Learning monocular depth estimation with unsupervised trinocular assumptions

Matteo Poggi, Fabio Tosi, Stefano Mattoccia
University of Bologna, Department of Computer Science and Engineering
Viale del Risorgimento 2, Bologna, Italy
{m.poggi, fabio.tosi5, stefano.mattoccia}@unibo.it

Abstract

Obtaining accurate depth measurements out of a single image represents a fascinating solution to 3D sensing. CNNs led to considerable improvements in this field, and recent trends replaced the need for ground-truth labels with geometry-guided image reconstruction signals enabling unsupervised training. Currently, for this purpose, state-of-the-art techniques rely on images acquired with a binocular stereo rig to predict inverse depth (i.e., disparity) according to the aforementioned supervision principle. However, these methods suffer from well-known problems near occlusions, left image border, etc inherited from the stereo setup. Therefore, in this paper, we tackle these issues by moving to a trinocular domain for training. Assuming the central image as the reference, we train a CNN to infer disparity representations pairing such image with frames on its left and right side. This strategy allows obtaining depth maps not affected by typical stereo artifacts. Moreover, being trinocular datasets seldom available, we introduce a novel interleaved training procedure enabling to enforce the trinocular assumption outlined from current binocular datasets. Exhaustive experimental results on the KITTI dataset confirm that our proposal outperforms state-of-the-art methods for unsupervised monocular depth estimation trained on binocular stereo pairs as well as any known methods relying on other cues.

1. Introduction

Depth plays a crucial role in many computer vision applications and active 3D sensors are becoming very popular. Nonetheless, such sensors may have severe shortcomings. For instance, the Kinect 1 is not suited at all for outdoor environments flooded by sunlight. Moreover, such sensor allows only for close range depth measurements. On the other hand, a popular active depth sensor perfectly suited for outdoor environments is LIDAR (e.g., Velodyne). However, sensors based on such technology are typically expensive and often cumbersome for some practical applications.

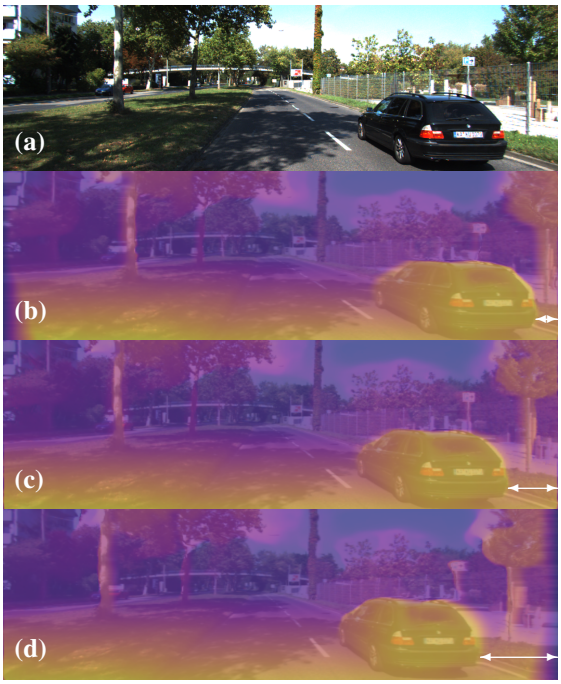

Figure 1. Overview of 3Net. a) Given a single reference image from KITTI 2015 training set [29], our network learns depth representations according to two additional points of view on left (b) and right (d) of the input (a), enabling to infer a more accurate depth map (c). White arrows highlight the different points of view.

Thus, inferring depth with passive sensors based on standard imaging technology would be highly desirable being cheap, lightweight and suited for indoor and outdoor environments. In this context, acquiring images from different viewpoints allows inferring depth exploiting geometry constraints. On the other hand, estimating depth from a single image is indeed an ill-posed problem. Nonetheless, this latter approach would overcome some major constraints such as the need for simultaneous acquisition in binocular stereo or handling dynamic objects in depth-from-motion approaches. Although a geometrically ambiguous problem, Convolutional Neural Networks (CNNs) achieved outstanding results in monocular depth estimation by casting it as

