Detector: Basic Idea

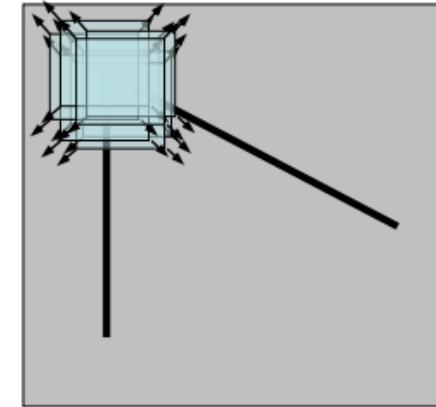
Explore intensity changes within a window
as the window changes location



“flat” region:
no change in
all directions



“edge”:
no change
along the edge
direction

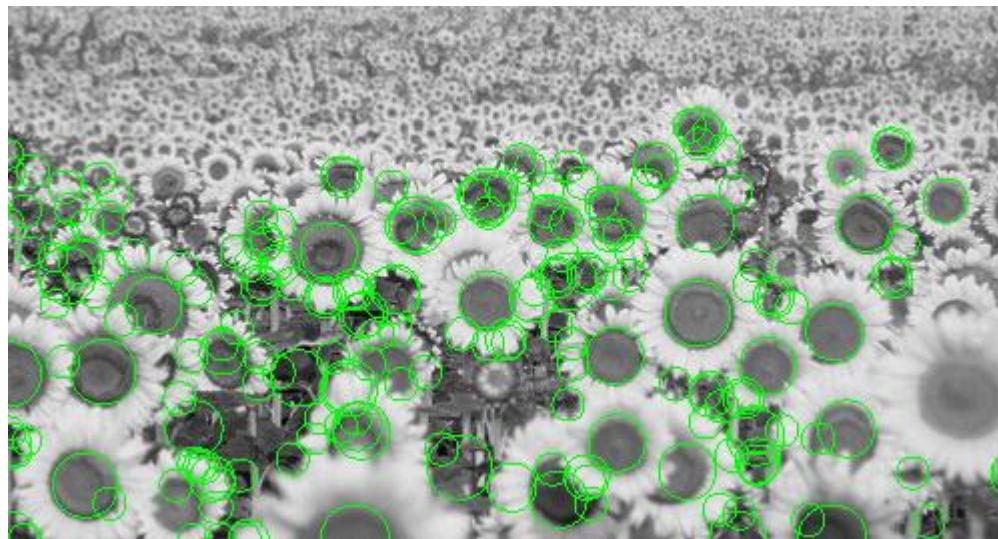


“corner”:
significant
change in all
directions

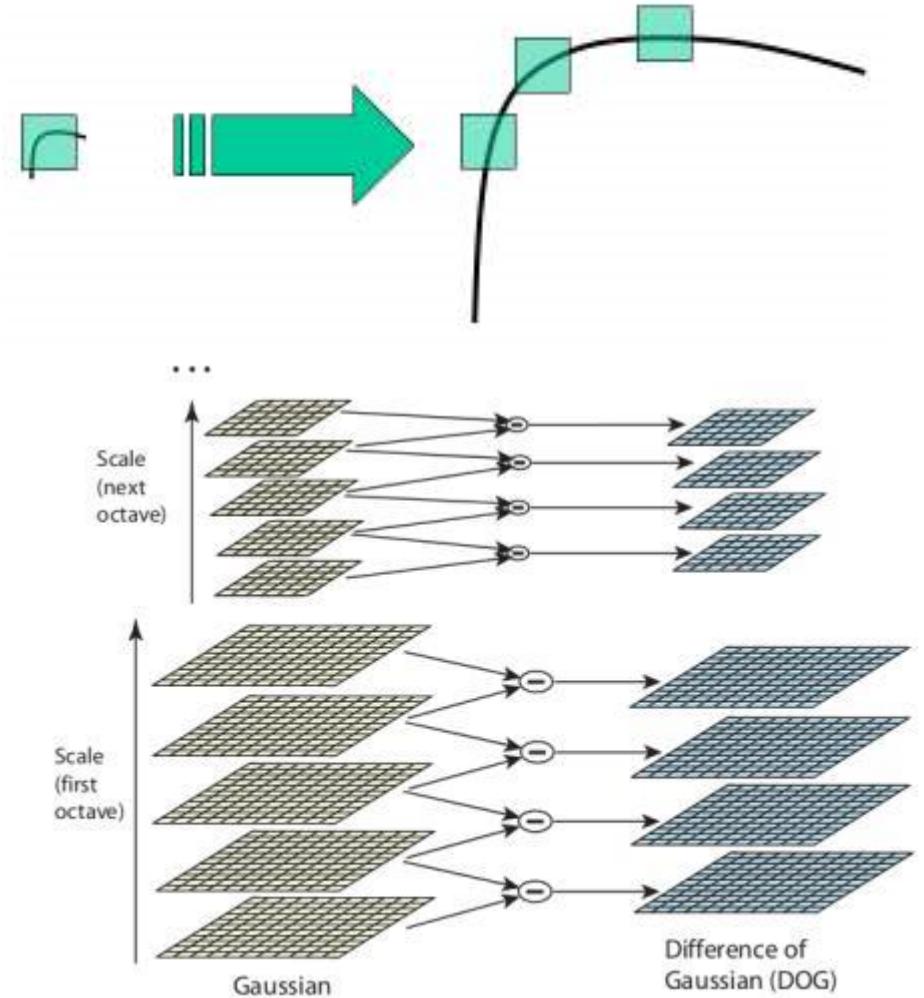


Blob Detector: Difference of Gaussians (DoG)

- Harris corner is rotation invariant
 - But not scale-invariant



- DoG: find extrema in both 2D-space and scale-space

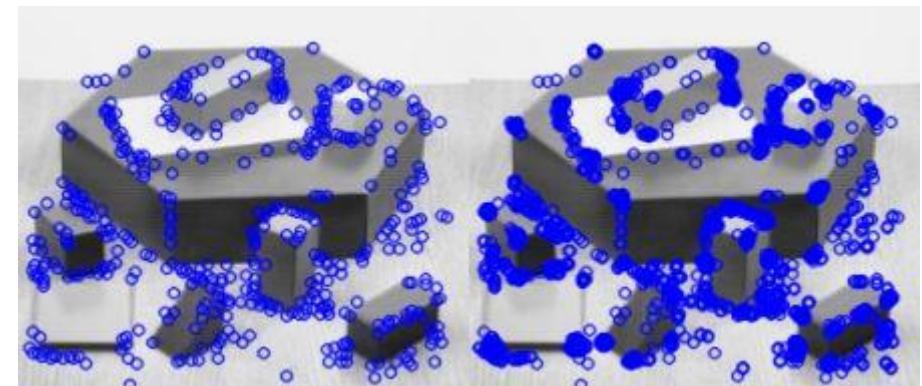
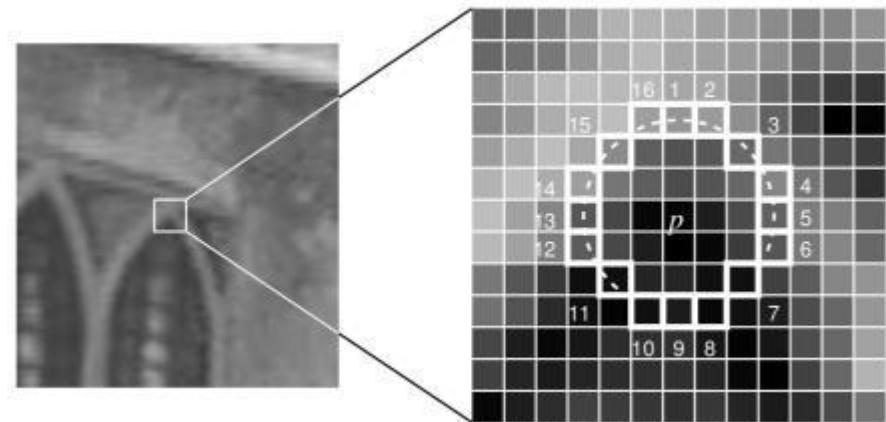


Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.



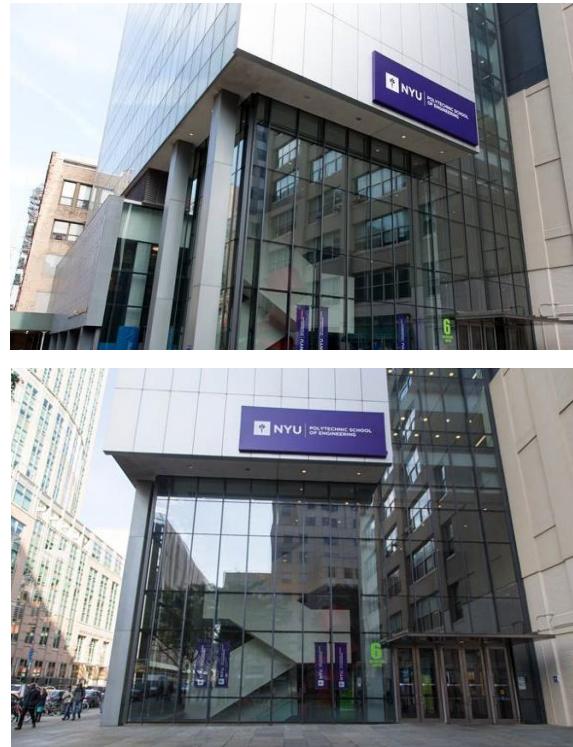
FAST: Machine Learning for Corner Detection

- FAST (Features from Accelerated Segment Test)
 - Corner: if there exists $n=12$ contiguous pixels in the circle which are all brighter or all darker than the center for a threshold t
- High-speed test
 - quickly exclude many non-corners
- Decision-tree based improvement
- Non-maximum suppression

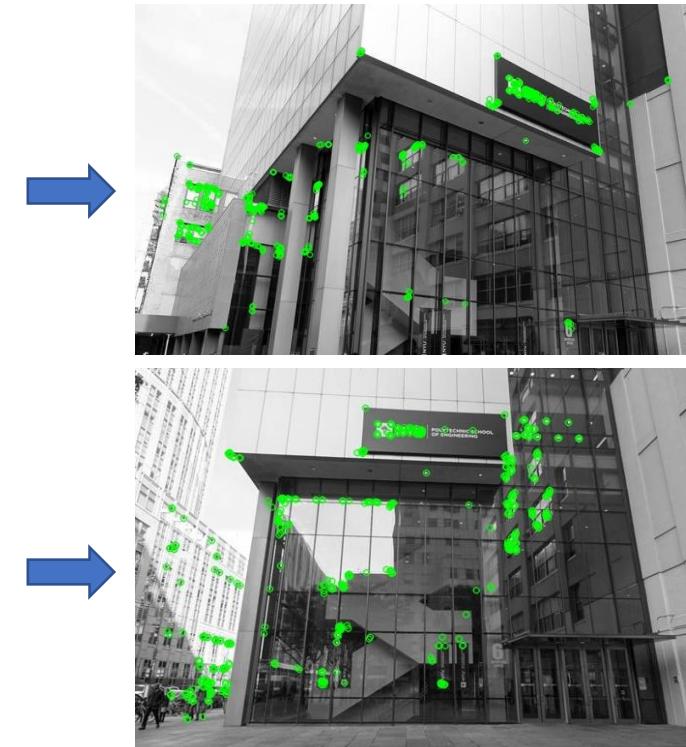


Edward Rosten, Reid Porter, and Tom Drummond, "Faster and better: a machine learning approach to corner detection" in IEEE Trans. Pattern Analysis and Machine Intelligence, 2010, vol 32, pp. 105-119.

