Deep Features and Matching

3D Vision

University of Illinois

Derek Hoiem

Deep Features and Matching

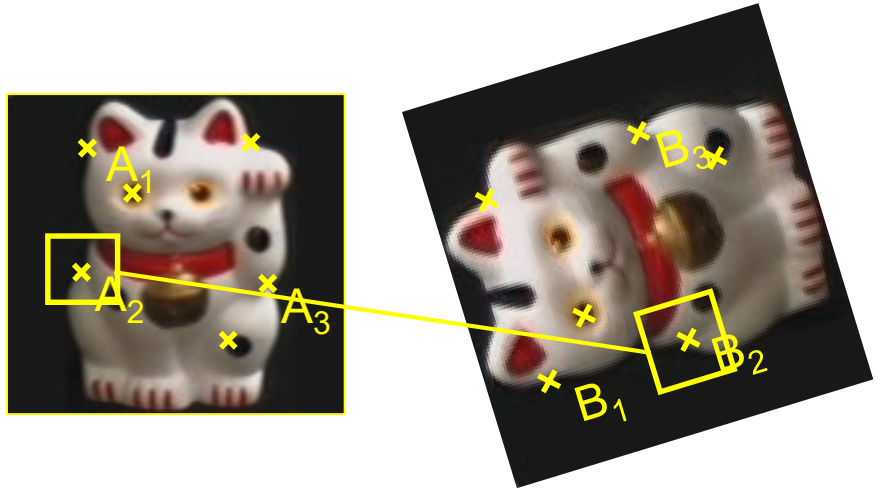
- Review traditional features (SIFT)
- Superpoint
- Superglue
- SfM studies on feature effectiveness

+ How to do research

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Key trade-offs



Detection

More Repeatable

Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

Minimize wrong matches

More Flexible

Robust to expected variations
Maximize correct matches

Many Existing Detectors Available

Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
Laplacian, DoG	[Lindeberg ‘98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid ‘01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid ‘04]
EBR and IBR	[Tuytelaars & Van Gool ‘04]
MSER	[Matas ‘02]
Salient Regions	[Kadir & Brady ‘01]
Others…	

Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

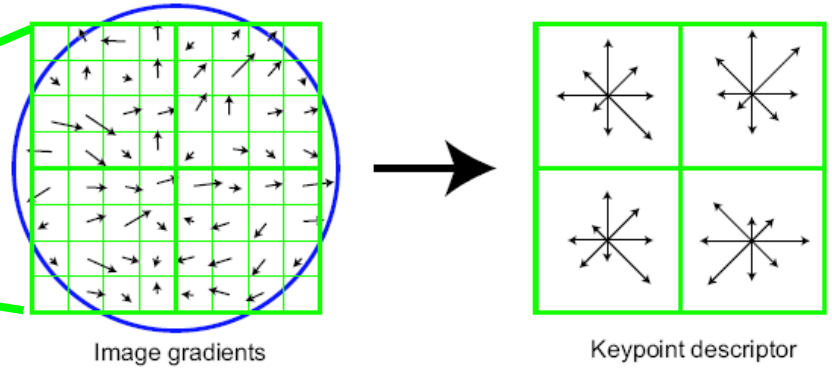
Tuytelaars Mikolajczyk 2008

Local Descriptors

- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient

- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

Local Descriptors: SIFT Descriptor



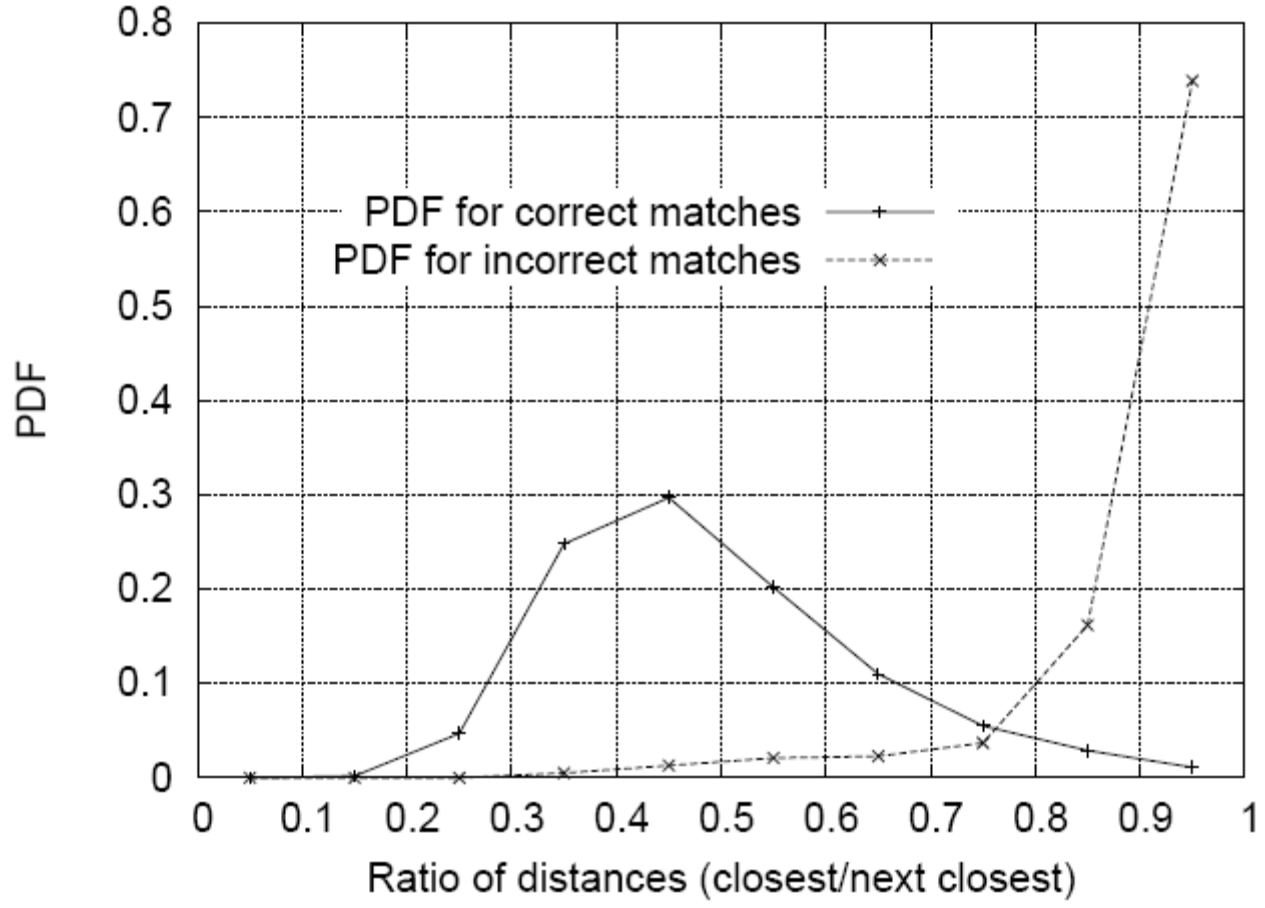
Histogram of oriented gradients

- **Captures important texture information**
- **Robust to small translations / affine deformations**

[Lowe, ICCV 1999]

Matching SIFT Descriptors

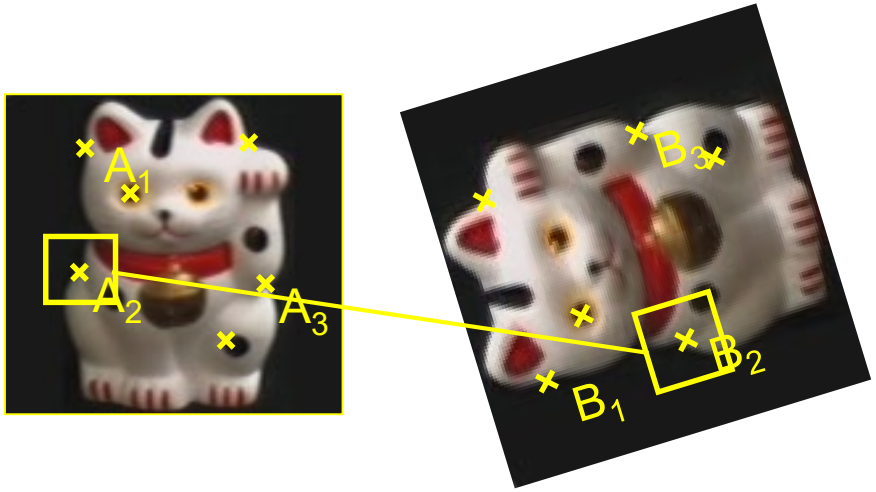
- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor



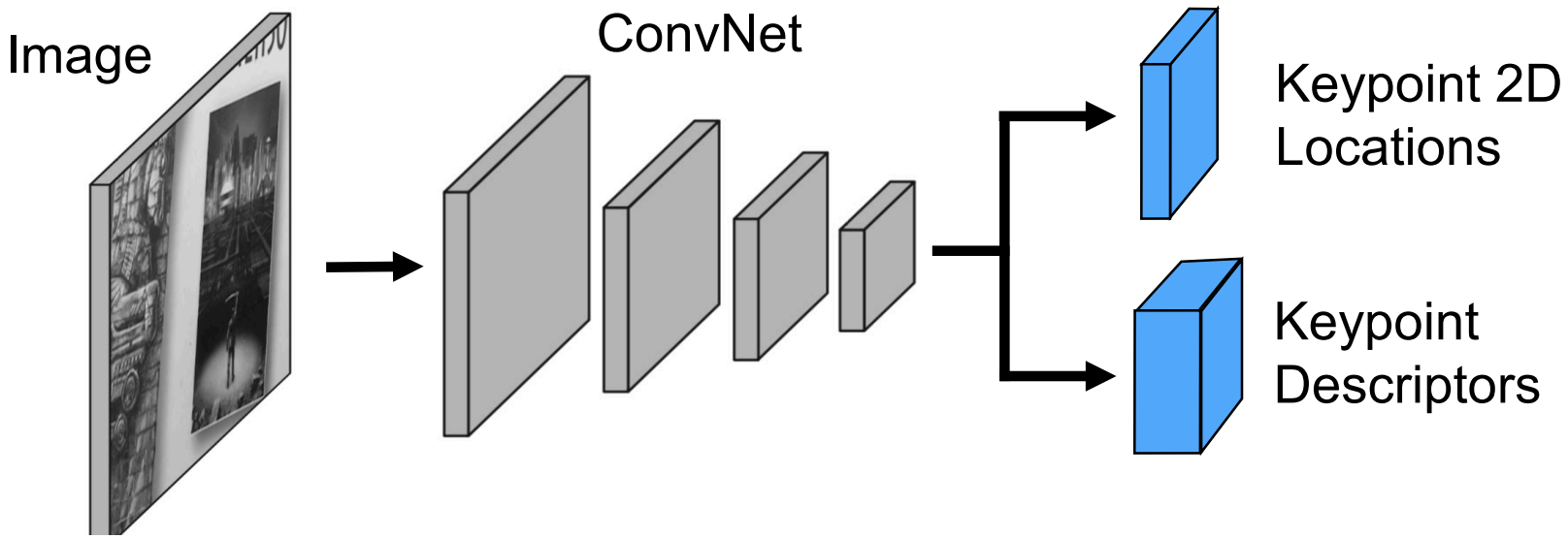
Lowe IJCV 2004

Challenges for learning keypoint detection/representation

- How to get ground truth for keypoints/matches?
- Maintaining precision, given that CNNs typically are low-res feature maps
- Getting diverse enough data for training, or robustness to new data



