Deep visual odometry

CV&ML reading group, 2020/01/31

Presenter: You-Yi Jau, M.S. in UCSD

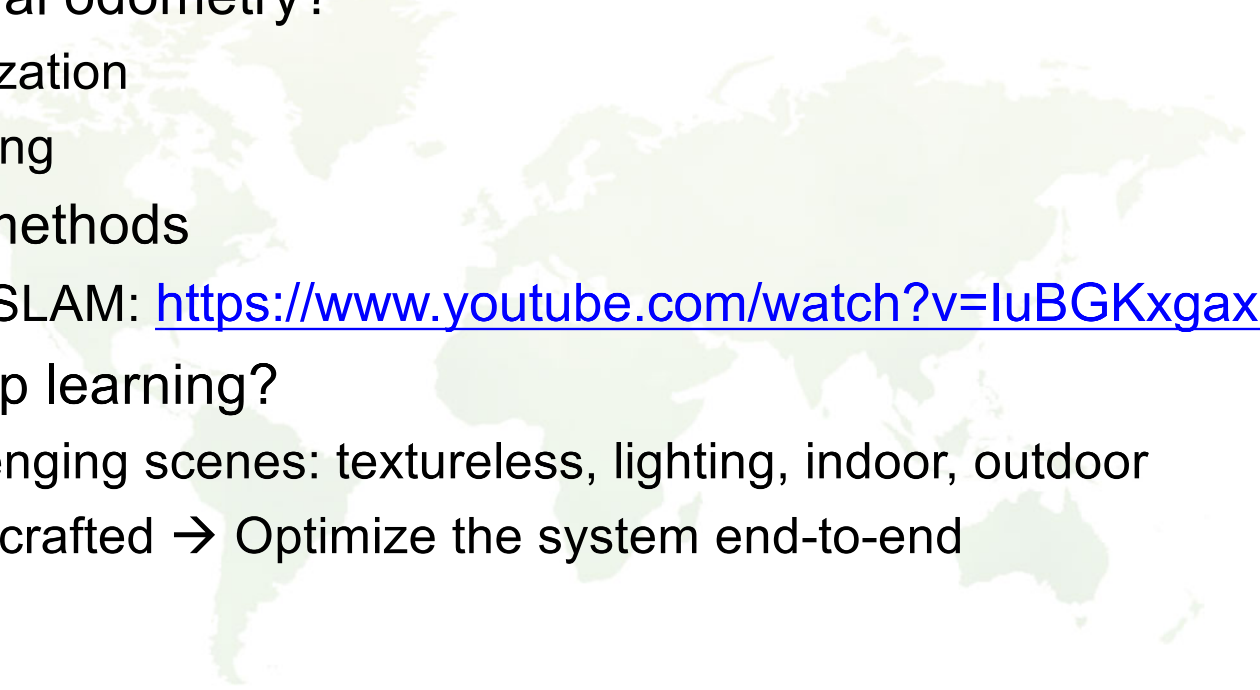
Outlines

- Motivation and problem description
- Brief summary of recent works
- Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
- Optional: Visual Odometry Revisited: What Should Be Learnt?

Outlines

- Motivation and problem description
- Brief summary of recent works
- Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
- Optional: Visual Odometry Revisited: What Should Be Learnt?

Motivation and problem description



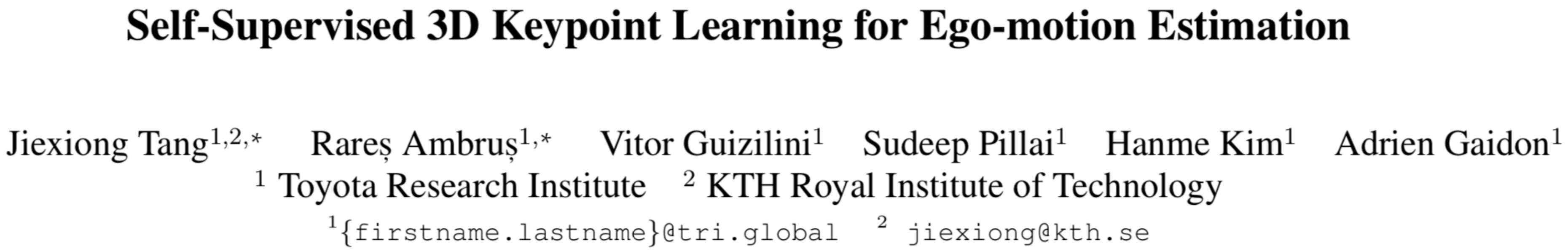
- Why visual odometry?
  - Localization
  - Mapping
- Classic methods
  - ORB-SLAM: https://www.youtube.com/watch?v=IuBGKxgaxS0
- Why deep learning?
  - Challenging scenes: textureless, lighting, indoor, outdoor
  - Hand-crafted → Optimize the system end-to-end

Brief summary of recent works

- Pure deep learning visual odometry
  - Beyond tracking: LSTM, relative pose, absolute pose
    - Beyond tracking: Selecting memory and refining poses for deep visual odometry
    - F Xue, X Wang, S Li, Q Wang, J Wang, H Zha - … of the IEEE Conference on Computer …, 2019
  - DSVO: predict left frame from right frame
    - Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry
    - N Yang, R Wang, J Stuckler, D Cremers - … of the European Conference on Computer …, 2018
- Not pure deep learning visual odometry
    - Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
    - Visual Odometry Revisited: What Should Be Learnt?

Outlines

- Motivation and problem description
- Brief summary of recent works
- Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
- Optional: Visual Odometry Revisited: What Should Be Learnt?



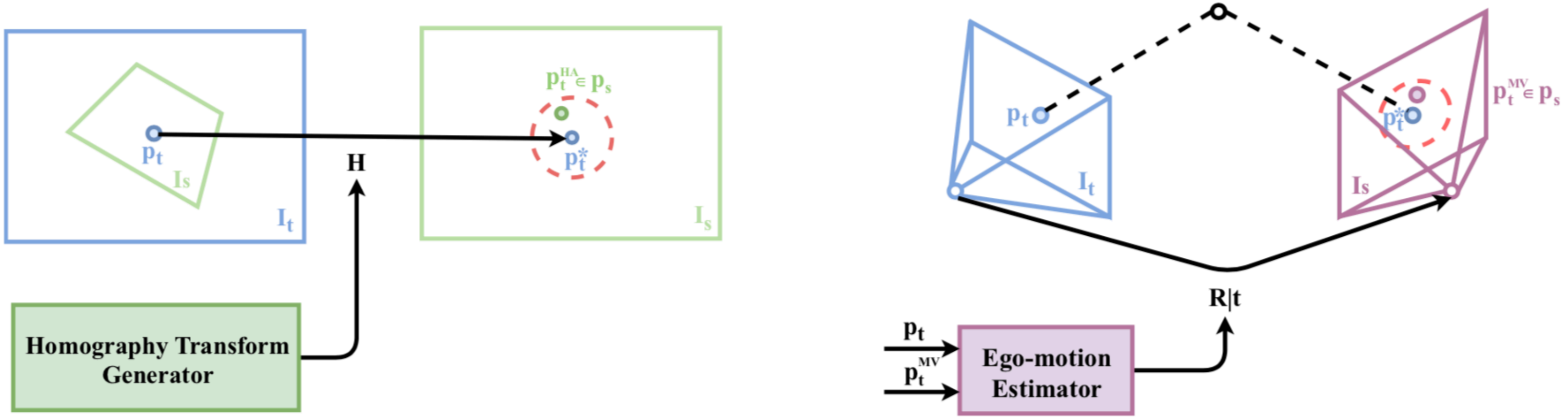
Self-Supervised 3D Keypoint Learning for Ego-motion Estimation

Jiexiong Tang[1,2,*] Rareș Ambruș[1,*] Vitor Guizilini[1] Sudeep Pillai[1] Hanme Kim[1] Adrien Gaidon[1]
[1] Toyota Research Institute [2] KTH Royal Institute of Technology
[1]{firstname.lastname}@tri.global [2] jiexiong@kth.se

Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
J Tang, R Ambrus, V Guizilini, S Pillai, H Kim, A Gaidon - arXiv preprint arXiv …, 2019

Highlights

- Self-supervised training with multi-view adaptation



(a) Homography Adaptation (b) Multi-View Adaptation

Figure 3. **Adaptations for Keypoint Learning**. We contrast between *Homography Adaptation* where $\mathbf{p}_t^*$ can be trivially computed and *Multi-View Adaptation* where we first compute $\mathbf{x}_{t\rightarrow s} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}$ via the correspondence set $\left(\mathbf{p}_t, \mathbf{p}_t^{MV}\right)$.

Motivation and problem description

- Keypoint matching is essential for Sfm, visual odometry (VO), SLAM
- Hand-engineered (SIFT) feature detectors and descriptors are dominant algorithms
- Deep learning methods drew attention (Superpoint)

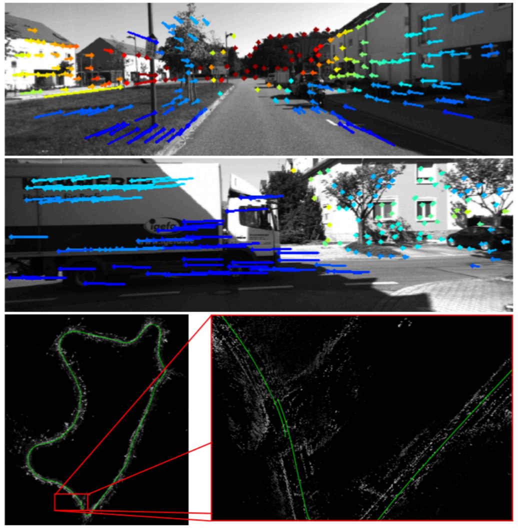


Figure 1. **Self-Supervised 3D Keypoints for Robust VO. Top/Middle:** The illustration shows the proposed self-supervised 3D keypoints learned *purely* from unlabeled monocular videos together with matched sparse-flow. By using the structured pose estimation with 3D keypoints, dynamic objects can be effectively handled by outlier rejection. **Bottom:** The proposed 3D keypoint estimation can be trivially integrated into a state-of-the-art visual SLAM back-end such as DSO for accurate, scale-aware, long-range monocular visual odometry.

