Self-Supervised Deep Visual Odometry with Online Adaptation

Shunkai Li Xin Wang Yingdian Cao Fei Xue Zike Yan Hongbin Zha
Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
PKU-SenseTime Machine Vision Joint Lab
{lishunkai, xinwang_cis, yingdianc, feixue, zike.yan}@pku.edu.cn zha@cis.pku.edu.cn

Abstract

Self-supervised VO methods have shown great success in jointly estimating camera pose and depth from videos. However, like most data-driven methods, existing VO networks suffer from a notable decrease in performance when confronted with scenes different from the training data, which makes them unsuitable for practical applications. In this paper, we propose an online meta-learning algorithm to enable VO networks to continuously adapt to new environments in a self-supervised manner. The proposed method utilizes convolutional long short-term memory (convLSTM) to aggregate rich spatial-temporal information in the past. The network is able to memorize and learn from its past experience for better estimation and fast adaptation to the current frame. When running VO in the open world, in order to deal with the changing environment, we propose an online feature alignment method by aligning feature distributions at different time. Our VO network is able to seamlessly adapt to different environments. Extensive experiments on unseen outdoor scenes, virtual to real world and outdoor to indoor environments demonstrate that our method consistently outperforms state-of-the-art self-supervised VO baselines considerably.

1. Introduction

Simultaneous localization and mapping (SLAM) and visual odometry (VO) play a vital role for many real-world applications, such as autonomous driving, robotics and mixed reality. Classic SLAM/VO [13, 14, 17, 29] methods perform well in regular scenes but fail in challenging conditions (*e.g.* dynamic objects, occlusions, textureless regions) due to their reliance on low-level features. Since deep learning is able to extract high-level features and infer in an end-to-end fashion, learning-based VO [22, 40, 41, 44] methods have been proposed in recent years to alleviate the limitation of classic hand-engineered algorithms.

However, learning-based VO suffers from a notable decrease in accuracy when confronted with scenes different from the training dataset [8, 37] (Fig. 1). When applied a pre-trained VO network to the *open world*, the inability to generalize itself to new scenes presents a serious problem for its practical applications. This requires the VO network to continuously adapt to the new environment.

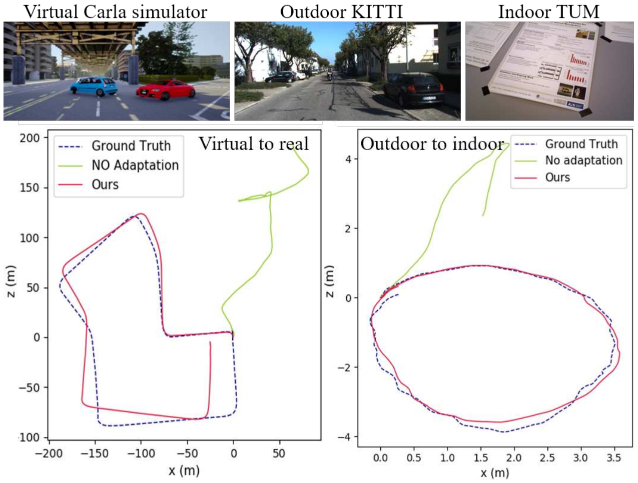

Figure 1. We demonstrate the domain shift problem for self-supervised VO. Previous methods fail to generalize when the test data are different from the training data. In contrast, our method performs well when tested on changing environments, which demonstrates the advantage of fast online adaptation

In contrast to fine-tuning a pre-trained network with ground truth data on the target domain [37], it is unlikely to collect enough data in advance when running VO in the open world. This requires the network to adapt itself in *real-time* to changing environments. In this online learning setting, there is no explicit distinction between training and testing phases — **we learn as we perform**. This is much different from conventional learning methods where a pre-trained model is fixed during inference.

During online adaptation, the VO network can only learn from the current data instead of the entire training data with batch training and multiple epoches [11]. The learning objective is to find an optimal model that is well adapted to the current data. However, because of the limited temporal perceptive field [26], the current optimal model may not be well suited for subsequent frames. This makes the optimal parameters oscillate with time, leading to slow convergence during online adaptation [9, 11, 20].

In order to address these issues, we propose an online meta-learning scheme for self-supervised VO that achieves online adaptation. The proposed method motivates the network to perform consistently well at different time by incorporating online adaptation process into the learning objective. Besides, the past experience can be used to accelerate the adaptation to a new environment. Therefore, instead of learning only from the current data, we employ convolutional long short-term memory (convLSTM) to aggregate rich spatial-temporal information in the video that enables the network to use past experience for better estimation and also adapt quickly to the current frame. In order to achieve fast adaptation in changing environments, we propose a feature alignment method to align non-stationary feature distributions at different time. The proposed network automatically adapts to changing environments without ground truth data collected in advance for external supervision. Our contributions can be summarized as follows:

- We propose an online meta-learning algorithm for VO to continuously adapt to unseen environments in a self-supervised manner.
- The VO network utilizes past experience incorporated by convLSTM to achieve better estimation and adapt quickly to the current frame.
- We propose a feature alignment method to deal with the changing data distributions in the open world.

Our VO network achieves 32 FPS on a Geforce 1080Ti GPU with online refinement, making it adapt in real-time for practical applications. We evaluate our algorithm across different domains, including outdoor, indoor and synthetic environments, which consistently outperforms state-of-the-art self-supervised VO baselines.