a learning task in both supervised and unsupervised manner. In particular, the latter paradigm addresses the hunger for data typical of deep learning tasks by training networks to produce a depth representation minimizing the warping error between images acquired from multiple points of view rather than the error with respect to difficult to source ground-truth depth labels. In this field, the work of Godard et al. [12] represents state-of-the-art for unsupervised monocular depth estimation. Deploying stereo imagery for training, a CNN learns to infer disparity from a single reference image and warps the target image accordingly to minimize the appearance error between the warped and the reference image. This strategy yields state-of-the-art performance [12, 41]. The CNN is trained to infer disparity from a single reference image and the target image is warped accordingly minimizing the appearance error between warped and reference image. This way, the depth representation learned by the network is affected by artifacts in specific image regions inherited from the stereo setup (e.g., the left border using the left image as the reference) and in occluded areas. The post-processing step proposed in [12] partially compensates for these artifacts. However, it requires a double forward of the input image and its horizontally flipped version thus obtaining two predictions with artifacts, respectively, on the left and right side of depth discontinuities. Such issues are softened in the final map at the cost of doubling processing time and memory footprint.

In this paper, we propose to explicitly take into account these artifacts training our network on imagery acquired by a trinocular setup. By assuming the availability of three horizontally aligned images at training time, our network learns to process the frame in the middle and produce inverse depth (i.e., disparity) maps according to all the available viewpoints. By doing so, we can attenuate the aforementioned occlusion artifacts because they occur in different regions of the estimated outputs. However, since trinocular setups are generally uncommon and hence datasets seldom available, we will show how to rely on popular stereo datasets such as CityScapes [3] and KITTI [11] to enforce our trinocular training assumption. Experimental results clearly prove that, deploying stereo pairs with a smart strategy aimed at emulating a trinocular setup, our Three-view Network (3Net) is able anyway to learn a three-view representation of the scene as shown intuitively in Figure 1 and how it leads to more robust monocular depth estimation compared to state-of-the-art methods trained on the same binocular stereo pair with a conventional paradigm. Figure 1 highlights the behavior of 3Net: we can see how disparity maps (b) and (d), from the point of view of two frames respectively on the left and right side of the reference image, show mirrored artifacts in occluded regions. Combining the two opposite views enables to compensate for these issues and produces a more accurate map (c) centered on the reference frame. Please note that KITTI does not explicitly contain trinocular views as those shown in Figure 1 and that this behavior is learned by 3Net trained only on standard binocular data. Indeed, images and depth maps in (b) and (d) are inferred by our network. Exhaustive experimental results on the KITTI 2015 stereo dataset [29] and the Eigen split [6] of the KITTI dataset [11] clearly show that 3Net, trained on standard binocular stereo pairs, improves state-of-the-art methods for unsupervised monocular depth estimation, regardless of the cues deployed for training.

2. Related Work

In this section, we review the literature concerning single view depth estimation in both supervised and unsupervised manner. Moreover, we also consider early works on multi-baseline stereo setup being these approaches relevant to our proposal.

Supervised depth-from-mono. The following techniques share the need for difficult to source ground-truth depth measurements for training, thus posing a substantial limitation to their practical deployment. Saxena et al. [33] estimated depth and local planes using a MRF framework. Ladicky et al. [21] proved that semantic can help depth estimation using a boosting classifier. More recently, CNN has emerged as mainstream strategy to estimate depth from a single image [6, 24, 22, 23]. Ummenhofer et al. [35] proposed DeMoN, a deep model to infer both depth and ego-motion from a pair of subsequent frames acquired by a single camera. Fu et al. [8] introduced a novel strategy to discretize depth and cast the learning process as an ordinal regression problem, while Xu et al. [39] integrated CRF models into deep architectures to improve depth prediction. Luo et al. [25] formulated the monocular depth estimation problem as a view synthesis procedure followed by a deep stereo matching approach. Kumar et al. [4] introduced a Recurrent Neural Network (RNN) aimed at predicting depth from monocular video sequences. Lastly, Atapour et al. [1] exploited image style transfer and adversarial training to predict depth from real images training the network on a large amount of synthetic data.