Motivation and problem description

- Contribution
  - Fully self-supervised framework for the learning of depth-aware keypoint detection and description from unlabeled videos
  - 3D multi-view adaptation, a novel adaptation technique that exploits the temporal context in videos
  - Integrating with a state-of-the-art tracking method such as Direct Sparse Odometry (DSO)

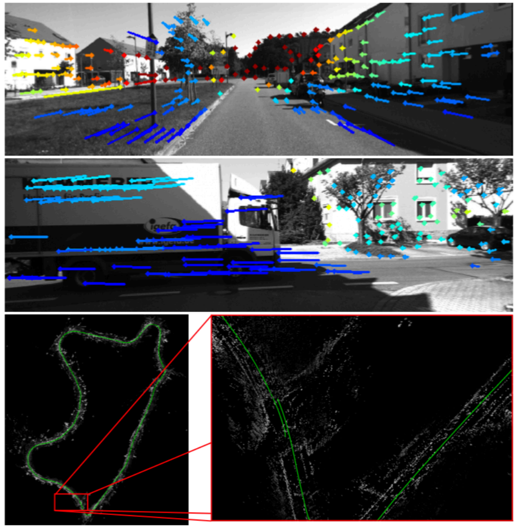


Figure 1. **Self-Supervised 3D Keypoints for Robust VO. Top/Middle:** The illustration shows the proposed self-supervised 3D keypoints learned *purely* from unlabeled monocular videos together with matched sparse-flow. By using the structured pose estimation with 3D keypoints, dynamic objects can be effectively handled by outlier rejection. **Bottom:** The proposed 3D keypoint estimation can be trivially integrated into a state-of-the-art visual SLAM back-end such as DSO for accurate, scale-aware, long-range monocular visual odometry.

Prior works

- Learning-based methods for keypoint estimation
  - Supervised method: LIFT (Kwang et al.), LF-Net (Yuki et al.)
  - Self-supervised: Superpoint (Daniel et al.), Unsuperpoint (Peter et al.)
- Learning-based methods for visual odometry
  - Sfm-learner (Zhou et al.)
  - Visual odometry revisited (Huangying et al.)

Overview: Pipeline

- Depth prediction
- Keypoint prediction

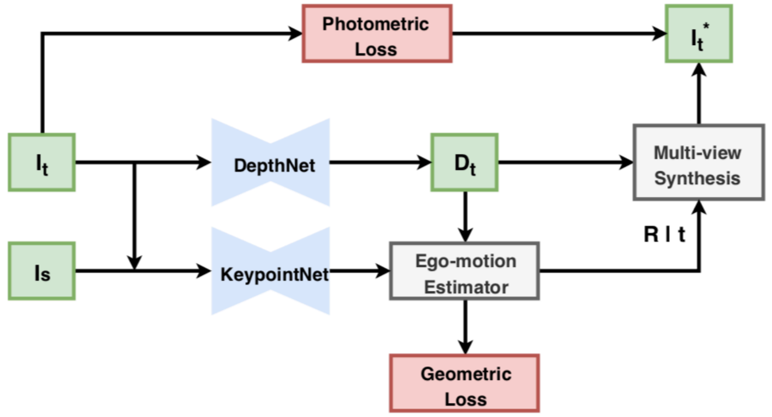


Figure 2. **Monocular SfM-based 3D Keypoint Learning**. We illustrate the overall architecture of our proposed method that uses two consecutive images (target I_t and source I_s) as input to self-supervise 3D keypoint learning for monocular ego-motion estimation. We train the *DepthNet* and *KeypointNet* simultaneously in an end-to-end fashion with a combination of photometric and multi-view geometric losses (Section 3.4), to develop a robust 3D keypoint estimator for long-term ego-motion estimation.

Networks

- Input/ Output
- DepthNet
  - RGB Image → "Scale-ambiguous" dense (inverse) depth map **D**
    - [20] Cle´ment Godard, Oisin Mac Aodha, and Gabriel J. Bros- tow. Unsupervised monocular depth estimation with left- right consistency. In CVPR, 2017.
- KeypointNet
  - RGB Image → **k = {p, f, s} =** {locations: Nx2, descriptors: Nx256, scores}
    - [4] Anonymous. Neural outlier rejection for self-supervised keypoint learning. In Submitted to International Conference on Learning Representations, 2020. under review.
    - Same author. (Guess accepted in ICLR 2020.)

Network architecture

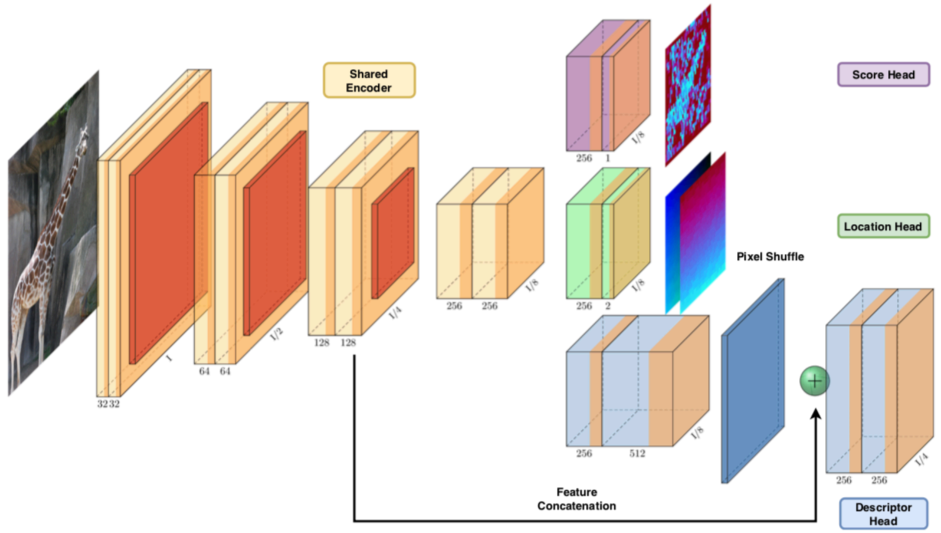
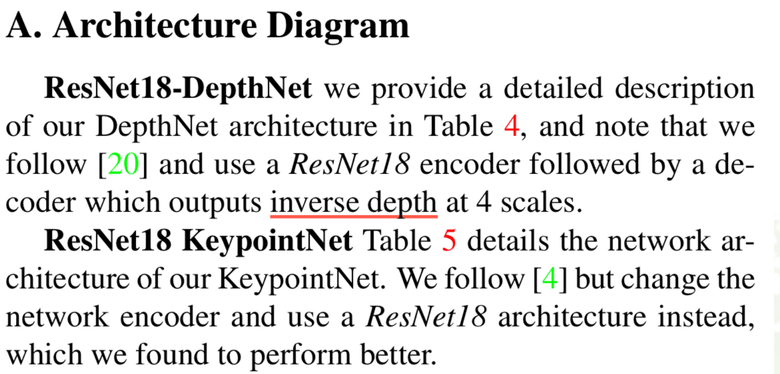


Figure 2: The proposed *KeyPointNet* architecture leverages a shared-encoder backbone with three output heads for the regression of keypoint locations (**p**), scores (**s**) and descriptions (**f**). To further improve keypoint description performance, KeyPointNet produces higher-resolution feature-maps for the keypoint descriptors via efficient sub-pixel convolutions at the keypoint descriptor head.

Neural Outlier Rejection for Self-Supervised Keypoint Learning

J Tang, H Kim, V Guizilini, S Pillai, R Ambrus - arXiv preprint arXiv:1912.10615, 2019

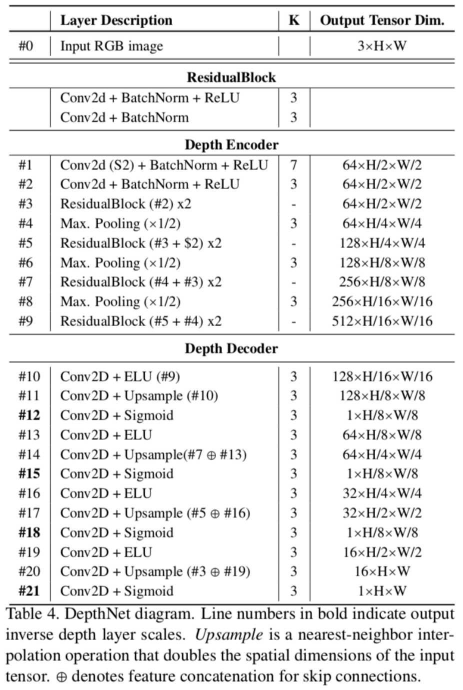
Network architecture



A. Architecture Diagram

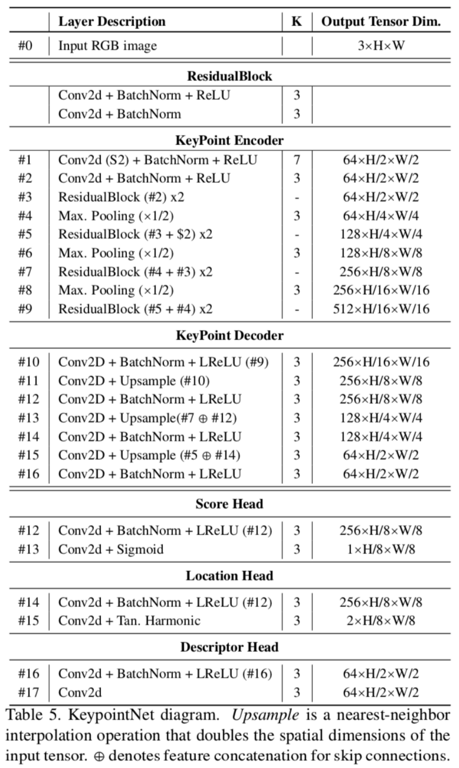
ResNet18-DepthNet we provide a detailed description of our DepthNet architecture in Table 4, and note that we follow [20] and use a *ResNet18* encoder followed by a decoder which outputs inverse depth at 4 scales.

