Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation

Andrea Pilzer[1], Stéphane Lathuilière[1], Nicu Sebe[1,2], and Elisa Ricci[1,3]

[1]DISI, University of Trento, via Sommarive 14, Povo (TN), Italy
[2]Huawei Technologies Ireland, Dublin, Ireland
[3]Technologies of Vision, Fondazione Bruno Kessler, via Sommarive 18, Povo (TN), Italy

{andrea.pilzer, stephane.lathuiliere, niculae.sebe, e.ricci}@unitn.it

Abstract

Nowadays, the majority of state of the art monocular depth estimation techniques are based on supervised deep learning models. However, collecting RGB images with associated depth maps is a very time consuming procedure. Therefore, recent works have proposed deep architectures for addressing the monocular depth prediction task as a reconstruction problem, thus avoiding the need of collecting ground-truth depth. Following these works, we propose a novel self-supervised deep model for estimating depth maps. Our framework exploits two main strategies: refinement via cycle-inconsistency and distillation. Specifically, first a student network is trained to predict a disparity map such as to recover from a frame in a camera view the associated image in the opposite view. Then, a backward cycle network is applied to the generated image to re-synthesize back the input image, estimating the opposite disparity. A third network exploits the inconsistency between the original and the reconstructed input frame in order to output a refined depth map. Finally, knowledge distillation is exploited, such as to transfer information from the refinement network to the student. Our extensive experimental evaluation demonstrate the effectiveness of the proposed framework which outperforms state of the art unsupervised methods on the KITTI benchmark.

1. Introduction

In the last few years, deep learning-based approaches for depth estimation [4, 20, 24, 37, 12, 42, 27, 31] have attracted a growing interest, motivated, on the one hand, by their ability to predict very accurate depth maps and, on the other hand, by the importance of recovering depth information in several applications, such as robot navigation, autonomous driving, virtual reality and 3D reconstruction.

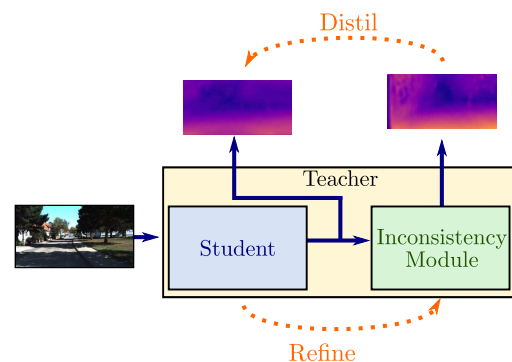Exploiting the availability of very large annotated

Figure 1. Outline of the proposed approach: from the right view image, we predict the left image from which we re-synthesize the right image. The inconsistencies are used by the inconsistency-module to improve the depth estimation. The refined depth maps are used to improve the Student Network via knowledge distillation.

datasets, Convolutional Neural Networks (ConvNets) trained in a supervised setting are now state-of-the-art in many computer vision tasks such as object detection [10], instance segmentation [30], human pose estimation [29]. However, a major weakness of these approaches is the need of collecting large-scale labeled datasets. In the case of depth estimation, acquiring data is especially costly. For instance, in the scenario of depth estimation for autonomous driving, it implies driving a car equipped with a laser LiDaR scanner for hours under diverse lighting and weather conditions. Self-supervised depth estimation, also referred to as unsupervised, recently emerged as an interesting paradigm and an effective alternative to supervised methods [26, 7, 28, 12, 33]. Roughly speaking, in the self-supervised setting, stereo image pairs are considered as input and a deep predictor is learned in order to estimate the associated disparity maps. Specifically, the predicted disparity is employed to synthesize, from a frame in a camera view (*e.g.* from the left camera), the opposite view through warping. The deep network is trained via gradient descent by minimizing the discrepancy between the original and

the reconstructed image. Importantly, even if stereo images pairs are required for training, depth can be recovered from a single image at test time.

In this paper, we follow this research thread and propose a novel self-supervised deep architecture for monocular depth estimation. The proposed approach, illustrated in Fig 1, consists of a first sub-network, referred to as the *student* network, which receives as input an image from a camera view and predicts a disparity map such as to recover the opposite view. On top of this network, we propose several contributions. First, from the generated image, we propose to re-synthesize the input image by estimating the opposite disparity. The resulting network forms a cycle. Second, a third network exploits the cycle inconsistency between the original and the reconstructed input images in order to refine the estimated depth maps. Our intuition is that inconsistency maps provide rich information which can be further exploited, as they indicate where the first two networks fail to predict disparity pixels. Finally, we propose to use the principle of distillation in order to transfer knowledge from the whole network, seen as a *teacher*, to the *student* network. Interestingly, our framework produce two outputs, corresponding to the depth maps estimated respectively by the *student* and the *teacher* networks. This is extremely relevant in practical applications, as the *student* network can be exploited in case of low computation power or real-time constraints.

Our extensive experiments on two large publicly available datasets, *i.e.* the KITTI [8] and the Cityscapes [2] datasets, demonstrate the effectiveness of the proposed framework. Notably, by combining the proposed cycle structure with our inconsistency-aware refinement, our unsupervised framework outperforms previous usupervised approaches, while obtaining comparable results with the state-of-the-art supervised methods on the KITTI dataset.

2. Related Work

In the last decade, deep learning models have greatly improved the performance of depth estimation methods. The vast majority of methods focus on a supervised setting and the problem of predicting depth maps is cast as a pixel-level regression problem [4, 24, 43, 21, 37, 39, 5]. The first ConvNet approach for monocular depth prediction was proposed in Eigen *et al.* [4], where the benefit of considering both local and global information was demonstrated. More recent works improved the performance of deep models by exploiting probabilistic graphical models implemented as neural networks [24, 35, 38, 37]. For instance, Wang *et al.* [35] proposed integrating hierarchical Conditional Random Fields (CRFs) into a ConvNet for joint depth estimation and semantic segmentation. Xu *et al.* [38, 37] exploited CRFs within a deep architecture in order to fuse information at multiple scales. However, supervised approaches rely on expensive ground-truth annotations and, consequently, lack flexibility for deployment in novel environments.