A Classic Vision Pipeline: Description & Matching



Feature Detection
(Harris/FAST/SIFT/SURF/ORB
/LSD)



Feature Description/Matching
(SIFT/SURF/ORB)



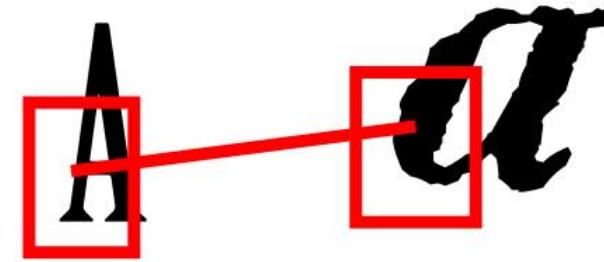
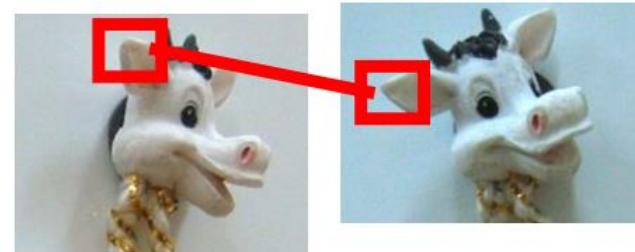
- Homography estimation
- F-matrix estimation
- PnP problem
- ...
- RANSAC to reject matching outliers



Feature Description

Depending on the application a descriptor must incorporate information that is:

- Invariant w.r.t:
 - Illumination
 - Pose
 - Scale
 - Intraclass variability

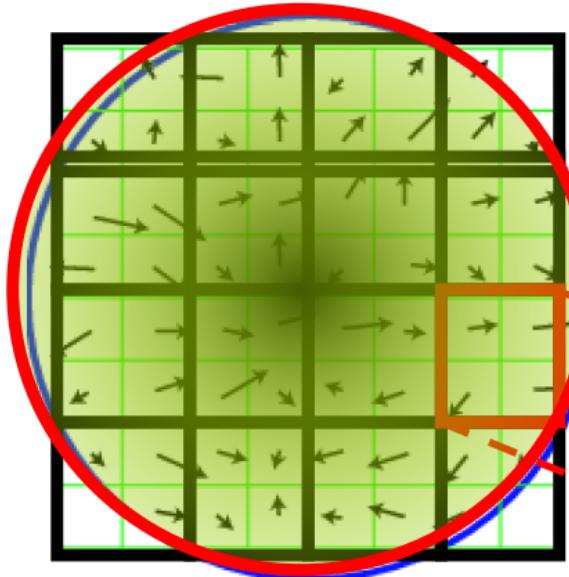


- **Highly distinctive** (allows a single feature to find its correct match with good probability in a large database of features)

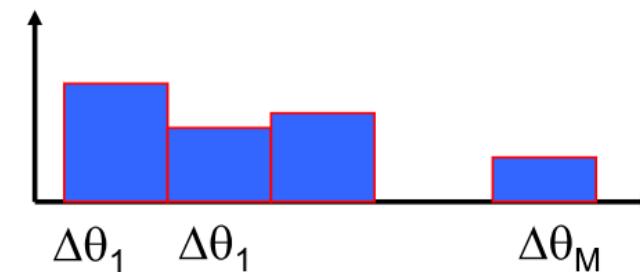


SIFT Descriptor

- A standard (but non-free) descriptor
- Location and scale given by DoG detector (SIFT keypoints)



- Compute gradient at each pixel
 - $N \times N$ spatial bins
 - Compute an histogram of M orientations for each bean
 - Gaussian center-weighting

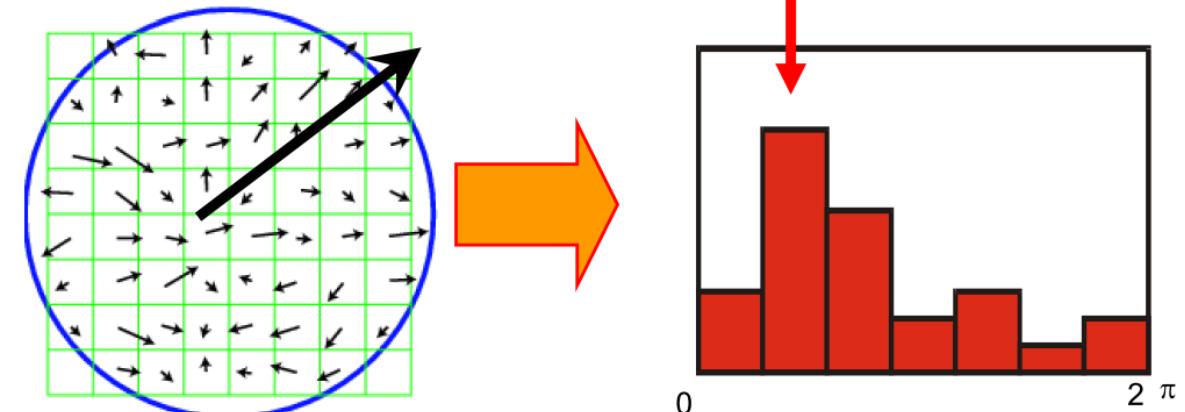


Typically $M = 8$; $N = 4$
 1×128 descriptor

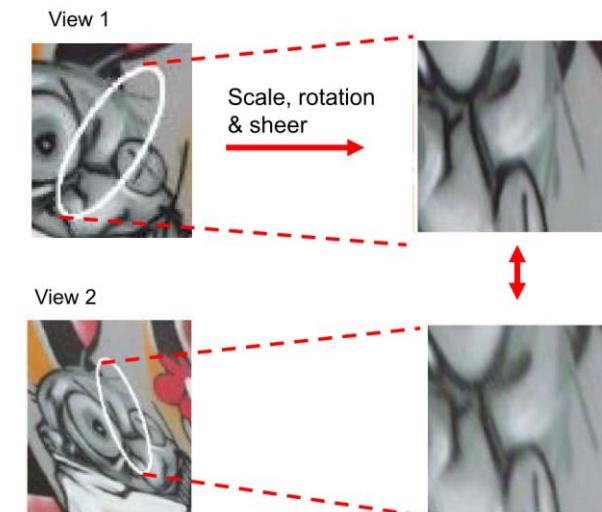


SIFT Descriptor is Robust to Small Variations

- Illumination
 - gradient & normalization
- Pose (small affine variation)
 - orientation histogram
- Scale
 - fixed by DOG
- Intra-class variability
 - histograms (small variations)



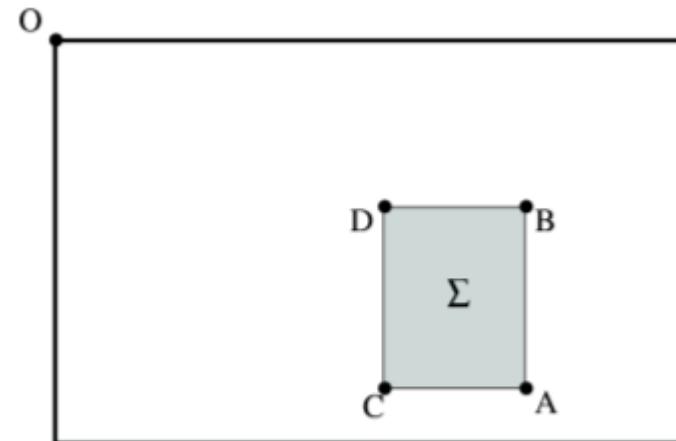
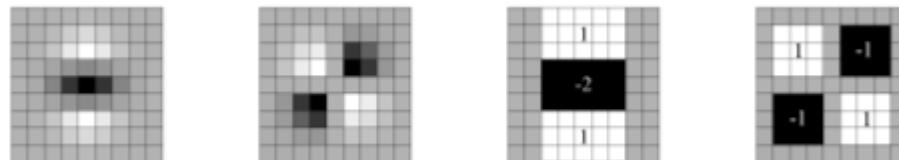
This makes the SIFT descriptor rotational invariant





From SIFT to SURF

- SIFT is good in terms of matching quality
 - But it is too slow for real-time applications
- SURF uses integral image to speed up the SIFT computation



Bay, H., Tuytelaars, T. and Van Gool, L., 2006, May. Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer, Berlin, Heidelberg.



Faster than SURF? BRIEF/ORB

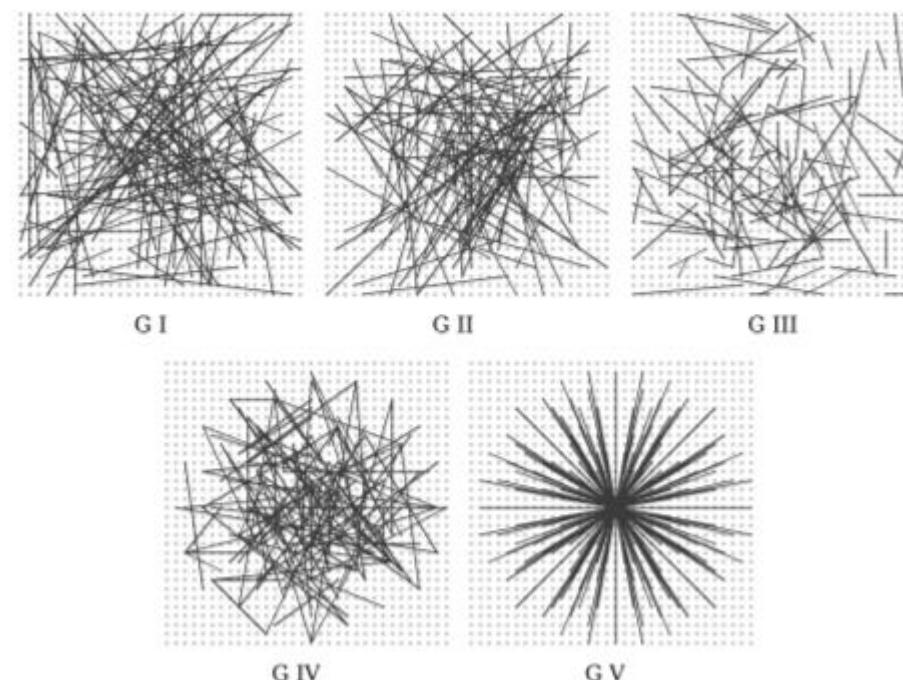
- Connecting feature description with machine learning

More specifically, we define test τ on patch \mathbf{p} of size $S \times S$ as

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where $\mathbf{p}(\mathbf{x})$ is the pixel intensity in a smoothed version of \mathbf{p} at $\mathbf{x} = (u, v)^\top$. Choosing a set of n_d (\mathbf{x}, \mathbf{y}) -location pairs uniquely defines a set of binary tests. We take our BRIEF descriptor to be the n_d -dimensional bitstring

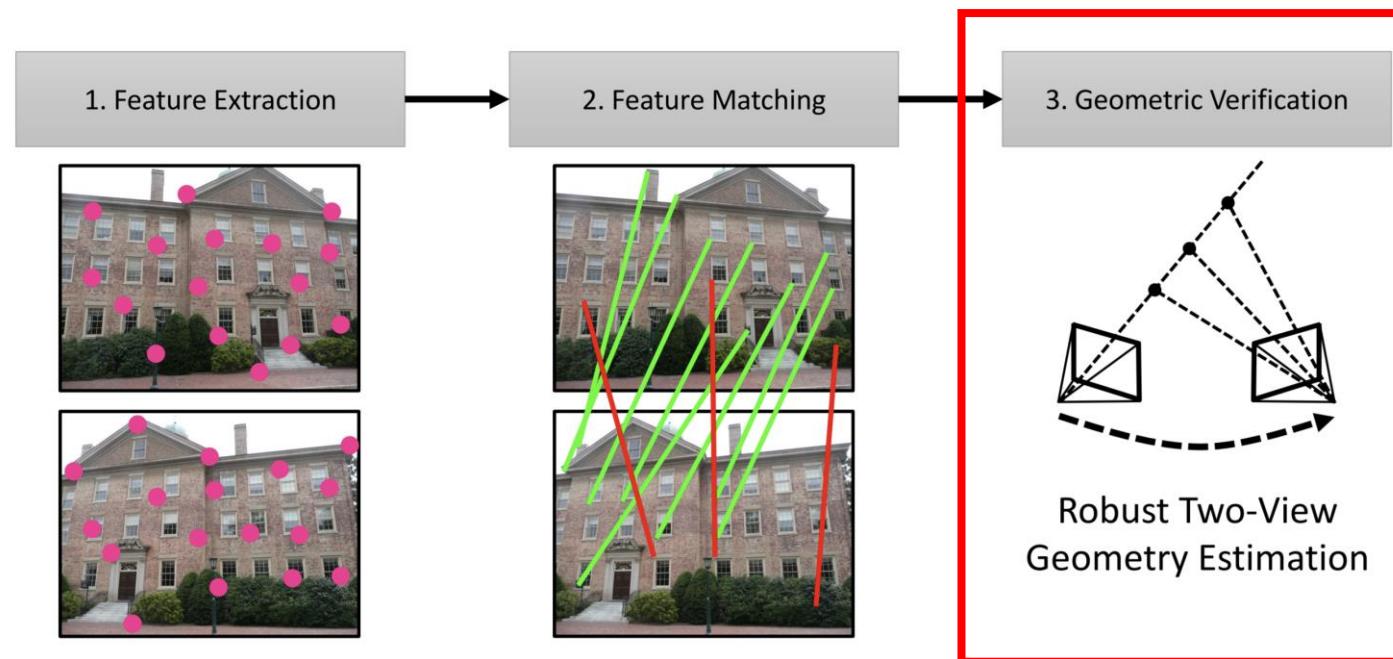
$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i). \quad (2)$$



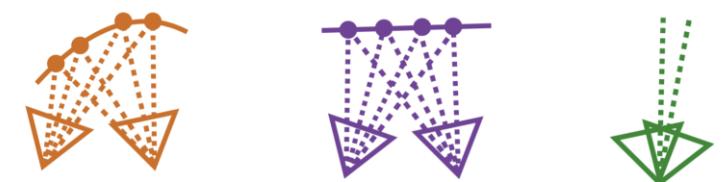
Calonder, Michael, et al. "Brief: Binary robust independent elementary features." European conference on computer vision. Springer, Berlin, Heidelberg, 2010.



