Online Depth Learning Against Forgetting in Monocular Videos

Zhenyu Zhang, Stéphane Lathuilière, Elisa Ricci, Nicu Sebe, Yan Yan, Jian Yang

Online Depth Learning against Forgetting in Monocular Videos

Zhenyu Zhang[1†], Stéphane Lathuilière[2,4], Elisa Ricci[2,3], Nicu Sebe[2], Yan Yan[1*†] and Jian Yang[1*†]
PCA Lab, Nanjing University of Science and Technology, China[1]
DISI, University of Trento, via Sommarive 14, Povo (TN), Italy[2]
Technologies of Vision, Fondazione Bruno Kessler, Via Sommarive 18, Povo (TN), Italy[3]
Télécom Paris, Multimedia Group, France[4]
zhangjesse, yanyan, csjyang@njust.edu.cn
stephane.lathuiliere, e.ricci, niculae.sebe@unitn.it

Abstract

Online depth learning is the problem of consistently adapting a depth estimation model to handle a continuously changing environment. This problem is challenging due to the network easily overfits on the current environment and forgets its past experiences. To address such problem, this paper presents a novel Learning to Prevent Forgetting (LPF) method for online mono-depth adaptation to new target domains in unsupervised manner. Instead of updating the universal parameters, LPF learns adapter modules to efficiently adjust the feature representation and distribution without losing the pre-learned knowledge in online condition. Specifically, to adapt temporal-continuous depth patterns in videos, we introduce a novel meta-learning approach to learn adapter modules by combining online adaptation process into the learning objective. To further avoid overfitting, we propose a novel temporal-consistent regularization to harmonize the gradient descent procedure at each online learning step. Extensive evaluations on real-world datasets demonstrate that the proposed method, with very limited parameters, significantly improves the estimation quality.

1. Introduction

Monocular depth estimation is a fundamental task in visual scene understanding, which has attracted increasing attention in computer vision and robotics [8, 58, 47, 54] communities. With the success of deep learning algorithms [46, 21], recent works usually propose methods based on deep neural networks [13, 27, 31, 53, 56, 59, 57]. Despite the attractive performance of these approaches, they mainly learn depth information from ground truth in a supervised manner. As such setting may not be practical in real-world applications due to the expensiveness of data collecting, many works turn to design unsupervised depth estimation methods [63, 17, 55, 1, 18], and show good performance compared with supervised approaches.

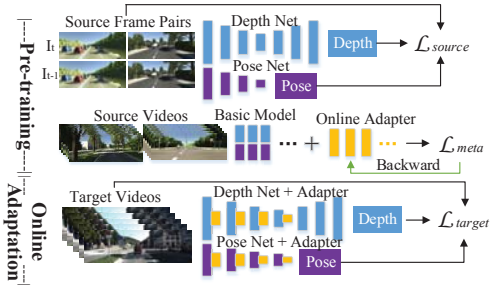


Figure 1. The proposed framework for online monocular depth learning. The model is firstly trained on synthetic dataset through un-and supervised loss $\mathcal{L}_{source}$. Then, a series of online adapters are learned from synthetic video sequences through an online-learning objective $\mathcal{L}_{meta}$. Finally, when adapting on the target real-world video, we only update the learned adapters and regressors (decoder) by unsupervised loss $\mathcal{L}_{target}$.

Despite the attractiveness of the above unsupervised methods, they may have limitations on **open-world** applications. Due to the classical paradigm in machine learning, after training phase, the model is frozen and used for inferencing without any change in the model parameters. However, in real-world applications, the deployment environment (*i.e. target* domain) may differ significantly from the training one (*i.e. source* domain), and keeps changing continuously over time. Several recent works are proposed to tackle this practical open-world problem for stereo matching [62, 50, 49], but few work focus on online monocular depth adaptation. Compared with online stereo, online

*Corresponding authors
†Zhenyu Zhang, Yan Yan and Jian Yang are with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology.

mono-depth learning is even more challenging from two aspects: (i) scale ambiguity and lack of support from geometrical information inherent to a monocular setting make the model extremely dependent on domain-specific visual features and prone to overfit on the current domain [11, 10]; (ii) the environmental variation (*e.g.*, speed or scene changes) introduces additional challenges both for depth or pose estimation, making the whole model fragile. On the top of these, as deep networks are not flexible in an online learning setting, there is catastrophic forgetting issue. In other words, the model, while updating to the novel domain, will easily forget the pre-learned knowledge. In such perspective, in this paper we argue that preventing forgetting while performing robust adaptation is the key point for online mono-depth learning.

To achieve such goal, as the training data on the source is usually of large amount, we could adjust the reliable pre-learned knowledge rather than update it entirely during adaptation. Firstly we propose to adjust the basic model to overcome domain shift. According to recent works [28, 2, 34] aligning domain shift through batch normalization (BN) [23] layers, although in online condition we never have access to full target data, we are inspired to smoothly adjust model statistics through online data stream of target videos. Besides the statistics, works [20, 37, 41, 40] on multi-domain or incremental learning inspire us to selectively tune a small subset of learned basic parameters while keeping all the other fixed. In our setting this is meant to ensure that visual appearance variations arising from scene changes will never influence the networks weights encoding the reliable knowledge. Following these ideas, we propose novel adapters that enable to adjust the source model online. Furthermore, inspired by recent learning to learn algorithm [12, 38], we develop a novel meta-learning based method to incorporate online learning procedure to the learning objective, which drives adapters for stable long-range adaptation.