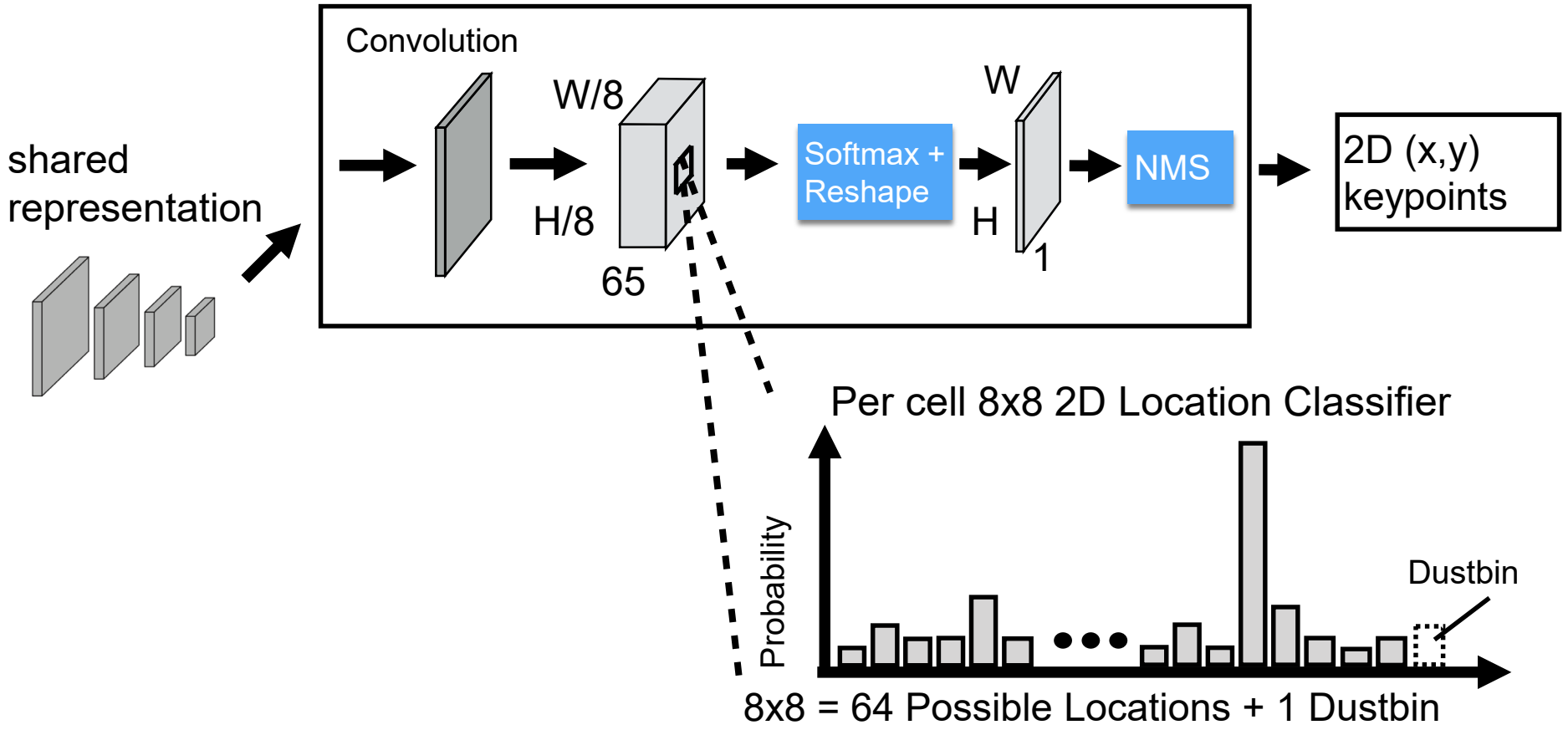
SuperPoint



- Points and descriptors computed jointly
  - Simple
  - Fast

Slide: T. Malisiewicz

Keypoint / Interest Point Decoder

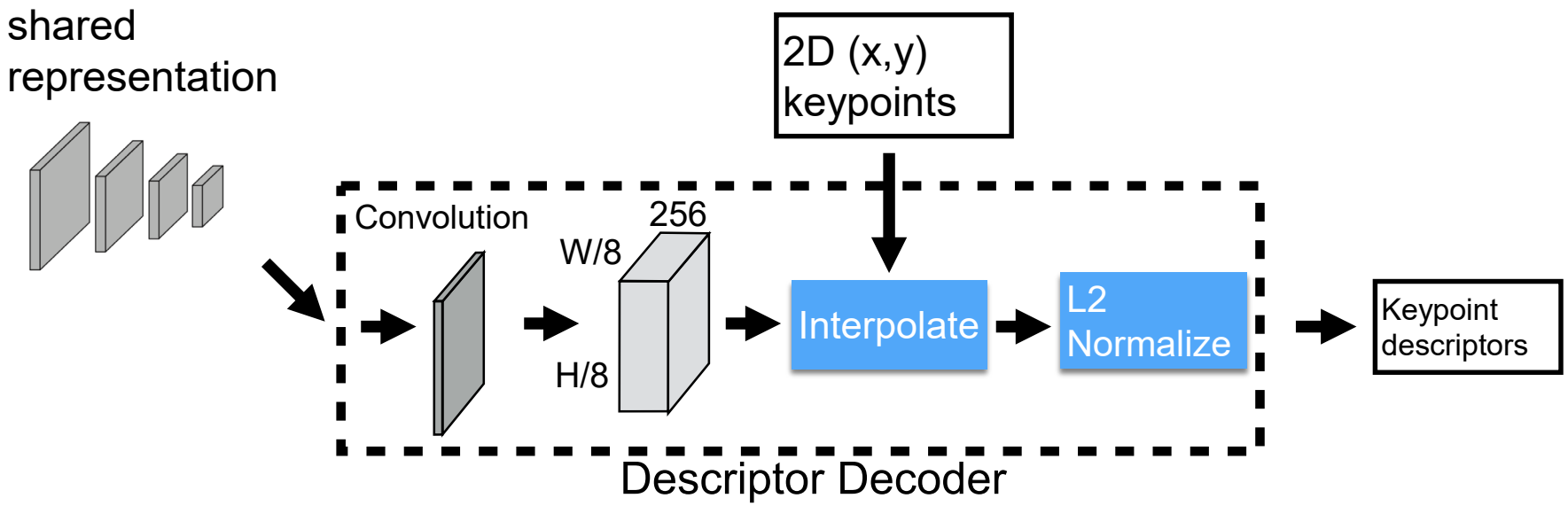


- No deconvolution layers
- Each output cell responsible for local 8x8 region

Slide: T. Malisiewicz

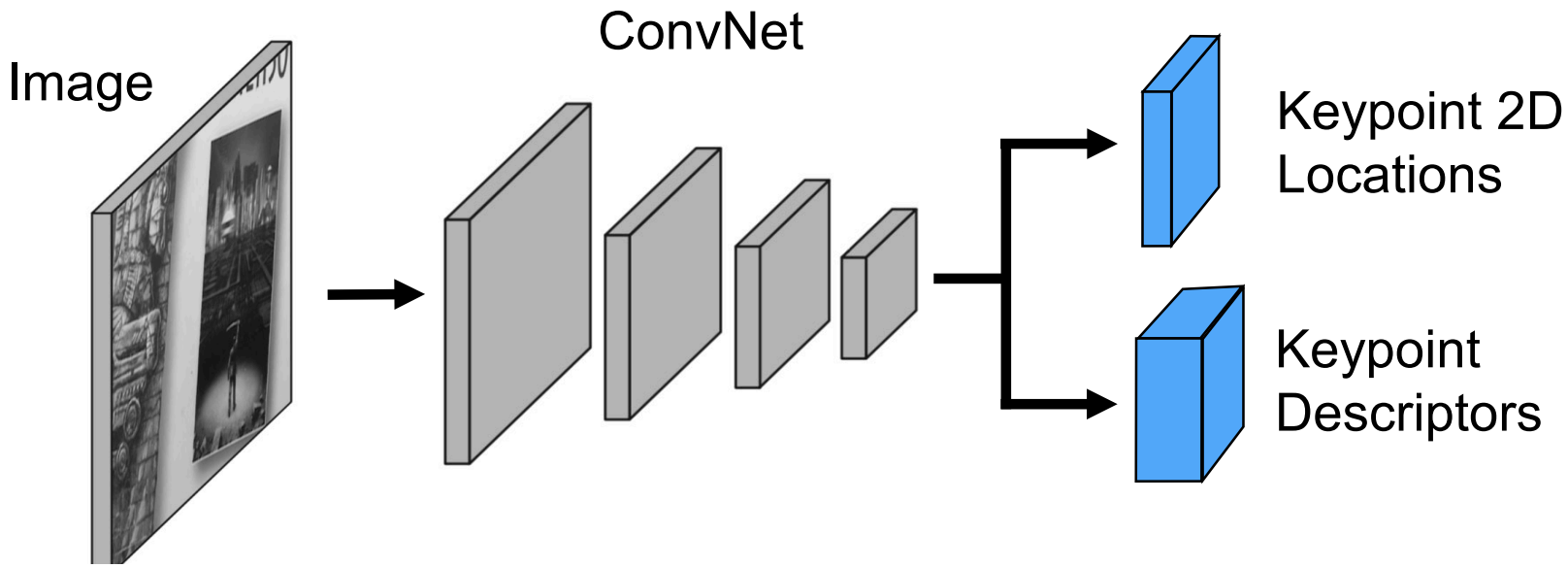
Descriptor Decoder

- Interpolate using 2D keypoint into coarse descriptor map



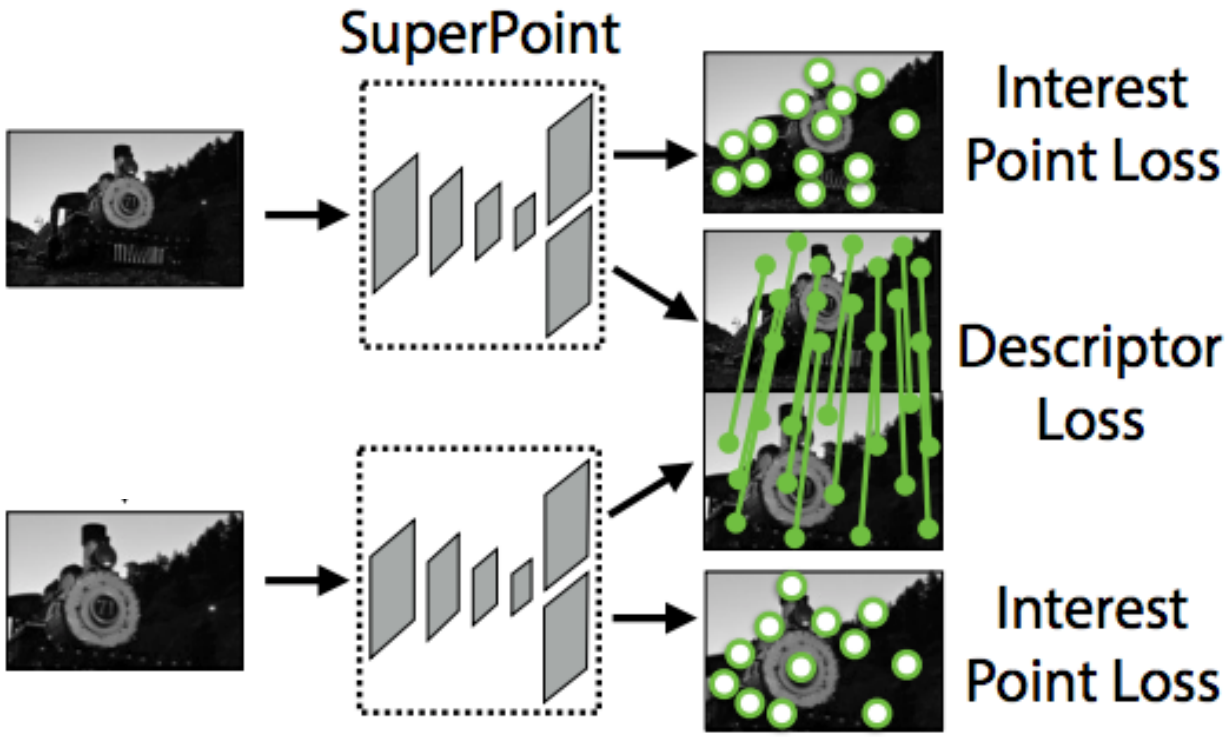
Slide: T. Malisiewicz

How to Train SuperPoint?