Data Association: Wrong Matches are Inevitable

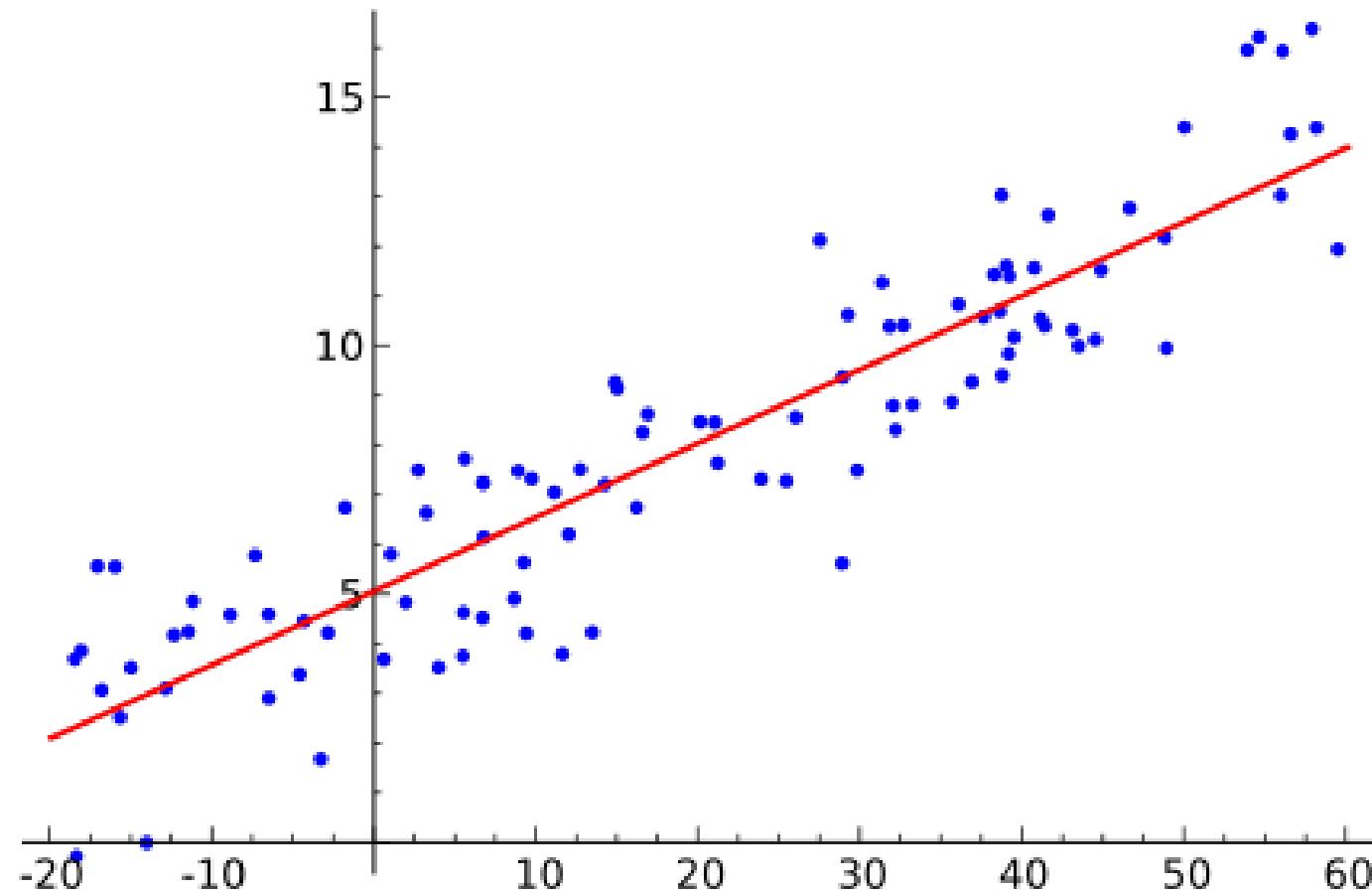


General	Planar	Panoramic
<ul style="list-style-type: none">• Fundamental matrix F (<i>uncalibrated</i>)• Essential matrix E (<i>calibrated</i>)• 7 correspondences• 5 correspondences	<ul style="list-style-type: none">• Homography H	<ul style="list-style-type: none">• Homography H• 4 correspondences• 4 correspondences



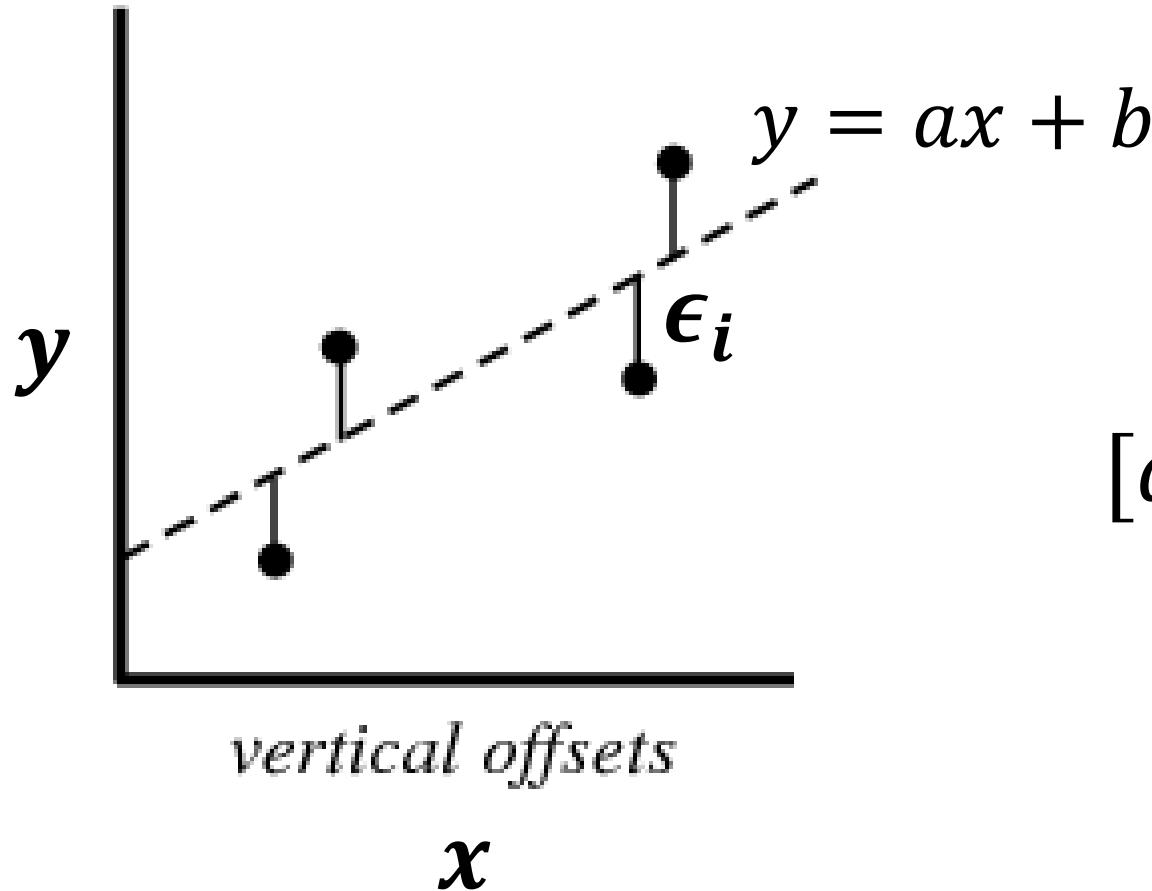


Fitting a 2D Line





Linear Regression (Linear Least Squares)



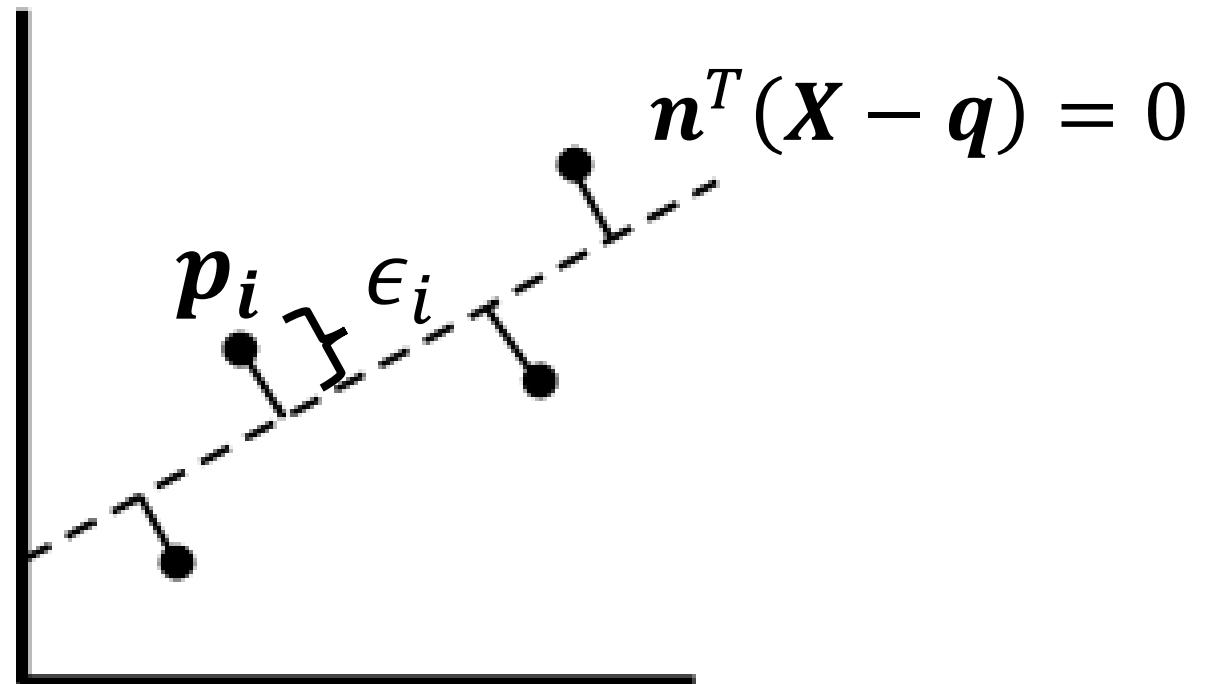
$$\epsilon = y - ax - b$$
$$[a, b]^* = \operatorname{argmin}_{a,b} \|\epsilon\|^2$$



Orthogonal Regression (Total Least Squares)

$$\epsilon_i = \mathbf{n}^T (\mathbf{p}_i - \mathbf{q})$$

$$\begin{aligned} [\mathbf{n}, \mathbf{q}]^* &= \operatorname{argmin}_{\mathbf{n}, \mathbf{q}} \sum_i \|\epsilon_i\|^2 \\ \text{s.t. } \|\mathbf{n}\|^2 &= 1 \end{aligned}$$



Orthogonal Regression for Line Estimation

- Given a set of 3D points $\{p_i\}$, we want to find out a line/plane (i.e., unit normal n and center q) that describes this set of points as

$$n^\top(p_i - q) = 0, \forall i$$

$$\begin{aligned} \text{cost}(n, q) &\triangleq \sum_i \text{dist}^2(p_i; n, q) \\ &= \sum_i (n^\top(p_i - q))^2 \\ &= n^\top [\dots, p_i - q, \dots] [\dots, p_i^\top - q^\top, \dots]^\top n \\ &= n^\top A(q) A(q)^\top n \end{aligned}$$



Orthogonal Regression for Line Estimation

- Solving \mathbf{q}

$$\mathbf{0} = \frac{\partial \text{cost}(n, q)}{\partial q} \equiv \sum_i (2nn^\top q - 2nn^\top p_i)$$

$$q^* = \frac{1}{|\{p_i\}|} \sum_i p_i$$

Orthogonal Regression for Line Estimation

- Solving \mathbf{n}

$$\text{cost}(\mathbf{n}; \mathbf{q}^*) \triangleq \mathbf{n}^\top \mathbf{A}(\mathbf{q}^*) \mathbf{A}(\mathbf{q}^*)^\top \mathbf{n} = \mathbf{n}^\top \mathbf{B}(\mathbf{q}^*) \mathbf{n}$$

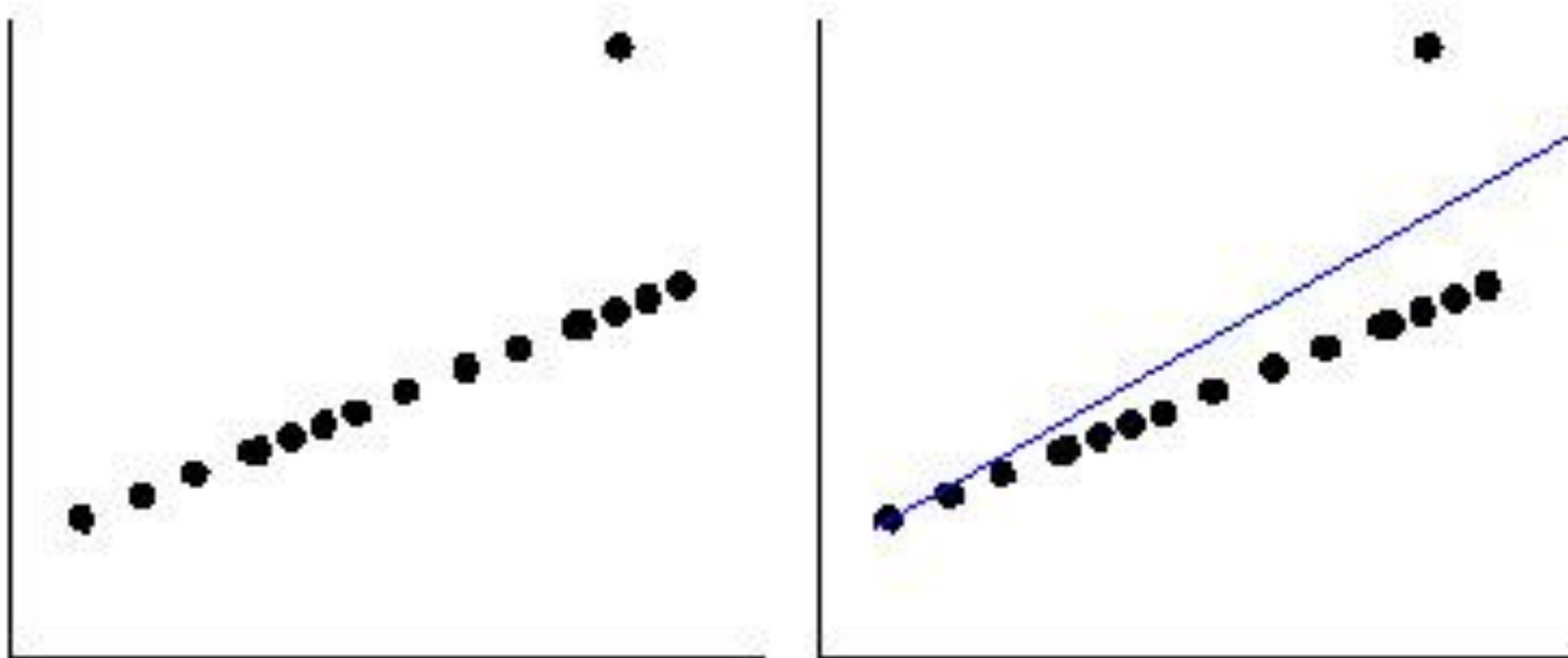
- Equivalent to solve:

$$\begin{aligned} \mathbf{n}^* &= \arg \min_{\mathbf{n}} && \mathbf{n}^\top \mathbf{B}(\mathbf{q}^*) \mathbf{n} \\ &&\text{s.t.} && \mathbf{n}^\top \mathbf{n} = 1 \end{aligned}$$

- Solve by SVD
 - Optimal \mathbf{n} is \mathbf{B} 's eigenvector corresponding to the smallest eigenvalue.
 - So, this is also referred to as the PCA-based solution.