Driven by the aforementioned motivation, in this paper we proposed a novel Learning to Prevent Forgetting (LPF) framework for online mono-depth learning. As illustrated in Fig. 1, we first train a monocular depth prediction model using synthetic data. Second, we employ a series of online adapters to adjust model statistics and weights, and train them with a novel meta-learning based strategy to properly update the basic knowledge. Specifically, we incorporate the learning objective $\mathcal{L}_{meta}$ with online adaptation procedures, thus $\mathcal{L}_{meta}$ will derive initial adapters to adapt temporal-continuous depth patterns in videos. Finally, while performing online adaptation on target real-world videos, we only update the learned online adapters and regressors (decoder) by unsupervised loss $\mathcal{L}_{target}$. In this way, we achieve our main purpose of adapting fast to a target video and long-range adaptation with less forgetting. We also propose a novel temporal-consistent regularization to harmonize the gradient descent during each online learning step.

In summary, this paper has four main contributions: **(i)** We propose a novel Learning to Prevent Forgetting framework for online depth learning in monocular videos, which is effective for rapid and long-range online adaptation on target data streams; **(ii)** We introduce a new adapter to handle the problem of domain shift in the case of continuous online data streams; **(iii)** We propose a novel meta-learning based method which permits to derive adapters for online learning condition; **(iv)** We perform an extensive evaluation to validate the effectiveness of our methods, showing that our approach achieve superior performance than state-of-the-art methods on real-world datasets.

2. Related Works

Monocular Depth Estimation. Early works on mono-depth estimation are mainly based on geometric priors [45, 30, 24]. With the availability of large-scale datasets, recently deep learning based methods have become the mainstream [9, 31, 27, 13, 56]. However, supervised methods require a large amount of pairs of images and ground truth depth maps. To overcome this limitation, unsupervised or self-supervised methods have been proposed [63, 17, 15, 55, 1, 18, 33, 3]. However, none of these papers focus on open-world setting, where the target sequence is gathered from a different environment with respect to the source and keeps changing with sequentially available data stream. Some recent works [5, 19] provide solution for open-world problem, but they need heavy extra annotation or computation like object motion and optical flow.

Domain Adaptation, Multi-domain and Continual Learning. There is a wide range of works on domain adaptation [7]. Recent deep learning-based methods mostly reduce the domain shift by considering distribution losses [32], alignment layers [28, 2, 35] and Generative Adversarial Network [43, 44]. Recently, cross-domain adaptation problems have also been studied for stereo depth estimation [50, 48]. However, these works did not explicitly address the domain shift problem while performing online adaptation. Some cross-domain monocular depth estimation works [61, 60] are also related to our paper, but they tackle no open-world problem.

Works that learn models for multi-domain problem through specific adapters [41, 40] are also related to our work. Besides, continual learning methods are loosely related to our works, where the task is to incrementally update a model trying to prevent catastrophic forgetting [36]. However, previous [26, 4, 29, 42] works mostly focus on classification problems, while we target a dense regression tasks. Further, our problem is in real-time condition with very limited learning stage.

Meta-Learning. Meta-learning, *i.e.* learning to learn, at-

tempts to design models that can adapt to new environments with few samples. In [51, 20, 12], meta-learning has been utilized for rapid generalization of models to novel domains and categories. Recently, Park *et al.* [38] introduced a method based on meta-learning to obtain an initial network for online tracking. A more closely related work to ours is [49], where an approach for learning to better adapt models to stereo videos is proposed. In contrast, our paper focuses on online mono-depth learning. Instead of learning a better initial model, our method learns how to exploit pre-existing knowledge for better online adaptation in a continuous data stream, thus aiming at robust adaptation to prevent forgetting. Therefore, the task, the goal and the approach considered in our work are radically different from previous papers.

3. Preliminary

In this section, we present the employed unsupervised framework for monocular depth estimation and its online learning algorithm.

Unsupervised Framework: We follow the approach firstly introduced in [63] for unsupervised depth estimation in monocular videos. The framework contains two sub-networks: depth net for predicting depth maps D_t of target frame I_t, and pose net for predicting relative poses P_{t-1}, P_{t+1} between adjacent frame pairs (I_{t-1}, I_t) and (I_{t+1}, I_t). Many works extended this method by jointly learning optical flow [55] and employing geometric constraints [33, 3]. In this paper we mainly implement and validate our method based on [63] and more recent work [1] but we would like to remark that, in principle, our framework can also integrate most of theses recent advances.

Online Learning: Here we discuss the paradigm for online mono-depth learning. To better formulate the problem, we first employ a source dataset $\mathcal{V}_\mathcal{S}$ (usually synthetic and with ground truth) to pre-train our monocular depth estimation model with a supervised loss $\mathcal{L}_s$ (e.g., L_1 regression) and unsupervised loss $\mathcal{L}_u$ (defined in [63, 1]). Then, the model is deployed and evaluated on target videos $\mathcal{V}_\mathcal{T}$. In previous works, $\mathcal{V}_\mathcal{T}$ is usually fixed and from the same domain as $\mathcal{V}_\mathcal{S}$. However, in practical applications, $\mathcal{V}_\mathcal{T}$ is usually from different domains (e.g., real-world scenes) and keeps changing (e.g., models implemented on a car have to work in changing environments). In such open-world condition, we process the video frames sequentially and continuously adapt our model at each time step in order to predict more accurate depth maps with t increasing. Similar to [49, 62] on open-world stereo, we follow the paradigm for learning and evaluating. At time t, we first predict depth from I_t (the current frame). Then, we evaluate our prediction according to the supervised loss $\mathcal{L}_s$, and finally, we update the model using $\mathcal{L}_u$. The learning process at each time step is obtained via gradient descent as follows:

$$[\theta^d_{t+1}, \theta^p_{t+1}] \leftarrow [\theta^d_t, \theta^p_t] - \alpha \nabla_{\theta^p, \theta^d} \mathcal{L}_u([\theta^d_t, \theta^p_t], I_t, I_{t-1}), \tag{1}$$

where θ^d, θ^p are parameters of depth net and pose net.