ResNet18 KeypointNet Table 5 details the network architecture of our KeypointNet. We follow [4] but change the network encoder and use a *ResNet18* architecture instead, which we found to perform better.



	Layer Description	K	Output Tensor Dim.
#0	Input RGB image		3×H×W
	ResidualBlock		
	Conv2d + BatchNorm + ReLU	3	
	Conv2d + BatchNorm	3	
	Depth Encoder		
#1	Conv2d (S2) + BatchNorm + ReLU	7	64×H/2×W/2
#2	Conv2d + BatchNorm + ReLU	3	64×H/2×W/2
#3	ResidualBlock (#2) x2	-	64×H/2×W/2
#4	Max. Pooling (×1/2)	3	64×H/4×W/4
#5	ResidualBlock (#3 + $2) x2	-	128×H/4×W/4
#6	Max. Pooling (×1/2)	3	128×H/8×W/8
#7	ResidualBlock (#4 + #3) x2	-	256×H/8×W/8
#8	Max. Pooling (×1/2)	3	256×H/16×W/16
#9	ResidualBlock (#5 + #4) x2	-	512×H/16×W/16
	Depth Decoder		
#10	Conv2D + ELU (#9)	3	128×H/16×W/16
#11	Conv2D + Upsample (#10)	3	128×H/8×W/8
#12	Conv2D + Sigmoid	3	1×H/8×W/8
#13	Conv2D + ELU	3	64×H/8×W/8
#14	Conv2D + Upsample(#7 ⊕ #13)	3	64×H/4×W/4
#15	Conv2D + Sigmoid	3	1×H/8×W/8
#16	Conv2D + ELU	3	32×H/4×W/4
#17	Conv2D + Upsample (#5 ⊕ #16)	3	32×H/2×W/2
#18	Conv2D + Sigmoid	3	1×H/8×W/8
#19	Conv2D + ELU	3	16×H/2×W/2
#20	Conv2D + Upsample (#3 ⊕ #19)	3	16×H×W
#21	Conv2D + Sigmoid	3	1×H×W

Table 4. DepthNet diagram. Line numbers in bold indicate output inverse depth layer scales. *Upsample* is a nearest-neighbor interpolation operation that doubles the spatial dimensions of the input tensor. ⊕ denotes feature concatenation for skip connections.



	Layer Description	K	Output Tensor Dim.
#0	Input RGB image		3×H×W
	ResidualBlock		
	Conv2d + BatchNorm + ReLU	3	
	Conv2d + BatchNorm	3	
	KeyPoint Encoder		
#1	Conv2d (S2) + BatchNorm + ReLU	7	64×H/2×W/2
#2	Conv2d + BatchNorm + ReLU	3	64×H/2×W/2
#3	ResidualBlock (#2) x2	-	64×H/2×W/2
#4	Max. Pooling (×1/2)	3	64×H/4×W/4
#5	ResidualBlock (#3 + $2) x2	-	128×H/4×W/4
#6	Max. Pooling (×1/2)	3	128×H/8×W/8
#7	ResidualBlock (#4 + #3) x2	-	256×H/8×W/8
#8	Max. Pooling (×1/2)	3	256×H/16×W/16
#9	ResidualBlock (#5 + #4) x2	-	512×H/16×W/16
	KeyPoint Decoder		
#10	Conv2D + BatchNorm + LReLU (#9)	3	256×H/16×W/16
#11	Conv2D + Upsample (#10)	3	256×H/8×W/8
#12	Conv2D + BatchNorm + LReLU	3	256×H/8×W/8
#13	Conv2D + Upsample(#7 ⊕ #12)	3	128×H/4×W/4
#14	Conv2D + BatchNorm + LReLU	3	128×H/4×W/4
#15	Conv2D + Upsample (#5 ⊕ #14)	3	64×H/2×W/2
#16	Conv2D + BatchNorm + LReLU	3	64×H/2×W/2
	Score Head		
#12	Conv2d + BatchNorm + LReLU (#12)	3	256×H/8×W/8
#13	Conv2d + Sigmoid	3	1×H/8×W/8
	Location Head		
#14	Conv2d + BatchNorm + LReLU (#12)	3	256×H/8×W/8
#15	Conv2d + Tan. Harmonic	3	2×H/8×W/8
	Descriptor Head		
#16	Conv2d + BatchNorm + LReLU (#16)	3	64×H/2×W/2
#17	Conv2d	3	64×H/2×W/2

Table 5. KeypointNet diagram. *Upsample* is a nearest-neighbor interpolation operation that doubles the spatial dimensions of the input tensor. ⊕ denotes feature concatenation for skip connections.

Loss function

- Photometric loss
  - Dense
  - Image synthesis
- Geometric loss
  - Sparse
  - Re-projection in the monocular two-view setting

Key idea of loss function

- Homography adaptation vs. multi-view adaptation (use both)

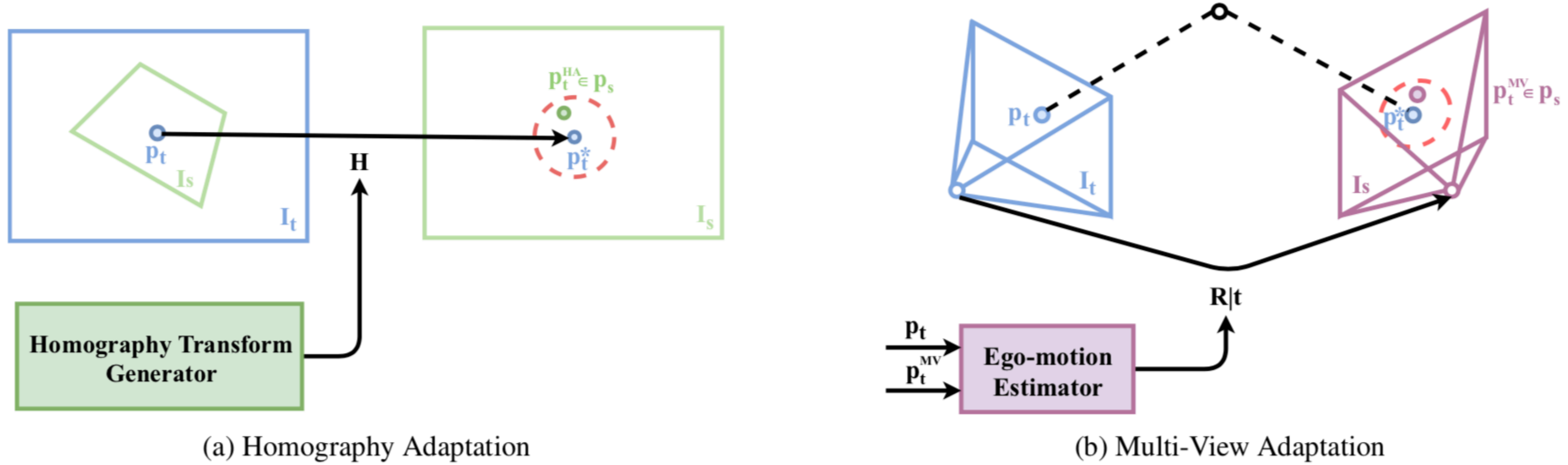


Figure 3. **Adaptations for Keypoint Learning**. We contrast between *Homography Adaptation* where $\mathbf{p}_t^*$ can be trivially computed and *Multi-View Adaptation* where we first compute $\mathbf{x}_{t\to s} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}$ via the correspondence set $\left(\mathbf{p}_t, \mathbf{p}_t^{MV}\right)$.