Slide: T. Malisiewicz

Setting up the Training



- Siamese training -> pairs of images
- Descriptor trained via metric learning
- Keypoints trained via supervised keypoint labels

Slide: T. Malisiewicz

How to get Keypoint Labels for Natural Images?

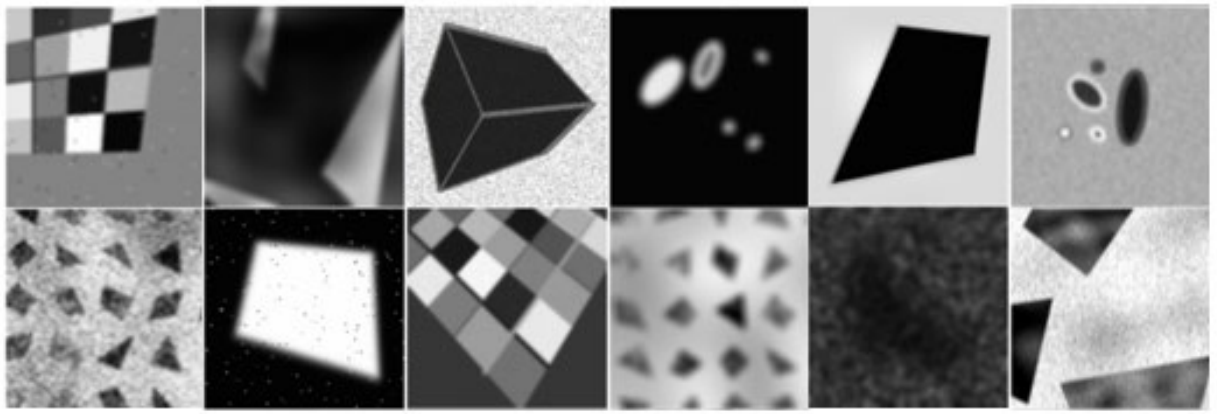
- Need large-scale dataset of annotated images

  – Too hard for humans to label

Slide: T. Malisiewicz

Self-Supervised Approach

Synthetic Shapes (has interest point labels)



First train on this

“Homographic Adaptation”

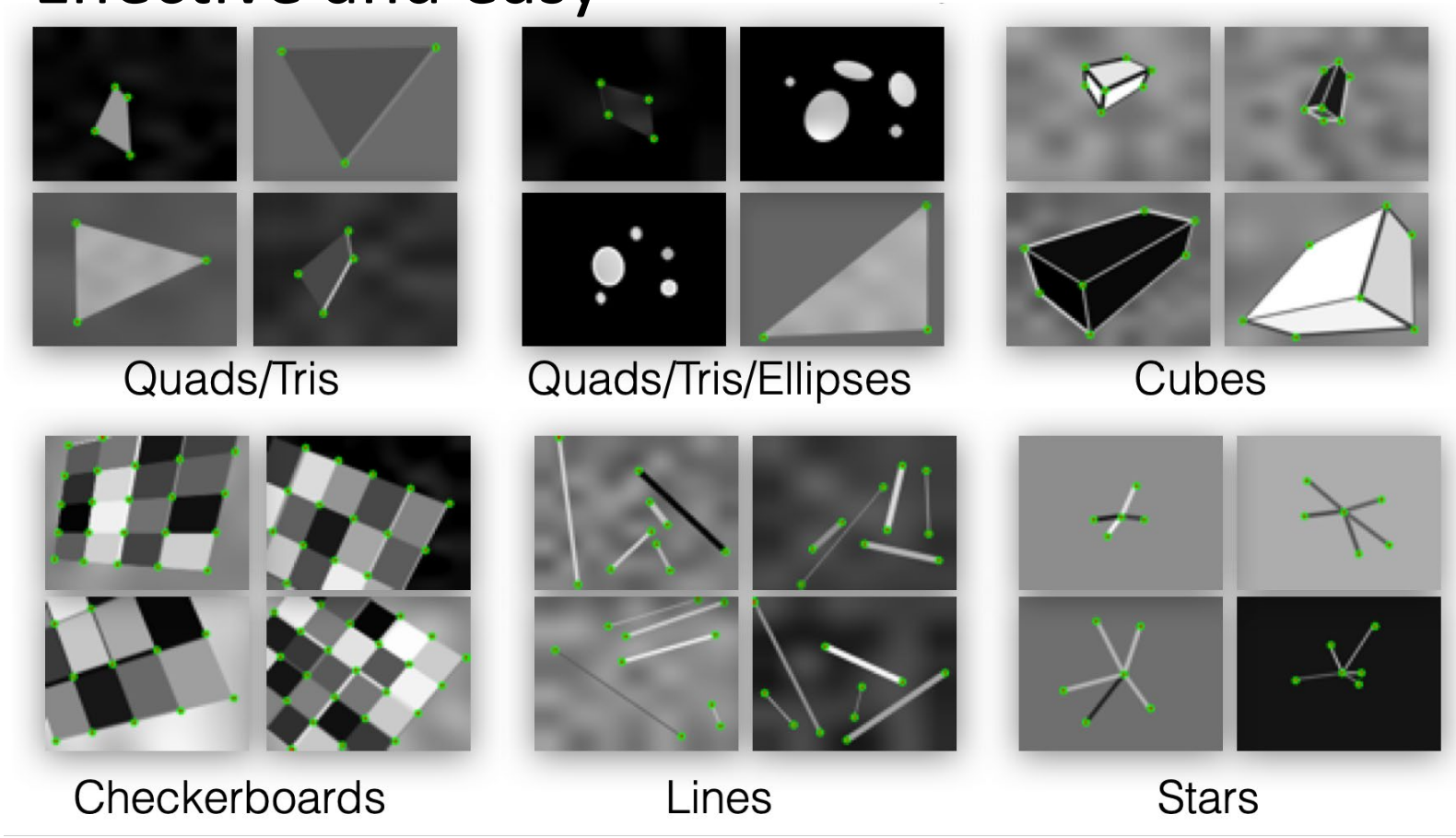
MS-COCO (no interest point labels)

Use resulting detector to label this

Slide: T. Malisiewicz

Synthetic Training

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



Slide: T. Malisiewicz

Training Loss

- Predict which pixel (if any) contains the keypoint in each 8x8 patch – softmax cross-entropy
- Matching points should have similar descriptors; non-matching should be dissimilar – margin-based loss

$$\mathcal{L}(\mathcal{X}, \mathcal{X}', \mathcal{D}, \mathcal{D}'; Y, Y', S) = \mathcal{L}_p(\mathcal{X}, Y) + \mathcal{L}_p(\mathcal{X}', Y') + \lambda \mathcal{L}_d(\mathcal{D}, \mathcal{D}', S)$$

0.0001

$$\mathcal{L}_p(\mathcal{X}, Y) = \frac{1}{H_c W_c} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} l_p(\mathbf{x}_{hw}; y_{hw}),$$

where

$$l_p(\mathbf{x}_{hw}; y) = -\log\left(\frac{\exp(\mathbf{x}_{hwy})}{\sum_{k=1}^{65}\exp(\mathbf{x}_{hwk})}\right).$$

$$\mathcal{L}_d(\mathcal{D}, \mathcal{D}', S) = \frac{1}{(H_c W_c)^2} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} \sum_{\substack{h'=1 \\ w'=1}}^{H_c, W_c} l_d(\mathbf{d}_{hw}, \mathbf{d}'_{h'w'}; s_{hwh'w'}),$$

where

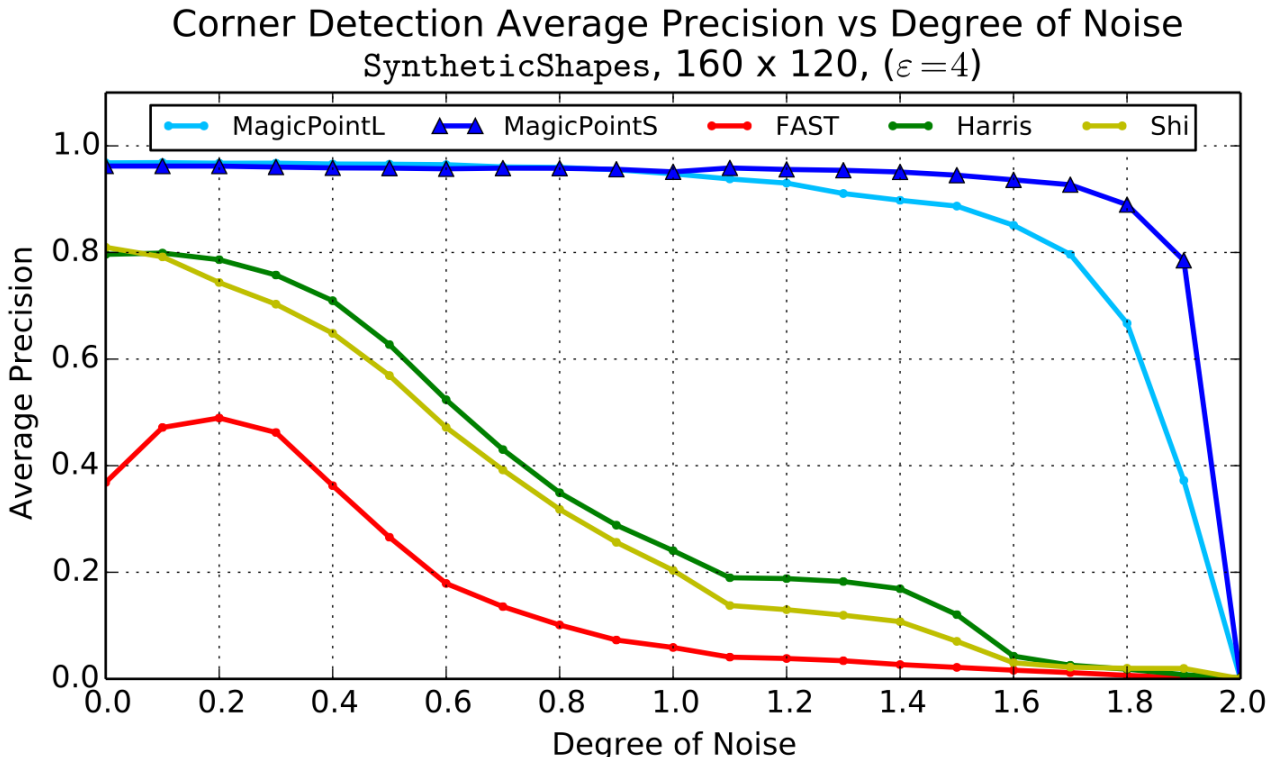
250, 1

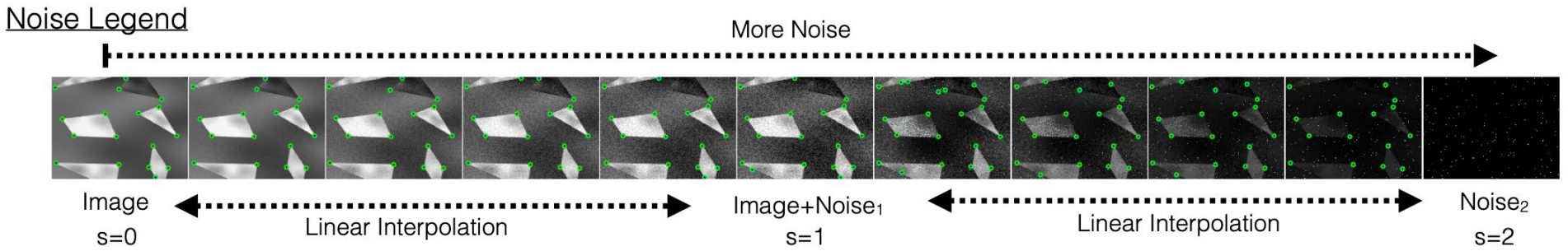
$$l_d(\mathbf{d}, \mathbf{d}'; s) = \lambda_d * s * \max(0, m_p - \mathbf{d}^T\mathbf{d}') + (1-s) * \max(0, \mathbf{d}^T\mathbf{d}' - m_n).$$