Recently, several works proposed to tackle the depth estimation problem within an unsupervised learning framework [19, 27, 33, 41, 31]. For instance, Garg *et al.* [7] attempted to learn depth maps in an indirect way. They used a ConvNet to predict the right-to-left disparity map from the left image and then reconstructed the right image according to the predicted disparity. They also introduced a network architecture operating based on a coarse-to-fine principle, *i.e.* they employed an encoder-decoder network where the decoder first estimates a low resolution disparity map and then refines it in order to obtain a map at higher resolution. Improving upon [7], Godard *et al.* [12] proposed to use a single generative network to estimate both the left-to-right and the right-to-left disparity maps. Consistency between the two disparities was exploited in form of a loss in order to better constrain the model. Other recent works demonstrated that temporal information and, in particular, considering multiple consecutive frames contribute to improve depth estimation [34, 40, 11, 42]. In particular, Zhou *et al.* [42] exploited temporal information to jointly learn the depth and the camera ego-motion from monocular sequences. Similarly, in [11], a deep network was designed in order to estimate both the depth and the camera pose from three consecutive frames. In this paper we focus on improving *frame-level* unsupervised depth estimation and we do not exploit any additional information such as supervision from related tasks (*e.g.* ego-motion estimation) or temporal consistency. In this respect, our work can be regarded as complementary to [42, 11].

The idea of exploiting cycle-consistency for depth estimation was recently investigated in [31]. Specifically, Pilzer *et al.* [31] introduced a deep architecture for stereo depth estimation which is organized in form of a cycle: two sub-networks, corresponding to the two half-cycles, estimate respectively the left-to-right and right-to-left disparities. They also showed that cycle consistency, together with an adversarial loss, can greatly improve the quality of the predicted depth maps. The main difference with our proposal is that the architecture in [31] is designed for stereo depth estimation whereas we focus on the monocular setting. Moreover, contrary to [31], our architecture exploits cycle inconsistency both at training and at test time. Simultaneously, Tosi *et al.* [32] proposed disparity refinement and Yang *et al.* [39] proposed to compute the error maps between the original input images and their cycle-reconstructed versions and considered them as an additional input to a second network which produces refined depth estimates. Opposite to our approach, the deep model in [39] is trained using supervision derived by Stereo Direct Sparse Odometry [36]. Furthermore, to construct the cycle, we exploit a backward network and introduce a distillation loss.
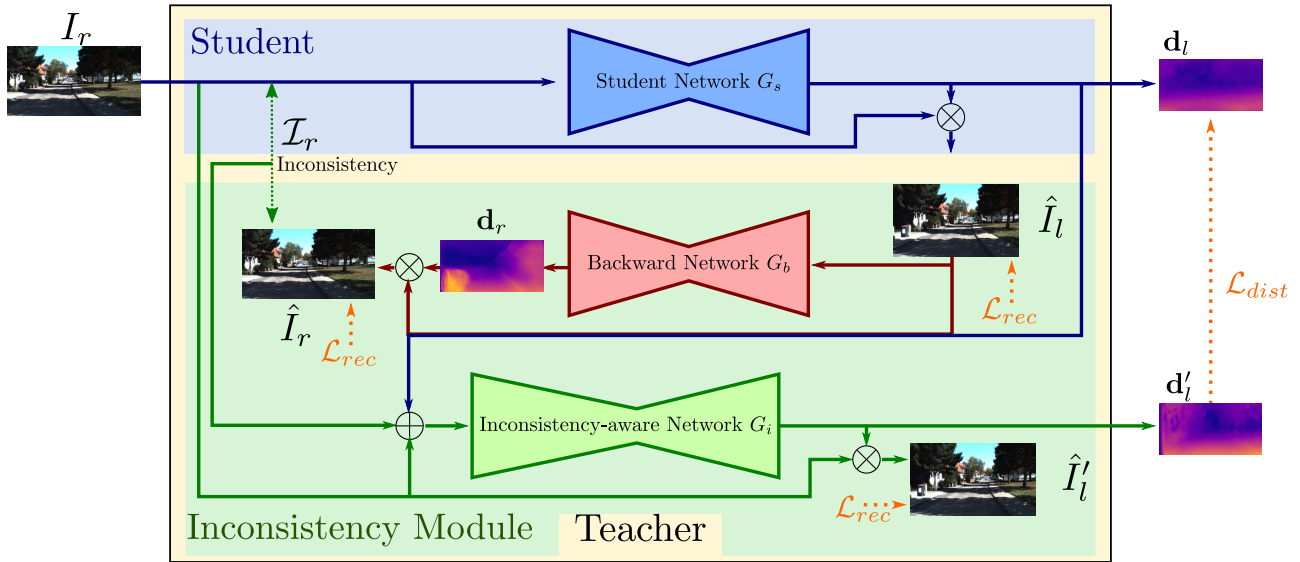
Figure 2. The proposed approach is composed of two modules. A first network G_s predicts the right-to-left disparity map d_l from the right image and synthesizes the left image as described in Sec. 3.4. In the second module, a generator network G_b predicts the left-to-right disparity map d_r in order to re-synthesize the right image. The model obtained in this way forms a cycle. The cycle inconsistency is used by a third network to predict the final disparity map. We use a set of losses (orange dot arrows) detailed in Sec. 3.4

Recently, knowledge distillation attracted a lot of attention [15]. This methodology consists in compressing a large deep network (usually referred to as the *teacher*) into a much smaller model (*student*) operating on the same modality. The *student* network is trained such that its outputs match those of the *teacher*. Knowledge distillation has been exploited for many computer vision tasks such as domain adaptation [14], object detection [1], learning from noisy labels [23] or facial analysis [25]. However, to the best of our knowledge, this work is the first attempt to exploit distillation for depth estimation. We claim that distillation is especially relevant for depth estimation since, in practical applications such as autonomous driving, real-time constraints may impose limitations in term of network size. Note that, we employ an unusual distillation scenario in which the *student* network is a sub-network of the *teacher*.