4. Learning to Prevent Forgetting

In this section we introduce our proposed Learning to Prevent Forgetting (LPF) framework for online mono-depth learning, including online adapters for depth estimation (in Section 4.1 and 4.2) and corresponding algorithm to learn them (in Section 4.3). A novel Temporal-Consistent Regularization for stable online learning is also introduced in Section 4.4.

4.1. Online Statistics Adapter

As discussed in Section 1, scale ambiguity and lack of geometric prior information make monocular depth estimation frameworks over-reliant on appearance cues and domain-specific information. As a consequence, they are especially susceptible to domain shift when deployed on new target data. According to [28, 2], domain discrepancy between source domain $\mathcal{S}$ and target domain $\mathcal{T}$ can be reduced by transforming statistics recorded in batch normalization (BN) layer. However, in our case we have no access to the full data but sequential stream of $\mathcal{T}$. This inspires us to develop a new adapter that smoothly updates statistics over data stream in an online fashion. After pre-training on source dataset, the model gathers statistics $\mathcal{B}_\mathcal{S} = (\mu_\mathcal{S}, \Sigma_\mathcal{S})$ aligned with $\mathcal{S}$. Here for sake of notation, we only analyse mean μ and covariance matrix Σ in one BN layer but the approach described below applies to all the others. When adapting on the target, using source statistics $\mathcal{B}_\mathcal{S}$ would make the model suffer from domain shift since the statistics $\mathcal{B}_\mathcal{T}$ are different from $\mathcal{B}_\mathcal{S}$ statistics. However, in the beginning of the sequence, we have not disposed of enough frames to have a robust estimation of $\mathcal{B}_\mathcal{T}$. Furthermore, by only using the observed statistics of target frames, the model would completely forget the source knowledge. Motivated by this, we design a more robust way for aligning model statistics. Based on the theory in [52] which analyses statistics across different layers, we can also model the statistics along time axis into a Kalman filtering process. At time t, given the state transition matrix $\mathbf{A}^t$ and feature x^t with m examples, we can estimate statistics as:

$$\begin{aligned}
\hat{\mu}^{t|t-1} &= \mathbf{A}^t \hat{\mu}^{t-1|t-1}, \\
\hat{\mu}^{t|t} &= (1 - a^t)\hat{\mu}^{t|t-1} + a^t \bar{x}^t, \\
\hat{\Sigma}^{t|t-1} &= \mathbf{A}^t \hat{\Sigma}^{t-1|t-1} (\mathbf{A}^t)^T + R, \\
\hat{\Sigma}^{t|t} &= (1 - a^t)\hat{\Sigma}^{t|t-1} + a^t S^t,
\end{aligned} \tag{2}$$

where $\hat{\mu}^{t|t-1}$ and $\hat{\Sigma}^{t|t-1}$ are the calculated mean and covariance matrix from time $t-1$, R and S^t are the covariance

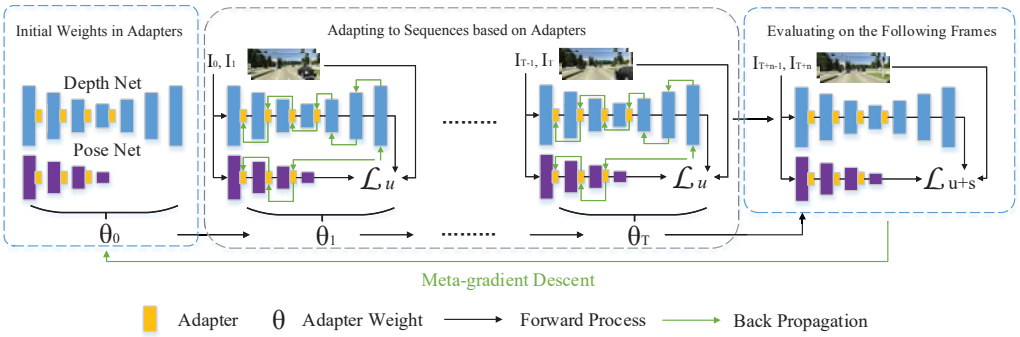


Figure 2. One single training loop of our proposed meta-learning based method for learning online adapters. The adapters start with weight θ_0, and then continuously adapt to a video sequence of length T on source dataset, with the weight updated to θ_T through unsupervised loss $\mathcal{L}_u$. After that we evaluate the adapter using frames that randomly selected from the following time points T + n, and perform gradient descent with respect to the initial weight θ_0. In this way, we learn a series of better initial adapters for adapting temporal-continuous depth patterns. More details can be seen in Section 4.3.