0.2

Early Version of SuperPoint (MagicPoint)

"Toward Geometric Deep SLAM"
DeTone et. al. 2017





Slide: T. Malisiewicz

Generalizing to Real Data

- Worked well on geometric structures, but didn’t work very well in general
- Solution: self-train on natural images

Unlabeled
Input Image

Synthetic Warp +
Run Detector

Homographic Adaptation

Point Set #1

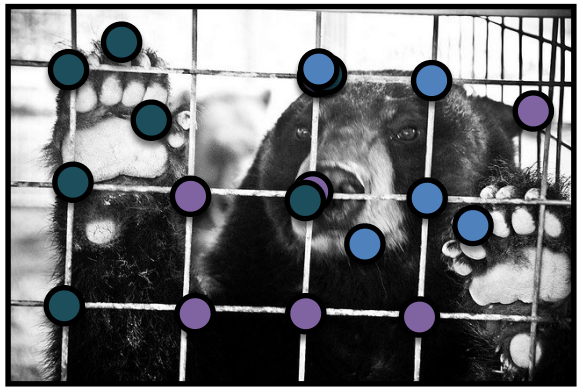
Point Set #2

Point Set #3

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

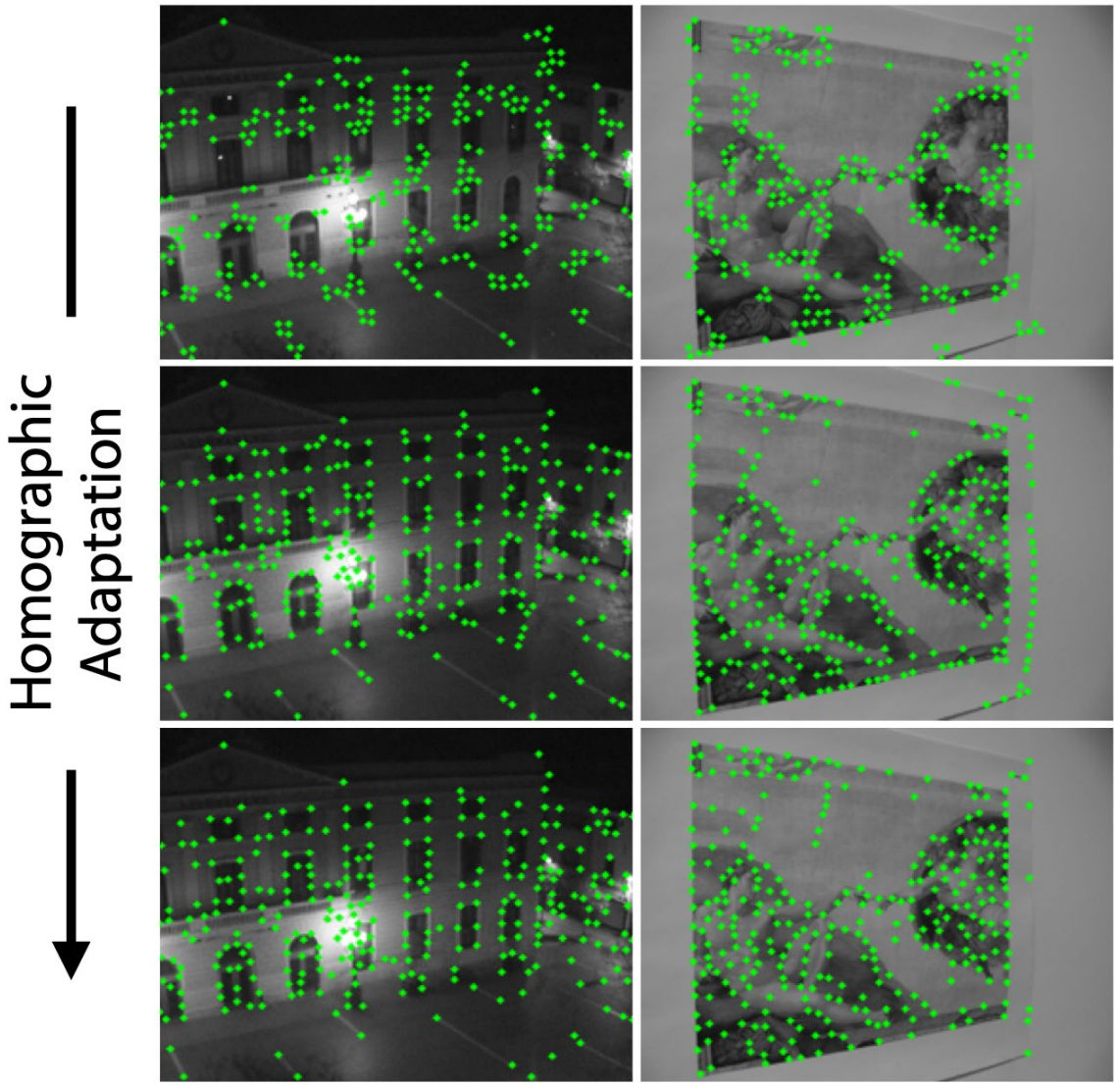
Point Aggregation

Detected Point Superset



Slide: T. Malisiewicz

Iterative Homographic Adaptation



- Label, train, repeat, …
- Resulting points:
  - Higher coverage
  - More repeatable

Slide: T. Malisiewicz

Training details

- Pretrain on synthetic images for 200000 iterations
- Generate 100 random homographies in 240x320 COCO images and pool keypoints
- Train to predict these pooled keypoints on original image

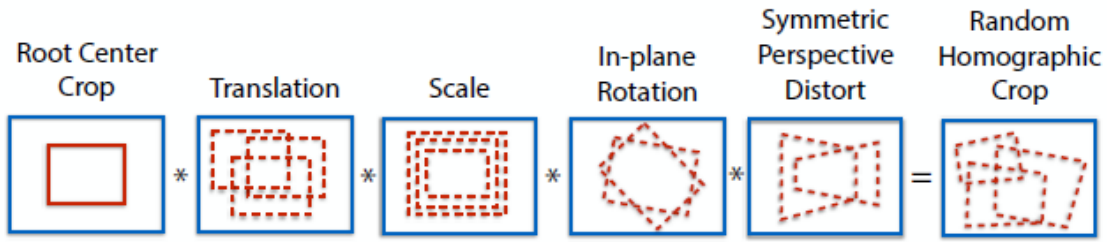


Figure 6. **Random Homography Generation.** We generate random homographies as the composition of less expressive, simple transformations.

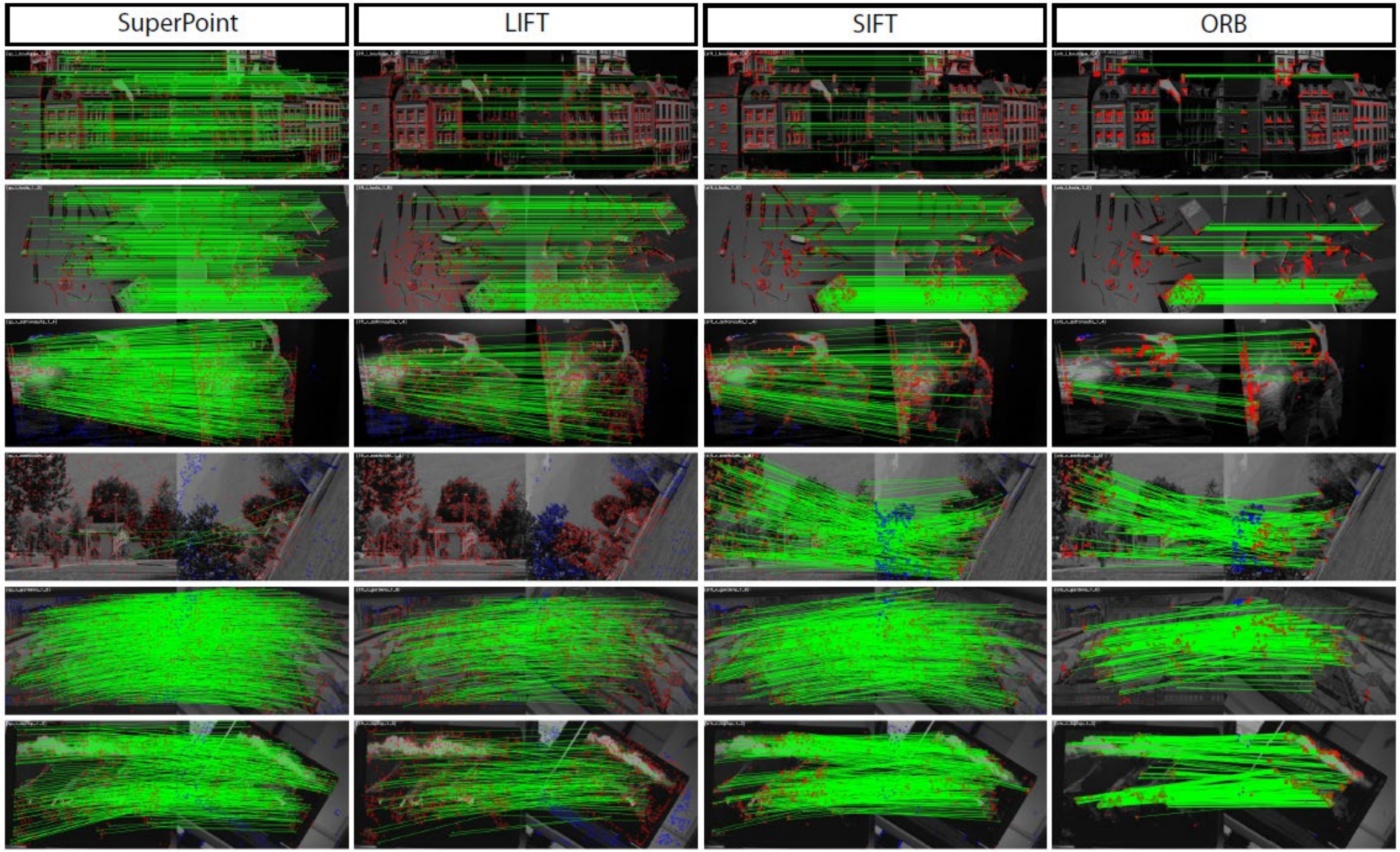
HPatches Evaluation

- Homography estimation task
- Dataset of 116 scenes each with 6 images = 696 images
- Indoor and outdoor planar scenes
- Compared against LIFT, SIFT and ORB

50% of dataset:
Illumination
Change

50% of dataset:
Viewpoint
Change

Slide: T. Malisiewicz


SuperPoint
LIFT
SIFT
ORB

	57 Illumination Scenes		59 Viewpoint Scenes	
	NMS=4	NMS=8	NMS=4	NMS=8
SuperPoint	**.652**	**.631**	.503	**.484**
MagicPoint	.575	.507	.322	.260
FAST	.575	.472	.503	.404
Harris	.620	.533	**.556**	.461
Shi	.606	.511	.552	.453
Random	.101	.103	.100	.104

Table 3. **HPatches Detector Repeatability**. SuperPoint is the most repeatable under illumination changes, competitive on viewpoint changes, and outperforms MagicPoint in all scenarios.