Loss function in detail

- Differentiable pose estimation from 2D-3D correspondences
  - 3D residuals
  - SVD for closed-form solution
  - Why need to be differentiable

$$
\begin{aligned}
r(\mathbf{R},\mathbf{t}) &= \left\|\mathbf{p}_s^d - \mathbf{R}\cdot\mathbf{p}_t^d + \mathbf{t}\right\|_2 ,\\
\text{where } \mathbf{p}_t^d &= \pi^{-1}(\mathbf{p}_t, D_t(\mathbf{p_t})) ,\\
\mathbf{p}_s^d &= \pi^{-1}(\mathbf{p}_s, d_s) ,\\
d_s &= \left[\mathbf{R_0}\cdot\mathbf{p}_t^d + \mathbf{t_0}\right]_z .
\end{aligned} \tag{2}
$$

Loss function in detail

- Keypoint loss
  - Descriptor matched keypoints ←→ 3D adapted keypoints (localization loss)

$$\mathcal{L}_p = \sum \|\mathbf{p}_t^* - \mathbf{p}_t^{MV}\|_2 \ . \qquad (6)$$

- Descriptor loss
  - Triplet loss, hard sample mining

$$\mathcal{L}_f = \sum_i \max(0, \|\mathbf{f}, \mathbf{f}_+\|_2 - \|\mathbf{f}, \mathbf{f}_-\|_2 + m) \ , \qquad (7)$$

- Score loss
  - Same scores across frames
  - High scores on good keypoints
  - 2D Re-projection error

$$\mathcal{L}_s = \sum \left[\frac{(s + s^{MV})}{2} \cdot (d(\mathbf{p}_\mathbf{t}^*, \mathbf{p_t}^{MV}) - \bar{d}) + (s - s^{MV})^2 \right] , \qquad (8)$$

Loss function in detail

- Photometric loss
  - Loss on SSIM
- Depth smoothness loss
  - Edge-aware term
- Depth Consistency
  - Discourage scale-drift between dense depth predictions in adjacent frames

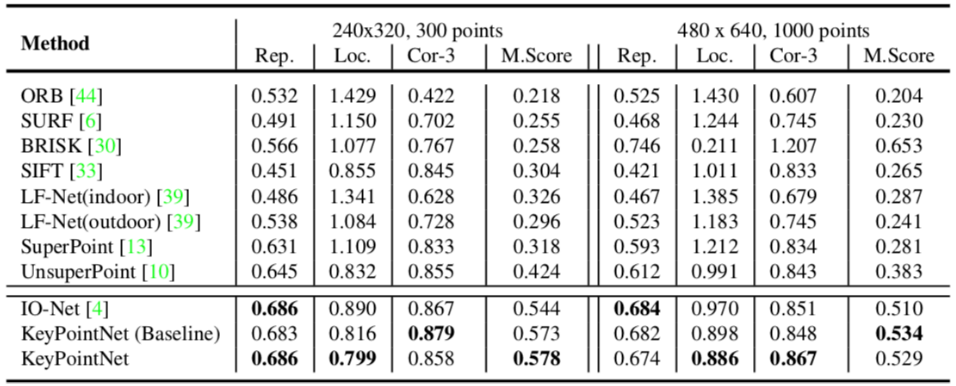
$$\mathcal{L}_c = \frac{\|D_t(\mathbf{p}_t) - D_s(\mathbf{p}_t^{MV})\|}{D_t(\mathbf{p}_t) + D_s(\mathbf{p}_t^{MV})} \,. \qquad (13)$$

- Overall loss

$$\mathcal{L} = \alpha(\mathcal{L}_p + \mathcal{L}_f + \mathcal{L}_s) + \mathcal{L}_c + \beta(\mathcal{L}_D + \mathcal{L}_{sm}) \,, \qquad (14)$$

Detector and descriptor evaluation

- Work the best among those



Method	240x320, 300 points				480 x 640, 1000 points			
	Rep.	Loc.	Cor-3	M.Score	Rep.	Loc.	Cor-3	M.Score
ORB [44]	0.532	1.429	0.422	0.218	0.525	1.430	0.607	0.204
SURF [6]	0.491	1.150	0.702	0.255	0.468	1.244	0.745	0.230
BRISK [30]	0.566	1.077	0.767	0.258	0.746	0.211	1.207	0.653
SIFT [33]	0.451	0.855	0.845	0.304	0.421	1.011	0.833	0.265
LF-Net(indoor) [39]	0.486	1.341	0.628	0.326	0.467	1.385	0.679	0.287
LF-Net(outdoor) [39]	0.538	1.084	0.728	0.296	0.523	1.183	0.745	0.241
SuperPoint [13]	0.631	1.109	0.833	0.318	0.593	1.212	0.834	0.281
UnsuperPoint [10]	0.645	0.832	0.855	0.424	0.612	0.991	0.843	0.383
IO-Net [4]	**0.686**	0.890	0.867	0.544	**0.684**	0.970	0.851	0.510
KeyPointNet (Baseline)	0.683	0.816	**0.879**	0.573	0.682	0.898	0.848	**0.534**
KeyPointNet	**0.686**	**0.799**	0.858	**0.578**	0.674	**0.886**	**0.867**	0.529

Table 1. **Keypoint and descriptor performance on HPatches** [5]. Repeatability and Localization Error measure keypoint performance while Correctness (pixel threshold 3) and Matching score measure descriptor performance. Higher is better for all metrics except Localization Error.

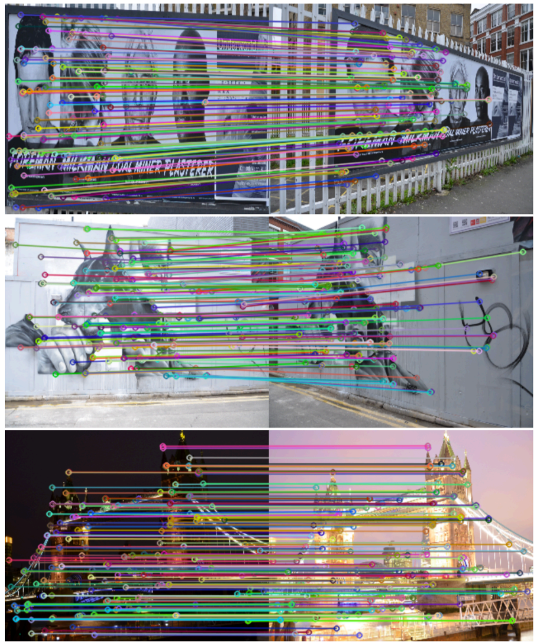


Figure 6. Qualitative matching results of our method on the HPatches dataset [5].

Experiments

- Frame-to-frame (F2F)
- Deep Semi-Direct Sparse Odometry (DS-DSO)

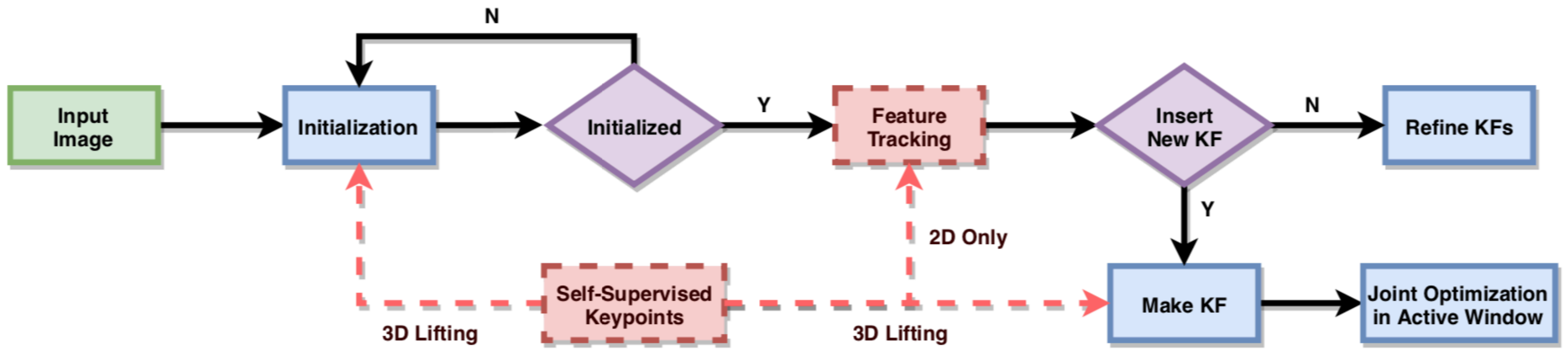
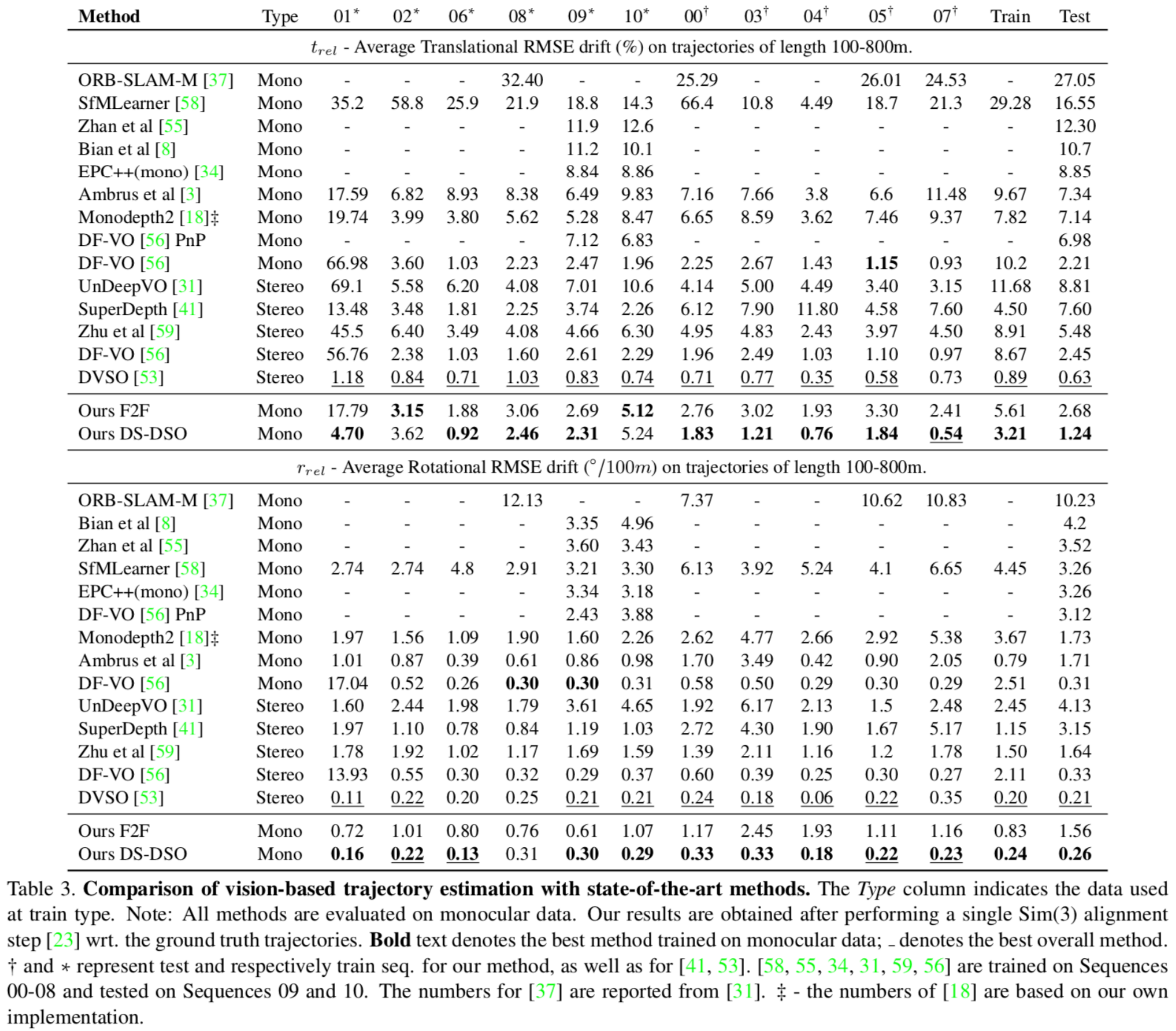