matrix of bias and observed feature. $\bar{x}^t = \frac{1}{m}\sum_i^m x_i^t$, and a^t is a balancing weight and $\hat{\mu}^{t|t}$ and $\hat{\Sigma}^{t|t}$ are the final estimations. As $\mathbf{A}^t$ and R are difficult to get during the online adaptation, we simply assume $\mathbf{A}^t$ is an identity matrix and bias is zero. In addition, Σ is usually vectors in convolution neural networks so that $S^t = \frac{1}{m}\sum_i^m (x_i - \bar{x}^t)$. In this way, we can simply Eqn. 2 as

$$\begin{aligned}\hat{\mu}^{t|t} = \mu^t &= (1-a^t)\hat{\mu}^{t|t-1} + a^t\bar{x}^t, \\ &= (1-a^t)\mu^{t-1} + a^t\tilde{\mu}^t \\ \hat{\Sigma}^{t|t} = \Sigma^t &= (1-a^t)\hat{\Sigma}^{t|t-1} + a^t S^t \\ &= (1-a^t)\Sigma^{t-1} + a^t\tilde{\Sigma}^t \end{aligned} \tag{3}$$

where $\tilde{\mu}^t, \tilde{\Sigma}^t$ are the observed mean and variance at time t respectively, and μ^t, Σ^t are the final estimations of the statistics. a^t is a learnable dynamic weight to decide how much the layer should adapt to the current frame. Note that, different from standard BN operation, we perform Eqn. 3 during forward process of training step, i.e., the input x of a layer in training step will be transformed by

$$\hat{x} = \omega\frac{x-\mu^t}{\sqrt{\Sigma^{t^2}+\epsilon}} + \rho. \tag{4}$$

ω, ρ are scaling and shifting factor, and ϵ is a small constant. In this way, $\mathcal{B}_\mathcal{S}$ is smoothly aligned to $\mathcal{B}_\mathcal{T}$ with time t increasing, and pre-learned knowledge is stably updated.

4.2. Online Weight Adapter

To preserve the pre-learned knowledge during online adaptation, we use adapters to adjust the feature representation without updating the main network parameters. This will weaken the misguidance caused by scene changing and benefit long-range adaptation. Besides, these adapters need to be efficient enough with very limited parameters to avoid computational overload. Here, we borrow some ideas from multi-domain and incremental learning works [40, 41] to design our adapters. The employed adapters are shown in Fig. 3. Considering a given convolutional layer of the main model, $\phi(\cdot, \lambda)$ denotes its computed function parametrized with weights λ. Let f_{in} and f_{out} be the input and output feature maps. We further define $\varphi(\cdot, \gamma)$ as the adapter with parameter γ. In this work we consider three different adapters. First, the original layer $\phi(\cdot, \lambda)$ can be adjusted as follows:

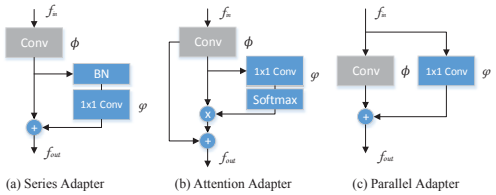


Figure 3. The proposed online weight adapters. Each adapter is able to adjust basic model through operations with very limited cost. The total amount of parameter is 4.7M, about 1/9 of DispNet encoder.

$$f_{out} = \phi(f_{in}) + \varphi(\phi(f_{in})). \tag{5}$$

Considering that φ is a 1×1 convolution, we obtain the series adapters introduced in [40] (see Fig. 3(a)). Second, if we employ a 1×1 convolution, a sigmoid activation and a scaling operation into φ, it becomes the attention module proposed in [22]. This adapter is able to re-weight the output of ϕ using attention mechanism. In our case we slightly modify this attention adapter by removing the squeezing operation and computing softmax response along each channel, in order to enhance the most related spatial information. This adapter is referred to as attention adapter and is shown in Fig. 3(b). Finally, $\phi(\cdot, \lambda)$ can also be adapted as follows:

$$f_{out} = \phi(f_{in}) + \varphi(f_{in}), \tag{6}$$

that is known as parallel adapter [41] (See Fig. 3(c)). The use of these low-cost adapters avoid losing reliable previ-

ous knowledge, which is beneficial to fast and long-range adaptation.

4.3. Learning to Learn Online Adapter

Through the adapters proposed in Section 4.1 and 4.2, the model obtains capacity to adjust statistics and weight for online adaptation. However, standard off-line training on the source is not satisfactory for learning these adapters which need to work in online mode. To overcome this limitation, we propose a novel learning objective in Fig. 2, which is able to evaluate how well the adapter is for online adaptation. Given adapter weights θ_0 and a video sequence with length T, we first perform online learning through $\mathcal{L}_u$ on every frame pair I_{t-1}, I_t, and finally obtain adapter weight θ_{T}. In practical condition, as our goal is to make the model perform well in the following frames, we then evaluate the adapter with θ_{T} on a randomly selected following frame pair I_{T+n-1}, I_{T+n}. The un-and supervised loss $\mathcal{L}_{s+u}$ reveals the prediction quality after online adaptation with initial adapter θ_0. Based on recent meta-learning theory proposed in [12], we can perform gradient descent w.r.t. θ_0 to derive the initial adapter that can perform fast and stable online adaptation.

We now provide the technical details of our meta-learning approach. We consider a sampled video $\mathcal{V}_k$ from source dataset $\mathcal{V}_\mathcal{S}$ and composed of the frames $[I_0, ..., I_{T+N-1}, I_{T+N}]$. Let $\mathcal{B}_\mathcal{S}$ be the obtained main model statistics after pre-training. The approach is detailed in Alg. 1. T is the length of each video sequence for online learning, and N is the number of following frames. K is the number of selected video sequences $\mathcal{V}_k$, and α, β are learning rate for online adaptation and meta-gradient descent step, respectively. In one single loop, we start initial adapter weight and model statistics as $\theta_0 = \theta, \mathcal{B}_0 = \mathcal{B}_\mathcal{S}$ in line 5. Then in Line 6 to 9 we adapt the model on a selected sequence and finally get θ_T and $\mathcal{B}_T$. In line 10 to 12, we randomly select following frames I_{T+n-1}, I_{T+n} to simulate possible future changes, and perform evaluation on them to obtain score L. Finally after learning and evaluating on all $\{\mathcal{V}_k\}_{k=1}^K$ we conduct a gradient descent step w.r.t. θ_0 to find a good initial weight for online adaptation. Our method is different from the approach in [49] from two aspects: i) it meta-learns and updates adapters rather than the whole model, which preserves and adjusts (rather than totally changes) reliable basic knowledge against drastic changes; ii) it performs evaluation and meta-gradient descent after adapting on video sequences rather than single frame, aiming to achieve good long-range adaptation for future frames.