But Least Squares is NOT Robust to Outliers!



<http://www.unige.ch/ses/sococ/cl////stat/action/nonlin5.jpg>



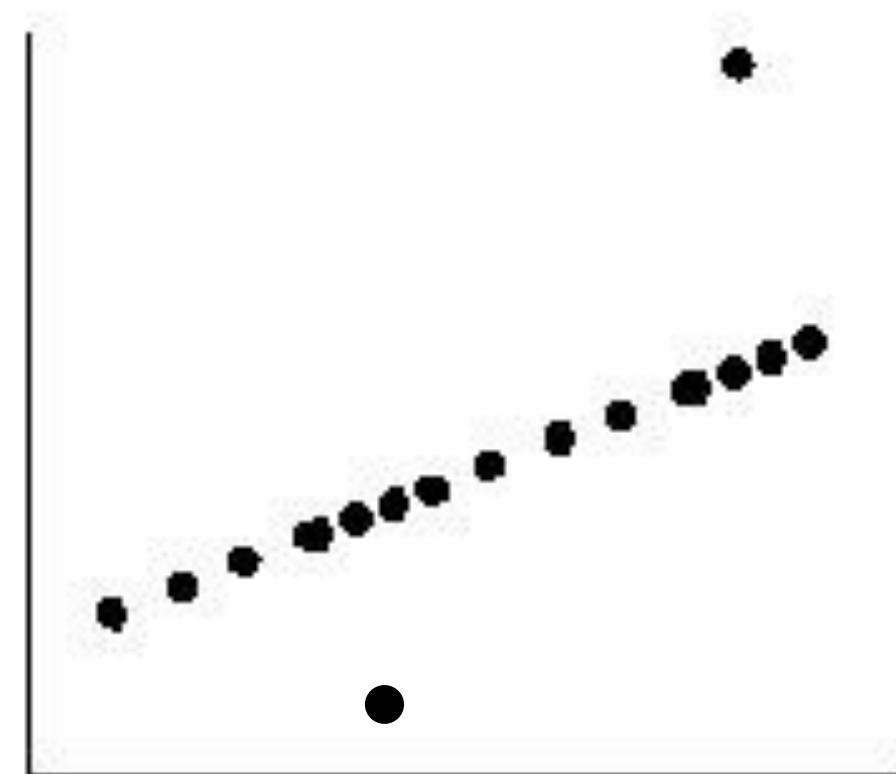
How to Solve This Intuitively?

- Mitigating outlier's influence/weight in the estimation
 - Iteratively Re-weighted Least Squares (IRLS)
- Detect outlier and remove it from estimation
 - RANdom SAmple Consensus (RANSAC)



How to Detect an Outlier?

- Enumeration strategy
 - Leave one out
- Voting strategy
 - Hypothesis and test

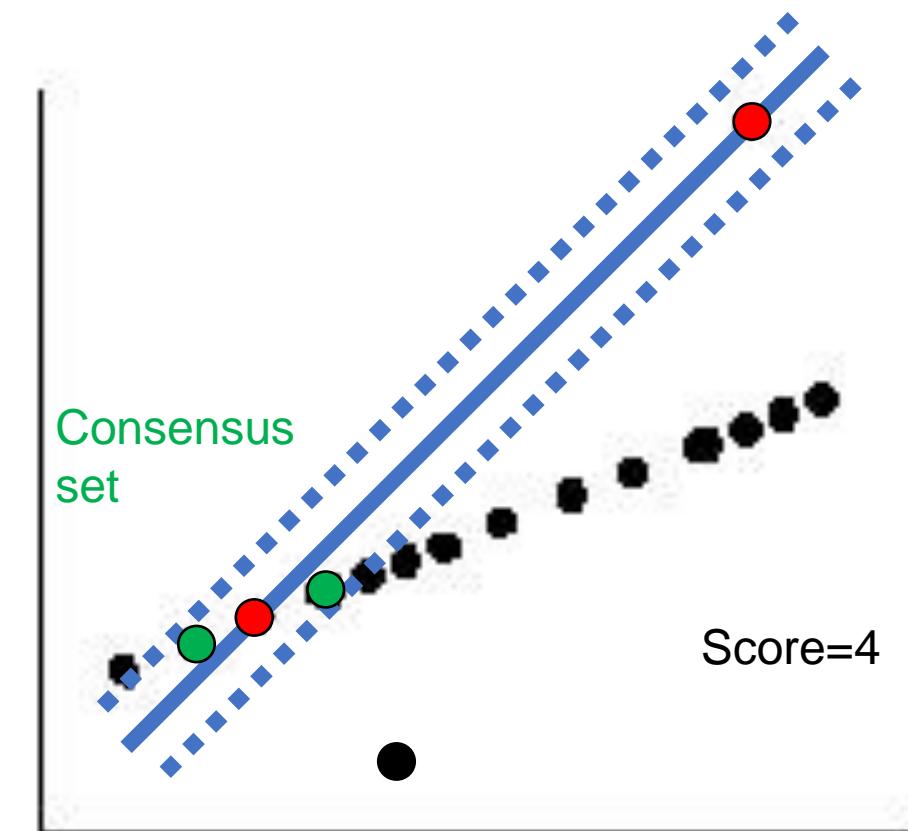


<http://www.unige.ch/ses/sococ/cl////stat/action/nonlin5.jpg>



Hypothesis and Test

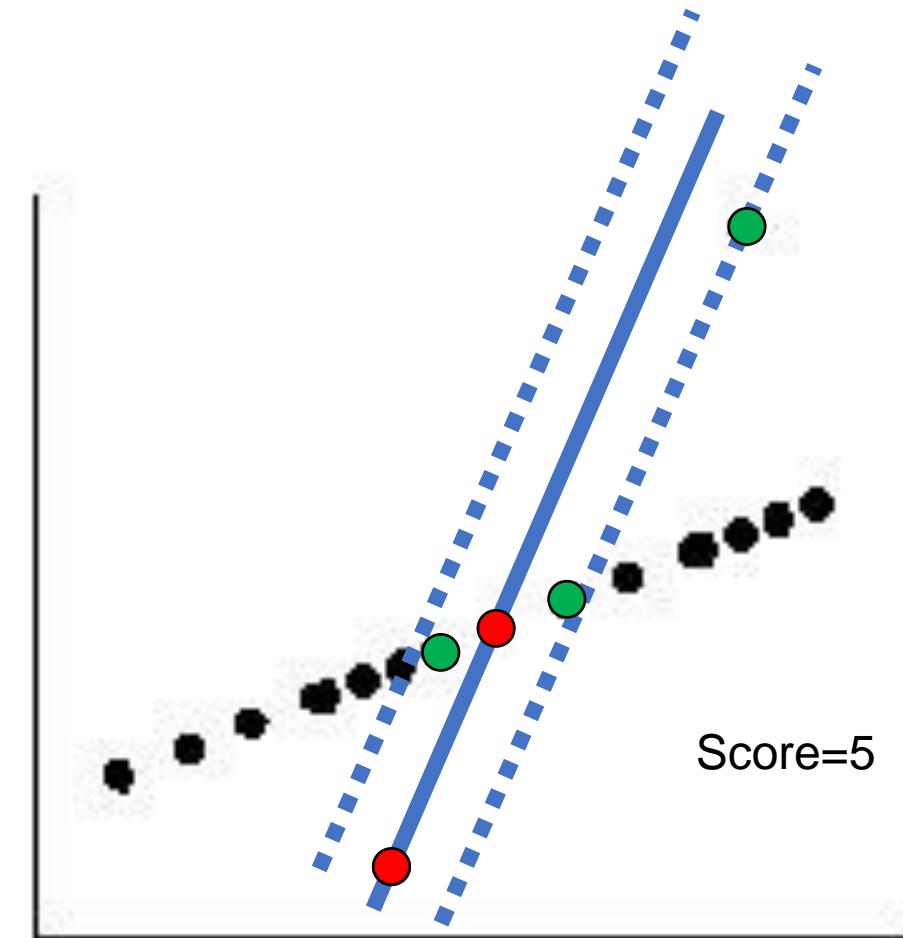
- Randomly select two points to generate a hypothesis line
- Test how good the current hypothesis is
 - Consensus set = {supporting points}
 - Score by #(supporting points)=
 $|{\text{Consensus set}}|$





Hypothesis and Test

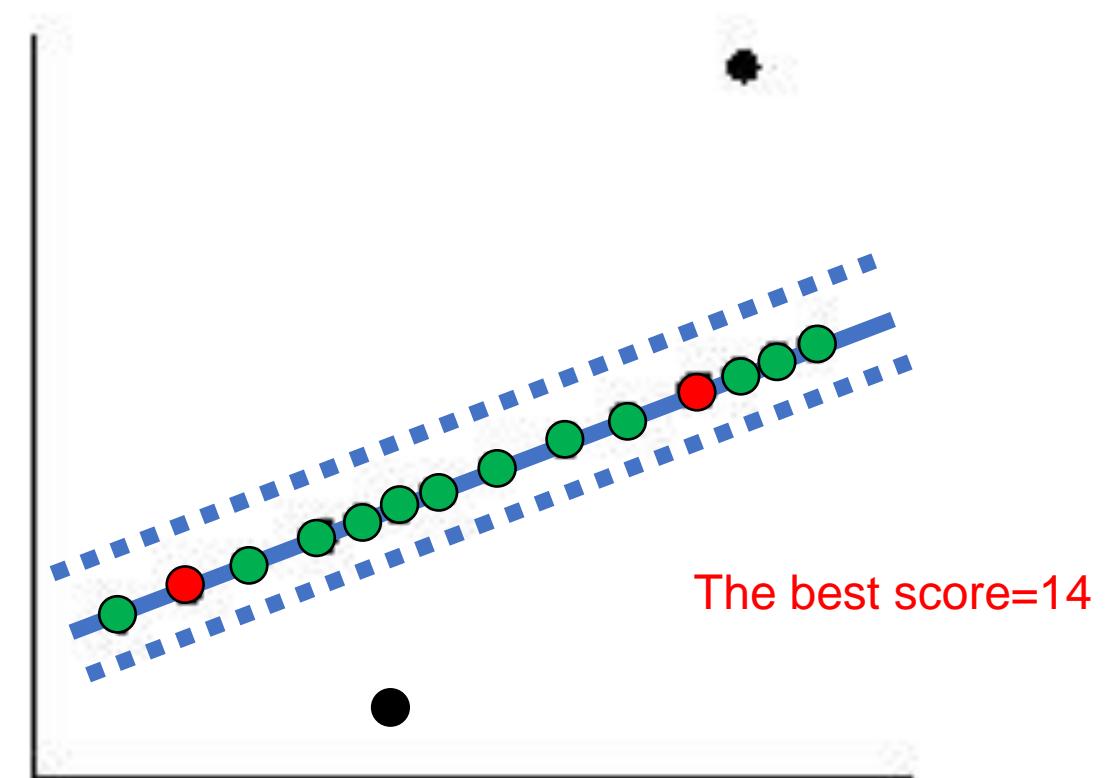
- Randomly select two points to generate a hypothesis line
- Test how good the current hypothesis is





Hypothesis and Test

- Randomly select two points to generate a hypothesis line
- Test how good the current hypothesis is





RANSAC Framework

Objective

Robust fit of a model to a data set S which contains outliers.