3. Proposed Method

3.1. Overview

The aim of this work is to estimate the depth of a scene from a single image. However, at training time, we consider that we dispose of pairs of images $\{\mathbf{I}_l, \mathbf{I}_r\}$ of size $H \times W$, derived from a stereo pair and corresponding to the same time instant. Here, $\mathbf{I}_l$ denotes the left camera view and $\mathbf{I}_r$ is the right camera view. Given $\mathbf{I}_r$, we are interested in predicting a correspondence map $\mathbf{d}_l \in \mathbb{R}^{H \times W}$, namely the right-to-left disparity, in which each pixel value represents the offset of the corresponding pixel between the right and the left images. Finally, assuming that the images are rectified, the depth at a pixel location (x, y) of the left image can be recovered from the predicted disparity with $d_l = \frac{f.b}{d(x,y)}$,

where b is the distance between the two cameras and f is the camera focal length.

An overview of the proposed framework is shown in Fig. 2. A first network G_s predicts the right-to-left disparity map $\mathbf{d}_l$ from the right image $\mathbf{I}_r$, and synthesizes the left image by warping $\mathbf{I}_r$ according to $\mathbf{d}_l$. Roughly speaking, the network G_s is trained to minimize the discrepancy between the real and the reconstructed left image (Sec. 3.4).

We employ a second generator network G_b that takes as input the synthesized left image and predicts a left-to-right disparity map $\mathbf{d}_r$ that is used to re-synthesize the right image. The model obtained in this way forms a cycle. This cycle design has three advantages. First, at training time, by sharing weights between G_s and G_b, the networks learn to predict disparity maps from the images of the training set (in the forward half-cycle G_s) but also from the synthesized images (in the backward half-cycle G_b). In that sense, the use of the cycle can be seen as a sort of data augmentation. Second, in order to re-synthesize correctly the right image, the second network G_b requires a correct input left image. Thus, G_b imposes a global constraint on the estimated disparity $\mathbf{d}_l$ oppositely to standard pixel-wise discrepancy losses, such as $\mathcal{L}_1$ or $\mathcal{L}_2$ that act only locally. Third, by comparing the input right image $\mathbf{I}_r$ and the output right image $\hat{\mathbf{I}}_r$ synthesized after applying our cycle framework, we can measure the cycle inconsistency. At a given location of the input image, if we observe no inconsistency, G_s and G_b must have predicted correctly the disparity maps. Conversely, in case of inconsistency, G_s or G_b (or both) must have predicted incorrectly the disparity maps. Note that inconsistencies may also appear on objects regions that are visible in only one of the two views. In-

terestingly, these regions are usually located on the object edges. Therefore, looking at cycle inconsistency also provides information about object edges that can help to predict better depth maps. Importantly, this inconsistency can be measured both at training and testing times, even if at testing time, we dispose only of the right image.

The main contribution of this work consists in exploiting the cycle inconsistency by training a third network in order to improve the prediction performance and output a refined depth map $\mathbf{d}_l'$. In addition, since employing our inconsistency-aware network leads to more accurate depth predictions, we propose to use the disparity maps predicted by G_i in order to improve G_s training via a knowledge distillation approach.

Note that, another possible cycle approach, as proposed in [39], would consist in using a single network to predict the two disparity maps. The two disparities can be used to obtain the synthesized left image and then the re-synthesized right image. Nevertheless, this approach has a major disadvantage with respect to our approach, *i.e.*, since only the warping operator in employed between the two synthesized images, and consequently the receptive field of $\hat{\mathbf{I}}_r$ in $\mathbf{I}_l$ is very small. In particular, when implementing the warping operator via bilinear sampling, the receptive field of the warping operator in only 2×2. Therefore, the right image reconstruction loss can act on the reconstructed left image only locally. Conversely, our backward network G_b imposes a global consistency on $\mathbf{d}_l$ thanks to its large receptive field.

The outputs of our method correspond to the estimated depth maps $\mathbf{d}_l$ and $\mathbf{d}_l'$. While the estimated depth $\mathbf{d}_l'$ corresponding to the teacher model is typically more accurate, in some applications, *e.g.* in resource-constrained settings, it could be convenient to exploit only a small student network.

In the following, we describe the design of our cycled network. Then, we introduce our novel inconsistency-aware network. Finally, we present the optimization objective including our proposed distillation approach.

3.2. Unsupervised Monocular Cycled Network

In this work, we adopt a setting in which the model is trained without the need of ground truth depth maps. This approach is often referred to as unsupervised or self-supervised depth estimation. Roughly speaking, it consists in training a network to predict a disparity map that can be used to generate the left image from the right image. Formally speaking, we employ a first network G_s that takes as input the right image $\mathbf{I}_r$ and predicts the right-to-left disparity $\mathbf{d}_l$. Following [12], we adopt a U-Net architecture for G_s. We employ a warping function $f_w(\cdot)$ that synthesizes the left view image by sampling from $\mathbf{I}_r$ according to $\mathbf{d}_l$:

$$\hat{\mathbf{I}}_l = f_w(\mathbf{d}_l, \mathbf{I}_r). \tag{1}$$

Importantly, $f_w(\cdot)$ is implemented using the bilinear sampler from the spatial transformer network [16] resulting in a fully differentiable model. Consequently, the network can be trained via gradient descent by minimizing the discrepancy between $\hat{\mathbf{I}}_l$ and $\mathbf{I}_l$ (see Sec. 3.4 for details about network training).