Algorithm 1 Learning to learn online adapter

Require: Initial weight θ for adapters, training set $\mathcal{V}_\mathcal{S}$, hyper-parameter T, N, K, α, β.

1: **while** not done **do**
2: Sample $\{\mathcal{V}_k\}_{k=1}^K$ from $\mathcal{V}_\mathcal{S}$
3: Initialize evaluation score $L = 0$
4: **for** all $\mathcal{V}_k$ **do**
5: $\theta_0 = \theta$, $\mathcal{B}_0 = \mathcal{B}_\mathcal{S}$
6: **for** $t \leftarrow 1, 2, ..., T$ **do**
7: $\mathcal{B}_t \leftarrow \mathcal{B}_{t-1}$ by Eqn. 3
8: $\theta_t \leftarrow \theta_{t-1} - \alpha \nabla_{\theta_{t-1}} \mathcal{L}_u(\theta_{t-1}, \mathcal{B}_t; I_{t-1}, I_t)$
9: **end for**
10: Uniform sampling $n \in [1, N]$
11: $\mathcal{B}_{T+n} \leftarrow \mathcal{B}_T$, I_{T+n-1}, I_{T+n} by Eqn. 3
12: $L = L + \mathcal{L}_{s+u}(\theta_T, \mathcal{B}_{T+n}; I_{T+n-1}, I_{T+n})$
13: **end for**
14: $\theta = \theta_0 - \beta \nabla_{\theta_0} L$
15: **end while**

4.4. Online Adaptation on Target Videos

After meta-training the adapters through Alg. 1, we can perform online adaptation on the target videos. However, although updating the adapters with fixed original knowledge prevents forgetting to some extent, the model may still get influenced from various environmental changing on real-world scenes and tends to overfit on the current frame. To further guarantee a stable adaptation process, we propose a Temporal-consistent Regularization (denote by $\mathcal{L}_r$). At the time step t of online adaptation, besides the current frame we also make the model predict depth map $\dot{D}_{t-\Delta t}$ from a randomly selected previous frame $I_{t-\Delta t}$. Then, we force the prediction $\dot{D}_{t-\Delta t}$ to be similar to the previous prediction $D_{t-\Delta t}$ at time step $t - \Delta t$, which can be achieved by computing

$$\mathcal{L}_r = ||\dot{D}_{t-\Delta t} - D_{t-\Delta t}||_1. \tag{7}$$

Here we just let $1 < \Delta t \leq 5$, thus only small memory is needed to store the previous frames and predictions. In this way, the model is constrained to preserve its ability learned from previous frames. Even if drastic environmental variation happens at time t, $\mathcal{L}_r$ can harmonize the gradient and penalize model's overfitting to current time step. Finally, the total unsupervised loss for adaptation on the target videos can be written as:

$$\mathcal{L}_{target} = \mathcal{L}_u + \delta \mathcal{L}_r, \tag{8}$$

where δ is a weight to balance the regularization. Importantly, we also employ $\mathcal{L}_r$ in our meta learning algorithm (second step of Fig. 2) in order to simulate the learning process on the target that will use this loss.

5. Experiment

5.1. Datasets

Virtual-KITTI: Virtual Kitti [14] (vKitti) is a synthetic dataset for urban driving environment. It contains 6 differ-

Table 1. Analysis on Each Component of LPF for Fast Adaptation

Method	Training	Lower is better				Higher is better		
		Abs Rel	Sq Rel	RMSE	RMSE_{log}	< 1.25	$< 1.25^2$	$< 1.25^3$
Basic (no adaptation)	standard	0.3641	6.2917	9.9467	0.4124	0.5070	0.7703	0.8867
Basic + Naive	standard	0.2242	2.1311	7.1179	0.2991	0.6558	0.8709	0.9486
Basic + SA	Standard	0.2150	1.9834	6.9069	0.2883	0.6628	0.8783	0.9539
Basic + SA + WA	Standard	0.2143	1.9576	6.9055	0.2885	0.6625	0.8790	0.9543
Basic + SA + WA + $\mathcal{L}_r$	Standard	0.2105	1.8732	6.7656	0.2820	0.6758	0.8802	0.9553
Basic + SA	$\mathcal{L}_{meta}$	0.2087	1.9003	6.8342	0.2833	0.6695	0.8901	0.9560
Basic + SA + WA	$\mathcal{L}_{meta}$	0.2045	1.7549	6.7022	0.2791	0.6783	0.8957	0.9591
Basic + SA + WA + $\mathcal{L}_r$	$\mathcal{L}_{meta}$	**0.2033**	**1.6076**	**6.5613**	**0.2778**	**0.6935**	**0.8965**	**0.9621**

ent scenes in monocular videos with ground truth depth and different weather conditions. We treat it as source domain and use the videos in all conditions (except for foggy and raining) from all 6 scenes for pre-training our model. The total training set contains 85k images.

Cityscapes: Cityscpes [6] is a urban dataset for autonomous driving and scene understanding. We use the sequential data from 41 different monocular videos for pre-training our model. Although it is in real-world scenes, the environment is still very different from our target domain and we treat it as source domain to validate our method.

KITTI: KITTI [16] is a widely-used real-world dataset for autonomous driving. Following the setting in [63, 1], we use Eigen's test split [10] as target domain for evaluation. Note that, as our method is for online depth learning, we perform online evaluation and adaptation using all the frames in all target videos.