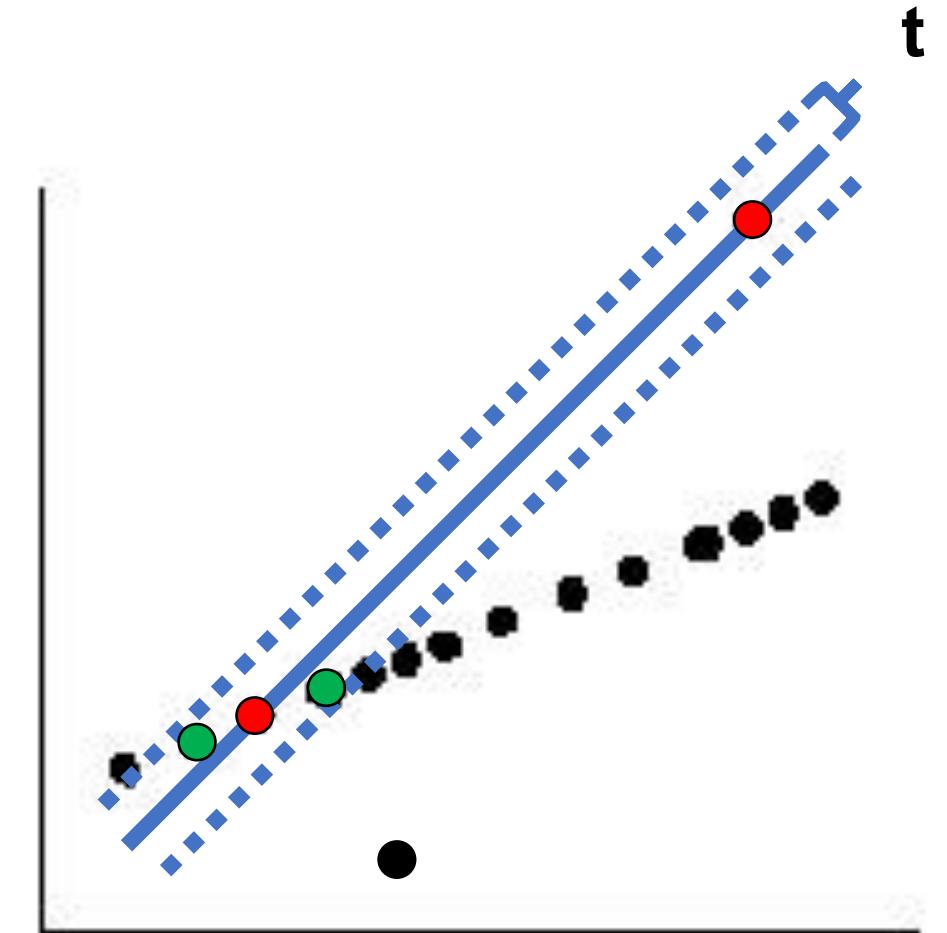
Algorithm

- (i) Randomly select a sample of s data points from S and instantiate the model from this subset.
- (ii) Determine the set of data points S_i which are within a distance threshold t of the model.
The set S_i is the consensus set of the sample and defines the inliers of S .
- (iii) If the size of S_i (the number of inliers) is greater than some threshold T ,
re-estimate the model using all the points in S_i and terminate.
- (iv) If the size of S_i is less than T , select a new subset and repeat the above.
- (v) After N trials the largest consensus set S_i is selected, and the model is
re-estimated using all the points in the subset S_i .

Often omitted in
implementation

RANSAC Details – Distant Threshold t

- If data is known to be distributed as a Gaussian of standard deviation σ :
 - Use the 3σ rule
- Otherwise:
 - Determined manually from experience
 - Try-and-error





RANSAC Details – #Samples N



- Basic idea: max the probability p of **at least one** successful sampling
 - Success sample: all the s sampled data points are inliers
- Assume outlier ratio ϵ is known
- Probability of one successful sampling (sampling with replacement): $(1 - \epsilon)^s$
- Probability of at least one successful sampling: $1 - [1 - (1 - \epsilon)^s]^N$
- Thus: $N = \ln(1 - p) / \ln[1 - (1 - \epsilon)^s]$

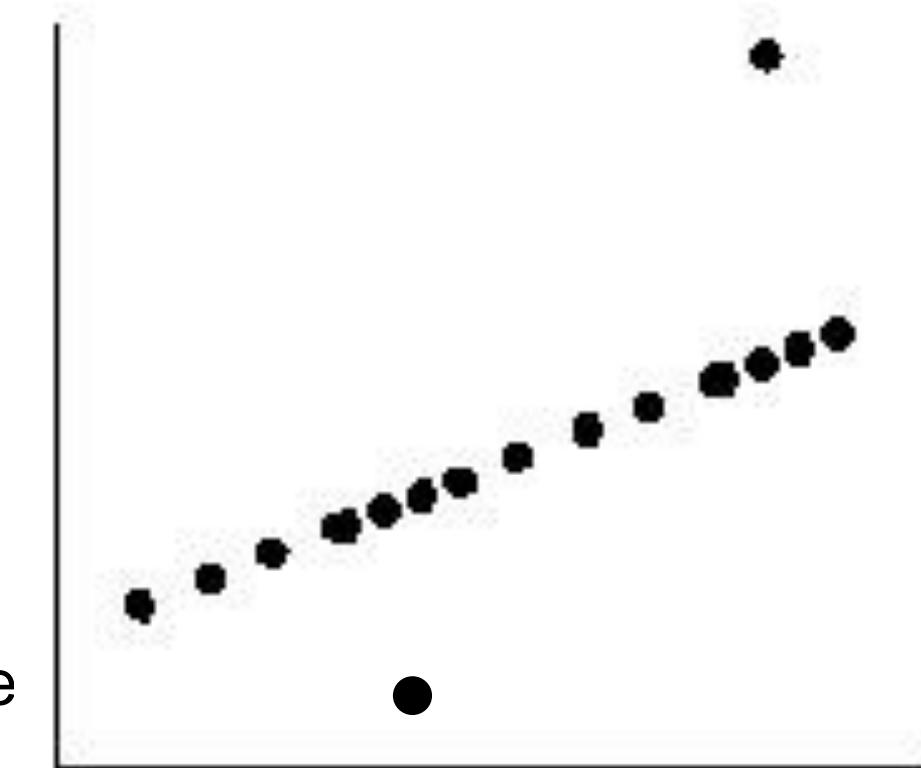


Why Minimal Solution is Important

- For $p=0.99$



Sample size	Proportion of outliers ϵ							
	5%	10%	20%	25%	30%	40%	50%	
2	2	3	5	6	7	11	17	
3	3	4	7	9	11	19	35	
4	3	5	9	13	17	34	72	
5	4	6	12	17	26	57	146	
6	4	7	16	24	37	97	293	
7	4	8	20	33	54	163	588	
8	5	9	26	44	78	272	1177	



RANSAC Details – Adaptive Sampling

- What if outlier ratio ϵ is NOT known?

- $N = \infty$, sample_count= 0.
- While $N >$ sample_count Repeat
 - Choose a sample and count the number of inliers.
 - Set $\epsilon = 1 - (\text{number of inliers})/(\text{total number of points})$
 - Set N from ϵ and (4.18) with $p = 0.99$.
 - Increment the sample_count by 1.
- Terminate.



More Examples – Homography

Objective

Compute the 2D homography between two images.

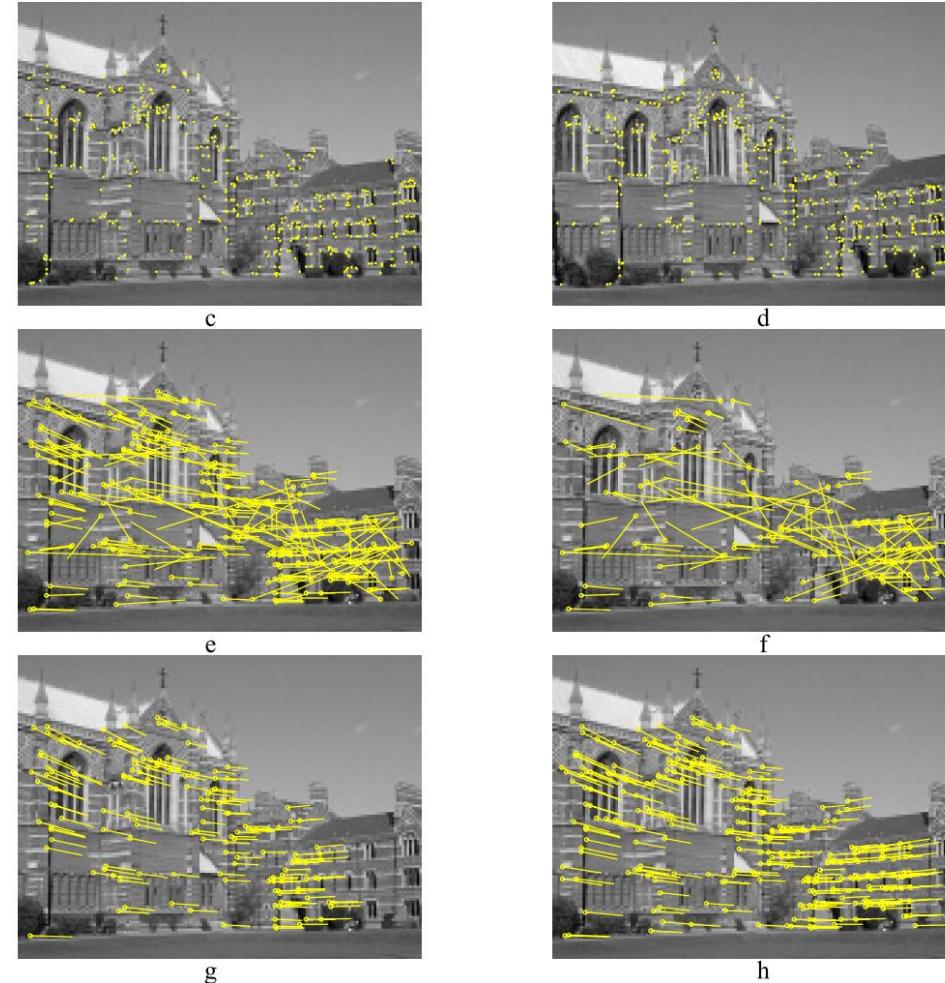
Algorithm

- (i) **Interest points:** Compute interest points in each image.
- (ii) **Putative correspondences:** Compute a set of interest point matches based on proximity and similarity of their intensity neighbourhood.
- (iii) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5:
 - (a) Select a random sample of 4 correspondences and compute the homography H .
 - (b) Calculate the distance d_{\perp} for each putative correspondence.
 - (c) Compute the number of inliers consistent with H by the number of correspondences for which $d_{\perp} < t = \sqrt{5.99} \sigma$ pixels.

Choose the H with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.

- (iv) **Optimal estimation:** re-estimate H from all correspondences classified as inliers, by minimizing the ML cost function (4.8–p95) using the Levenberg–Marquardt algorithm of section A6.2(p600).
- (v) **Guided matching:** Further interest point correspondences are now determined using the estimated H to define a search region about the transferred point position.

The last two steps can be iterated until the number of correspondences is stable.





More Examples – F-matrix

Objective Compute the fundamental matrix between two images.

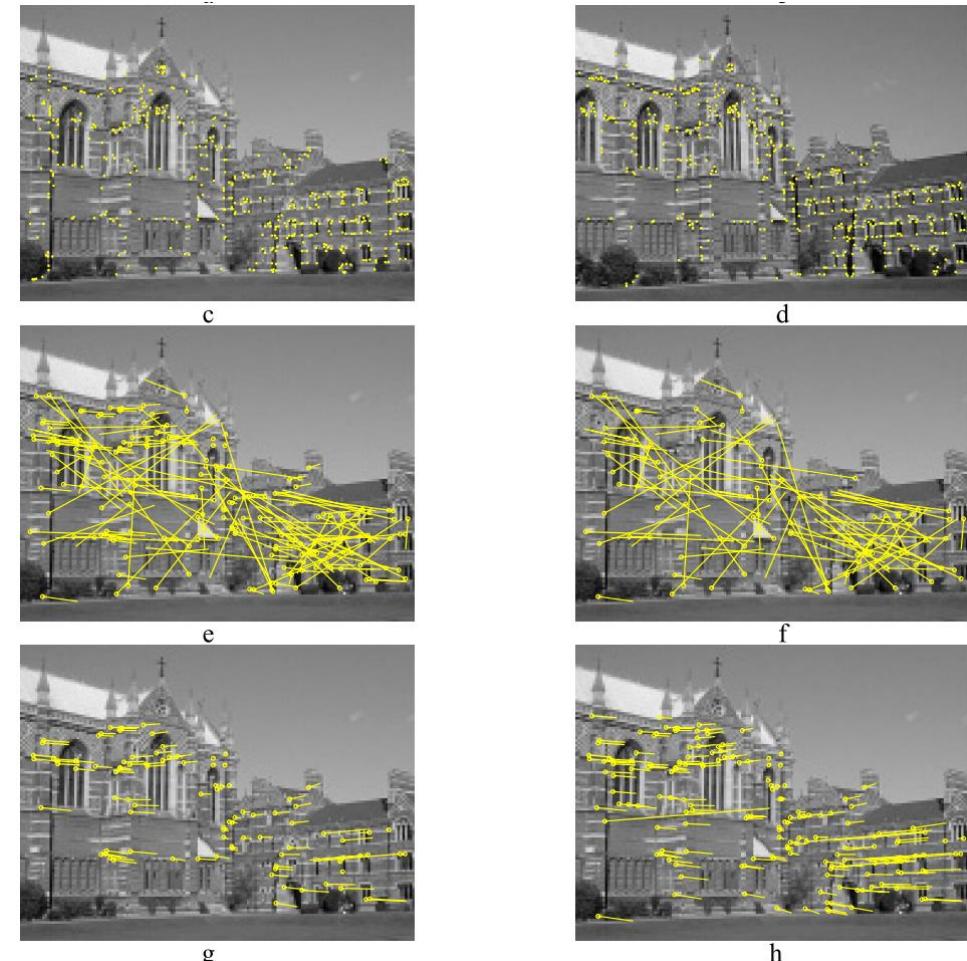
Algorithm

- (i) **Interest points:** Compute interest points in each image.
- (ii) **Putative correspondences:** Compute a set of interest point matches based on proximity and similarity of their intensity neighbourhood.
- (iii) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5(p121):
 - (a) Select a random sample of 7 correspondences and compute the fundamental matrix F as described in section 11.1.2. There will be one or three real solutions.
 - (b) Calculate the distance d_{\perp} for each putative correspondence.
 - (c) Compute the number of inliers consistent with F by the number of correspondences for which $d_{\perp} < t$ pixels.
 - (d) If there are three real solutions for F the number of inliers is computed for each solution, and the solution with most inliers retained.