Inspired by [31], we employ a second network G_b in order to re-synthesize the right image according to:

$$\hat{\mathbf{I}}_r = f_w(\mathbf{d}_r, \hat{\mathbf{I}}_l). \tag{2}$$

where:

$$\mathbf{d}_r = G_b(\hat{\mathbf{I}}_l) \tag{3}$$

The G_b and G_s networks share their encoder parameters. Note that, differently from the stereo depth model proposed in [31], our second half-cycle network takes only the synthesized left image as input. This crucial difference allows the use of this cycle in the monocular setting at testing time. Concerning the decoder networks, we adopt an architecture composed of a sequence of up-convolution layers in which the disparity is estimated and gradually refined from low to full resolutions similarly to [12]. We obtain the estimated left and the right disparity maps at each scale $\mathbf{d}_l^n$ and $\mathbf{d}_r^n$, $n \in \{0, 1, 2, 3\}$, with sizes $[H/2^n, W/2^n]$. More precisely, $\mathbf{d}_r^n$ is computed from the decoder feature map $\xi_r^{\,n}$ of size $[H/2^n, W/2^n]$ via a convolutional layer. Then, $\mathbf{d}_r^n$ is concatenated with $\xi_{\mathbf{r}}^{\,n}$ obtaining a tensor that is input to an up-convolution layer in order to estimate the disparity at the next resolution $\mathbf{d}_r^{n-1}$.

3.3. Inconsistency-Aware Network

We define the inconsistency tensor as the difference between the input image $\mathbf{I}_r$ and the image $\hat{\mathbf{I}}_r$ predicted by the backward network G_b:

$$\mathcal{I}_r = \mathbf{I}_r - \hat{\mathbf{I}}_r \tag{4}$$

The proposed inconsistency-aware network G_i takes as input the concatenation of $\mathbf{I}_r$, $\mathcal{I}_r$ and $\mathbf{d}_l$. We employ a network architecture similar to the half-cycle monocular network described in Sec. 3.2. However, we propose to provide to the encoder network the disparity maps $\mathbf{d}_l^n, n \in \{1, 2, 3\}$ estimated by G_s at each scale. More precisely, we concatenate along the channel axis each disparity $\mathbf{d}_l^n$ with network features of corresponding dimensionality.

The inconsistency-aware network G_i estimates the right-to-left disparity $\mathbf{d}_l' = G_i(\mathbf{I}_r, \mathcal{I}_r, \mathbf{d}_l, \mathbf{d}_l^{\{1,2,3\}})$ and we reconstruct the left view image $\hat{\mathbf{I}}_l{}'$ by applying the warping function f_w:

$$\hat{\mathbf{I}}_l' = f_w(\mathbf{d}_l', \mathbf{I}_r) \tag{5}$$

Similarly to G_s and G_b, G_i estimates low resolution disparity maps $\mathbf{d'}_{\mathbf{l}}^{\,n}, n \in \{1, 2, 3\}$ that are gradually refined from low to full resolutions.

3.4. Network Training and Knowledge Self-Distillation

In this section, we detail the losses employed to train the proposed network in an end-to-end fashion.

Reconstruction. First, we employ a reconstruction and stucture similarity loss for each network. Following [12], we adopt the $\mathcal{L}_1$ loss to measure the discrepancy between the synthesized and the real images and the structure similarity loss $\mathcal{L}_{SSIM}$ to measure the discrepancy between the synthesized and the real images structure. By summing the losses of the three networks G_s, G_b and G_i, we obtain:

$$\mathcal{L}_{rec}^{(0)} = \lambda_s[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_l, \mathbf{I}_l) + (1-\alpha)||\hat{\mathbf{I}}_l - \mathbf{I}_l||_1]$$
$$+ \lambda_b[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_r, \mathbf{I}_r) + (1-\alpha)||\hat{\mathbf{I}}_r - \mathbf{I}_r||_1] \quad (6)$$
$$+ \lambda_t[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_l', \mathbf{I}_l) + (1-\alpha)||\hat{\mathbf{I}}_l' - \mathbf{I}_l||_1]$$

where λ_s, λ_b and λ_t are adjustment parameters and $\alpha = 0.85$. Similarly, we also compute a reconstruction loss $\mathcal{L}_{rec}^{(n)}$ for the low resolution disparity maps. Following [11], we upsample the low resolution $\mathbf{d}_l^n$, $\mathbf{d}_r^n$ and $\mathbf{d}_l'^n$ to $H \times W$ and use the warping operator f_w to re-synthesize full resolution images that are compared with the real images according to the $\mathcal{L}_1$ loss. The total reconstruction loss is:

$$\mathcal{L}_{rec} = \sum_{n=0}^{4} \mathcal{L}_{rec}^{(n)} \quad (7)$$

Self-Distillation. Finally, we propose to introduce a knowledge distillation loss. As detailed in the experimental section (Sec 4), the inconsistency-aware network outperforms by a significant margin the simple half-cycle network G_s. This boost is at the cost of a higher computation complexity. The idea of the proposed self-distillation loss consists in distilling knowledge from inconsistency-aware network to the half-cycle network G_s. Thus, we improve the performance of G_s without adding any computation complexity at testing time. To do so, we evaluate disparity and feature distillation. For the first, we impose that the network G_d predicts disparity maps similar to the output of inconsistency-aware network. It can be seen as a distillation approach where G_s plays the role of the *student* and the whole network (composed of G_s, G_b and G_i) is the *teacher*. However, in our particular case, the *student* network is a sub-network of the *teacher*. From this perspective, we name this approach self-distillation. The self-distillation loss is given by:

$$\mathcal{L}_{dist} = ||\mathbf{d}_l - \mathcal{S}(\mathbf{d}_l')||_1 \quad (8)$$

where $\mathcal{S}$ denotes the stop-gradient operation. In particular, the stop-gradient operation equals the identity function when computing the forward pass of the back-propagation algorithm but it has a null gradient when computing the backward pass. The purpose of the stop-gradient is to avoid

that $\mathbf{d}_l'$ converges to $\mathbf{d}_l$. On the contrary, the goal is to help $\mathbf{d}_l$ to become as accurate as $\mathbf{d}_l'$.