5.2. Implementation Details

The proposed method is implemented using PyTorch Library [39]. We validate the method using the framework proposed in SfM-Learner [63] and SC-SfM-Learner [1], which are widely-used or most recent mono-depth estimation approach. Following these two frameworks, we use DispNet and PoseNet for predicting depth and relative pose, respectively. For SfM-Learner, we use a pytorch implementation with input size of 128×416; for SC-SfM-Learner we just use the authors' released code with input size of 256×832. We add BN layer after each conv-layer in the encoder as needed by our model statistics adapter. This almost gives no changes to reproduce the results of original papers. We use the combination $\mathcal{L}_{u+s}$ of the supervised and unsupervised losses to train the basic model on source domain for 100 epochs, and use $\mathcal{L}_u$ to train for another 100 epochs to guarantee the learning of PoseNet. For training adapters, we select $T = 5, N = 5, K = 8$ and learning rate $\alpha = 1e-4, \beta = 1e-5$ in Alg. 1. We give a_t in Eqn. 3 a upper bound of 0.05 during training to avoid too largely changing, and the adapters are trained for 20 epochs. For online adaptation on the target, we use the same learning rate as α to update adapters and regressors. δ in Eqn. 7 is set to 0.2. Adam optimizer [25] is used during pre-training and online adaptation.

Table 2. Analysis on Different Weight Adapters

Method	Abs Rel	RMSE	< 1.25	$< 1.25^2$
Basic + Naive	0.2242	7.1179	0.6558	0.8709
LPF (WA-series)	0.2054	6.5423	0.6976	0.8960
LPF (WA-parallel)	0.2033	6.5613	0.6935	0.8965
LPF (WA-attention)	0.2072	6.6107	0.6920	0.8933

5.3. Evaluation Protocol

We use an evaluation protocol suited for online condition in which frames are sequentially fed into the network. At each time step, we first measure the performance of our model on the current input frame, and then adapt this frame by a step of back-propagation and weight update. We concatenate all of the videos in Eigen's testing split with a random order and start adaptation from the 1st frame. Such randomly concatenation can further simulate environment changes to some extent. We calculate average scores on all the frames as final results, and we also show the scores on last 20% frames of a video to analyse performance when the model has adapted on a series of frames. These scores can tell how fast and stable the model adapt to each video. We employ metrics used in [63, 1] to perform evaluation.

5.4. Fast Online Adaptation

In this section we analyse if the components and mechanisms in our method can help for fast online adaptation. Without specially noting, the models are built based on SfM-Learner [63] and pre-trained on vKitti dataset. More additional experiments can be seen in supplementary material.

Analysis on Method Component: For sake of notation, we use **Basic** to define the basic model without our approach, and **WA** and **SA** to define the proposed weight adapter and statistic adapter. We also use $\mathcal{L}_r$ to denote the proposed Temporal-Consistent Regularization in Eqn. 7. In **Naive** method, we employ the main model without adapters, and update all the parameters in learning steps. Concerning pre-training, **Standard** refers to the classical pretraining without the meta-learning formulation of Alg. 1. The results are illustrated in table 1. Here for sake of clarity, we just show results with parallel weight adapter (Eqn. 6). We observe that the model without online adaptation cannot provide satisfactory results, and naive online learning shows

Table 3. Comparison of different frameworks and source datasets.

Method	data	Abs Rel	RMSE	< 1.25	$< 1.25^2$
SfM-Learner [63]					
Naive	vKitti	0.2242	7.1179	0.6558	0.8709
	Cityscapes	0.2016	6.7935	0.7166	0.8982
LPF	vKitti	0.2033	6.5613	0.6935	0.8965
	Cityscapes	0.1751	6.0677	0.7499	0.9186
SC-Sfm-Learner [1]					
Naive	vKitti	0.1782	5.9408	0.7473	0.9021
	Cityscapes	0.1675	5.8185	0.7754	0.9163
LPF	vKitti	0.1528	5.5081	0.7762	0.9234
	Cityscapes	0.1383	5.3478	0.8194	0.9307

Table 4. Long-range Adaptation: K_{train}→vKitti→K_{test}

Method	Abs Rel	RMSE	< 1.25	$< 1.25^2$
SfM-Learner [63]				
Naive	0.2070	6.5248	0.7041	0.8806
Regressor	0.2002	6.4325	0.7156	0.8879
L2A [49]	0.1937	6.3804	0.7221	0.8980
LPF	**0.1794**	**6.1090**	**0.7307**	**0.9126**
SC-Sfm-Learner [1]				
Method	Abs Rel	RMSE	< 1.25	$< 1.25^2$
Naive	0.1735	5.6528	0.7743	0.9140
Regressor only	0.1702	5.5883	0.7769	0.9153
L2A [49]	0.1692	5.5500	0.7881	0.9197
LPF	**0.1505**	**5.4452**	**0.7990**	**0.9325**

only limited improvements. In the case of standard pre-training, although our **SA** and $\mathcal{L}_r$ benefit the adaptation procedure, the **WA** brings very limited improvement. A possible explanation is that standard offline training does not provide online adaptation ability to the model. In contrast, with our proposed meta learning method $\mathcal{L}_{meta}$ for pre-training, **SA** shows higher improvements and our **WA** is also able to show better performance, which reveals that our meta-learning method makes **WA** efficiently work. Our full method achieves best performance compared with other baselines in the table. These results demonstrate that our proposed LPF method obviously leads to a faster online adaptation on target videos.