Choose the F with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.

- (iv) **Non-linear estimation:** re-estimate F from all correspondences classified as inliers by minimizing a cost function, e.g. (11.6), using the Levenberg–Marquardt algorithm of section A6.2(p600).
- (v) **Guided matching:** Further interest point correspondences are now determined using the estimated F to define a search strip about the epipolar line.

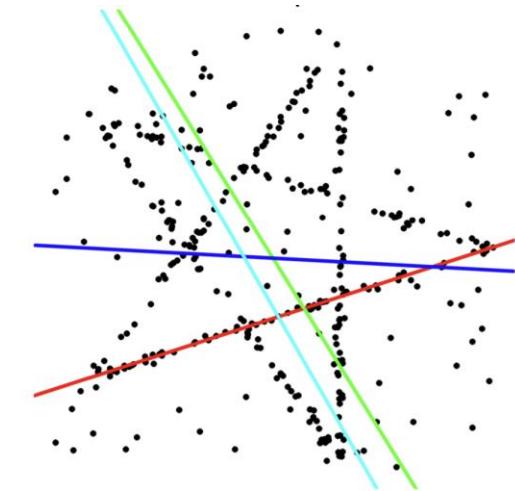
The last two steps can be iterated until the number of correspondences is stable.



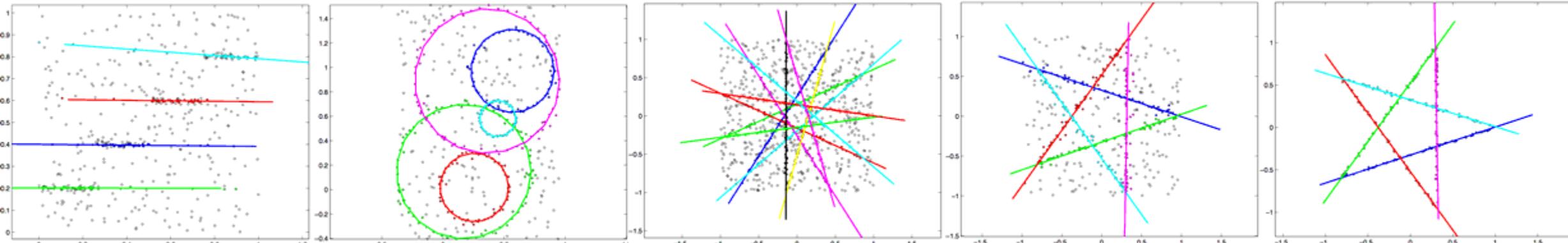


Issues in RANSAC

- RANSAC could be **time-consuming**
 - Too many trials
- RANSAC will fail if the problem is **multi-model**
 - i.e., the data is sampled from multiple models (lines/planes/homography/...)



Ordinary Least Squares (O.L.S.), Total Least Squares (T.L.S.) (via PCA), Least Median of Squares (LMedS), Random Sample Consensus (RANSAC)





The RANSAC Song

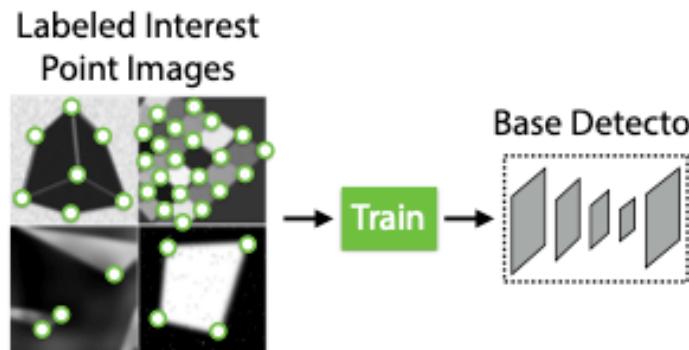


<https://youtu.be/1YNjMxxXO-E>



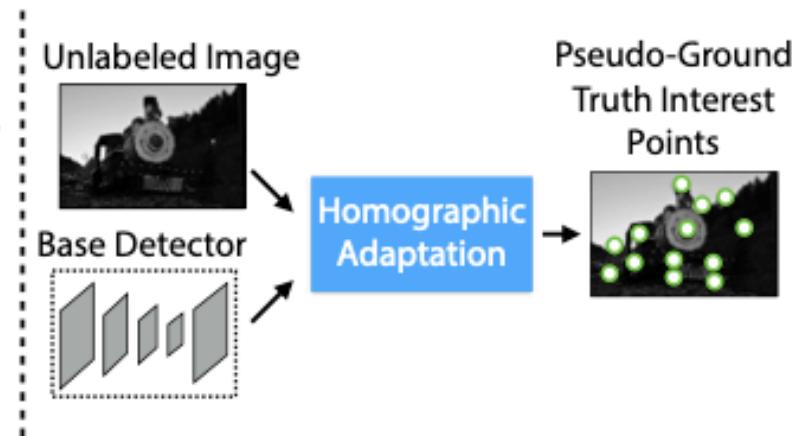
Recent Work with Deep Learning

(a) Interest Point Pre-Training



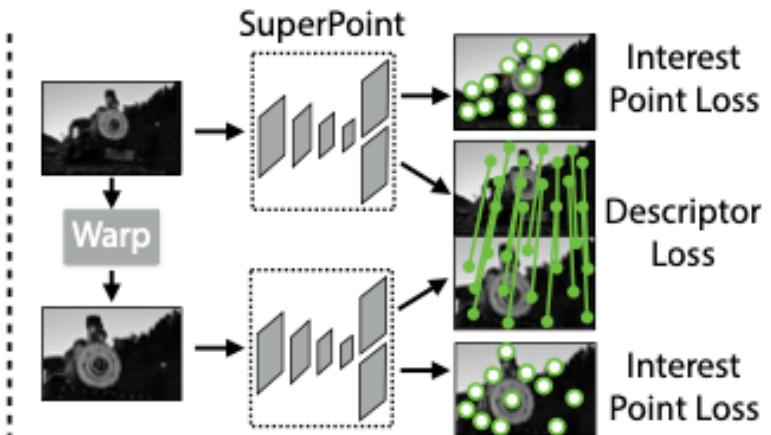
[see Section 4]

(b) Interest Point Self-Labeling



[see Section 5]

(c) Joint Training



[see Section 3]

Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

- **SuperPoint** - fairly “old” work by modern ML standard, but very relevant and is widely used in work on Feature Extraction.
- Built upon an earlier work by the same authors ([MagicPoint](#)) – extended with self-supervised learning + test-time augmentation (they called it Homographic Adaptation).
- Supersedes handcrafted feature extractor like SIFT / SURF / ORB / etc.



Recent Work with Deep Learning

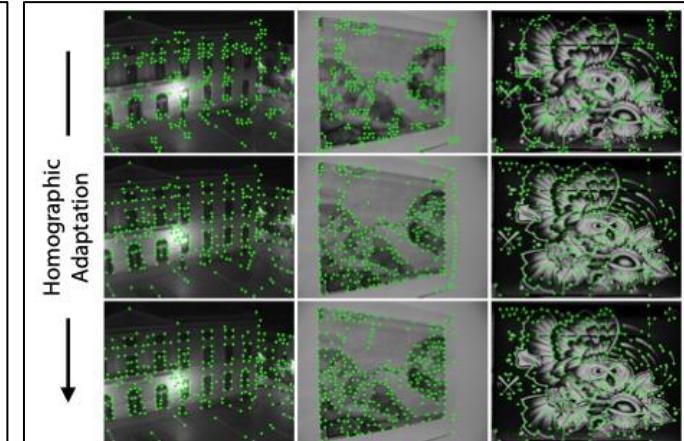
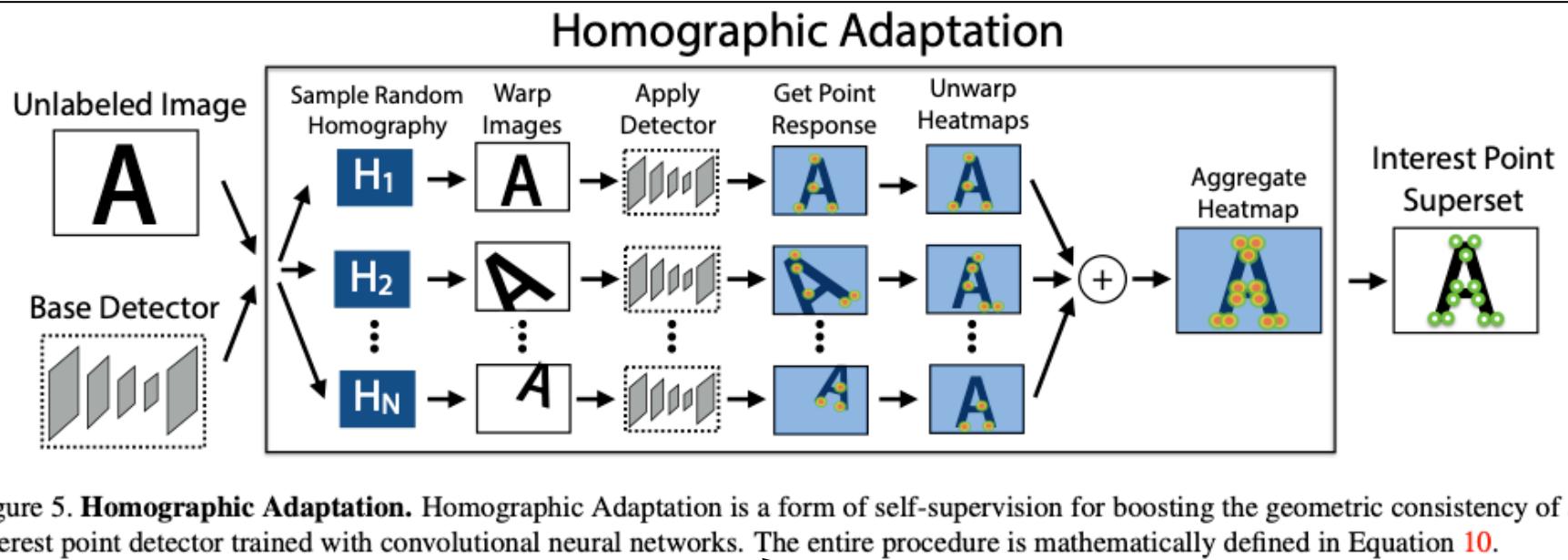
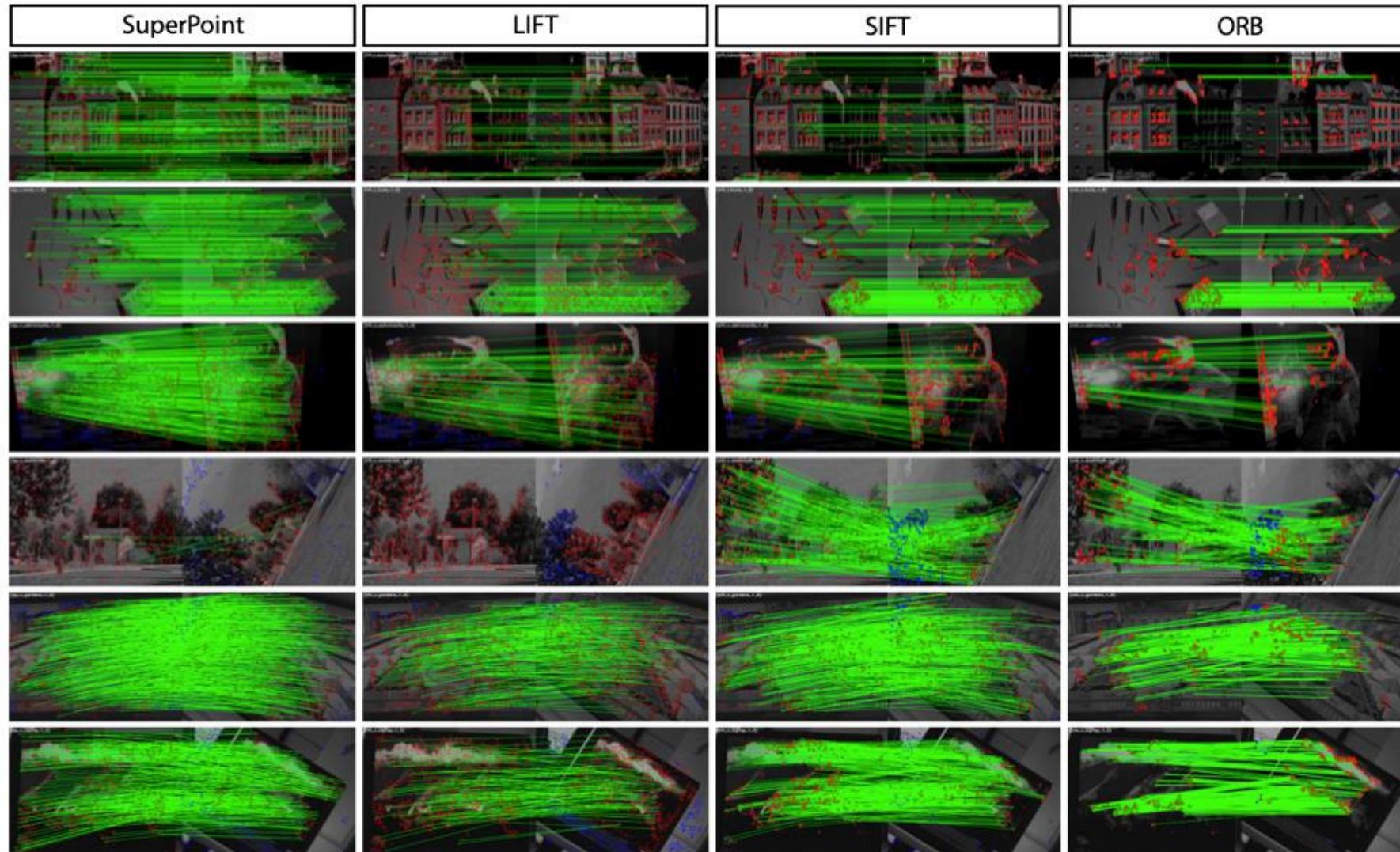