For the second, we impose that the decoder features $\xi_r'^n, n \in 0, 1, 2$ of the *teacher* are similar to the features ξ_r^n of the *student*. The self-distillation loss is given by:

$$\mathcal{L}_{dist} = ||\xi_r^n - \mathcal{S}(\xi_r'^n)||_2 \quad (9)$$

The total training loss is given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{rec} + \lambda_{dist}\mathcal{L}_{dist} \quad (10)$$

4. Experiments

We evaluate our proposed approach on two publicly available datasets and compare its performance with state of the art methods.

4.1. Experimental Setup

Datasets. We perform experiments on two large stereo images datasets, *i.e.* KITTI [9] and Cityscapes [3]. Both datasets are recorded from driving vehicles. Concerning the *KITTI* dataset, we employ the training and test split of Eigen *et al.* [4]. This split is composed of 22,600 training image pairs, and 697 test pairs. We consider data-augmentation with online random flipping of the images during training as in [12]. For Cityscapes, images were collected with higher resolution. To train our model we combine images from the densely and coarse annotated splits to obtain 22,973 image-pairs as in [31]. The test split is composed of 1,525 image-pairs of the densely annotated split. The evaluation is performed using the pre-computed disparity maps.

Evaluation Metrics. The quantitative evaluation is performed according to several standard metrics used in previous works [4, 12, 35]. Let P be the total number of pixels in the test set and $\hat{d}_i$, d_i the estimated depth and ground truth depth values for pixel i. We compute the following metrics:

- Mean relative error (abs rel): $\frac{1}{P}\sum_{i=1}^{P} \frac{||\hat{d}_i - d_i||}{d_i}$,
- Squared relative error (sq rel): $\frac{1}{P}\sum_{i=1}^{P} \frac{||\hat{d}_i - d_i||^2}{d_i}$,
- Root mean squared error (rmse): $\sqrt{\frac{1}{P}\sum_{i=1}^{P}(\hat{d}_i - d_i)^2}$,
- Mean log 10 error (rmse log): $\sqrt{\frac{1}{P}\sum_{i=1}^{P} ||\log\hat{d}_i - \log d_i||^2}$
- Accuracy with threshold τ, *i.e.* the percentage of $\hat{d}_i$ such that $\delta = \max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < \alpha^\tau$. We employ $\alpha = 1.25$ and $\tau \in [1, 2, 3]$ following [4].

4.2. Baselines for Ablation.

To perform the ablation study presented in Sec.4.3, we consider the following baselines:

- *half-cycle*: our basic building block, uses the forward branch that takes $\mathbf{I}_r$ as input and generates $\mathbf{d}_l$ to reconstruct the other stereo view $\hat{\mathbf{I}}_l$. Neither cycle-consistency nor self-distillation are used in this model.
- *cycle*: a backward network is added to the *half-cycle* model in order to reconstruct $\hat{\mathbf{I}}_r$ from the estimated $\hat{\mathbf{I}}_l$. Note that the backward network is used only at training time. At test time, the output is the same as for the *half-cycle* model.
- *teacher*, we stack the inconsistency-aware network after the *cycle* as described in Sec 3.3.
- *student*: the output of the inconsistency-aware network is distilled in order to refine the first *half-cycle*. At test time, the output and the computation complexity are the same as in the *half-cycle* model.

In Tables 1, 2 and 3 we indicate with *HC*, *C*, *T* and *S*, the *half-cycle*, *cycle*, *teacher* and *student* respectively; *feat* and *disp* denote self-distillations of features and disparities.

Training Procedure. The whole network is trained following an iterative procedure. First, we start by training the forward *half-cycle* network for 10 epochs. In a second step, we train the backward network decoder for 5 epochs without updating the first half-cycle network. The whole cycle is then jointly trained for further 10 epochs. Then, the inconsistency-aware module is pretrained for 5 epochs. Finally, the whole network is jointly fine-tuned for 10 epochs.

Parameters. The model is implemented with the deep learning library *TensorFlow*. Similarly to [12], the input images are down-sampled to a resolution of 512×256 from the original sizes which are 1226×370 for the KITTI dataset and for CityScapes. In all our experiments we use a batch size equal to 8 stereo image pairs and the Adam optimizer with learning rate set to 10^{-5} following the recommendations of [22].

The *half-cycle* and *cycle* networks are trained with the following loss parameters $\lambda_s = 1$, $\lambda_b = 0.1$ and $\lambda_t = 0$. When training the *teacher* network we use $\lambda_s = 0$, $\lambda_b = 0$ and $\lambda_t = 1$. We weight the distillation loss $\mathcal{L}_{dist}$ with $\lambda_{dist} = 0.005$ and $\lambda_{dist} = 0.1$ respectively, if feature distillation or disparity distillation is applied. The joint training of the full network is done with learning rate $l_r = 10^{-5}$, loss parameters $\lambda_s = 1$, $\lambda_b = 0.1$, $\lambda_t = 1$ and λ_{dist} equal to 0.005 in the case feature distillation and 0.1 in the case of disparity distillation, respectively.