2. Related works

Learning-based VO has been widely studied in recent years with the advent of deep learning and many methods with promising results have been proposed. Inspired by the framework of parallel tracking and mapping in classic SLAM/VO, DeepTAM [43] utilizes two networks for pose and depth estimation simultaneously. DeepVO [38] uses recurrent neural network (RNN) to leverage sequential correlations to estimate poses recurrently. However, these methods require ground truth which is expensive or impractical to obtain. To avoid the need of annotated data, self-supervised VO has been recently developed. SfMLearner [44] utilizes the 3D geometric constraint of pose and depth to learn by minimizing photometric loss. Yin *et al*. [41] and Ranjan *et al*. [32] extend this idea to joint estimation of pose, depth and optical flow to handle non-rigid cases which are against static-scene assumption. These methods focus on mimicking local structure from motion (SfM) with image pairs, but fail to exploit spatial-temporal correlations over long sequence. SAVO [22] formulates VO as a sequential generative task and utilizes RNN to reduce scale drift significantly. In this paper, we adopt the same idea as SfMLearner [44] and SAVO [22].

Online adaptation Most machine learning models suffer from a significant reduce in performance when the test data are different from the training set. An effective solution to alleviate this domain shift issue is online learning [35], where data are processed sequentially and data distribution changes continuously. Previous methods use online gradient update [12] and probabilistic filtering [6]. Recently, domain adaptation has been widely studied in computer vision. Long *et al*. [23] propose Maximum Mean Discrepancy loss to reduce the domain shift. Several works [5, 33] utilize Generative Adversarial Networks (GAN) to directly transfer images in the target domain to the source domain (*e.g.* day to night or winter to summer). Inspired by [5, 7], we propose a feature alignment method for online adaptation.

Meta-learning, or learning to learn, is a continued interest in machine learning. It exploits inherent structures in data to learn more effective learning rules for fast domain adaptation [27, 36]. A popular approach is to train a meta-learner that learns how to update the network [4, 15]. Finn *et al*. [15, 16] proposed Model Agnostic Meta-Learning (MAML) that constrains the learning rule for the model and uses stochastic gradient descent to quickly adapt networks to new tasks. This simple yet effective formulation has been widely used to adapt deep networks to unseen environments [1, 2, 21, 30, 39]. Our proposed method is most relevant to MAML, which extends it to the self-supervised, online learning setting.

3. Problem setup

3.1. Self-supervised VO

Our self-supervised VO follows the similar idea of SfMLearner [44] and SAVO [22] (shown in Fig. 2). The DepthNet predicts depth $\hat{D}_t$ of the current frame I_t. The PoseNet takes stacked monocular images I_{t-1}, I_t and $\hat{D}_{t-1}, \hat{D}_t$ to regress relative pose $\hat{T}_t^{t-1}$. Then view synthesis is applied to reconstruct $\hat{I}_t$ by differentiable image warping:

$$p_{t-1} \sim K\hat{T}_t^{t-1}\hat{D}_t(p_t)K^{-1}p_t, \tag{1}$$

where p_{t-1}, p_t are the homogeneous coordinates of a pixel in I_{t-1} and I_t, respectively. K denotes camera intrinsics. The MaskNet predicts a per-pixel mask $\hat{M}_t$ [44] according to the warping residuals $\|\hat{I}_t - I_t\|_1$.

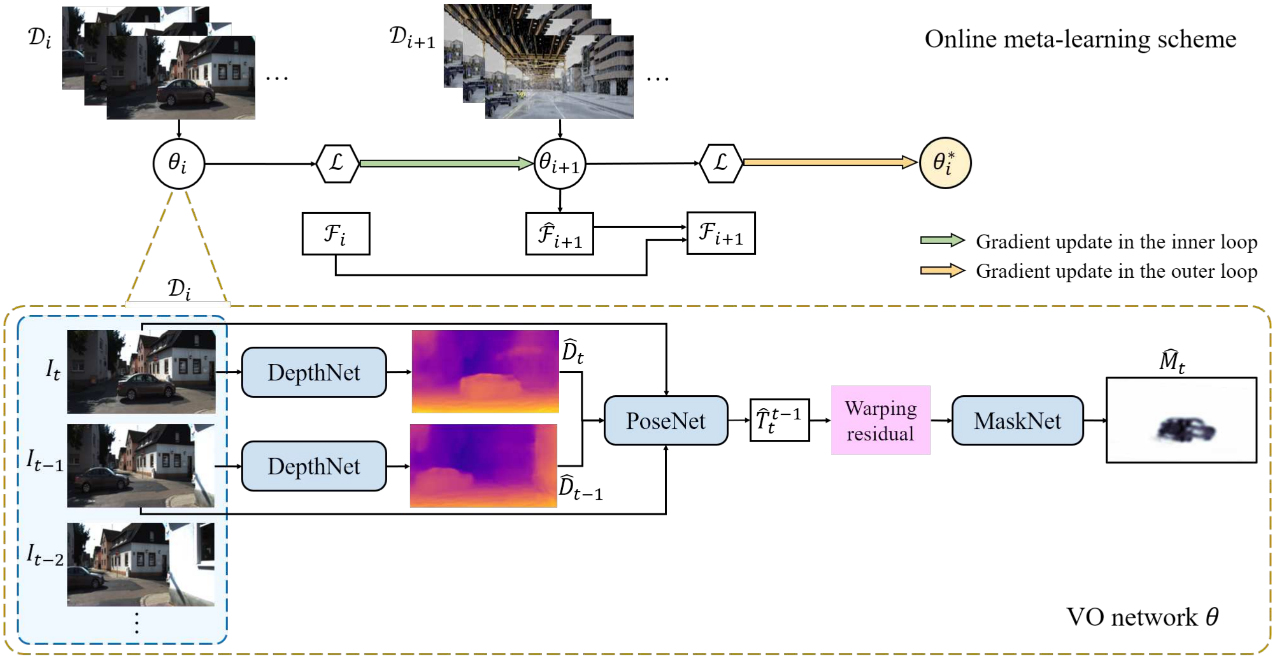

Figure 2. The framework of our method. The VO network estimates pose $\hat{T}_t^{t-1}$, depth $\hat{D}_t, \hat{D}_{t-1}$ and mask $\hat{M}_t$ from image sequences $\mathcal{D}_i$. At each iteration i, the network parameters θ_i are updated according to the loss $\mathcal{L}$ and performs inference for $\mathcal{D}_{i+1}$ at next time. The network learns to find a set of weights θ_i^* that perform well both for $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$. During online learning, spatial-temporal information is aggregated by convLSTM and feature alignment is adopted to align feature distributions $\hat{\mathcal{F}}_i, \hat{\mathcal{F}}_{i+1}$ at different time for fast adaptation