Figure 7. **Iterative Homographic Adaptation**. Top row: initial base detector (MagicPoint) struggles to find repeatable detections. Middle and bottom rows: further training with Homographic Adaption improves detector performance.

- Homographic Adaptation (test-time augmentation) is used to generate a super-set of features via random homography transformation on a source image.
- The super-set of features are then used to train a final classifier (SuperPoint) in a self-supervised manner.
- This self-supervised approach + augmentation enables more robust and consistent detection compared to MagicPoint alone.



Recent Work with Deep Learning





Recent Work with Deep Learning

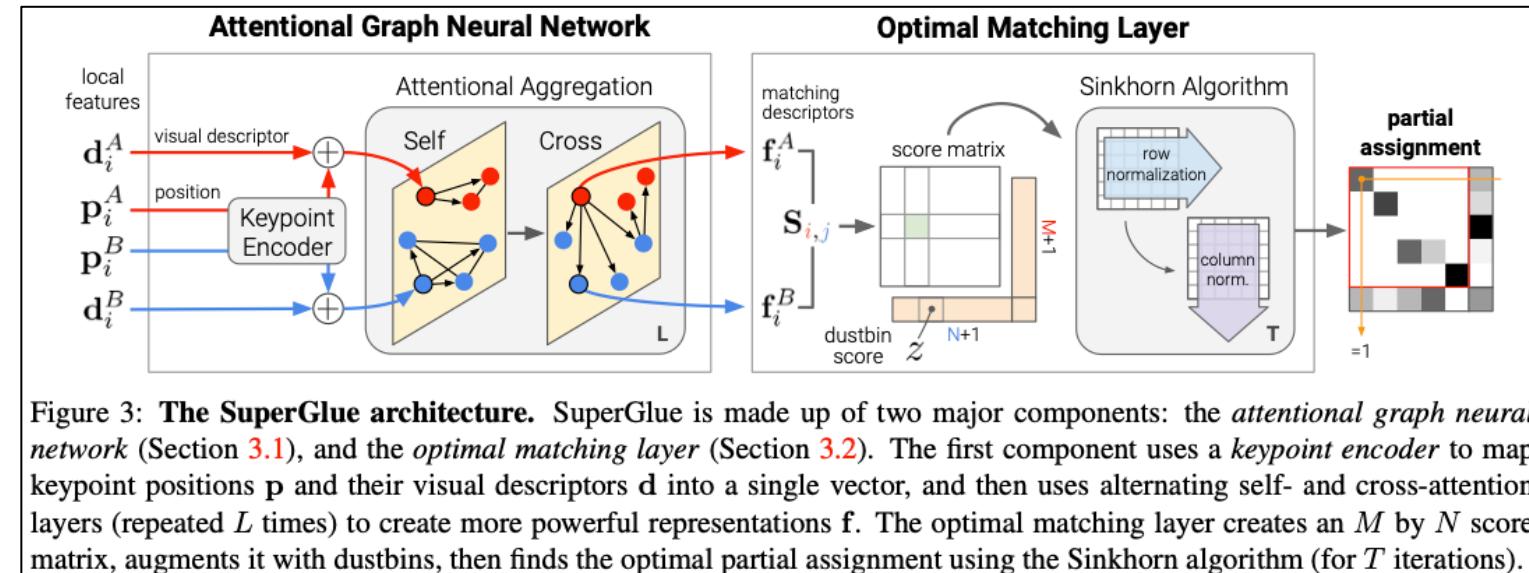
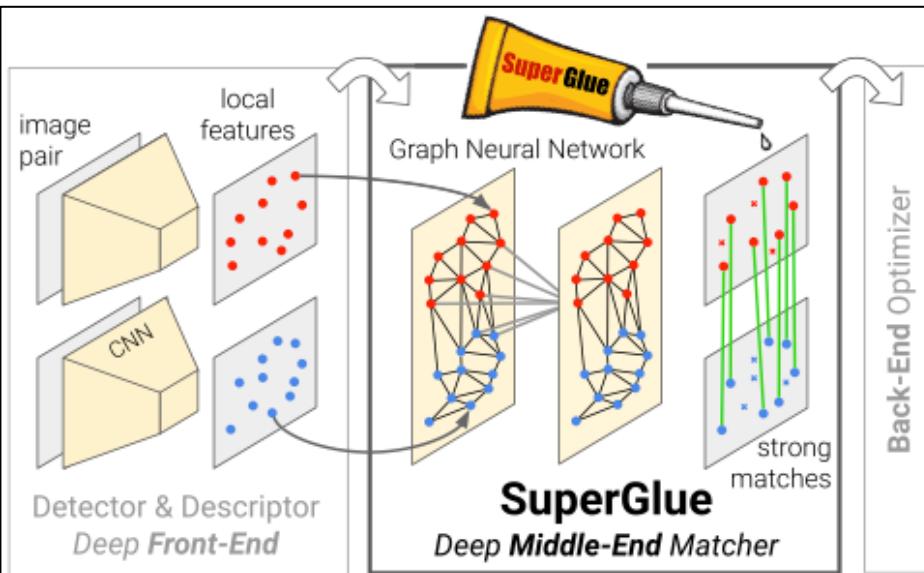


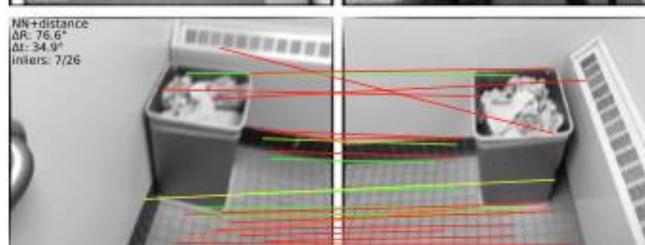
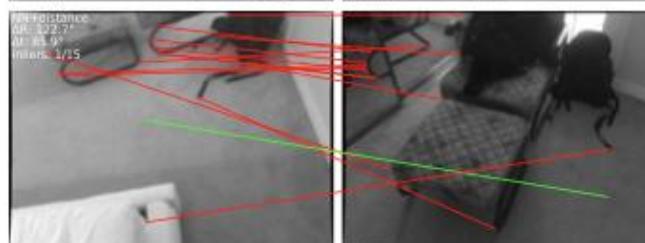
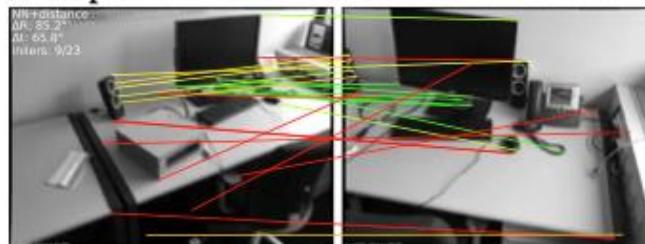
Figure 3: **The SuperGlue architecture.** SuperGlue is made up of two major components: the *attentional graph neural network* (Section 3.1), and the *optimal matching layer* (Section 3.2). The first component uses a *keypoint encoder* to map keypoint positions p and their visual descriptors d into a single vector, and then uses alternating self- and cross-attention layers (repeated L times) to create more powerful representations f . The optimal matching layer creates an M by N score matrix, augments it with dustbins, then finds the optimal partial assignment using the Sinkhorn algorithm (for T iterations).

- **SuperGlue** – robust feature matching via attentional Graph Neural Networks.
- SuperPoint introduces a method to find salient features, SuperGlue introduces a method to find robust correspondences with said features.
- Traditionally, this is usually done by some sort of Nearest Neighbor algorithm, which is prone to outliers, thereby necessitating additional post-processing – e.g. mutual check, Lowe's Ratio Test.
- How does SuperGlue tackle this?
 - By representing it via a Graph Neural Network which renders it end-to-end differentiable, and
 - By leveraging self and cross attentions to better find correspondences in a learnable way.
- Predictably, SuperPoint is used here for Feature Extraction.

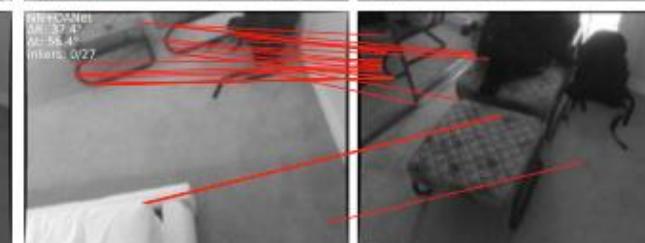
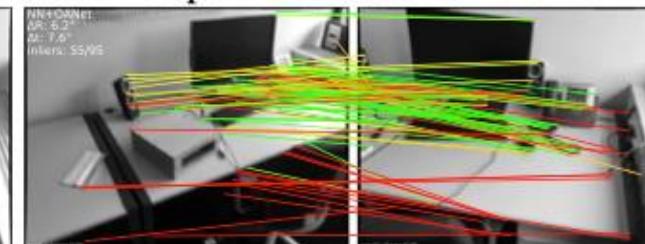


Recent Work with Deep Learning

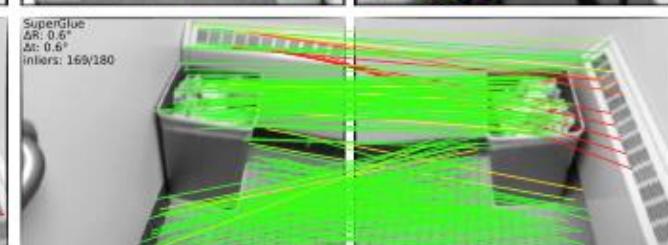
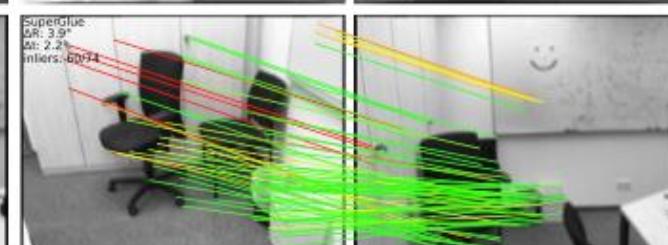
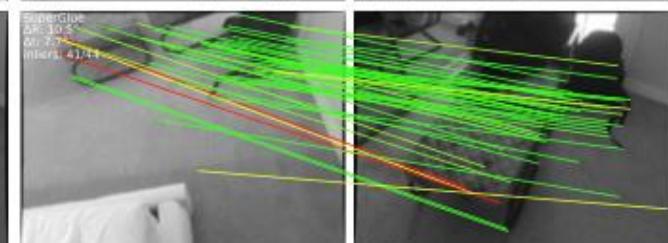
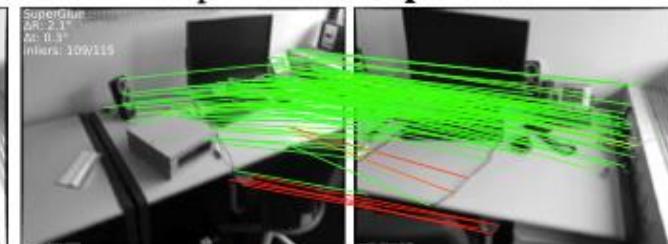
SuperPoint + NN + distance threshold