Besides, we also show the effectiveness of different **WA** in table 2. In each experiment we just change the weight adapter in our full method. **WA-series** and **WA-attention** mean adapters defined in Eqn. 5 but with different φ as described in Fig. 3(a) and (b). **WA-parallel** means adapter defined in Eqn. 6 and shown in Fig. 3(c). We observe that all of the three adapters show consistent improvement in the metrics. These results reveal that our LPF method can be implemented with different kinds of adapters and consistently improve performance.

Comparison with Different Frameworks and Dataset: We evaluate the effectiveness of our method on different unsupervised frameworks and datasets. In each experiment we use parallel adapter. The results are shown in table 3. We first observe that both framework are benefitted by our LPF method, which reveals that LPF can be successfully implemented into different frameworks. Then we observe that the improvement brought by LPF is consistent across different datasets, which further demonstrates the generalization of our method.

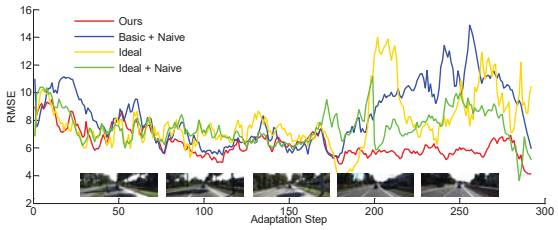


Figure 4. Illustration of online learning process of different model on a same Kitti video. According to the video picture and choppy performance of ideal offline model, there exists environmental changing near $t = 200$. Naive method cannot deal with it and brings slower convergence, while our LPF method obtains stable and robust adaptation process.

5.5. Long-range Adaptation across Domains

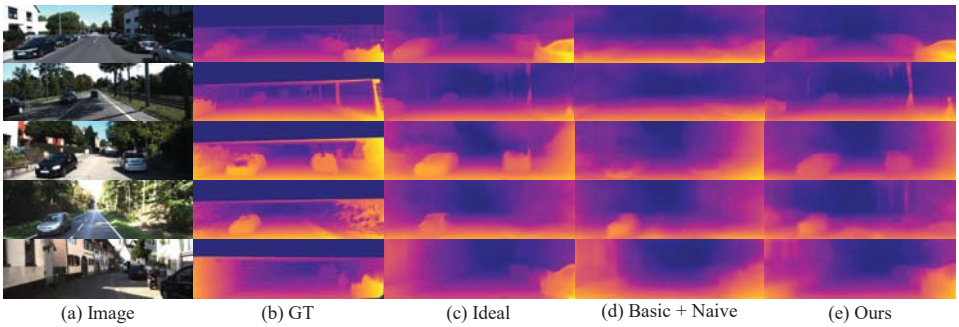
In this section, we analyse how LPF perform in the case of very long-range online adaptation. To simulate a long online learning scenario, we first pre-train the model on Kitti Eigen's training split (K_{train}). Then, we perform online adaptation on Virtual Kitti (vKitti) dataset using all of the videos that we described in Section 5.1. Finally, after adaptation on Virtual Kitti, we perform online learning on Kitti Eigen's testing split (K_{test}) and calculate the evaluation scores based on Section 5.3. This experiment is able to show whether our LPF method prevents forgetting in such a long-range cross-domain adaptation procedure. Indeed, adapting on vKitti may make the model lose the reliable knowledge learned on K_{train} and harm the performance on K_{test}. Results are reported in table 4. In this comparison, we include the L2A method [49] which can also be implemented in our model. Importantly, this method is not design to prevent catastrophic forgetting. Besides, we also consider a model where only the decoder is updated (*Regressor*). We observe that only updating the regression layer obtains scores that are slightly better than updating the whole network. This reveals that freezing the encoder is even better to address the catastrophic forgetting issue. L2A also provides improvement thanks to its meta-learning formulation that makes the method adapt fast. Nevertheless, our LPF method obtains the best performances in the two frameworks. It shows that the knowledge learned on K_{train} is still preserved after adapting on vKitti. It confirms that properly adjust the main knowledge rather than updating it mitigate forgetting.

5.6. Comparison with Ideal and SOTA Method

In this section we compare our method with ideal and state-of-the-art approaches. The experiment settings are the same as Section 5.4 to analyse fast online adaptation performance on target videos. The results are illustrated in table 5, where the methods are all implemented in the two considered frameworks. To show the upper bound of performance, we also illustrate scores of ideal condition where the

Table 5. Comparisons with Ideal and State-of-the-art Method. Evaluation scores are computed on all the frames of Eigen's testing videos.

		Online Evaluation Scores				Evaluation Scores on Last 20% frames			
Method	Training Set	Abs Rel ↓	RMSE ↓	$< 1.25 \uparrow$	$< 1.25^2 \uparrow$	Abs Rel ↓	RMSE ↓	$< 1.25 \uparrow$	$< 1.25^2 \uparrow$
SfM-Learner [63]									
Ideal (no adapt.)	Kitti	**0.2024**	6.5597	0.7180	0.8935	0.2091	6.5403	**0.7111**	0.8879
Ideal + Naive	Kitti	0.2032	**6.5080**	**0.7220**	**0.8989**	**0.2113**	**6.4522**	0.7075	**0.8962**
Naive	vKitti	0.2242	7.1179	0.6558	0.8709	0.2195	6.9022	0.6683	0.8798
L2A [49]	vKitti	0.2171	6.8024	0.6759	0.8762	0.2103	6.7256	0.6783	0.8791
LPF (Ours)	vKitti	**0.2033**	**6.5613**	**0.6835**	**0.8965**	**0.1962**	**6.3887**	**0.7147**	**0.8996**
SC-SfM-Learner [1]									
Ideal (no adapt.)	Kitti	0.1537	5.6295	0.8086	0.9338	0.1535	5.6412	0.8027	0.9335
Ideal + Naive	Kitti	**0.1468**	**5.2768**	**0.8203**	**0.9431**	**0.1399**	**5.1413**	**0.8320**	**0.9479**
Naive	vKitti	0.1782	5.9408	0.7473	0.9021	0.1662	5.7583	0.7596	0.9198
L2A [49]	vKitti	0.1708	5.8764	0.7548	0.9157	0.1615	5.6831	0.7728	0.9213
LPF (Ours)	vKitti	**0.1628**	**5.6581**	**0.7762**	**0.9234**	**0.1495**	**5.4327**	**0.7936**	**0.9301**