Figure 4. **The proposed DS-DSO pipeline**. DS-DSO leverages the depth initialization and robust feature tracking using our self-supervised depth-aware keypoint detection and description. The red block and arrows show that where the 3D keypoint is affecting the original DSO system, the purple texts show where 2D and 3D information is utilized.

Visual Odometry

- F2F worse than DF-VO[56]
- DS-DSO works on par with stereo methods
- Boosted by DSO

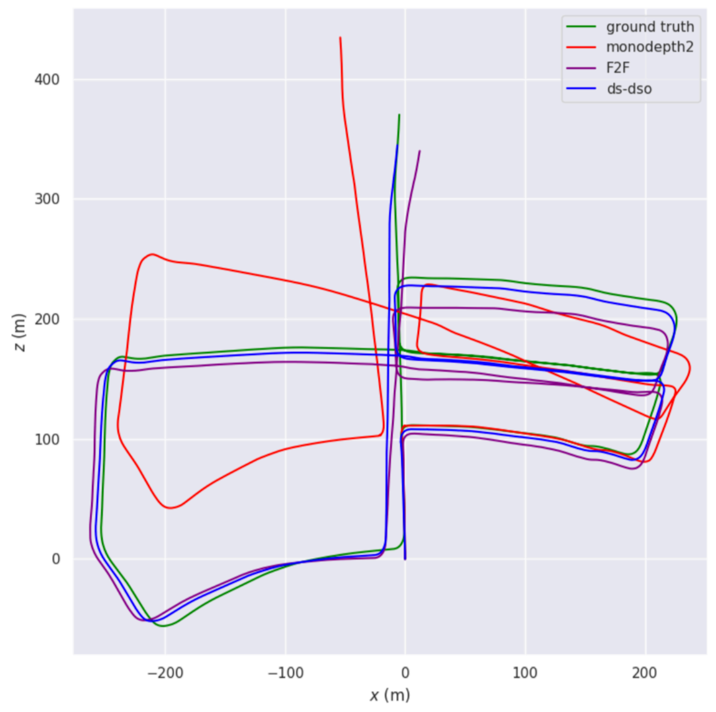


Method	Type	01*	02*	06*	08*	09*	10*	00†	03†	04†	05†	07†	Train	Test
t_{rel} - Average Translational RMSE drift (%) on trajectories of length 100-800m.														
ORB-SLAM-M [37]	Mono	-	-	-	32.40	-	-	25.29	-	-	26.01	24.53	-	27.05
SfMLearner [58]	Mono	35.2	58.8	25.9	21.9	18.8	14.3	66.4	10.8	4.49	18.7	21.3	29.28	16.55
Zhan et al [55]	Mono	-	-	-	-	11.9	12.6	-	-	-	-	-	-	12.30
Bian et al [8]	Mono	-	-	-	-	11.2	10.1	-	-	-	-	-	-	10.7
EPC++(mono) [34]	Mono	-	-	-	-	8.84	8.86	-	-	-	-	-	-	8.85
Ambrus et al [3]	Mono	17.59	6.82	8.93	8.38	6.49	9.83	7.16	7.66	3.8	6.6	11.48	9.67	7.34
Monodepth2 [18]‡	Mono	19.74	3.99	3.80	5.62	5.28	8.47	6.65	8.59	3.62	7.46	9.37	7.82	7.14
DF-VO [56] PnP	Mono	-	-	-	-	7.12	6.83	-	-	-	-	-	-	6.98
DF-VO [56]	Mono	66.98	3.60	1.03	2.23	2.47	1.96	2.25	2.67	1.43	**1.15**	0.93	10.2	2.21
UnDeepVO [31]	Stereo	69.1	5.58	6.20	4.08	7.01	10.6	4.14	5.00	4.49	3.40	3.15	11.68	8.81
SuperDepth [41]	Stereo	13.48	3.48	1.81	2.25	3.74	2.26	6.12	7.90	11.80	4.58	7.60	4.50	7.60
Zhu et al [59]	Stereo	45.5	6.40	3.49	4.08	4.66	6.30	4.95	4.83	2.43	3.97	4.50	8.91	5.48
DF-VO [56]	Stereo	56.76	2.38	1.03	1.60	2.61	2.29	1.96	2.49	1.03	1.10	0.97	8.67	2.45
DVSO [53]	Stereo	1.18	0.84	0.71	1.03	0.83	0.74	0.71	0.77	0.35	0.58	0.73	0.89	0.63
Ours F2F	Mono	17.79	**3.15**	1.88	3.06	2.69	**5.12**	2.76	3.02	1.93	3.30	2.41	5.61	2.68
Ours DS-DSO	Mono	**4.70**	3.62	**0.92**	**2.46**	**2.31**	5.24	**1.83**	**1.21**	**0.76**	**1.84**	**0.54**	**3.21**	**1.24**
r_{rel} - Average Rotational RMSE drift ($°/100m$) on trajectories of length 100-800m.														
ORB-SLAM-M [37]	Mono	-	-	-	12.13	-	-	7.37	-	-	10.62	10.83	-	10.23
Bian et al [8]	Mono	-	-	-	-	3.35	4.96	-	-	-	-	-	-	4.2
Zhan et al [55]	Mono	-	-	-	-	3.60	3.43	-	-	-	-	-	-	3.52
SfMLearner [58]	Mono	2.74	2.74	4.8	2.91	3.21	3.30	6.13	3.92	5.24	4.1	6.65	4.45	3.26
EPC++(mono) [34]	Mono	-	-	-	-	3.34	3.18	-	-	-	-	-	-	3.26
DF-VO [56] PnP	Mono	-	-	-	-	2.43	3.88	-	-	-	-	-	-	3.12
Monodepth2 [18]‡	Mono	1.97	1.56	1.09	1.90	1.60	2.26	2.62	4.77	2.66	2.92	5.38	3.67	1.73
Ambrus et al [3]	Mono	1.01	0.87	0.39	0.61	0.86	0.98	1.70	3.49	0.42	0.90	2.05	0.79	1.71
DF-VO [56]	Mono	17.04	0.52	0.26	**0.30**	**0.30**	0.31	0.58	0.50	0.29	0.30	0.29	2.51	0.31
UnDeepVO [31]	Stereo	1.60	2.44	1.98	1.79	3.61	4.65	1.92	6.17	2.13	1.5	2.48	2.45	4.13
SuperDepth [41]	Stereo	1.97	1.10	0.78	0.84	1.19	1.03	2.72	4.30	1.90	1.67	5.17	1.15	3.15
Zhu et al [59]	Stereo	1.78	1.92	1.02	1.17	1.69	1.59	1.39	2.11	1.16	1.2	1.78	1.50	1.64
DF-VO [56]	Stereo	13.93	0.55	0.30	0.32	0.29	0.37	0.60	0.39	0.25	0.30	0.27	2.11	0.33
DVSO [53]	Stereo	0.11	0.22	0.20	0.25	0.21	0.21	0.24	0.18	0.06	0.22	0.35	0.20	0.21
Ours F2F	Mono	0.72	1.01	0.80	0.76	0.61	1.07	1.17	2.45	1.93	1.11	1.16	0.83	1.56
Ours DS-DSO	Mono	**0.16**	**0.22**	**0.13**	0.31	**0.30**	**0.29**	**0.33**	**0.33**	**0.18**	**0.22**	**0.23**	**0.24**	**0.26**

Table 3. **Comparison of vision-based trajectory estimation with state-of-the-art methods.** The *Type* column indicates the data used at train type. Note: All methods are evaluated on monocular data. Our results are obtained after performing a single Sim(3) alignment step [23] wrt. the ground truth trajectories. **Bold** text denotes the best method trained on monocular data; _ denotes the best overall method. † and ⋆ represent test and respectively train seq. for our method, as well as for [41, 53]. [58, 55, 34, 31, 59, 56] are trained on Sequences 00-08 and tested on Sequences 09 and 10. The numbers for [37] are reported from [31]. ‡ - the numbers of [18] are based on our own implementation.

Visual odometry

- Fair? Why choose monodepth2?