SuperPoint + NN + OANet

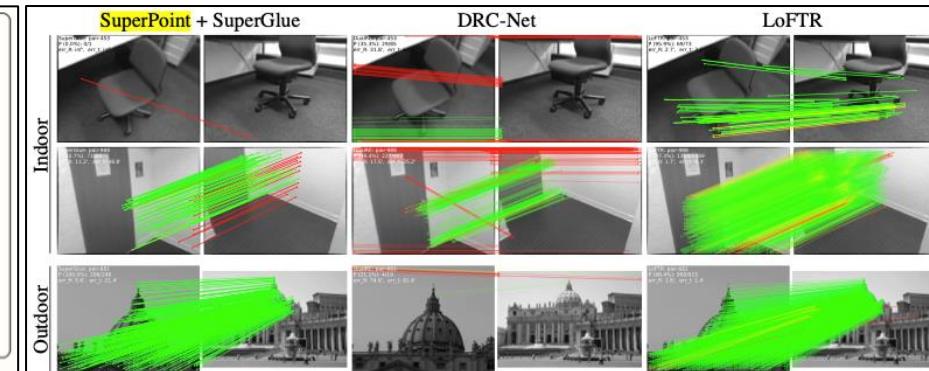
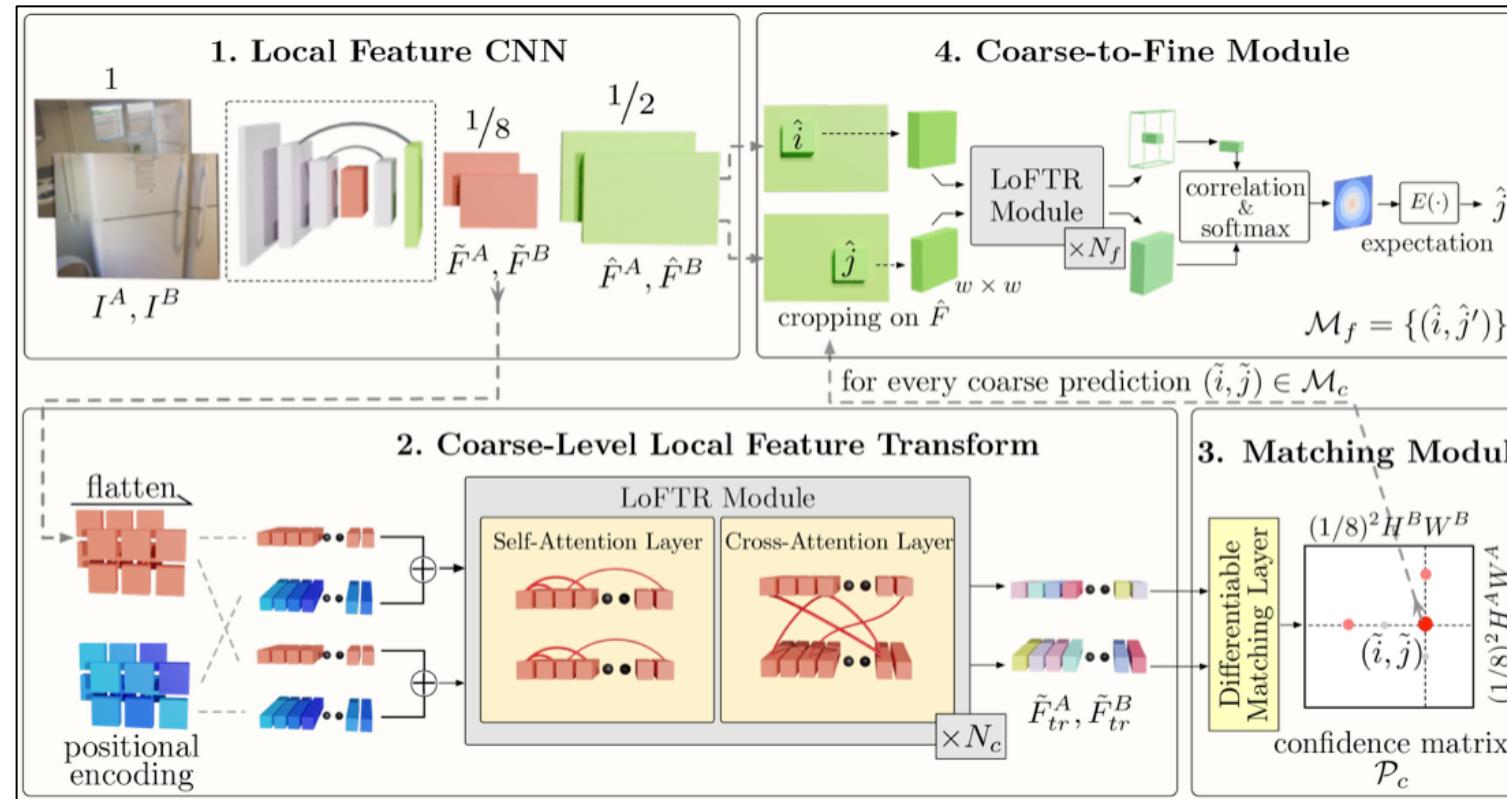


SuperPoint + SuperGlue





Recent Work with Deep Learning



Category	Method	Homography est. AUC			#matches
		@3px	@5px	@10px	
Detector-based	D2Net [11]+NN	23.2	35.9	53.6	0.2K
	R2D2 [32]+NN	50.6	63.9	76.8	0.5K
	DISK [47]+NN	52.3	64.9	78.9	1.1K
	SP [9]+SuperGlue [37]	53.9	68.3	81.7	0.6K
Detector-free	Sparse-NCNet [33]	48.9	54.2	67.1	1.0K
	DRC-Net [19]	50.6	56.2	68.3	1.0K
	LoFTR-DS	65.9	75.6	84.6	1.0K

- **LoFTR** – An “detector-free” Multi-View matching algorithm; think of it as SuperPoint + SuperGlue in one package.
- Utilizes attention as well via Transformers. It’s “detector-free” since there are no explicit feature extraction like SuperPoint.
- Since feature extraction and matching is done end-to-end in a single learnable pipeline, it seems to be able to find better features and correspondences compared to piecewise methods. Also utilizes a coarse-to-fine approach with 2 stages of attentional modules to help achieve this.



Recent Work with Deep Learning

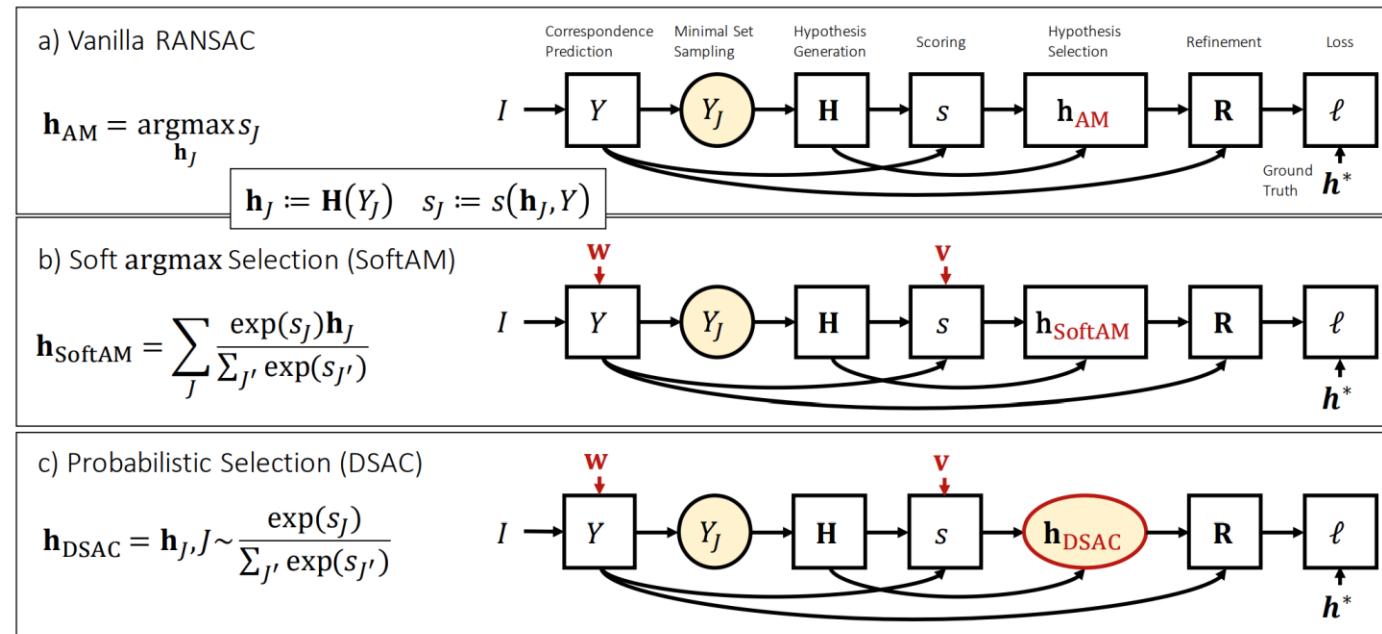
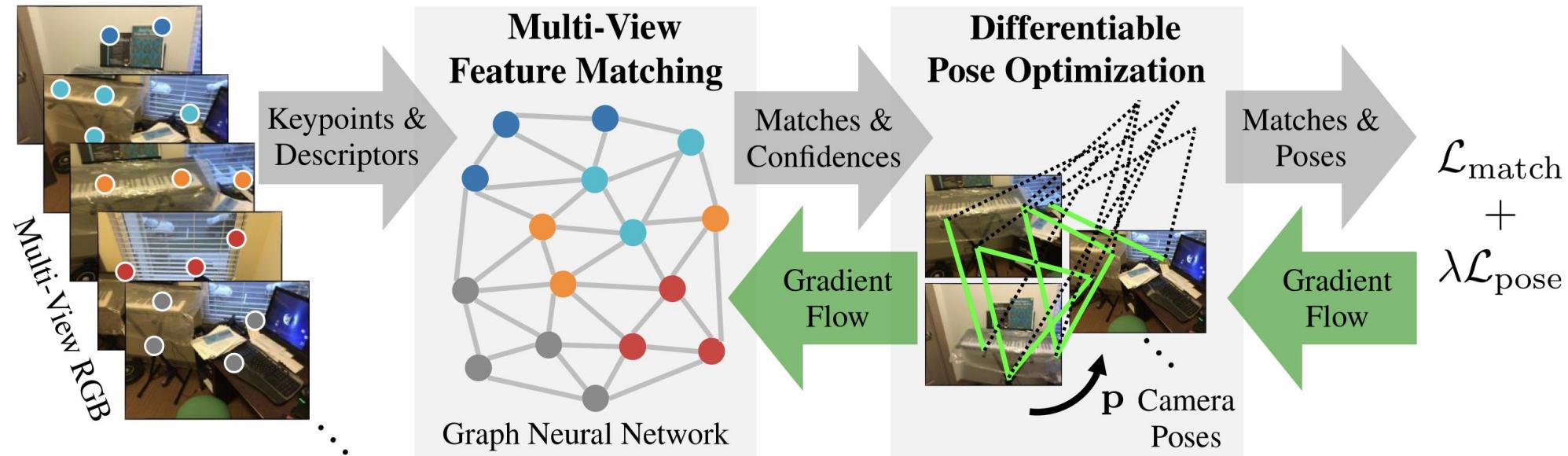


Figure 1. **Stochastic Computation Graphs [34]**. A graphical representation of three RANSAC variants investigated in this work. The variants differ in the way they select the final model hypothesis: **a**) non-differentiable, vanilla RANSAC with hard, deterministic argmax selection; **b**) differentiable RANSAC with deterministic, soft argmax selection; **c**) differentiable RANSAC with hard, probabilistic selection (named DSAC). Nodes shown as boxes represent deterministic functions, while circular nodes with yellow background represent probabilistic functions. Arrows indicate dependency in computation. All differences between a), b) and c) are marked in red.

- RANSAC can be viewed as a stochastic computation graph
- RANSAC itself is not directly differentiable, i.e., the partial derivative of the final loss w.r.t. any intermediate variables before hypothesis selection cannot be computed from backpropagation due to the use of argmax
- The DSAC paper proposed two possible solutions, Soft argmax, or even better, a probabilistic selection inspired by policy gradient approaches in reinforcement learning



Recent Work with Deep Learning



- An end-to-end framework that combines matching + outlier filtering. No RANSAC required. Uses SuperPoint for extraction. (Wait for next team to combine extraction as well for all-in-one!).
- Does so via joint-learning of multi-view matching + pose estimation. The task of pose estimation essentially constraints and direct the network to find proper correspondences across views.
- Uses a Graph Neural Network just like SuperGlue, and introduces a learnable pose-estimation schema for robust matching.



Next Week

- + Depth from Stereo (Stereo Calibration and Stereo Matching)
- + Monocular Depth Estimation by CNN
- * ICP
 - ++ Procrustes Analysis
- + TSDF & Kinect Fusion
- + Fast Plane Extraction (PEAC)

*: know how to code

++: know how to derive

+: know the concept



References for next week

- Co2011:
 - Section 14.3
- Sz2022:
 - Chapter 12
- FP2011
 - Chapter 7, Section 12.1, 14.3
- HZ2003
 - Section 11.12
- Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4), 376-380.
- Low, K.L., 2004. Linear least-squares optimization for point-to-plane icp surface registration. *University of North Carolina Chapel Hill*, 4(10).
- Garg, Ravi, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." In European conference on computer vision, pp. 740-756. Springer, Cham, 2016.
- Feng, C., Taguchi, Y. and Kamat, V.R., (2014). Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6218-6225).
- Park, J., Zhou, Q. Y., & Koltun, V., (2017). Colored point cloud registration revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 143-152).