Unsupervised depth-from-mono. Rethinking depth estimation as an image reconstruction task allowed to avoid the need for ground-truth depth labels and some works concerned with view synthesis paved the way for this purpose. Flynn et al. [7] proposed DeepStereo to generate new points of view training on images acquired by multiple cameras. Xie et al. [38] trained their Deep3D framework to create, from a single image, a target frame paired with the input according to a stereo setup by learning a disparity representation.

Unsupervised monocular depth estimation methods can be broadly categorized into two main categories according to the cues used to replace ground-truth labels. The first one

[10, 12] leverages images with known relative camera pose, typically acquired by a calibrated stereo rig, following the strategy outlined by Deep3D [38]. A seminal work using this methodology was proposed by Garg et al. [10]. Godard et al. [12] deploying spatial transformer networks [17] and left-right consistency were able to notably improve depth accuracy. More compact models [32] can be trained the same way and deployed on embedded systems as well.

The second category concerns the use of imagery acquired by an unconstrained moving camera [42, 27]. Differently, from the previous methodology, temporally adjacent frames acquired by a single moving camera may contain dynamic objects that need to be explicitly handled during re-projection. Moreover, camera pose is unknown and needs to be estimated together with depth. On the other hand, such a strategy does not require a stereo camera to collect training samples. On this track, Zhou et al. [42] proposed a model to infer depth from unconstrained video sequences by computing a reconstruction loss between subsequent frames and predicting, at the same time, the relative pose between them. This strategy was improved by Mahjourian et al. [27] thanks to a 3D point-cloud alignment loss and by Wang et al. [36] including a differentiable implementation of Direct Visual Odometry (DVO) with a novel depth normalization strategy. Yin et al. [40] proposed GeoNet, a framework for depth and optical flow estimation from monocular sequences. Finally, we mention the work of Zhan et al. [41] which combined both strategies (i.e., training on stereo sequences) and the semi-supervised works of Kuznietsov et al. [20] and Kumar et al. [5].

Multi-baseline stereo. It is generally recognized that using more than two views has the potential to improve the quality of depth estimation. An early work concerning multi-camera stereo was proposed by Minoru and Akira [16] deploying a triangular rig, while Okutomi and Kanade [31] achieved accurate depth measurements combining stereo from multiple baseline cameras horizontally aligned. Kang et al. [18] proposed a method to handle the increasing number of occlusions occurring in multi-view stereo setup, while Ayache and Lustman [2] designed a three cameras rig for robotic applications and Garcia et al. [9] proposed a pose detection algorithm based on a trinocular stereo system. In the last decade, along with the availability of off-the-shelf stereo cameras (e.g., Intel RealSense) some multi-baseline stereo systems too were commercially made available. For instance, the Bumblebee XB3 was used to acquire the RobotCar dataset [26], counting *millions* of images acquired driving for about 1000 Km. Honneger et al. [15] developed a multi-baseline camera with on-board FPGA, enabling real-time processing of dense disparity maps. Therefore, a trinocular stereo configuration for training, like the one we advocate in our work, would be undoubtedly feasible. Nonetheless, our strategy is feasible and useful even with conventional binocular datasets.

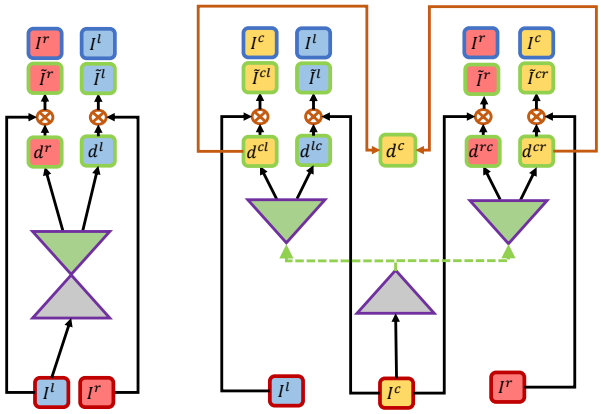

Figure 2. Training frameworks enforcing a) binocular [12] and b) trinocular assumptions.

3. Method overview

In this section, we propose a framework aimed at enforcing a trinocular assumption for training in an unsupervised manner a network for monocular depth estimation. We will outline the rationale behind this choice and the differences with known techniques in the literature. Then, deploying a conventional binocular stereo dataset, we will show how our strategy allows advancing state-of-the-art.