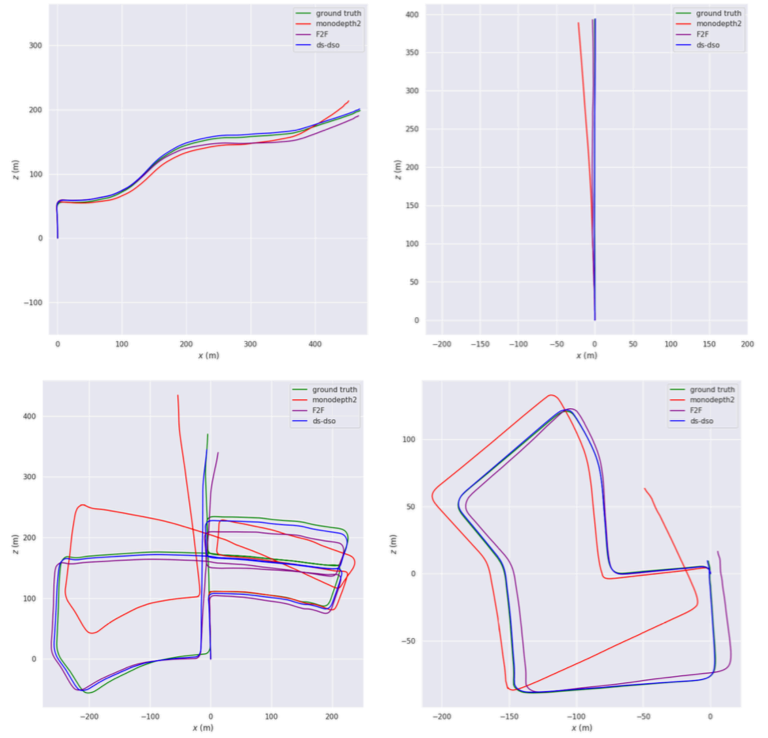
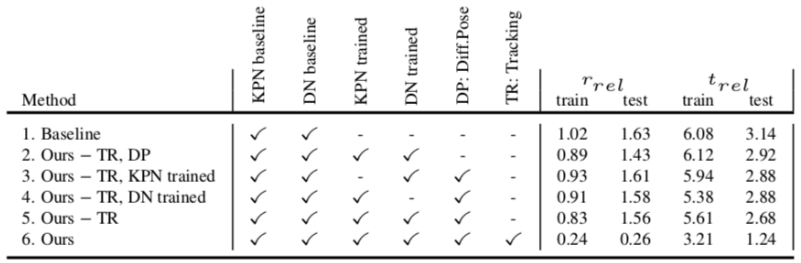


Figure 5. **Qualitative trajectory estimation results on the KITTI Odometry Seq. 03, 04, 05 and 07**. We compare trajectory estimation results obtained via hand-engineered keypoint matching methods against our depth-aware learned keypoint matching, with a common visual odometry backend such as DSO. As illustrated in the figure, our self-supervised method is able to accurately and robustly track stable keypoints for the task of long-term trajectory estimation.

Ablation study

- DSO improves a lot



Method	KPN baseline	DN baseline	KPN trained	DN trained	DP: Diff.Pose	TR: Tracking	r_{rel} train	r_{rel} test	t_{rel} train	t_{rel} test
1. Baseline	✓	✓	-	-	-	-	1.02	1.63	6.08	3.14
2. Ours − TR, DP	✓	✓	✓	✓	-	-	0.89	1.43	6.12	2.92
3. Ours − TR, KPN trained	✓	✓	-	✓	✓	-	0.93	1.61	5.94	2.88
4. Ours − TR, DN trained	✓	✓	✓	-	✓	-	0.91	1.58	5.38	2.88
5. Ours − TR	✓	✓	✓	✓	✓	-	0.83	1.56	5.61	2.68
6. Ours	✓	✓	✓	✓	✓	✓	0.24	0.26	3.21	1.24

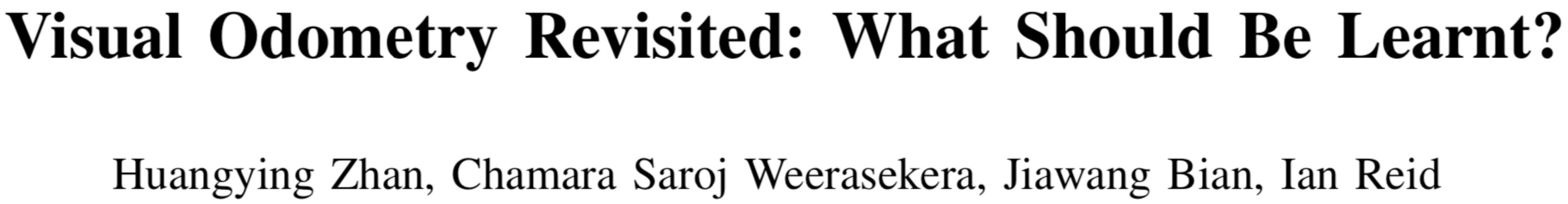
Table 2. **Ablative analysis**. We ablate from our method: **TR** - the tracking component (Section 4), **DP** - the differentiable pose component (Section 3.3), **KPN trained** - the trained version of the KeyPointNet (i.e. using only the baseline), **DN trained** - the trained version of the DepthNet. All t_{rel} results are obtained by performing a Sim(3) alignment step [23].

Future work and discussion

- First work to train detector and descriptor with multi-view adaptation
- Experiments look somehow good
- Only cover half of the visual odometry pipeline
- DS-DSO is not end-to-end optimized

Outlines

- Motivation and problem description
- Brief summary of recent works
- Self-Supervised 3D Keypoint Learning for Ego-motion Estimation
- Optional: Visual Odometry Revisited: What Should Be Learnt?



Visual Odometry Revisited: What Should Be Learnt?

Huangying Zhan, Chamara Saroj Weerasekera, Jiawang Bian, Ian Reid

Ian Reid is the Head of the School of Computer Science at the University of Adelaide

[2] Visual Odometry Revisited: What Should Be Learnt?
H Zhan, CS Weerasekera, J Bian, I Reid - arXiv preprint arXiv:1909.09803, 2019

Motivation

Method overview

- Image
- Depth prediction
- Forward/ backward optical flow
- Optical flow consistency
- Sparse correspondences from optical flow consistency
- 3D-2D correspondences to estimate [R, t]

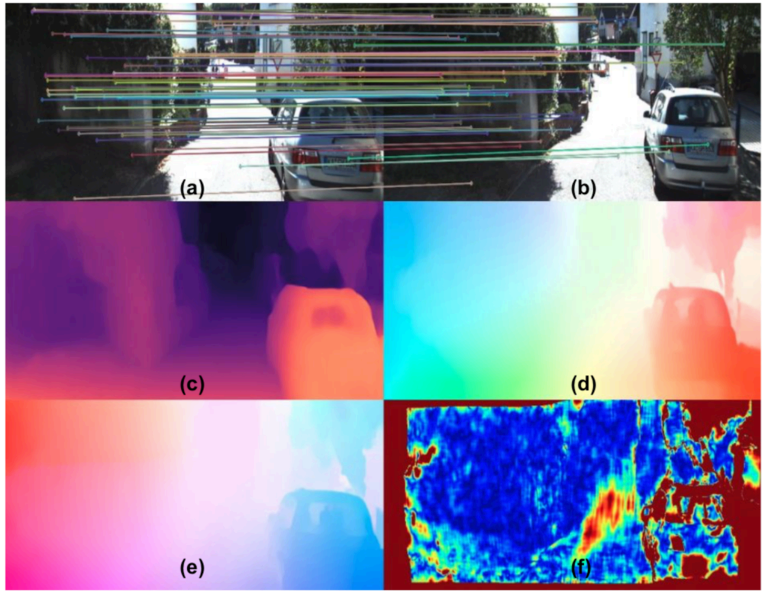


Fig. 1. Inputs and intermediate CNN outputs of the system. (a, b): Current and previous monocular input images with examples of auto-selected 2D-2D matches; (c) Single-view depth prediction; (d, e): Forward and backward optical flow prediction; (f) Forward-backward flow consistency using (d,e).

Method details and analysis

- Depth prediction
  - Unsupervised photometric loss
  - Training in stereo for scale consistency
- Optical flow
  - LiteFlowNet
  - Pre-trained in synthetic data
  - Unsupervised photometric loss