	Homography Estimation			Detector Metrics		Descriptor Metrics	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Rep.	MLE	NN mAP	M. Score
SuperPoint	.310	**.684**	**.829**	.581	1.158	**.821**	**.470**
LIFT	.284	.598	.717	.449	1.102	.664	.315
SIFT	**.424**	.676	.759	.495	**0.833**	.694	.313
ORB	.150	.395	.538	**.641**	1.157	.735	.266

Table 4. **HPatches Homography Estimation.** SuperPoint outperforms LIFT and ORB and performs comparably to SIFT using various ϵ thresholds of correctness. We also report related metrics which measure detector and descriptor performance individually.

Timing SuperPoint vs LIFT

- Speed important for low-compute Visual SLAM
  - SuperPoint total 640x480 time: ~ 33 ms
  - LIFT total 640x480 time: ~2 minutes

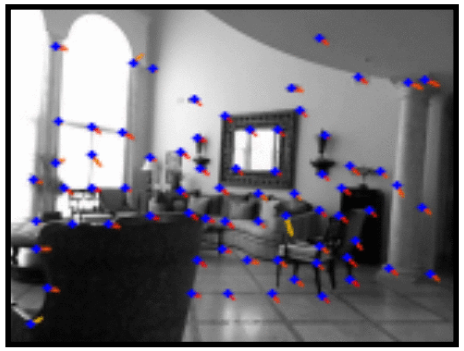
Slide: T. Malisiewicz

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!

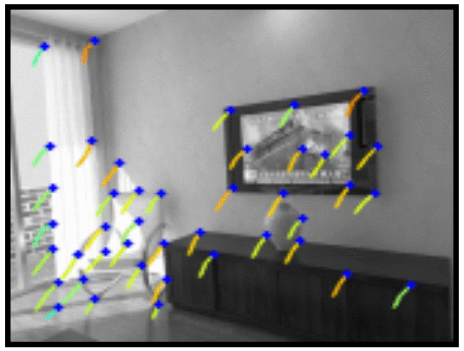
Freiburg (Kinect)

NYU (Kinect)



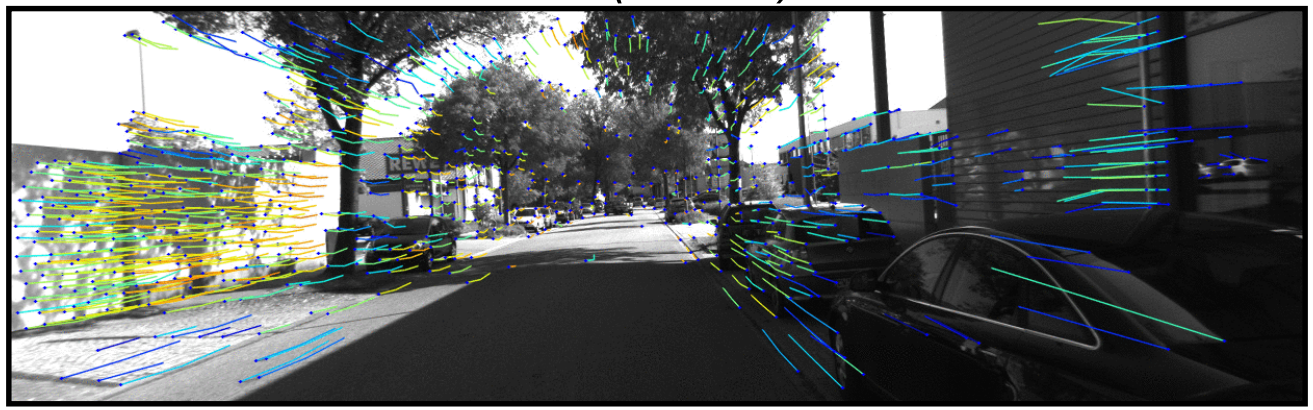
MonoVO (fisheye)

ICL-NUIM (synth)



MS7 (Kinect)

KITTI (stereo)



Slide: T. Malisiewicz

Do learned features actually work better for SfM/SLAM?

Comparative Evaluation of Hand-Crafted and Learned Local Features

Johannes L. Schönberger[1] Hans Hardmeier[1] Torsten Sattler[1] Marc Pollefeys[1,2]
[1] Department of Computer Science, ETH Zürich [2] Microsoft Corp.
{jsch,harhans,sattlert,pomarc}@inf.ethz.ch

2017

"Advanced hand-crafted features still perform on par or better than recent learned features in the practical context of image-based reconstruction."

"The current generation of learned descriptors shows a high variance across different datasets and applications."

		# Images	# Registered	# Sparse Points	# Observations	Track Length	Reproj. Error	# Inlier Pairs	# Inlier Matches	# Dense Points	Pose Error	Dense Error
Fountain	*SIFT*	11	11	10,004	44K	4.49	**0.30px**	49	76K	2,970K	0.002m (0.002m)	0.77 (0.90)
	SIFT-PCA		11	**14,608**	**70K**	**4.80**	0.39px	**55**	**124K**	**3,021K**	0.002m (0.002m)	0.77 (0.90)
	DSP-SIFT		11	**14,785**	**71K**	**4.80**	0.41px	54	**129K**	2,999K	0.002m (0.002m)	0.77 (0.90)
	ConvOpt		11	**14,179**	**67K**	**4.75**	0.37px	**55**	**114K**	**2,999K**	0.002m (0.002m)	0.77 (0.90)
	DeepDesc		11	13,519	61K	4.55	**0.35px**	**55**	93K	2,972K	0.002m (0.002m)	0.77 (0.90)
	TFeat		11	13,696	64K	4.68	**0.35px**	54	103K	2,969K	0.002m (0.002m)	0.77 (0.90)
	LIFT		11	10,172	46K	4.55	0.59px	**55**	83K	**3,019K**	0.002m (0.002m)	0.77 (0.90)
Herzjesu	*SIFT*	8	8	4,916	19K	4.00	**0.32px**	27	28K	2,373K	0.004m (0.004m)	0.57 (0.73)
	SIFT-PCA		8	**7,433**	**31K**	**4.19**	0.42px	**28**	**47K**	2,372K	0.004m (0.004m)	0.57 (0.73)
	DSP-SIFT		8	**7,760**	**32K**	**4.19**	0.45px	**28**	**50K**	**2,376K**	0.004m (0.004m)	0.57 (0.73)
	ConvOpt		8	6,939	28K	**4.13**	0.40px	**28**	42K	2,375K	0.004m (0.004m)	0.57 (0.73)
	DeepDesc		8	6,418	25K	3.92	**0.38px**	**28**	34K	**2,380K**	0.004m (0.004m)	0.57 (0.73)
	TFeat		8	6,606	27K	4.09	**0.38px**	**28**	38K	**2,377K**	0.004m (0.004m)	0.57 (0.73)
	LIFT		8	**7,834**	**30K**	3.95	0.63px	**28**	**46K**	2,375K	0.004m (0.004m)	0.57 (0.73)
South Building	*SIFT*	128	128	62,780	353K	5.64	**0.42px**	1K	1,003K	1,972K	–	–
	SIFT-PCA		128	**107,674**	**650K**	**6.04**	0.54px	**3K**	**2,019K**	1,993K	–	–
	DSP-SIFT		128	**110,394**	**664K**	**6.02**	0.57px	3K	**2,079K**	**1,994K**	–	–
	ConvOpt		128	**103,602**	**617K**	5.96	0.51px	**4K**	**1,856K**	**2,007K**	–	–
	DeepDesc		128	101,154	558K	5.53	**0.48px**	**6K**	1,463K	**2,002K**	–	–
	TFeat		128	94,589	566K	**5.99**	**0.49px**	3K	1,567K	1,960K	–	–
	LIFT		128	74,607	399K	5.35	0.78px	3K	1,168K	1,975K	–	–
Madrid Metropolis	*SIFT*	1,344	440	62,729	416K	**6.64**	**0.53px**	14K	1,740K	435K	–	–
	SIFT-PCA		**465**	**119,244**	**702K**	5.89	0.57px	**27K**	**3,597K**	**537K**	–	–
	DSP-SIFT		**476**	**107,028**	**681K**	**6.36**	0.64px	**21K**	**3,155K**	**570K**	–	–
	ConvOpt		**455**	**115,134**	**634K**	5.51	0.57px	**29K**	**3,148K**	**561K**	–	–
	DeepDesc		377	68,110	348K	5.11	**0.53px**	19K	1,570K	516K	–	–
	TFeat		439	90,274	512K	5.68	**0.54px**	18K	2,135K	522K	–	–
	LIFT		430	52,755	337K	**6.40**	0.76px	13K	1,498K	450K	–	–
Gendarmenmarkt	*SIFT*	1,463	950	169,900	1,010K	**5.95**	**0.64px**	28K	3,292K	1,104K	–	–
	SIFT-PCA		**953**	272,118	**1,477K**	**5.43**	0.69px	**43K**	**5,137K**	1,240K	–	–
	DSP-SIFT		**975**	**321,846**	**1,732K**	**5.38**	0.74px	**56K**	**7,648K**	**1,505K**	–	–
	ConvOpt		945	**341,591**	**1,601K**	4.69	0.70px	**56K**	**6,525K**	**1,342K**	–	–
	DeepDesc		809	244,925	949K	3.88	**0.68px**	31K	2,849K	921K	–	–
	TFeat		**953**	**297,266**	1,445K	4.86	**0.66px**	39K	4,685K	1,181K	–	–
	LIFT		942	180,746	964K	5.34	0.83px	27K	2,495K	**1,386K**	–	–
Tower of London	*SIFT*	1,576	702	142,746	963K	6.75	**0.53px**	18K	3,211K	1,126K	–	–
	SIFT-PCA		692	137,800	1,090K	**7.91**	0.60px	12K	2,455K	1,124K	–	–
	DSP-SIFT		**755**	**236,598**	**1,761K**	**7.44**	0.64px	**33K**	**8,056K**	**1,143K**	–	–
	ConvOpt		**719**	**274,987**	**1,732K**	6.30	0.62px	**39K**	**7,542K**	**1,129K**	–	–
	DeepDesc		551	196,990	964K	4.90	**0.55px**	25K	2,745K	653K	–	–
	TFeat		714	**206,142**	**1,424K**	6.91	**0.57px**	**28K**	**5,333K**	**1,182K**	–	–
	LIFT		**715**	147,851	1,045K	**7.07**	0.72px	23K	4,079K	729K	–	–
Alamo	*SIFT*	2,915	743	120,713	1,384K	11.47	0.54px	**23K**	**7,671K**	**611K**	–	–
	SIFT-PCA		**746**	108,553	1,377K	**12.69**	0.55px	12K	4,669K	564K	–	–
	DSP-SIFT		**754**	**144,341**	**1,815K**	**12.58**	0.66px	**16K**	**10,115K**	**629K**	–	–
	ConvOpt		703	102,044	1,001K	9.81	**0.48px**	3K	850K	452K	–	–
	DeepDesc		665	**152,537**	1,207K	7.92	**0.48px**	16K	4,196K	607K	–	–
	TFeat		683	**127,642**	**1,443K**	11.31	**0.52px**	16K	6,356K	**648K**	–	–
	LIFT		**768**	112,984	**1,477K**	**13.08**	0.73px	**23K**	**9,117K**	607K	–	–
Roman Forum	*SIFT*	2,364	1,407	242,192	1,805K	**7.45**	0.61px	**25K**	**6,063K**	**3,097K**	–	–
	SIFT-PCA		**1,463**	**244,556**	**1,834K**	**7.50**	0.61px	16K	4,322K	2,799K	–	–
	DSP-SIFT		**1,583**	**372,573**	**2,879K**	**7.73**	0.71px	**26K**	**9,685K**	**3,748K**	–	–
	ConvOpt		1,376	195,305	1,173K	6.01	**0.55px**	11K	2,111K	3,043K	–	–
	DeepDesc		1,173	174,532	1,275K	7.31	**0.60px**	9K	1,834K	2,434K	–	–
	TFeat		**1,450**	**271,902**	**1,963K**	7.22	**0.61px**	**19K**	**5,584K**	**3,477K**	–	–
	LIFT		1,434	220,026	1,608K	7.31	0.75px	17K	4,732K	2,898K	–	–
Cornell	*SIFT*	6,514	**4,999**	1,010,544	6,317K	6.25	**0.53px**	71K	25,603K	**12,970K**	**1.537m (0.793m)**	–
	SIFT-PCA		3,049	640,553	4,335K	**6.77**	**0.54px**	26K	13,793K	6,135K	11.498m (1.088m)	–
	DSP-SIFT		**4,946**	1,177,916	**7,233K**	6.14	0.67px	73K	26,150K	**11,066K**	**2.943m (1.001m)**	–
	ConvOpt		1,986	632,613	4,747K	**7.50**	0.57px	42K	18,615K	5,321K	5.824m (0.904m)	–
	DeepDesc		3,489	**1,225,780**	6,977K	5.69	**0.55px**	**73K**	**28,845K**	10,159K	3.832m (0.695m)	–
	TFeat		**5,428**	**1,499,117**	**9,830K**	**6.56**	0.59px	**89K**	**40,640K**	**15,605K**	**2.126m (0.593m)**	–
	LIFT		3,798	**1,455,732**	**7,377K**	5.07	0.71px	**81K**	**39,812K**	10,512K	3.113m (0.712m)	–