4.3. Results

Ablation Study. To demonstrate the validity of the proposed contributions we first conduct an ablation study on the KITTI dataset [9] and the CityScapes dataset [3]. Results are shown in Table 1 and Table 3, respectively.

We split the ablation in two parts where we employ two different reconstruction loss variants. For the first part, as in [12], we use a multi-scale reconstruction loss where the

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
HC	0.1487	1.2942	5.800	0.246	0.805	0.925	0.965
C	0.1451	1.2943	5.850	0.242	0.796	0.924	0.967
T *feat*	*0.1220*	*1.0433*	5.321	0.229	*0.834*	0.933	*0.968*
T *disp*	0.1234	1.0509	*5.283*	*0.228*	0.834	*0.934*	0.968
S *feat*	0.1438	1.2806	5.834	0.241	0.797	0.926	0.968
S *disp*	0.1438	1.2551	5.771	0.238	0.797	0.927	0.969
[11] $\mathcal{L}_1$ loss							
T *feat*	0.1017	0.8930	4.768	0.206	0.878	0.946	0.972
T *disp*	**0.0983**	**0.8306**	**4.656**	**0.202**	**0.882**	**0.948**	**0.973**
S *feat*	0.1474	1.2416	5.849	0.241	0.788	0.923	0.968
S *disp*	0.1424	1.2306	5.785	0.239	0.795	0.924	0.968

Table 1. Ablation study on KITTI dataset using the training and testing split proposed by Eigen *et al.* [4]. The upper part shows the results with the multiscale reconstruction $\mathcal{L}_1$ loss in [12], the bottom part with the $\mathcal{L}_1$ loss proposed in [11].

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
1-CN C	0.1533	1.3326	5.837	0.240	0.785	0.919	0.967
1-CN S *disp*	0.1503	1.2622	5.868	0.243	0.783	0.918	0.967
Ours S *disp*	**0.1438**	**1.2551**	**5.771**	**0.238**	**0.797**	**0.927**	**0.969**
1-CN T *disp*	0.1478	1.3609	5.952	0.243	0.793	0.921	0.966
Ours T *disp*	**0.1234**	**1.0509**	**5.283**	**0.228**	**0.834**	**0.934**	**0.968**

Table 2. Ablation study where our two-network *cycle* is replaced by the single-network *cycle* from Yang *et al.* [39] (referred as to 1-CN).

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
HC	0.4676	7.3992	5.741	0.493	0.735	0.890	0.945
C	0.4523	6.2604	5.381	0.557	0.736	0.888	0.946
T *feat*	0.4087	5.8777	4.394	0.334	0.846	0.940	0.967
T *disp*	*0.3988*	*5.8752*	*4.293*	*0.316*	*0.848*	*0.941*	*0.968*
S *feat*	0.4494	6.2599	5.343	0.421	0.739	0.891	0.947
S *disp*	0.4467	5.9012	5.297	0.473	0.736	0.890	0.946
[11] $\mathcal{L}_1$ loss							
T *feat*	0.3878	**5.8190**	**4.123**	0.397	0.861	**0.945**	**0.969**
T *disp*	**0.3846**	6.2007	4.476	**0.318**	**0.864**	**0.945**	**0.969**
S *feat*	0.4455	6.2748	5.366	0.468	0.739	0.891	0.946
S *disp*	0.4305	5.9552	5.281	0.519	0.740	0.891	0.946

Table 3. Ablation study on the Cityscapes dataset. The upper part shows the results with the multiscale reconstruction $\mathcal{L}_1$ loss in [12], the bottom part with the $\mathcal{L}_1$ loss proposed in [11].

smaller scale reconstruction is compared with a downsampled version of the stereo image. In contrast with that, for the second part, we employ a more effective reconstruction loss, upsampling to input scale all the disparities before warping as described in Sec. 3.4.

In Table 1 it is interesting to note that our intuition of self-constraining the monocular student network with cycled design improves, without requiring additional losses, in several of the metrics compared to the simple forward branch. This comes at the cost of doubling the forward propagation time at training but not at testing time. Moreover, the monocular cycled structure has the big advantage of automatically computing the inconsistency of the reconstruction both at training and testing time. Therefore, stacking a network aware of the inconsistencies and previous estimations, the *teacher* network, improves the performance. We observe that our proposed inconsistency-aware network brings an important improvement consistent over all the metrics, *e.g.* 14% and 18% in *Abs Rel* and *Sq Rel*, respec-
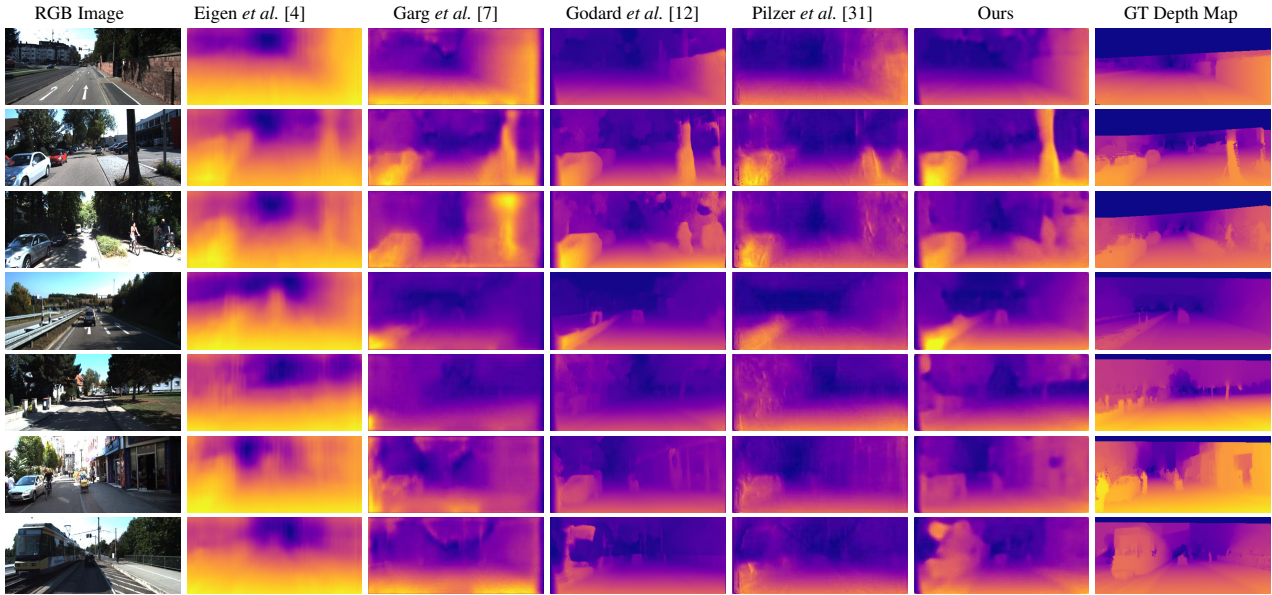
Figure 3. Qualitative comparison of different state-of-the-art models with our *teacher* network on the KITTI testing split proposed by [4]. The sparse KITTI ground truth depth maps are interpolated with bilinear interpolation for better visualization.