3.2. Online adaptation

As shown in Fig. 1, the performance of VO networks is fundamentally limited by their generalization ability when confronted with scenes different from the training data. The reason is they are designed under a **closed world** assumption: the training data $\mathcal{D}^{train}$ and test data $\mathcal{D}^{test}$ are i.i.d. sampled from a common dataset with fixed distribution. However, when running a pre-trained VO network in the **open world**, images are continuously collected in changing scenes. In this sense, the training and test data no longer share similar visual appearances, and the data at the current view may be different from previous views. This requires the network to online adapt to changing environments.

Given a model θ pretrained on $\mathcal{D}^{train}$, a naive approach for online learning is to update parameters θ by computing loss $\mathcal{L}$ on the current data $\mathcal{D}_i$:

$$\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i), \tag{2}$$

where $\theta_0 = \theta$ and α is the learning rate. Despite its simplicity, this approach has several drawbacks. The temporal perceptive field of the learning objective $\mathcal{L}(\theta_i, \mathcal{D}_i)$ is 1, which means it accounts only for the current input $\mathcal{D}_i$ and has no correlation with previous data. The optimal solution for current $\mathcal{D}_i$ is likely to be unsuitable for subsequent inputs. Therefore, the gradients $\nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i)$ at different iterations are stochastic without consistency [9, 26]. This leads to slow convergence and may introduce negative bias in the learning procedure.

4. Method

In order to address these issues, we propose to exploit *correlations* of different time for fast online adaptation. Our framework is illustrated in Fig. 2. The VO network θ_i takes N consecutive frames in the sliding window $\mathcal{D}_i$ to estimate pose and depth in a self-supervised manner (Sec. 3.1). Then it is updated according to the loss $\mathcal{L}$ and infers for frames $\mathcal{D}_{i+1}$ at the next time. The network learns to find a set of weights θ_i^* to perform well both for $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ (Sec. 4.1). During online learning, spatial-temporal information is incorporated by convLSTM (Sec. 4.2) and feature alignment is adopted (Sec. 4.3) for fast adaptation.

4.1. Self-supervised online meta-learning

In contrast to $\mathcal{L}(\theta_i, \mathcal{D}_i)$, we extend the online learning objective to $\mathcal{L}(\theta_{i+1}, \mathcal{D}_{i+1})$, which can be written as:

$$\min_{\theta_i} \mathcal{L}(\theta_i - \alpha \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i), \mathcal{D}_{i+1}). \tag{3}$$

Different from naive online learning, the temporal perceptive field of Eq. 3 becomes 2. It optimizes the performance on $\mathcal{D}_{i+1}$ after adapting to the task on $\mathcal{D}_i$. The insight is instead of minimizing the *training* error $\mathcal{L}(\theta_i, \mathcal{D}_i)$ on the current iteration i, we try to minimize the *test* error on the next iteration. Our formulation directly incorporates online adaptation into the learning objective, which motivates the network to learn θ_i at i to perform better at next time $i+1$.

Our objective of learning to adapt is similar in spirit to that of Model Agnostic Meta Learning (MAML) [15]:

$$\min_{\theta} \sum_{\tau \in \mathcal{T}} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_{\tau}^{train}), \mathcal{D}_{\tau}^{val}), \tag{4}$$

which aims to minimize the evaluation (adaptation) error on the validation set instead of minimizing the training error on the training set. τ denotes tasks sampled from the task set $\mathcal{T}$. More details of MAML can be found in [15].

As a nested optimization problem, our objective function is optimized via a two-stage gradient descent. At each iteration i, we take N consecutive frames in the sliding window as a mini-dataset $\mathcal{D}_i$ (shown within the blue area in Fig. 2):

$$\mathcal{D}_i = \{I_t, I_{t-1}, I_{t-2}, \ldots, I_{t-N+1}\}. \tag{5}$$

In the inner loop of Eq. 3, we evaluate the performance of VO in D_i by self-supervised loss $\mathcal{L}$ and update parameters θ_i according to Eq. 2. Then, in the outer loop, we evaluate the performance of the updated model θ_{i+1} on subsequent frames $\mathcal{D}_{i+1}$. We mimic this continuous adaptation process on both training and online test phases. During training, we minimize the sum of losses by Eq. 3 across all sequences in the training dataset, which motivates the network to learn base weights θ that enables fast online adaptation.

In order to provide more intuition on what it learns and the reason for fast adaptation, we take Taylor expansion on our training objective:

$$\begin{aligned}
&\min_{\theta_i} \mathcal{L}(\theta_i - \alpha \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i), \mathcal{D}_{i+1}) \\
&\approx \min_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_{i+1}) - \alpha \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i) \cdot \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_{i+1}) \\
&\quad + H_{\theta_i} \cdot [\alpha \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i)]^2 + \ldots \\
&\approx \min_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_{i+1}) - \alpha \left\langle \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_i), \nabla_{\theta_i} \mathcal{L}(\theta_i, \mathcal{D}_{i+1}) \right\rangle,
\end{aligned} \tag{6}$$

where H_{θ_i} denotes Hessian matrix and $\langle \cdot, \cdot \rangle$ denotes inner product. Since most neural networks use ReLU activations, the networks are locally linear, thus the second order derivative equals 0 in most cases [28]. Therefore, $H_{\theta_t} \approx 0$ and higher order terms are also omitted.