3.1. Trinocular assumption and network design

While traditional depth-from-mono frameworks learn to estimate $d(I)$ from an input image I by minimizing the prediction error with respect to a ground-truth map $\hat{d}(I)$ whose pixels are labelled with real depth measurements, the introduction of image-reconstruction based losses moved this task to an unsupervised learning paradigm. In particular, estimated depth is used to project across different points of view exploiting 3D geometry and camera pose thus obtaining supervision signals through the minimization of the re-projection error. According to the literature reviewed in Section 2, the training methodology based on images acquired with a stereo camera, as in [10, 12], removes the need to infer pose estimation required when gathering data with a single unconstrained camera.

Coaching a CNN to infer depth emulating a stereo system for training introduces artifacts in the learned representation (i.e., disparity) intrinsically because of well-known issues arising when dealing with pixels having no direct matches across the two images, such as on left border or occlusions. Godard et al. [12] deal with this problem using a simple, yet effective, trick. By processing a horizontally flipped input image and then back-flipping the result,

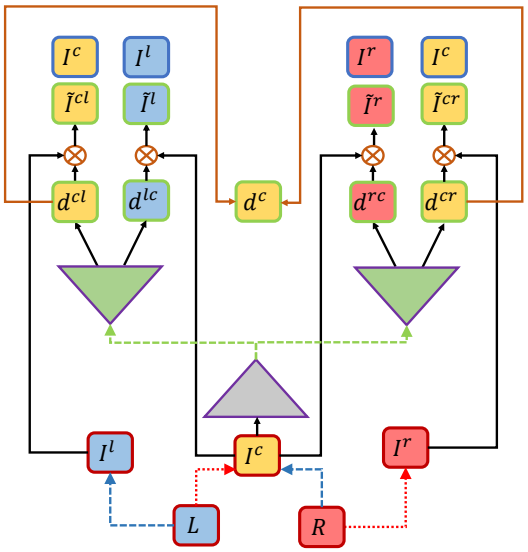

Figure 3. Scheme of interleaved training. A binocular stereo pair is used to train the network enforcing a trinocular assumption by first setting $L \rightarrow I^l$, $R \rightarrow I^c$ (blue arrows) and optimizing the model according to losses on $\tilde{I}^l$, $\tilde{I}^{cl}$, then setting $L \rightarrow I^c$, $R \rightarrow I^r$ (red arrows) and optimizing the model according to losses on $\tilde{I}^{cr}$, $\tilde{I}^r$.

artifacts will appear on the opposite side w.r.t. the result obtained on the un-flipped frame (e.g., on the right border rather than on the left). Thus, combining the two predictions allows removing artifacts partially. Nevertheless, this strategy requires two forwards, hence doubling memory footprint and runtime, which would not be necessary if the CNN could learn to estimate disparity concerning a frame acquired on the left w.r.t reference image. Guided by this intuition, we rethink the training protocol of [12] to exploit a trinocular configuration, on which the image we want to learn the depth of is the central frame of three horizontally aligned points of view. Figure 2 gives an overview of our framework b) and the one by Godard et al. [12] leveraging binocular stereo a). While a) trains the network to estimate a depth representation for I^l by means of disparity map d^l, used to warp I^r to $\tilde{I}^l$ and measure the appearance difference with I^l, we process I^c to obtain d^{lc} and d^{cr}, disparity maps assuming as target I^l and I^r, then we warp these latter two images to obtain $\tilde{I}^{cl}$ and $\tilde{I}^{cr}$ to finally compute supervision signals as re-projection error w.r.t. I^c. Finally, in our framework, d^{lc} and d^{cl} are combined to attenuate occlusions and obtain the final d^c from a single forward pass, conversely to [12] which requires two forwards. Eventually, as [12] estimates d^r to enforce losses between $\tilde{I}^r$,I^r and the LRC consistency, our network generates d^{lc} and d^{rc} to exploit losses between $\tilde{I}^l$,I^l and $\tilde{I}^r$,I^r.