Method	Sup	Video	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			lower is better				higher is better		
Eigen *et al.* [4]	Y	N	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Xu *et al.* [37]	Y	N	0.132	0.911	-	*0.162*	0.804	0.945	0.981
Jiang *et al.*[17]	Y	N	0.131	0.937	5.032	0.203	0.827	0.946	0.981
Gan *et al.* [6]	Y	N	0.098	0.666	*3.933*	0.173	*0.890*	0.964	0.985
Guo *et al.* [13]	Y	N	*0.097*	*0.653*	4.170	0.170	0.889	*0.967*	*0.986*
Yang *et al.* [39]	Y	Y	*0.097*	*0.734*	*4.442*	*0.187*	*0.888*	*0.958*	*0.980*
Zou *et al.*[44]	N	Y	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Godard *et al.*[11]	N	Y	0.115	1.010	5.164	0.212	0.858	0.946	0.97
Zhou *et al.* [42]	N	N	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Garg *et al.* [7]	N	N	0.169	1.08	5.104	0.273	0.740	0.904	0.962
Kundu *et al.* [18], 50m	N	N	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Godard *et al.* [12]	N	N	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Pilzer *et al.* [31]	N	N	0.152	1.388	6.016	0.247	0.789	0.918	0.965
Ours Student	N	N	0.1424	1.2306	5.785	0.239	0.795	0.924	0.968
Ours Teacher	N	N	**0.0983**	**0.8306**	**4.656**	**0.202**	**0.882**	**0.948**	**0.973**

Table 4. Comparison with the state of the art. Training and testing are performed on the KITTI [9] dataset. Supervised and semi-supervised methods are marked with Y in the supervision (Sup.) column, unsupervised methods with N. Methods using a frame sequence in input and, thus, exploiting temporal information either at training or testing time, are marked with Y in the *Video* column. Numbers are obtained on Eigen [4] test split with Garg [7] image cropping. Depth predictions are capped at the common threshold of 80 meters, if capped at 50 meters we specify it. Best scores among static unsupervised methods are in bold. Best scores among other method categories are in italic.

tively, comparing *cycle* and *teacher*.

Student-teacher distillation leads to a consistent improvement over all metrics, demonstrating that self-distillation improves the *student*, while keeping the performance of teacher constant. Regarding the two distillation strategies, we found that network with disparity distillation converges faster than that with the feature distillation. This is not unexpected, given the much more compact size of the disparity compared to the several channels of the features.

For demonstrating the validity of the design of our cycle network, we perform an ablation study where our two-network *cycle* structure is replaced by the single-network *cycle* proposed by Yang *et al.* [39]. In this experiment, we use our proposed inconsistency-aware module to exploit the inconsistency estimated by the single network cycle in [39]. Contrary to [39], we trained the models without supervision in order to compare the two different approaches in the unsupervised setting. We use the $\mathcal{L}_1$ loss from [12] for fair comparison. Results are reported in Table 2. We observe that the inconsistency estimates obtained with the single-network cycle of [39] are associated with worse performance with respect to those of our method.

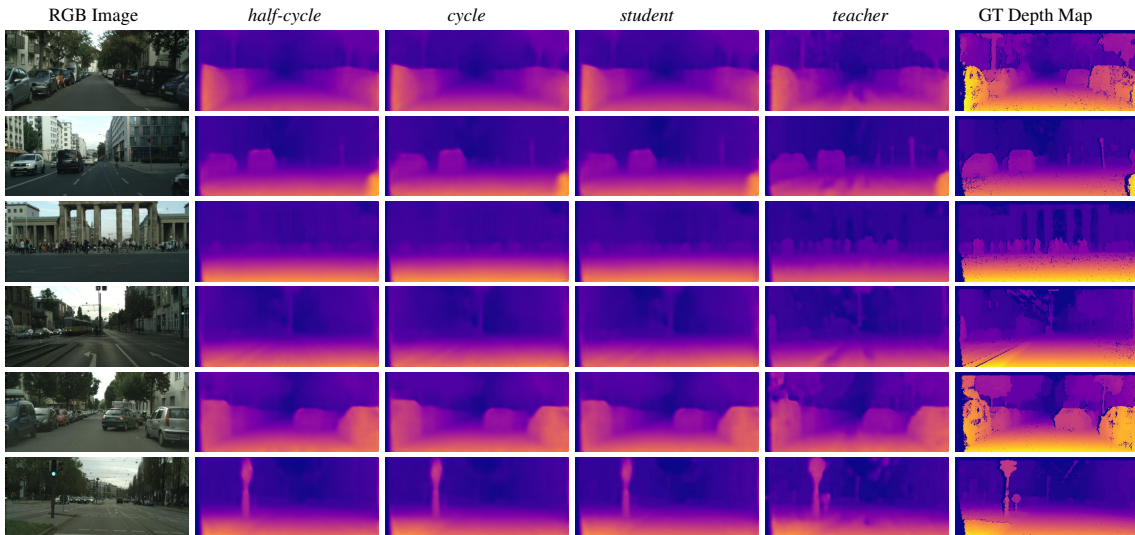| RGB Image | half-cycle | cycle | student | teacher | GT Depth Map |