Table 3. Results for our reconstruction benchmark. Pose error as mean (median) over all images. Dense error for 2cm (10cm) threshold [19]. **First**, **second**, **third** best results highlighted in bold. Number of images, sparse points, and dense points visualized in Figs. 1, 2, and 3.

Image Matching Across Wide Baselines: From Paper to Practice

Yuhe Jin · Dmytro Mishkin · Anastasiia Mishchuk · Jiri Matas · Pascal Fua · Kwang Moo Yi · Eduard Trulls

2021

Local featured type	Number of Images				
	100	200	400	800	all
SIFT [54]	0.06°	0.09°	0.06°	0.07°	0.09°
SIFT (Upright) [54]	0.07°	0.07°	0.04°	0.06°	0.09°
HardNet (Upright) [62]	0.06°	0.06°	0.06°	0.04°	0.05°
SuperPoint [34]	0.31°	0.25°	0.33°	0.19°	0.32°
R2D2 [80]	0.12°	0.08°	0.07°	0.08°	0.05°

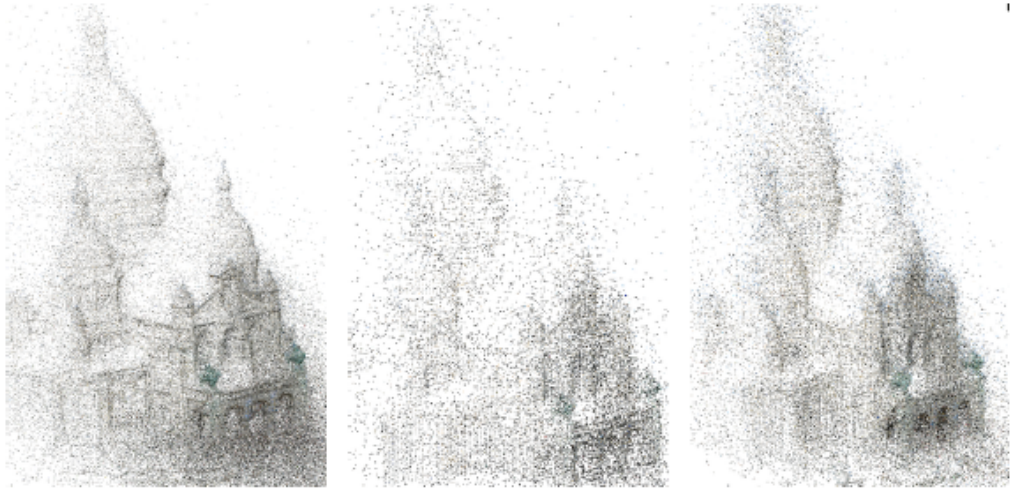
Table 2 Standard deviation of the pose difference of three COLMAP runs with different number of images. Most of them are below 0.1°, except for SuperPoint.

Local feature type	Number of images			
	100 vs. all	200 vs. all	400 vs. all	800 vs. all
SIFT [54]	0.58° / 0.22°	0.31° / 0.08°	0.23° / 0.05°	0.18° / 0.04°
SIFT (Upright) [54]	0.52° / 0.16°	0.29° / 0.08°	0.22° / 0.05°	0.16° / 0.03°
HardNet (Upright) [62]	0.35° / 0.10°	0.33° / 0.08°	0.23° / 0.06°	0.14° / 0.04°
SuperPoint [34]	1.22° / 0.71°	1.11° / 0.67°	1.08° / 0.48°	0.74° / 0.38°
R2D2 [80]	0.49° / 0.14°	0.32° / 0.10°	0.25° / 0.08°	0.18° / 0.05°

Table 3 Pose convergence in SfM. We report the mean/median of the difference (in degrees) between the poses extracted with the full set of 1179 images for "Sacre Coeur", and different subsets of it, for four local feature methods – to keep the results comparable we only look at the 100 images in common across all subsets. We report the maximum among the angular difference between rotation matrices and translation vectors. The estimated poses are stable, with as little as 100 images.

Reference	Compared to			
	SIFT (Upright)	HardNet (Upright)	SuperPoint	R2D2
SIFT [54]	0.20° / 0.05°	0.26° / 0.05°	1.01° / 0.62°	0.26° / 0.09°

Table 4 Difference between poses obtained with different local features. We report the mean/median of the difference (in degrees) between the poses extracted with SIFT (Upright), HardNet (Upright), SuperPoint, or R2D2, and those extracted with SIFT. We use the maximum of the angular difference between rotation matrices and translation vectors. SIFT (Upright), HardNet (Upright), and R2D2 give near-identical results to SIFT.



(a) SIFT (b) SuperPoint (c) R2D2

Fig. 6 COLMAP with different local features. We show the reconstructed point cloud for the scene "Sacre Coeur" using three different local features: SIFT, SuperPoint, and R2D2, using all images available (1179). The reconstructions with SIFT and R2D2 are both dense, albeit somewhat different. The reconstruction with SuperPoint is quite dense, considering it can only extract a much smaller number of features effectively, but its poses appear less accurate.

Method	NL↑	SR↑	RC↑	TL↑	mAA(5°)↑	mAA(10°)↑	ATE↓	Rank
CV-SIFT	2577.6	96.7	94.1	3.95	.5309	.6261	.4721	14
VL-SIFT	3030.7	97.9	95.4	4.17	.5273	.6283	.4669	13
VL-Hessian-SIFT	3209.1	97.4	94.1	4.13	.4857	.5866	.5175	16
VL-DoGAff-SIFT	3061.5	98.0	96.2	4.11	.5263	.6296	.4751	12
VL-HesAffNet-SIFT	3327.7	97.7	95.2	4.08	.5049	.6069	.4897	15
CV-$\sqrt{\text{SIFT}}$	3312.1	98.5	96.6	4.13	.5778	.6765	.4485	9
CV-SURF	2766.2	94.8	92.6	3.47	.3897	.4846	.6251	18
CV-AKAZE	4475.9	99.0	95.4	3.88	.4516	.5553	.5715	17
CV-ORB	3260.3	97.2	91.1	3.45	.2697	.3509	.7377	22
CV-FREAK	2859.1	92.9	91.7	3.53	.3735	.4653	.6229	20
L2-Net	3424.9	98.6	96.2	4.21	.5661	.6644	.4482	10
DoG-HardNet	4001.4	99.5	97.7	4.34	.6090	.7096	.4187	1
DoG-HardNetAmos+	3550.6	98.8	96.9	4.28	.5879	.6888	.4428	6
Key.Net-HardNet	3366.0	98.9	96.7	4.32	.5391	.6483	.4622	11
Key.Net-SOSNet	5505.5	100.0	98.7	4.46	.5989	.7038	.4286	2
GeoDesc	3839.0	99.1	97.2	4.26	.5782	.6803	.4445	8
ContextDesc	3732.5	99.3	97.6	4.22	.6036	.7035	.4228	3
DoG-SOSNet	3796.0	99.3	97.4	4.32	.6032	.7021	.4226	4
LogPolarDesc	4054.6	99.0	96.4	4.32	.5928	.6928	.4340	5
D2-Net (SS)	5893.8	99.8	97.5	3.62	.3435	.4598	.6361	21
D2-Net (MS)	6759.3	99.7	98.2	3.39	.3524	.4751	.6283	19
R2D2 (wasf-n8-big)	4432.9	99.7	97.2	4.59	.5775	.6832	.4333	7
DoG-AffNet-HardNet	4671.3	99.9	98.1	4.56	.6296	.7267	.4021	1*
DoG-MKD-Concat	3507.4	98.5	96.1	4.17	.5461	.6476	.4668	11*
DoG-TFeat	2905.3	97.1	94.8	4.04	.5270	.6261	.4873	14*

Method	NL↑	SR↑	RC↑	TL↑	mAA(5°)↑	mAA(10°)↑	ATE↓	Rank
CV-SIFT	1081.2	87.6	87.4	3.70	.3718	.4562	.6136	13
CV-$\sqrt{\text{SIFT}}$	1174.7	90.3	89.4	3.82	.4074	.4995	.5589	12
CV-SURF	1186.6	90.2	88.6	3.55	.3335	.4184	.6701	15
CV-AKAZE	1383.9	94.7	90.9	3.74	.3393	.4361	.6422	14
CV-ORB	683.3	74.9	73.0	3.21	.1422	.1914	.8153	19
CV-FREAK	1075.2	87.2	86.3	3.52	.2578	.3297	.7169	17
L2-Net	1253.3	94.7	92.6	3.96	.4369	.5392	.5419	9
DoG-HardNet	1338.2	96.3	93.7	4.03	.4624	.5661	.5093	6
Key.Net-HardNet	1276.3	97.8	95.7	4.49	.5050	.6161	.4902	2
Key.Net-SOSNet	1475.5	99.3	96.5	4.42	.5229	.6340	.4853	1
GeoDesc	1133.6	93.6	91.3	4.02	.4246	.5244	.5455	10
ContextDesc	1504.9	95.6	93.3	3.92	.4529	.5568	.5327	7
DoG-SOSNet	1317.4	96.0	93.8	4.05	.4739	.5784	.5194	5
LogPolarDesc	1410.2	96.0	93.8	4.05	.4794	.5849	.5090	4
D2-Net (SS)	2357.9	98.9	94.7	3.39	.2875	.3943	.7010	16
D2-Net (MS)	2177.3	98.2	93.4	3.01	.1921	.3007	.7861	20
LF-Net	1385.0	95.6	90.4	4.14	.4156	.5141	.5738	11
SuperPoint	1184.3	95.6	92.4	4.34	.4423	.5464	.5457	8
R2D2 (wasf-n16)	1228.4	99.4	96.2	4.29	.5045	.6149	.4956	3
DoG-AffNet-HardNet	1788.7	98.7	95.7	4.19	.4771	.5854	.5114	4*