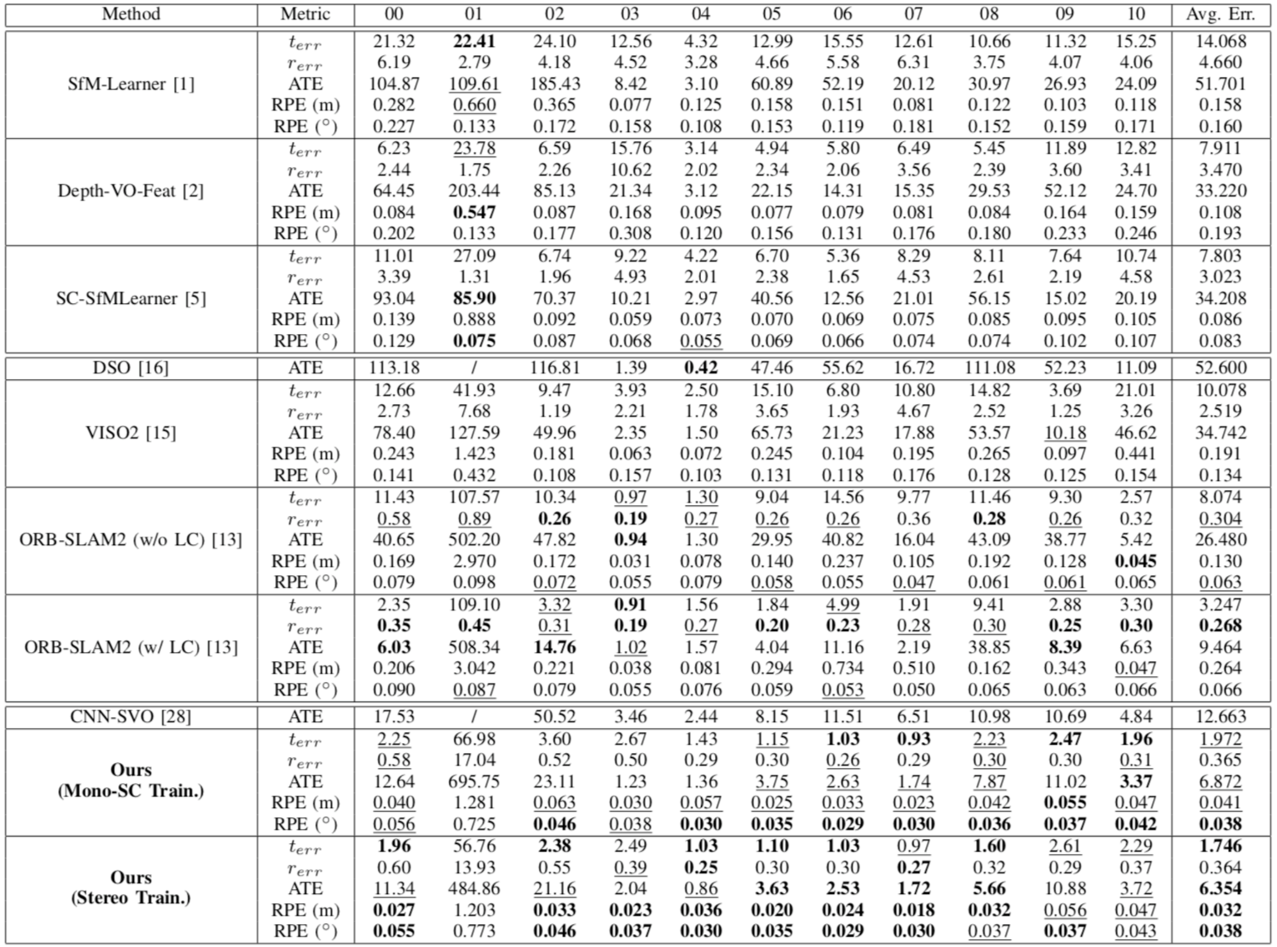
[35] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth prediction,” October 2019.

[48] T.-W. Hui, X. Tang, and C. C. Loy, “Liteflownet: A lightweight con-volutional neural network for optical flow estimation,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018, pp. 8981–8989.

Experiments

TABLE I

QUANTITATIVE RESULT ON KITTI ODOMETRY SEQ. 00-10. THE BEST RESULT IS IN BOLD AND SECOND BEST IS UNDERLINED.



Method	Metric	00	01	02	03	04	05	06	07	08	09	10	Avg. Err.
SfM-Learner [1]	t_{err}	21.32	**22.41**	24.10	12.56	4.32	12.99	15.55	12.61	10.66	11.32	15.25	14.068
	r_{err}	6.19	2.79	4.18	4.52	3.28	4.66	5.58	6.31	3.75	4.07	4.06	4.660
	ATE	104.87	109.61	185.43	8.42	3.10	60.89	52.19	20.12	30.97	26.93	24.09	51.701
	RPE (m)	0.282	0.660	0.365	0.077	0.125	0.158	0.151	0.081	0.122	0.103	0.118	0.158
	RPE (°)	0.227	0.133	0.172	0.158	0.108	0.153	0.119	0.181	0.152	0.159	0.171	0.160
Depth-VO-Feat [2]	t_{err}	6.23	23.78	6.59	15.76	3.14	4.94	5.80	6.49	5.45	11.89	12.82	7.911
	r_{err}	2.44	1.75	2.26	10.62	2.02	2.34	2.06	3.56	2.39	3.60	3.41	3.470
	ATE	64.45	203.44	85.13	21.34	3.12	22.15	14.31	15.35	29.53	52.12	24.70	33.220
	RPE (m)	0.084	**0.547**	0.087	0.168	0.095	0.077	0.079	0.081	0.084	0.164	0.159	0.108
	RPE (°)	0.202	0.133	0.177	0.308	0.120	0.156	0.131	0.176	0.180	0.233	0.246	0.193
SC-SfMLearner [5]	t_{err}	11.01	27.09	6.74	9.22	4.22	6.70	5.36	8.29	8.11	7.64	10.74	7.803
	r_{err}	3.39	1.31	1.96	4.93	2.01	2.38	1.65	4.53	2.61	2.19	4.58	3.023
	ATE	93.04	**85.90**	70.37	10.21	2.97	40.56	12.56	21.01	56.15	15.02	20.19	34.208
	RPE (m)	0.139	0.888	0.092	0.059	0.073	0.070	0.069	0.075	0.085	0.095	0.105	0.086
	RPE (°)	0.129	**0.075**	0.087	0.068	0.055	0.069	0.066	0.074	0.074	0.102	0.107	0.083
DSO [16]	ATE	113.18	/	116.81	1.39	**0.42**	47.46	55.62	16.72	111.08	52.23	11.09	52.600
VISO2 [15]	t_{err}	12.66	41.93	9.47	3.93	2.50	15.10	6.80	10.80	14.82	3.69	21.01	10.078
	r_{err}	2.73	7.68	1.19	2.21	1.78	3.65	1.93	4.67	2.52	1.25	3.26	2.519
	ATE	78.40	127.59	49.96	2.35	1.50	65.73	21.23	17.88	53.57	10.18	46.62	34.742
	RPE (m)	0.243	1.423	0.181	0.063	0.072	0.245	0.104	0.195	0.265	0.097	0.441	0.191
	RPE (°)	0.141	0.432	0.108	0.157	0.103	0.131	0.118	0.176	0.128	0.125	0.154	0.134
ORB-SLAM2 (w/o LC) [13]	t_{err}	11.43	107.57	10.34	0.97	1.30	9.04	14.56	9.77	11.46	9.30	2.57	8.074
	r_{err}	0.58	0.89	**0.26**	**0.19**	0.27	0.26	0.26	0.36	**0.28**	0.26	0.32	0.304
	ATE	40.65	502.20	47.82	**0.94**	1.30	29.95	40.82	16.04	43.09	38.77	5.42	26.480
	RPE (m)	0.169	2.970	0.172	0.031	0.078	0.140	0.237	0.105	0.192	0.128	**0.045**	0.130
	RPE (°)	0.079	0.098	0.072	0.055	0.079	0.058	0.055	0.047	0.061	0.061	0.065	0.063
ORB-SLAM2 (w/ LC) [13]	t_{err}	2.35	109.10	3.32	**0.91**	1.56	1.84	4.99	1.91	9.41	2.88	3.30	3.247
	r_{err}	**0.35**	**0.45**	0.31	**0.19**	0.27	**0.20**	**0.23**	0.28	0.30	**0.25**	**0.30**	**0.268**
	ATE	**6.03**	508.34	**14.76**	1.02	1.57	4.04	11.16	2.19	38.85	**8.39**	6.63	9.464
	RPE (m)	0.206	3.042	0.221	0.038	0.081	0.294	0.734	0.510	0.162	0.343	0.047	0.264
	RPE (°)	0.090	0.087	0.079	0.055	0.076	0.059	0.053	0.050	0.065	0.063	0.066	0.066
CNN-SVO [28]	ATE	17.53	/	50.52	3.46	2.44	8.15	11.51	6.51	10.98	10.69	4.84	12.663
Ours (Mono-SC Train.)	t_{err}	2.25	66.98	3.60	2.67	1.43	1.15	**1.03**	**0.93**	2.23	**2.47**	**1.96**	1.972
	r_{err}	0.58	17.04	0.52	0.50	0.29	0.30	0.26	0.29	0.30	0.30	0.31	0.365
	ATE	12.64	695.75	23.11	1.23	1.36	3.75	2.63	1.74	7.87	11.02	**3.37**	6.872
	RPE (m)	0.040	1.281	0.063	0.030	0.057	0.025	0.033	0.023	0.042	**0.055**	0.047	0.041
	RPE (°)	0.056	0.725	**0.046**	0.038	**0.030**	**0.035**	**0.029**	**0.030**	**0.036**	**0.037**	**0.042**	**0.038**
Ours (Stereo Train.)	t_{err}	**1.96**	56.76	**2.38**	2.49	**1.03**	**1.10**	**1.03**	0.97	**1.60**	2.61	2.29	**1.746**
	r_{err}	0.60	13.93	0.55	0.39	**0.25**	0.30	0.30	**0.27**	0.32	0.29	0.37	0.364
	ATE	11.34	484.86	21.16	2.04	0.86	**3.63**	**2.53**	**1.72**	**5.66**	10.88	3.72	**6.354**
	RPE (m)	**0.027**	1.203	**0.033**	**0.023**	**0.036**	**0.020**	**0.024**	**0.018**	**0.032**	0.056	0.047	**0.032**
	RPE (°)	**0.055**	0.773	**0.046**	**0.037**	**0.030**	**0.035**	**0.029**	**0.030**	0.037	**0.037**	0.043	**0.038**

Future work and discussion

References

[1] Self-Supervised 3D Keypoint Learning for Ego-motion Estimation

J Tang, R Ambrus, V Guizilini, S Pillai, H Kim, A Gaidon - arXiv preprint arXiv …, 2019

[2] Visual Odometry Revisited: What Should Be Learnt?