As shown in Eq. 6, the network learns to minimize the prediction error $\mathcal{L}(\theta_i, \mathcal{D}_{i+1})$ with θ_i while maximizing the similarity between the gradients at $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$. Since the camera is continuously moving, the scenes $\mathcal{D}_i, \mathcal{D}_{i+1}$ may vary from different time. Naive online learning treats different scenes *independently* by fitting only the current scene but ignores the way to perform VO in different scenes are similar. As gradient indicates the direction to update the network, this leads to inconsistent gradients at $i, i+1$ and slow convergence. In contrast, the second term enforces consistent gradient directions by *aligning* gradient for $\mathcal{D}_{i+1}$ with previous information, indicating that we are training the network θ_i at i to perform consistently well for both i and $i+1$. This meta-learning scheme alleviates stochastic gradient problem in online learning. Eq. 6 describes the dynamics of sequential learning in non-stationary scenes. The network learns to adjust at current state by $\mathcal{L}(\theta_i, \mathcal{D}_i)$ to better perform at next time. Consequently, the learned θ is less sensitive to the non-stationary data distributions of sequential inputs, enabling fast adaptation to unseen environments.

4.2. Spatial-temporal aggregation

As stated in Sec. 1, online learning suffers from slow convergence due to the inherent limitation of temporal perceptive field. In order to make online updating more effective, we let the network perform current estimation based on previous information. Besides, predicting pose from only image pairs is prone to error accumulation. This trajectory drift problem can be mitigated by exploiting spatial-temporal correlations over long sequence [22, 40].

In this paper, we use convolutional LSTM (convLSTM) to achieve fast adaptation and reduce accumulated error. As shown in Fig. 3, we embed recurrent units into the encoder of DepthNet and PoseNet to allow the convolutional network to leverage not only spatial but also temporal information for depth and pose estimation. The length N of convLSTM is the number of frames in $\mathcal{D}_i$. ConvLSTM acts as the memory of the network. As new frames are processed, the network is able to memorize and learn from its past experience, so as to update parameters to quickly adapt to unseen environments. This approach not only enforces correlations among different time steps, but also learns the temporally dynamic nature of the moving camera from video inputs.

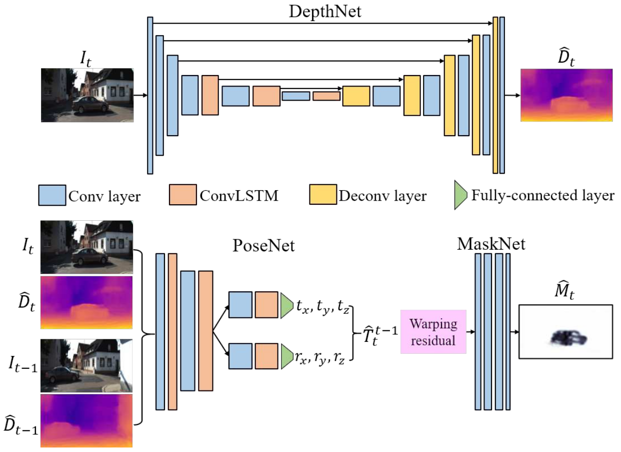

Figure 3. Network architecture of DepthNet, PoseNet and MaskNet in self-supervised VO framework. The height of each block represents the size of its feature maps

4.3. Feature alignment

One basic assumption of conventional machine learning is that the training and test data are independently and identically (i.i.d.) drawn from the same distribution. However, this assumption does not hold when running VO in the open world, since the test data (target domain) are usually different from the training data (source domain). Besides, as the camera is continuously moving in the changing environment, the captured scenes $\mathcal{D}_i$ also vary in time. As highlighted in [7, 25], aligning feature distributions of two domains will improve performance in domain adaptation.

Inspired by [7], we extend this domain adaptation method to the online learning setting by aligning feature distributions in different time. When training on the source domain, we collect the statistics of features $f_j \in \{f_1, ..., f_n\}$ in a feature map tensor by Layer Normalization (LN) [3]:

$$\begin{aligned} \mathcal{F}_s &= (\mu_s, \sigma_s^2), \\ \mu_s = \frac{1}{n}\sum_{j=1}^{n} f_j, \quad \sigma_s^2 &= \frac{1}{n}\sum_{j=1}^{n}(f_j - \mu_s)^2, \\ n =& H \times W \times C, \end{aligned} \tag{7}$$

where H, W, C are the height, width and channels of each feature map. When adapted to the target domain, we initialize feature statistics at $i = 0$:

$$\mathcal{F}_0 = \mathcal{F}_s. \tag{8}$$

Then at each iteration i, feature statistics $\hat{\mathcal{F}_i} = (\hat{\mu_i}, \hat{\sigma_i^2})$ are computed by Eq. 7. Given previous statistics $\mathcal{F}_{i-1} = (\mu_{i-1}, \sigma_{i-1}^2)$, feature distribution at i is aligned by:

$$\begin{aligned} \mu_i &= (1-\beta)\mu_{i-1} + \beta\hat{\mu}_i, \\ \sigma_i^2 &= (1-\beta)\sigma_{i-1}^2 + \beta\hat{\sigma}_i^2, \end{aligned} \tag{9}$$

where β is a hyperparameter. After feature alignment, the features $f_j \in \{f_1, ..., f_n\}$ are normalized to [3]:

$$\hat{f_j} = \gamma \frac{f_j - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \delta, \tag{10}$$

where ϵ is a small constant for numerical stability. γ and δ are the learnable scale and shift in normalization layers [3].