(a) Image (b) GT (c) Ideal (d) Basic + Naive (e) Ours

Figure 5. Illustration of visual results. The ideal model is pre-trained on Kitti, while the models of (d) and (e) are pre-trained on vKitti and online adapted to Kitti videos. Results of our method are superior than naive baseline, and even competitive with those of ideal method.

model is offline pre-trained on Kitti Eigen's training split. For methods implemented on Sfm-Learner, we first observe the performance gain brought by naive online learning is very limited even in ideal condition, and L2A [49] is able to improve the online evaluation scores. Then compared with L2A, our method further improve the performance and obtain best results. These analyses reveal that our LPF method outperforms L2A and naive method. Compared with upper bound i.e., the ideal scenario of the models, our method obtains competitive scores, or even better performance especially on the last % 20 frames. These results further demonstrate that our LPF method leads to a stable and fast online adaptation process even if the model never sees data in target domain before. For the models implemented on SC-Sfm-Learner, we observe that our LPF method still obtain best results among online learning models in all metrics. Compared with ideal models, our method is slightly weaker but closest to the upper bound than other online approaches. It is worthy noting that the performance gain on SC-Sfm-Learner is comparatively smaller than that on Sfm-Leaner. One possible explanation is that SC-Sfm-Learner is able to capture more geometric constraints between two adjacent frames and provide more reliable visual cues, which makes the model depend less on appearance information and robust to environmental changes. Even though, our LPF method still shows its power for online depth learning.

To further analyse our method, we illustrate online learning process of different models in Fig. 4. All models are built on Sfm-Leaner. We observe that at beginning all models perform similarly. However, when environment changes largely around $t = 200$, two models of naive method show unstable behavior which leads to worse performance. In contrast, our LPF method properly deals with such environmental changing and shows a more robust and faster learning procedure. Finally, we show qualitative results in Fig. 5. We observe our LPF method predicts more accurate depth maps than naive model, and shows close performance to ideal approach and ground truth. These qualitative results are inline with the quantitative results.

6. Conclusion

In this paper we propose a novel Learning to Prevent Forgetting (LPF) framework for unsupervised online adaptation in monocular videos. Two adapters are designed for adjusting model statistics and weights against forgetting issue. A novel meta-learning based algorithm is developed to learn adapters for better online learning procedure. Extensive experiments demonstrate that LPF contributes to fast and stable long-range online adaptation, and obtains competitive or better performance than ideal models.

7. Acknowledgement

This work was supported by the National Science Fund of China under Grant Nos. U1713208, 61806094 and "111" Program B13022.

References

[1] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *NeurIPS*, 2019.

[2] Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, pages 5077–5085, 2017.

[3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, volume 33, pages 8001–8008, 2019.

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, pages 532–547, 2018.

[5] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7063–7072, 2019.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[7] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[8] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019.

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015.

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, pages 2366–2374, 2014.

[11] Jose M. Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Camconvs: Camera-aware multi-scale convolutions for single-view depth. In *CVPR*, June 2019.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[13] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[14] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, pages 4340–4349, 2016.

[15] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*. Springer, 2016.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. 2019.

[19] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. *arXiv preprint arXiv:1904.04998*, 2019.

[20] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *CVPR*, pages 4805–4814, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[24] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] 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.

[27] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[28] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[30] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, pages 1253–1260. IEEE, 2010.

[31] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 2016.

[32] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[33] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, pages 5667–5675, 2018.

[34] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the wild through online domain adaptation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1103–1109, 2018.

[35] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *CVPR*, pages 3771–3780, 2018.

[36] 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. 1989.

[37] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, pages 11321–11329, 2019.

[38] Eunbyung Park and Alexander C Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *ECCV*, pages 569–585, 2018.

[39] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[40] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

[41] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pages 8119–8127, 2018.

[42] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017.

[43] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, 2018.

[44] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, pages 3752–3761, 2018.

[45] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[47] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *CVPR*, pages 6243–6252, 2017.

[48] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *ICCV*, pages 1605–1613, 2017.

[49] Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr. Learning to adapt for stereo. In *CVPR*, pages 9661–9670, 2019.

[50] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *CVPR*, 2019.

[51] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[52] Guangrun Wang, Ping Luo, Xinjiang Wang, Liang Lin, et al. Kalman normalization: Normalizing internal representations across network layers. In *Advances in Neural Information Processing Systems*, pages 21–31, 2018.

[53] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018.

[54] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *ECCV*, pages 817–833, 2018.

[55] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018.

[56] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, 2018.

[57] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for rgb-d scene understanding. *TPAMI*, 2019.

[58] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019.

[59] Zhenyu Zhang, Chunyan Xu, Jian Yang, Ying Tai, and Liang Chen. Deep hierarchical guidance and regularization learning for end-to-end depth estimation. *Pattern Recognition*, 83:430–442, 2018.

[60] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, pages 9788–9798, 2019.

[61] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, pages 767–783, 2018.

[62] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *ECCV*, pages 101–116, 2018.

[63] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.