Table 8 Test – Multiview results with 2k features. Same as Table 6. * **The last group of results is obtained after the paper publication and not described in the text of the paper. Their rank does not influence other entries ranks.**

Root-SIFT performs close to the best (within 15%). DoG-HardNet performs best

Follow-up works

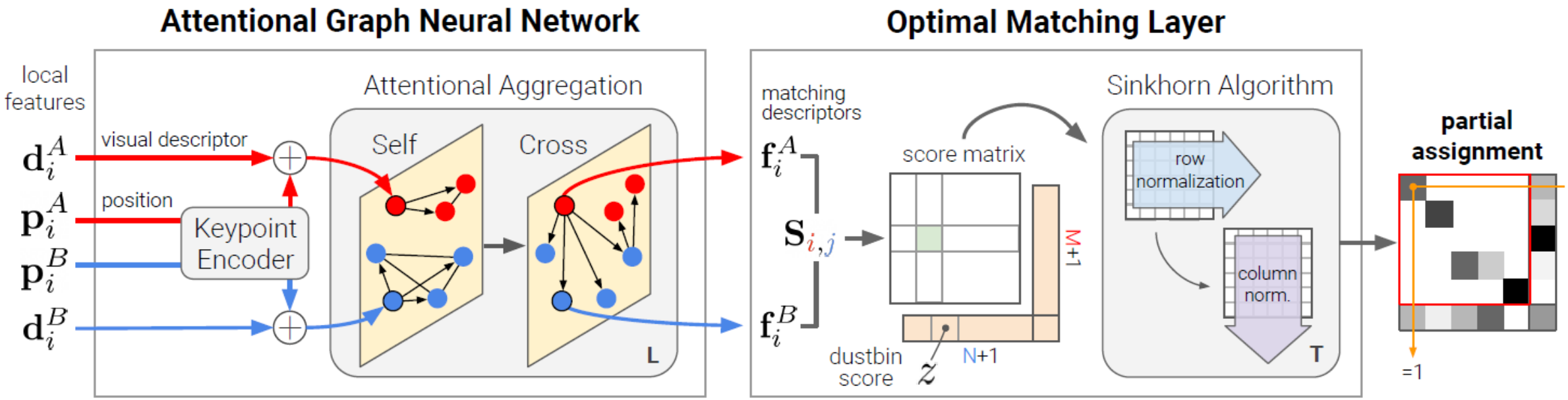
- Deep ChArUco (2019)
  - https://www.youtube.com/watch?v=Smorg9dffc0

- Self-Improving Visual Odometry (2018)
  - https://arxiv.org/abs/1812.03245

SuperGlue: Learning Feature Matching with Graph Neural Networks

Paul-Edouard Sarlin[1*] Daniel DeTone[2] Tomasz Malisiewicz[2] Andrew Rabinovich[2]
[1] ETH Zurich [2] Magic Leap, Inc.

2020



Self-attention and Cross-attention

- Alternately, aggregate features of keypoints within image and across images
  - Multi-attention head

concatenate

$$ {}^{(\ell+1)}\mathbf{x}_i^A = {}^{(\ell)}\mathbf{x}_i^A + \mathrm{MLP}\left(\left[{}^{(\ell)}\mathbf{x}_i^A \,||\, \mathbf{m}_{\mathcal{E}\to i}\right]\right) $$

$$ \mathbf{m}_{\mathcal{E}\to i} = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{v}_j, $$

$$ \mathbf{q}_i = \mathbf{W}_1 \, {}^{(\ell)}\mathbf{x}_i^Q + \mathbf{b}_1 $$

$$ \begin{bmatrix}\mathbf{k}_j \\ \mathbf{v}_j\end{bmatrix} = \begin{bmatrix}\mathbf{W}_2 \\ \mathbf{W}_3\end{bmatrix} {}^{(\ell)}\mathbf{x}_j^S + \begin{bmatrix}\mathbf{b}_2 \\ \mathbf{b}_3\end{bmatrix} $$

- Final features for each keypoint go through one more weight layer

$$ \mathbf{f}_i^A = \mathbf{W} \cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b}, \quad \forall i \in \mathcal{A}, $$

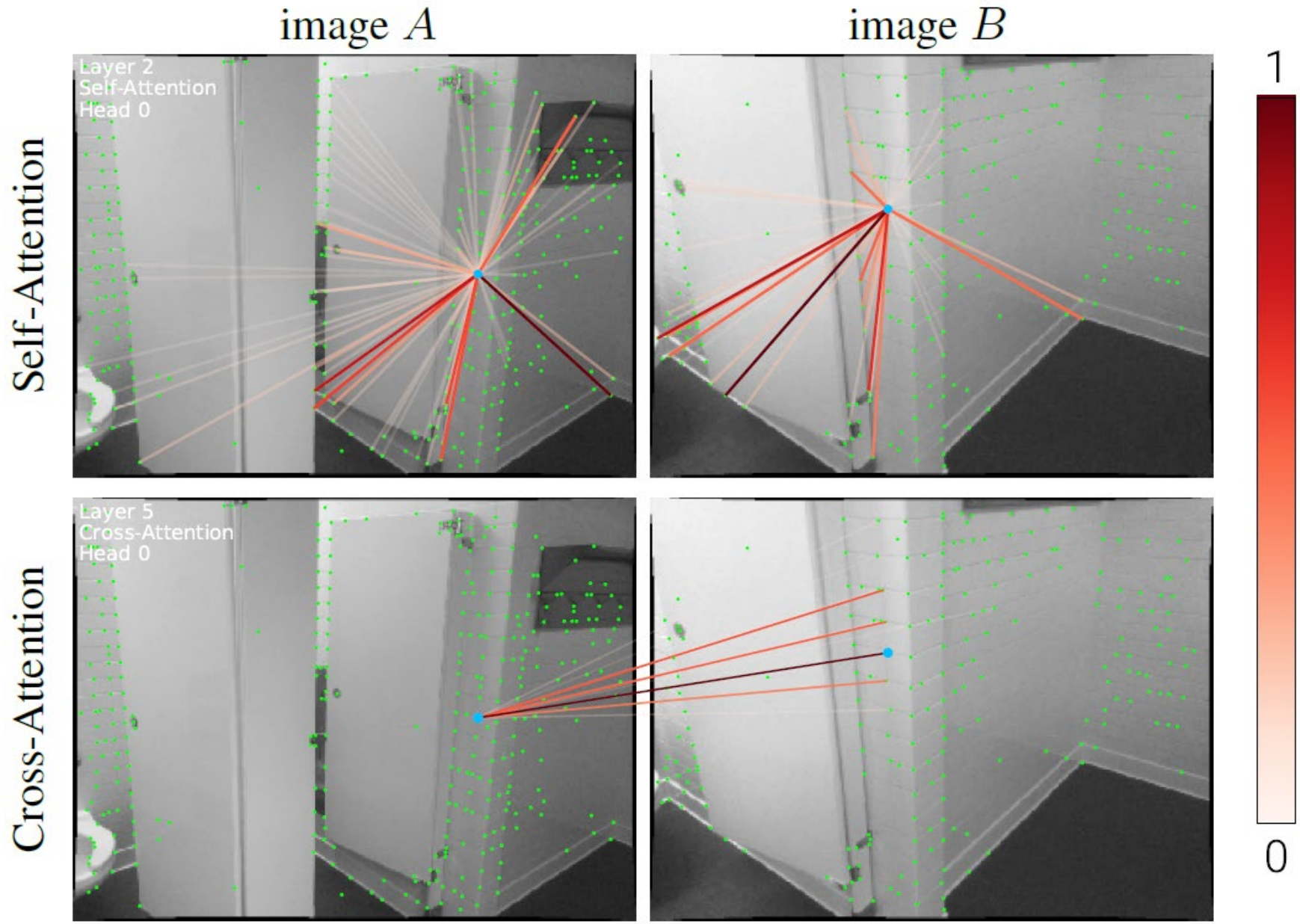


Figure 4: **Visualizing self- and cross-attention.** Atten-

Matching

- Maximize inner product of matched keypoints $\sum_{i,j} \mathbf{S}_{i,j}\mathbf{P}_{i,j}$ (assignment)

$$\mathbf{S}_{i,j} =< \mathbf{f}_i^A, \mathbf{f}_j^B >, \forall (i,j) \in \mathcal{A} \times \mathcal{B}$$

- Each keypoint also has a constant learned score for being assigned to the “dustbin”

- Sinkhorn algorithm: iteratively normalize exp(S) across rows and columns (i.e. soft assignment)

- In training minimize

$$\text{Loss} = -\sum_{(i,j)\in\mathcal{M}} \log \bar{\mathbf{P}}_{i,j} - \sum_{i\in\mathcal{I}} \log \bar{\mathbf{P}}_{i,N+1} - \sum_{j\in\mathcal{J}} \log \bar{\mathbf{P}}_{M+1,j}$$

Experiments

- Train/test on ScanNet for indoor
- Train on MegaDepth, test on PhotoTourism
- Can use SIFT or Superpoint for keypoints