The insight of this approach is to enforce correlation of non-stationary feature distributions in changing environments. Learning algorithms perform well when feature distribution of the test data is the same as the training data. When changed to a new environment, despite the extracted features are different, we deem that feature distributions of two domains should be the same (Eq. 8). Despite the view is changing when running VO in an open world, $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ are observed continuously in time, thus their feature distributions should be similar (Eq. 9). This feature normalization and alignment approach acts as *regularization* that simplifies the learning process, which makes the learned weights θ consistent for non-stationary environments.

4.4. Loss functions

Our self-supervised loss $\mathcal{L}$ is the same as most previous methods. It consists of:

Appearance loss We measure the reconstructed image $\hat{I}$ by photometric loss and structural similarity metric (SSIM):

$$\begin{aligned} \mathcal{L}_a = \lambda_m \mathcal{L}_m(\hat{M}) + (1-\alpha_s)\frac{1}{N}\sum \hat{M}\|\hat{I} - I\|_1 \\ + \frac{1}{N}\sum_{x,y}\alpha_s \frac{1 - \text{SSIM}(\hat{I}(x,y), I(x,y))}{2}. \end{aligned} \tag{11}$$

The regularization term $\mathcal{L}_m(\hat{M})$ prevents the learned mask $\hat{M}$ converges to a trivial solution [44]. The filter size of SSIM is set 5×5 and α_s is set 0.85.

Depth regularization We introduce an edge-aware loss to enforce discontinuity and local smoothness in depth:

$$\begin{aligned} \mathcal{L}_r =& \frac{1}{N}\sum_{x,y}\|\nabla_x \hat{D}(x,y)\| e^{-\|\nabla_x I(x,y)\|} + \\ & \|\nabla_y \hat{D}(x,y)\| e^{-\|\nabla_y I(x,y)\|}. \end{aligned} \tag{12}$$

Thus the self-supervised loss $\mathcal{L}$ is:

$$\mathcal{L} = \lambda_a \mathcal{L}_a + \lambda_r \mathcal{L}_r. \tag{13}$$

5. Experiments

5.1. Implementation details

The architecture of our network is shown in Fig. 3. The DepthNet uses a U-shaped architecture similar to [44]. The PoseNet is splited into 2 parts followed by fully-connected layers to regress Euler angles and translations of 6-DoF pose, respectively. The length of convLSTM N is set 9. Layer Normalization and ReLUs are adopted in each layer except for the output layers. Detailed network architecture can be found in the supplementary materials.

Our model is implemented by PyTorch [31] on a single NVIDIA GTX 1080Ti GPU. All sub-networks are jointly trained in a self-supervised manner. Images are resized to 128×416 during both training and online adaptation. The Adam [19] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ is used and the weight decay is set 4×10^{-4}. Weighting factors $\lambda_m, \lambda_a, \lambda_r$ are set 0.01, 1 and 0.5, respectively. The feature alignment parameter β is set 0.5. The batch size is 4 for training and 1 for online adaptation. The learning objective (Eq. 3) is used for both training and online adaptation. We pre-train the network for 20,000 iterations. The learning rate α of the inner loop and outer loop are both initialized to 10^{-4} and reduced by half for every 5,000 iterations.

5.2. Outdoor KITTI

First, we test our method on KITTI odometry [18] dataset. It contains 11 driving scenes with ground truth poses. We follow the same train/test split as [22, 41, 44] using sequences 00-08 for training and 09-10 for online test.

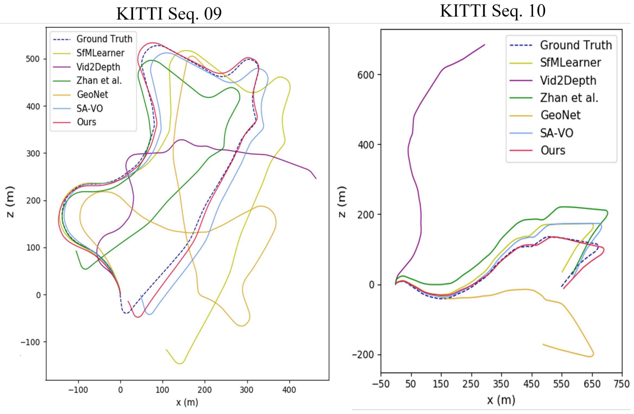

Figure 4. Trajectories of different methods on KITTI dataset. Our method shows a better odometry estimation due to online updating

Method	Seq. 09		Seq. 10	
	t_{err}	r_{err}	t_{err}	r_{err}
SfMLearner [44]	11.15	3.72	5.98	3.40
Vid2Depth [24]	44.52	12.11	21.45	12.50
Zhan *et al*. [42]	11.89	3.62	12.82	3.40
GeoNet [41]	23.94	9.81	20.73	9.10
SAVO [22]	9.52	3.64	6.45	2.41
Ours	**5.89**	**3.34**	**4.79**	**0.83**

Table 1. Quantitative comparison of visual odometry results on KITTI dataset. t_{err}: average translational root mean square error (RMSE) drift (%); r_{err}: average rotational RMSE drift (°/100m)

Instead of calculating absolute trajectory error (ATE) on image pairs in previous methods, we recover full trajectories and compute translation error t_{err} by KITTI evaluation toolkit, rotation error r_{err}. We compare our method with several state-of-the-art self-supervised VO baselines: SfMLearner [44], GeoNet [41], Zhan *et al*. [42], Vid2Depth [24] and SAVO [22]. As stated in [24], a scaling factor is used to align trajectories with ground truth to solve the scale ambiguity problem in monocular VO. The estimated trajectories of sequences 09-10 are plotted in Fig. 4 and quantitative evaluations are shown in Table 1. Our method outperforms all the other baselines by a clear margin, the accumulated error is reduced by online adapation.