H Zhan, CS Weerasekera, J Bian, I Reid - arXiv preprint arXiv:1909.09803, 2019

[3] Beyond tracking: Selecting memory and refining poses for deep visual odometry

F Xue, X Wang, S Li, Q Wang, J Wang, H Zha - … of the IEEE Conference on Computer …, 2019

[4] Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry

N Yang, R Wang, J Stuckler, D Cremers - … of the European Conference on Computer …, 2018

Thank you

ALLPPT.com

Backup

Pipeline

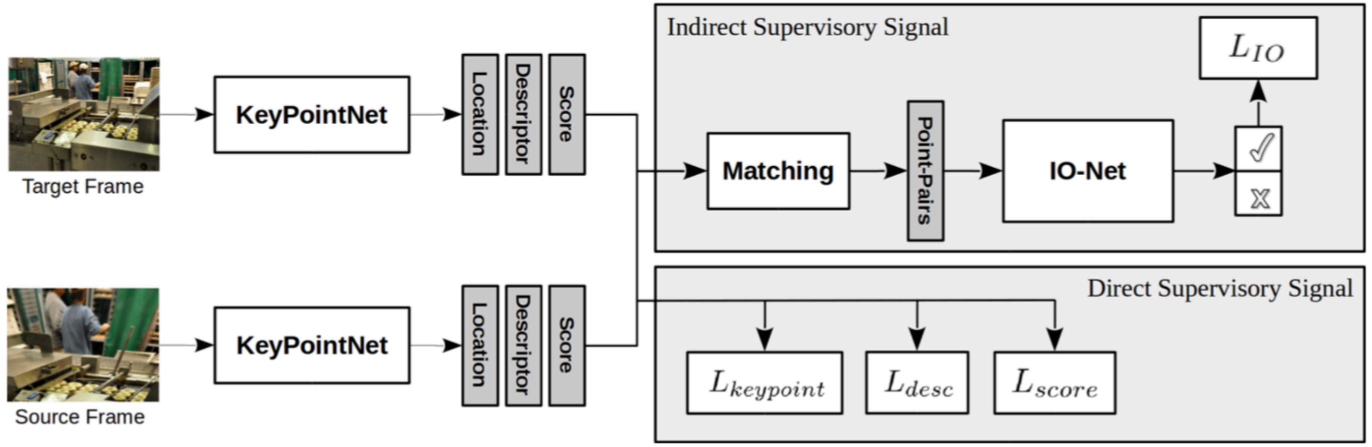


Figure 1: Our proposed framework for self-supervised keypoint detector and descriptor learning using *KeyPointNet* and *IO-Net*. The KeyPointNet is optimized in an end-to-end differentiable manner by imposing an explicit loss on each of the 3 target outputs (score, location and descriptor). Additionally, the IO-Net produces an indirect supervisory signal to KeyPointNet targets by propagating gradients from the classification of matching input point-pairs.

Neural Outlier Rejection for Self-Supervised Keypoint Learning

J Tang, H Kim, V Guizilini, S Pillai, R Ambrus - arXiv preprint arXiv:1912.10615, 2019

Network architecture

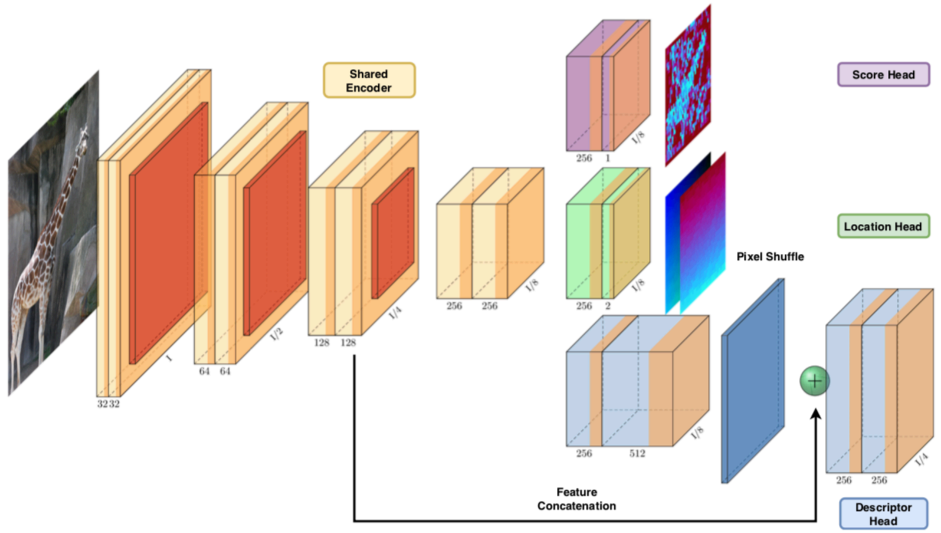


Figure 2: The proposed *KeyPointNet* architecture leverages a shared-encoder backbone with three output heads for the regression of keypoint locations ($\mathbf{p}$), scores ($\mathbf{s}$) and descriptions ($\mathbf{f}$). To further improve keypoint description performance, KeyPointNet produces higher-resolution feature-maps for the keypoint descriptors via efficient sub-pixel convolutions at the keypoint descriptor head.

Neural Outlier Rejection for Self-Supervised Keypoint Learning

J Tang, H Kim, V Guizilini, S Pillai, R Ambrus - arXiv preprint arXiv:1912.10615, 2019

Keypoint regression

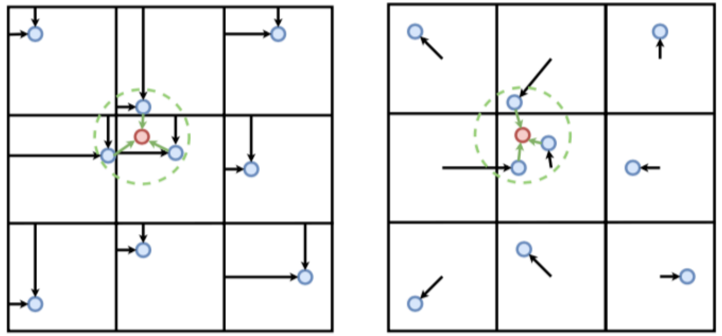


Figure 3: Cross-border detection illustration. We illustrate how a warped point (in red) can be associated to multiple predicted points (in blue) based on a distance threshold (dashed circle). (a) The keypoint location head in Christiansen et al. (2019) forces keypoint predictions in the same cell, causing convergence issues since these points can only be pulled up to the cell-boundary. (b) Instead, in our method, we design the network to predict the localization from the cell-center and allow keypoints to be outside the border for better matching and aggregation.

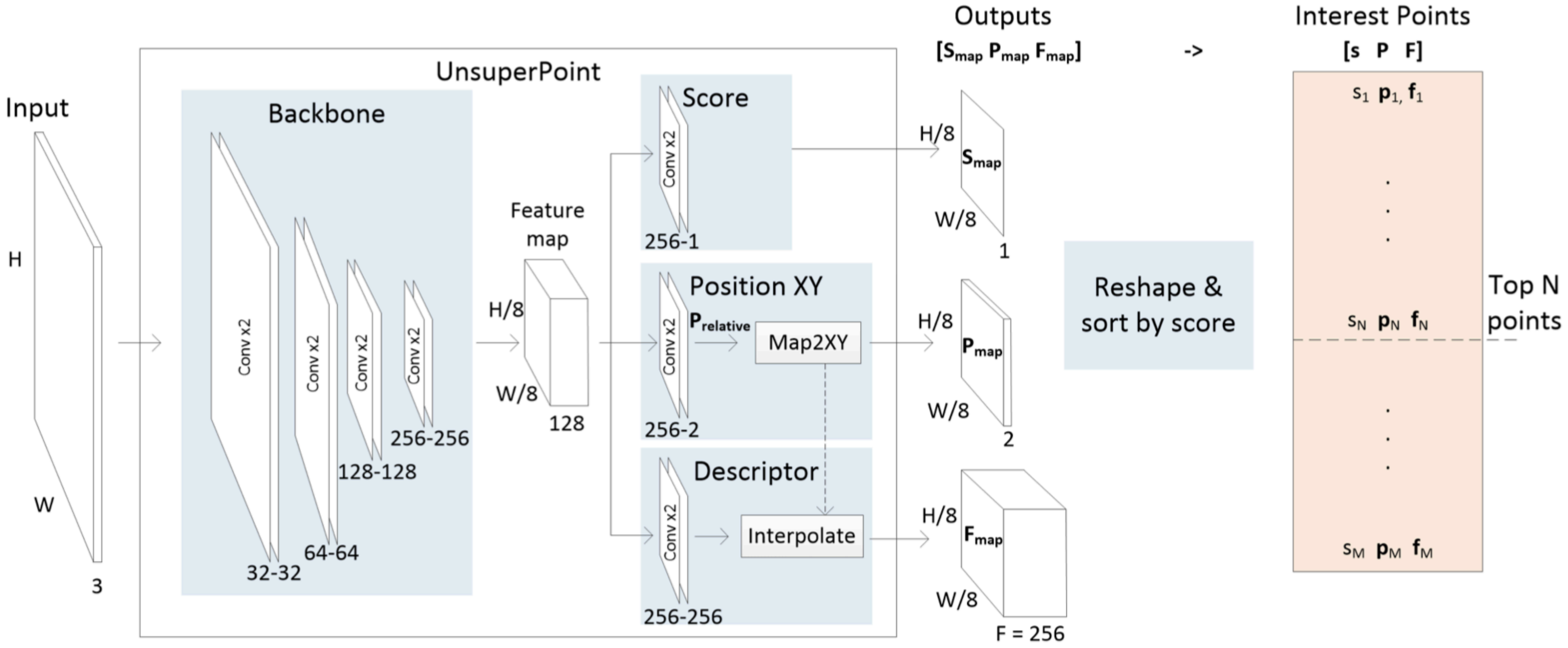


Figure 1: UnsuperPoint takes an input image and outputs an interest point vector. The score, position and descriptor share most computations by a shared backbone. Each interest point m is described by a score s_m, a position $\mathbf{p}_m$ and a descriptor $\mathbf{f}_m$.