Figure 4. Qualitative comparison of different baseline models of the proposed approach on the Cityscapes testing dataset.

We also performed an ablation study on the Cityscapes dataset in Table 3, following the evaluation procedure proposed in [31]. The results confirm the trends observed on KITTI. The *cycle* network improves over the *half-cycle* in five metrics out of seven. The *teacher*, effectively exploiting inconsistencies, is associated with an improvement on all error metrics (ranging from 7% to 20%). Distillation further provides a boost in performance of about 1.5% to 5%. In the second part of the ablation study, the *teacher* further improves its estimations gaining over 20% over the initial *cycle* setting. More interesting is the gain in performance of the *student* that improves from 2% to 5%.

In Fig. 4, we present qualitative results for Cityscapes. *half-cycle* and *cycle* images are smooth and do not present artifacts. The *teacher* provides more accurate depth maps with sharper edges for small objects and better background estimations (*e.g.* third row, people in the back). After distillation also the *student* inherits this ability and we observe more detailed predictions compared to the original *cycle*.

4.4. Comparison with State-of-the-Art

In Table 4 we compare with several state-of-the-art works, considering both supervised learning-based (Eigen *et al.* [4], Xu *et al.* [37], Jiang *et al.* [17], Gan *et al.* [6], Guo *et al.* [13], Yang *et al.* [39]) and unsupervised learning-based (Zhou *et al.* [42], Garg *et al.* [7], Kundu *et al.* [18], Godard *et al.* [12], Pilzer *et al.* [31], Godard *et al.* [11] and Zou *et al.*[44]) methods.

The *teacher* network reaches state-of-the-art performance for the frame-level unsupervised setting, even improving over the state-of-the-art method that use depth supervision as [37], and is competitive with those using depth and video clues [6, 13, 39]. Note that Yang *et al.* [39] consider a similar setting to ours proposing to use errors to refine the depth estimation with a stacked network. Our

method has several advantages though: it is unsupervised, it does not consider multiple video frames and it avoids the use of several losses whose hyper-parameters are hard to tune. Furthermore, as demonstrated by our experiments in Table 2, our approach adopts a more effective network structure for computing cycle inconsistencies. The *student* network, after distillation, improves on unsupervised approaches with similar network capacity like [7, 12, 31] and it is only outperformed by previous unsupervised methods that exploit additional information during training like [11].

Qualitative results in Figure 3 show that our model predicts more accurately challenging areas, *i.e.* sky, trees in background and shadowed areas difficult to interpret, compared to competitive unsupervised models [7, 12, 31]. Note that small details are better reconstructed by [12] but, overall, our estimations look smoother and have fewer large errors, as the train windshield in row seven.

5. Conclusions

We proposed a monocular depth estimation network which computes the inconsistencies between input and cycle-reconstructed images and exploit them to generate state-of-the-art depth predictions through a refinement network. We proved that distillation is an effective paradigm for depth estimation and improve the student network performance by transferring information from the refinement network. In future work we plan to further improve the distillation process by accounting for teacher and student confidence in the estimates. In this way we expect to better guide the learning process and correct more effectively prediction inconsistencies.

6. Acknowledgement

References

[1] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, 2017.

[2] 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*, 2016.

[3] 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*, 2016.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

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

[6] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *ECCV*, 2018.

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

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

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

[10] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[11] Clément Godard, Oisin Mac Aodha, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018.

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

[13] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, September 2018.

[14] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Workshop on Deep Learning*, 2015.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *NIPS*. 2015.

[17] Huaizu Jiang, Gustav Larsson, Michael Maire Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In *ECCV*, 2018.

[18] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *CVPR*, 2018.

[19] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *CVPR*, 2017.

[20] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *CVPR*, 2014.

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

[22] Stéphane Lathuilire, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *To appear in TPAMI*, 2019.

[23] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, 2017.

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

[25] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, Xiaoou Tang, et al. Face model compression by distilling knowledge from neurons. In *AAAI*, 2016.

[26] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *CVPR*, 2016.

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

[28] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[29] Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Pose partition networks for multi-person pose estimation. In *ECCV*, 2018.

[30] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Semi-convolutional operators for instance segmentation. In *ECCV*, 2018.

[31] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *3DV*, 2018.

[32] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation by infusing traditional stereo knowledge. In *CVPR*, 2019.

[33] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[34] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[35] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[36] Rui Wang, Martin Schwörer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *ICCV*, 2017.

[37] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *TPAMI*, 2018.

[38] 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.

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

[40] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. *arXiv preprint arXiv:1806.10556*, 2018.

[41] 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. *arXiv preprint arXiv:1803.03893*, 2018.

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

[43] Wei Zhuo, Mathieu Salzmann, Xuming He, and Miaomiao Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015.

[44] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.