The comparison of the running speed with other VO methods can be found in Table 2. Since we are studying the online learning problem, the running time includes forward propagation, loss computing, back propagation and network updating. Our method achieves real-time online adaptation and outperforms state-of-the-art baselines considerably.

Method	SfMLearner	GeoNet	Vid2Depth	SAVO	Ours
FPS	24	21	**37**	17	32

Table 2. Running speed of different VO methods.

5.3. Synthetic to real

Synthetic datasets (*e.g.* virtual KITTI, Synthia and Carla) have been widely used for research since they provide ground truth labels and controllable environment settings. However, there's a large gap between the synthetic and real-world data. In order to test the domain adaptation ability, we use Carla simulator [10] to collect synthetic images under different weather conditions in the virtual city for training, and use KITTI 00-10 for online testing.

It can be seen from Fig. 1, 5 and Table 3 that previous methods all failed when shifted to real-world environments. This is probably because the features of virtual scenes are much different from the real world despite they are both collected in the driving scenario. In contrast, our method significantly outperforms previous arts, which is able to bridge the domain gap and quickly adapt to the real-world data.

5.4. Outdoor KITTI to indoor TUM

In order to further evaluate the adaptability of our method, we test various baselines on TUM-RGBD [34] dataset. KITTI is captured by moving cars with planar motion, high quality images and sufficient disparity. Instead, TUM dataset is collected by handheld cameras in indoor scenes with much more complicated motion patterns, which is significantly different from KITTI. It includes various challenging conditions (Fig. 6) such as dynamic objects, non-texture scenes, abrupt motions and large occlusions.

We pretrain these methods on KITTI 00-08 and test on TUM dataset. Despite the ground truth depth is available, we only use monocular RGB images during test. It can be seen (Table 4 and Fig. 6) that our method consistently outperforms all the other baselines. Despite the large domain shift and significant difference in motion patterns (*i.e.* large, planar motion vs small motion in 3 axes), our method can still recover trajectories well. On the contrary, GeoNet [41] and Zhan *et al*. [42] tend to fail. Despite SAVO [22] utilizes LSTM to alleviate accumulated error to some extent, our method performs better due to online adaptation.

5.5. Ablation studies

In order to demonstrate the effectiveness of each component, we present ablation studies on various versions of our method on KITTI dataset (shown in Table 5).

First, we evaluate the backbone of our method (the first row) which includes convLSTM and feature alignment but no meta-learning process during training and online test. It can be seen from Table 1 and Table 5 that, even without meta-learning and online adaptation, our network backbone still outperforms most pervious methods. The results indicate that convLSTM is able to reduce accumulated error and feature alignment improves the performance when confronted with unseen environments.

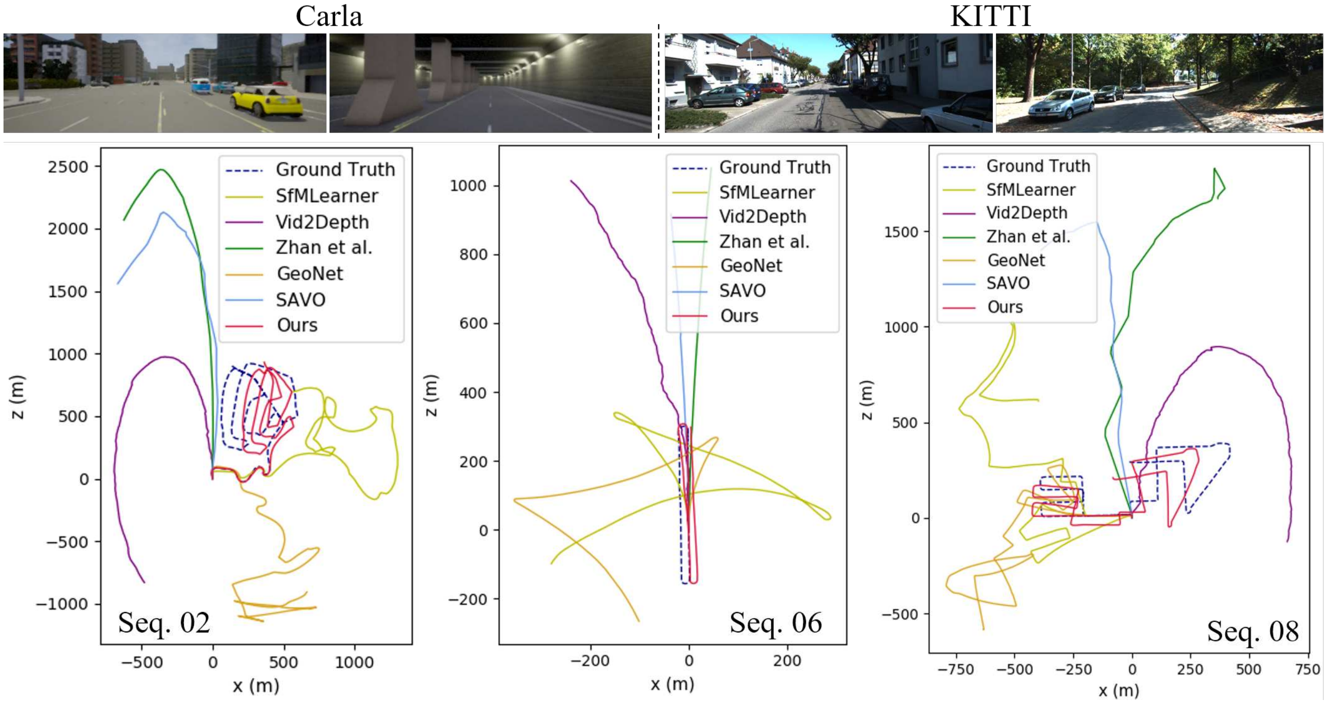

Figure 5. Trajectories of different methods pretrained on Carla and test on KITTI dataset. Our method significantly outperforms all the other baselines when changed from virtual to the real-world data