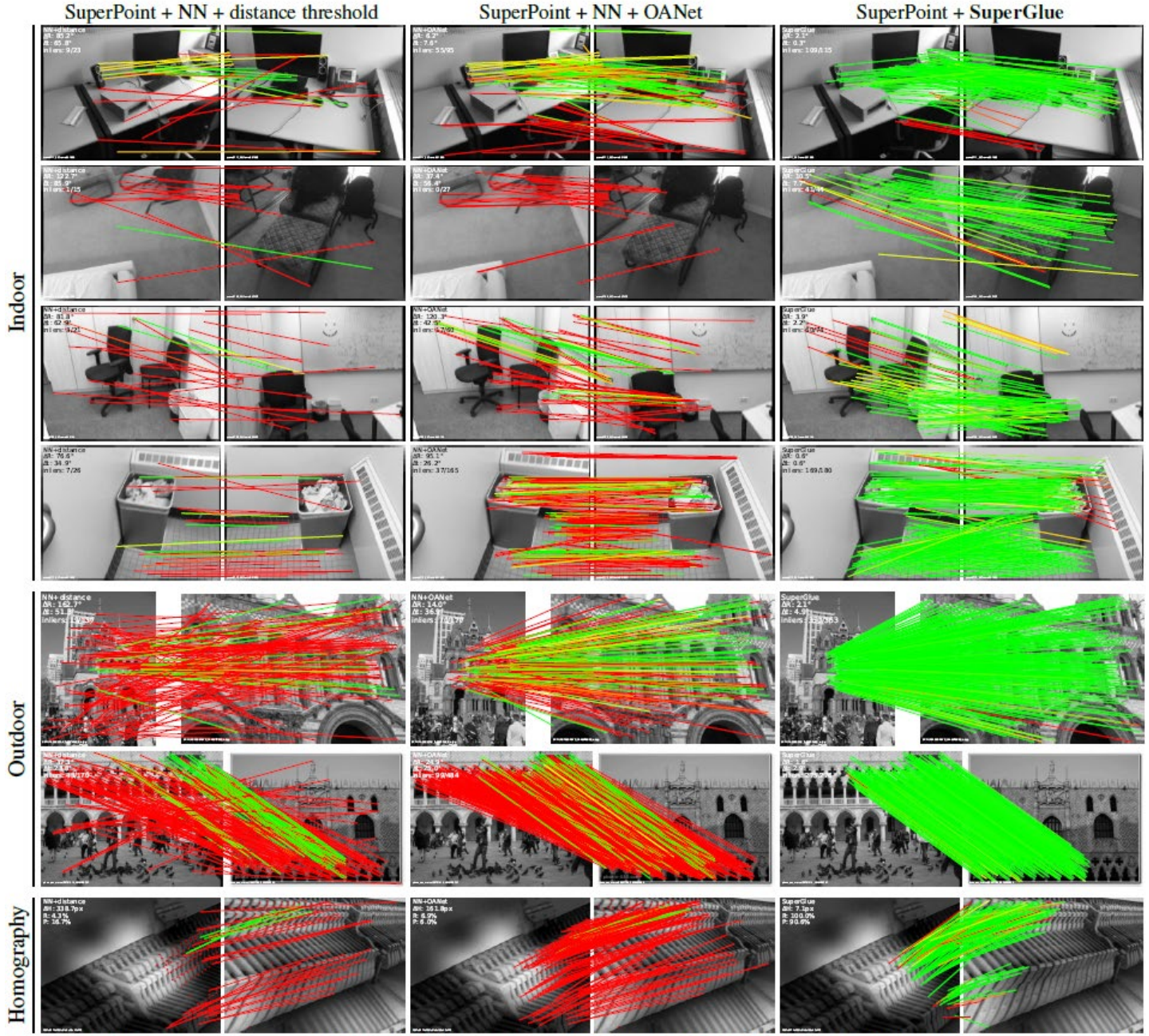


Figure 6: **Qualitative image matches.** We compare SuperGlue to the Nearest Neighbor (NN) matcher with two outlier rejectors, handcrafted and learned, in three environments. SuperGlue consistently estimates more correct matches (green lines) and fewer mismatches (red lines), successfully coping with repeated texture, large viewpoint, and illumination changes.

Local features	Matcher	Pose estimation AUC @5°	@10°	@20°	P	MS
ORB D2-Net ContextDesc	NN + GMS NN + mutual NN + ratio test	5.21 5.25 6.64	13.65 14.53 15.01	25.36 27.96 25.75	72.0 46.7 51.2	5.7 12.0 9.2
SIFT	NN + ratio test NN + NG-RANSAC NN + OANet **SuperGlue**	5.83 6.19 6.00 **6.71**	13.06 13.80 14.33 **15.70**	22.47 23.73 25.90 **28.67**	40.3 61.9 38.6 **74.2**	1.0 0.7 4.2 **9.8**
SuperPoint	NN + mutual NN + distance + mutual NN + GMS NN + PointCN NN + OANet **SuperGlue**	9.43 9.82 8.39 11.40 11.76 **16.16**	21.53 22.42 18.96 25.47 26.90 **33.81**	36.40 36.83 31.56 41.41 43.85 **51.84**	50.4 63.9 50.3 71.8 74.0 **84.4**	18.8 14.6 19.0 25.5 25.7 **31.5**

Table 2: **Wide-baseline indoor pose estimation.** We report the AUC of the pose error, the matching score (MS) and precision (P), all in percents %. SuperGlue outperforms all handcrafted and learned matchers when applied to both SIFT and SuperPoint.

Local features	Matcher	Pose estimation AUC @5°	@10°	@20°	P	MS
ContextDesc	NN + ratio test	20.16	31.65	44.05	56.2	3.3
SIFT	NN + ratio test NN + NG-RANSAC NN + OANet **SuperGlue**	15.19 15.61 18.02 **23.68**	24.72 25.28 28.76 **36.44**	35.30 35.87 40.31 **49.44**	43.4 64.4 55.0 **74.1**	1.7 1.9 3.7 **7.2**
SuperPoint	NN + mutual NN + GMS NN + OANet **SuperGlue**	9.80 13.96 21.03 **34.18**	18.99 24.58 34.08 **50.32**	30.88 36.53 46.88 **64.16**	22.5 47.1 52.4 **84.9**	4.9 4.7 8.4 **11.1**

Table 3: **Outdoor pose estimation.** Matching SuperPoint

Runtime

- Not really good enough yet for SLAM or SfM due to being sub-realtime on GPU

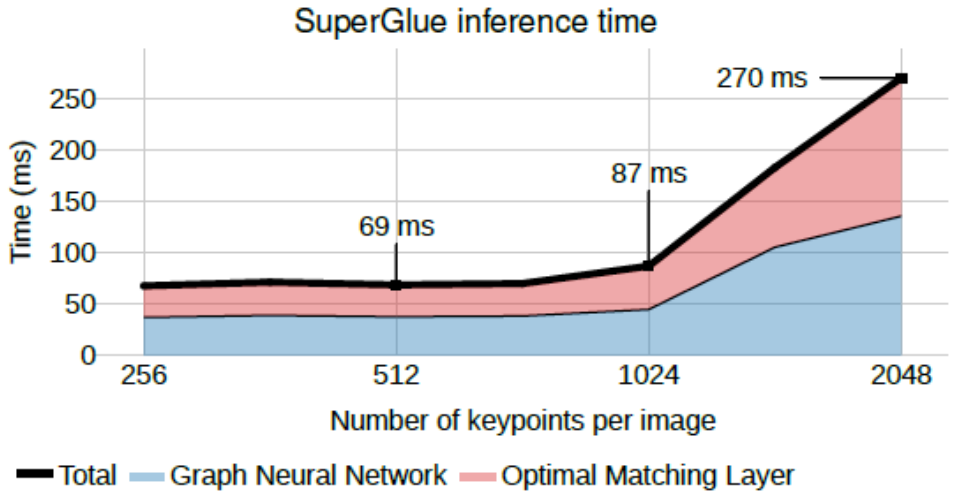


Figure 11: **SuperGlue detailed inference time.** SuperGlue's two main blocks, the Graph Neural Network and the Optimal Matching Layer, have similar computational costs. For 512 and 1024 keypoints per image, SuperGlue runs at 14.5 and 11.5 FPS, respectively.

Localization toolbox using superglue

- https://github.com/cvg/Hierarchical-Localization/

- Contains notebooks for using matches in localization and SfM (w/ colmap)

Open problems / research ideas

- Learned keypoint detectors, descriptors, matchers have promise but need to overcome several drawbacks
  - Keypoint precision
  - Generalization of descriptors to new scenes
  - Speed of matching, e.g. vs. 64 core CPU, or NN ratio matching on GPU

- Detector and descriptor evaluation metrics are not good indicators of final SfM/SLAM performance, so need to optimize or at least test in context of entire system

Summary (of technical part)

- Learned features have potential to be more robust to noise, low-light, wide baselines, and other adverse conditions
- DoG-SIFT remains competitive, performing close to the best solutions in SfM evaluations
- Where learning can best play a role in SfM continues to be explored, e.g. Pixel-Perfect SfM (ICCV 2021)

Tips for choosing and vetting research directions

In choosing research directions,

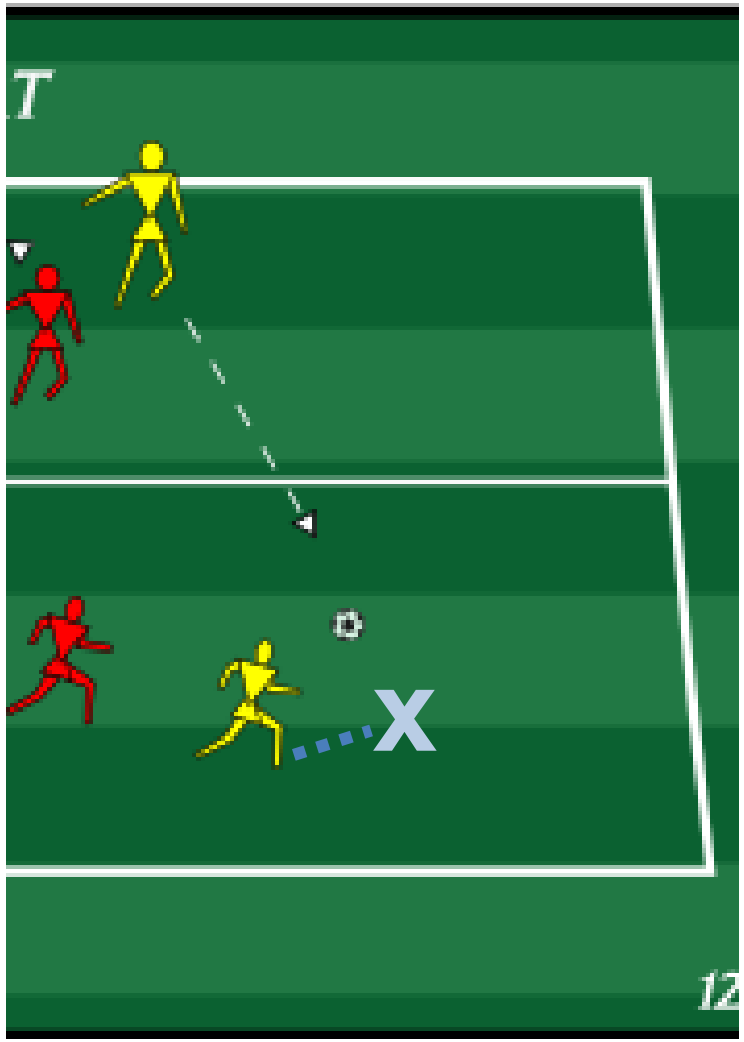
be **part of the team**

Don’t crowd the ball. Find your position.

- What is your unique perspective or angle in this research?
- How can your research influence others? How can you best advance the community as a whole?
- Sometimes you *should* run at the ball (i.e. work on the popular topic)

Aim ahead of your target

- What will be the big problem once the currently popular problems are “solved”?



Celebrate the wins of your community

https://edition.cnn.com/2016/12/29/sport/2016-best-sports-year-ever/index.html

How do you avoid getting stuck?

Vet a project carefully before you start

- Before you touch a keyboard for a research project, write down:
  - What you are trying to solve (or answer)
  - Why you are trying to solve it
  - What is your key idea
  - How is that different from existing work
  - How will you measure success
  - Write the intro – is it exciting?
  - What is the simplest proof of concept to verify the key idea?
- Make sure that someone else finds this plan credible
- Don’t be afraid to discard an idea if it doesn’t seem that great after all
- Set a date to review progress and decide whether to continue

Research Proposal Template

Problem to solve

What research question are you trying to answer, or problem are you trying to solve?

Related work

What are some of the most closely related papers, and what progress have they made? What problems remain?

Method overview

What is the basic idea of your approach?

Proposed contributions

What do you expect the statement of contributions to be?

Significance

What is the potential impact of your work? Why will anyone care?

How to test?

How will you demonstrate your contributions?

Expected result

What do you expect to be the results of your experiments?

Target venue

Where will you submit the work?

When to start fresh

- Are you still excited about your research idea?
- Did your proof of concept pan out, or does the idea keep evolving?
- Do you need more time to see if the result is amazing, or are you hoping with a bit more work it will be above threshold?
- Imagine a friend were telling you about your project – would you think it's a good idea for your friend to keep working on it?

- If it's time to stop: spend a few days to write up a tech report of your ideas, experiments, and any results