		SfMLearner [44]		Vid2Depth [24]		Zhan *et al*. [42]		GeoNet [41]		SAVO [22]		Ours	
Seq	frames	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
00	4541	61.55	27.13	61.69	28.41	63.30	28.24	44.08	14.89	60.10	28.43	**14.21**	**5.93**
01	1101	83.91	10.36	48.44	10.30	35.68	9.78	43.21	8.42	64.68	9.91	**21.36**	**4.62**
02	4661	71.48	27.80	70.56	25.72	84.63	24.67	73.59	12.53	69.15	24.78	**16.21**	**2.60**
03	801	49.51	36.81	41.92	27.31	50.05	16.44	43.36	14.56	66.34	16.45	**18.41**	**0.89**
04	271	23.80	10.52	39.34	3.42	12.08	**1.56**	17.91	9.95	25.28	1.84	**9.08**	4.41
05	2761	87.72	30.71	63.62	30.71	89.03	29.66	32.47	13.12	59.90	29.67	**24.82**	**6.33**
06	1101	59.53	12.70	84.33	32.75	93.66	30.91	40.28	16.68	63.18	31.04	**9.77**	**3.58**
07	1101	51.77	18.94	74.62	48.89	99.69	49.08	37.13	17.20	63.04	49.25	**12.85**	**2.30**
08	4701	86.51	28.13	70.20	28.14	87.57	28.13	33.41	11.45	62.45	27.11	**27.10**	**7.81**
09	1591	58.18	20.03	69.20	26.18	83.48	25.07	51.97	13.02	67.06	25.76	**15.21**	**5.28**
10	1201	45.33	16.91	49.10	23.96	53.70	22.93	46.63	13.80	58.52	23.02	**25.63**	**7.69**

Table 3. Quantitative comparisons of different methods pretraining on synthetic data in Carla simulator and testing on KITTI

Sequence	Structure	Texture	Abrupt motion	Zhan *et al*. [42]	GeoNet [41]	SAVO [22]	Ours
fr2/desk	✓	✓	-	0.361	0.287	0.269	**0.214**
fr2/pioneer_360	✓	✓	✓	0.306	0.410	0.383	**0.218**
fr2/pioneer_slam	✓	✓	✓	0.309	0.301	0.338	**0.190**
fr2/360_kidnap	✓	✓	✓	0.367	0.325	0.311	**0.298**
fr3/cabinet	✓	-	-	0.316	0.282	0.281	**0.272**
fr3/long_off_hou_valid	✓	✓	-	0.327	0.316	0.297	**0.237**
fr3/nstr_tex_near_loop	-	✓	-	0.340	0.277	0.440	**0.255**
fr3/str_ntex_far	✓	-	-	0.235	0.258	0.216	**0.177**
fr3/str_ntex_near	✓	-	-	0.217	0.198	0.204	**0.128**

Table 4. Quantitative evaluation of different methods pretraining on KITTI and testing on TUM-RGBD dataset. We evaluate relative pose error (RPE) which is presented as translational RMSE in [m/s]

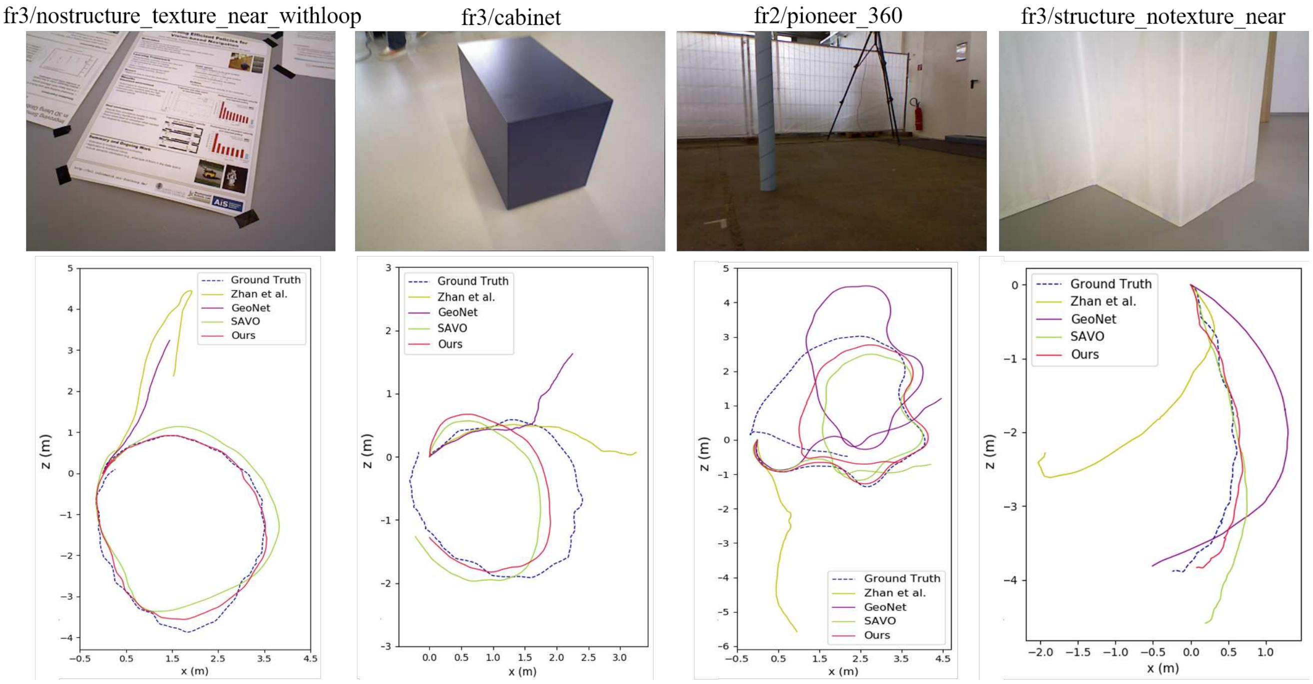

Figure 6. Raw images (top) and trajectories (bottom) recovered by different methods on TUM-RGBD dataset

				Seq. 09		Seq. 10	
Online	Pretrain	LSTM	FA	t_{err}	r_{err}	t_{err}	r_{err}
-	Standard	✓	✓	10.93	3.91	11.65	4.11
Naive	Standard	✓	✓	10.22	5.33	8.24	3.22
Meta	Meta	-	-	9.25	4.20	7.58	3.13
Meta	Meta	✓	-	6.36	3.84	5.37	1.41
Meta	Meta	-	✓	7.52	4.12	5.98	2.72
Meta	Meta	✓	✓	**5.89**	**3.34**	**4.79**	**0.83**

Table 5. Quantitative comparison of ablation study on KITTI dataset for various versions of our method. FA: feature alignment

Then we compare the efficiency of naive online learning (the second row) and meta-learning (the last row). It can be seen that, although naive online learning is able to reduce estimation error to some extent, it converges much slower than the meta-learning scheme, indicating that it takes much longer time to adapt the network to the new environment.

Finally, we study the effect of convLSTM and feature alignment during meta-learning (last four rows). Compared with baseline meta-learning scheme, convLSTM and feature alignment give the VO performance a further boost. Besides, convLSTM tends to perform better than feature alignment during online adaptation. One possible explaination is convLSTM incorporates spatial-temporal correlations and past experience over long sequence. It associates different states recurrently, making the gradient computation graph more intensively connected during back propagation. Meanwhile, convLSTM correlates the VO network at different time, enforcing to learn a set of weights θ that are consistent in the dynamic environment.

Besides, we study how the size of sliding window N is influencing the VO performance. The change of N has no much impact on the running speed (30-32 FPS), but as N increases, the adaptation gets faster and better. When N is greater than 15, the adaptation speed and accuracy becomes lower. Therefore, we set $N = 15$ as the best choice.

6. Conclusions

In this paper, we propose an online meta-learning scheme for self-supervised VO to achieve fast online adaptation in the open world. We use convLSTM to aggregate spatial-temporal information in the past, enabling the network to use past experience for better estimation and fast adaptation to the current frame. Besides, we put forward a feature alignment method to deal with changing feature distributions in the unconstrained open world setting. Our network dynamically evolves in time to continuously adapt to changing environments on-the-fly. Extensive experiments on outdoor, virtual and indoor datasets demonstrate that our network with online adaptation ability outperforms state-of-the-art self-supervised VO methods.

Acknowledgments This work is supported by the National Key Research and Development Program of China (2017YFB1002601) and National Natural Science Foundation of China (61632003, 61771026).

References

[1] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments. In *ICLR*, 2018.

[2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to Learn by Gradient Descent by Gradient Descent. In *NeurIPS*, 2016.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the Optimization of a Synaptic Learning Rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas, 1992.

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In *CVPR*, 2017.

[6] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming Variational Bayes. In *NeurIPS*, 2013.

[7] Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic Domain Alignment Layers. In *ICCV*, 2017.

[8] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In *AAAI*, 2019.

[9] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-Supervised GANs via Auxiliary Rotation Loss. In *CVPR*, 2019.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[11] Quang Pham Doyen Sahoo, Jing Lu, and Steven CH Hoi. Online Deep Learning: Learning Deep Neural Networks on the Fly. In *IJCAI*, 2018.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[13] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.

[14] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *ECCV*, 2014.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 2017.

[16] Chelsea Finn and Sergey Levine. Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate Any Learning Algorithm. In *ICLR*, 2018.

[17] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. S-VO: Fast Semi-Direct Monocular Visual Odometry. In *ICRA*, 2014.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for Stochastic Optimization. In *ICLR*, 2015.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[21] Ke Li and Jitendra Malik. Learning to optimize. In *ICLR*, 2017.

[22] Shunkai Li, Fei Xue, Xin Wang, Zike Yan, and Hongbin Zha. Sequential Adversarial Learning for Self-Supervised Deep Visual Odometry. In *ICCV*, 2019.

[23] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*, 2017.

[24] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In *CVPR*, 2018.

[25] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the Wild through Online Domain Adaptation. In *IROS*, 2018.

[26] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[27] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A Simple Neural Attentive Meta-Learner. In *ICLR*, 2018.

[28] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the Number of Linear Regions of Deep Neural Networks. In *NeurIPS*, 2014.

[29] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[30] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to Adapt in Dynamic, Real-World Environments Through Meta-Reinforcement Learning. In *ICLR*, 2019.

[31] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. PyTorch. `https://github.com/pytorch/pytorch`, 2017.

[32] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *CVPR*, 2019.

[33] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to Adapt: Aligning Domains Using Generative Adversarial Networks. In *CVPR*, 2018.

[34] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IROS*, 2012.

[35] Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

[36] Sebastian Thrun and Lorien Pratt. Learning to Learn: Introduction and Overview. pages 3–17, 1998.

[37] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-Time Self-Adaptive Deep Stereo. In *CVPR*, 2019.

[38] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *ICRA*, 2017.

[39] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning. In *CVPR*, 2019.

[40] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry. In *CVPR*, 2019.

[41] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018.

[42] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *CVPR*, 2018.

[43] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. DeepTAM: Deep Tracking and Mapping. In *ECCV*, 2018.

[44] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *CVPR*, 2017.