Figure 2 also highlights a further main difference between the two frameworks. While a traditional UNet architecture is used by previous works in literature [42, 12], we build two separate decoders respectively in charge of estimating d^{cl} and d^{cr} separately. This strategy adds a negligible overhead regarding memory and runtime requirements, being the encoder the most computationally expensive module of the framework (i.e., the decoder mostly applies upsampling operations). According to our experiments, training a single decoder to infer a disparity representation for both points of view yields slightly worse results.

3.2. Interleaved training for binocular images

To effectively learn mirrored representation and compensate for occlusions/borders, the framework outlined so far relies on a set of three horizontally aligned images at training time. Although sensors designed to acquire such imagery are currently available, for instance the aforementioned Bumblebee XB3, it is still quite uncommon to find publicly available images obtained in such configuration. Indeed, in this sense, the Oxford RobotCar dataset [26] represents an exception providing a large amount of street scenes captured with the trinocular XB3 sensor. Unfortunately, the provided calibration parameters only allow obtaining aligned views between left-right and center-right cameras, hence not permitting to align the three views as we desire. Nonetheless, we describe in this section how to train our framework leveraging the proposed trinocular assumption with a much more common binocular setup (e.g., KITTI dataset). Given a stereo pair made of images L and R, Figure 3 depicts how to enforce the trinocular assumption by scheduling an *interleaved training* of the network. We update the parameters of the network by optimizing its four outputs d^{cl}, d^{lc}, d^{rc} and d^{cr} in two steps:

1. Firstly, we assign L to I^l and R to I^c as shown by the blue arrows in Figure 3. In other words, we assume that the stereo pair represents the left and center images of a *virtual* trinocular system in which the right frame is missing. In this case, we use as supervision signal the reconstruction error between $\tilde{I}^{cl}$, I^c and $\tilde{I}^l$, I^l, producing gradients that flow to the left decoder and the encoder.

2. Then, as shown by the red arrows in Figure 3 we change the role of L and R assuming them, respectively, as I^c and I^r. In this phase, we suppose to have the center and right images available hence implicitly assuming that in our virtual trinocular system the left image is missing. Thus, using the supervision given by re-projection errors on pairs $\tilde{I}^r$, I^r and $\tilde{I}^{cr}$, I^c, we optimize the parameters of the right decoder and the (shared) encoder.

It is worth to note that, following this protocol, every time we run a training iteration on a stereo pair the network learns all the depth representations output of our framework. Moreover, the two learned disparity pairs from the

two views are obtained according to the same baseline (i.e., the same of the training stereo pairs), making them consistent and hence easy to combine in d^c. Therefore, the network learns a trinocular representation even if it actually never sees the scene with such setup. Indeed, this strategy is very effective as supported by experimental evidence in Section 5.

4. Implementation details

In this section, we provide a detailed description of our framework, designed with the TensorFlow APIs. The source code is available at https://github.com/mattpoggi/3net.

4.1. Network architecture

For our 3Net we follow a quite established design strategy adopted by other methods in this field [24, 22, 42, 12], based on an encoder-decoder architecture. The peculiarity of our approach consists in two different decoders, as depicted in Figure 3, in charge of learning disparity representations w.r.t. two points of view located respectively on the left and right side of the input image. In our network, depicted in Figure 3, each decoder generates outputs at four different scales, respectively: full, half, quarter and $\frac{1}{8}$ resolution. As encoder, we tested VGG [34] and ResNet50 [13] to obtain the most fair and complete comparison w.r.t. [12], being it our baseline and state-of-the-art. To obtain the final map d_c, we merge the contribution of d_{cl} and d_{cr} using the same post-processing procedure applied in [12], thus keeping 5% left-most pixels from d_{cl}, 5% right-most from d_{cr} and averaging the remaining ones.

4.2. Training losses

We train 3Net to minimize a multi-component loss made of appearance, smoothness and consistency-check terms similarly to [12], namely $\mathcal{L}_{ap}, \mathcal{L}_{ds}$ and $\mathcal{L}_{lcr}$.

$$\mathcal{L}_{total} = \beta_{ap}(\mathcal{L}_{ap}) + \beta_{ds}(\mathcal{L}_{ds}) + \beta_{lcr}(\mathcal{L}_{lcr}) \quad (1)$$

The first term uses a weighted sum of SSIM [37] and L1 between all four warped pairs and real images as shown on top of Figure 3. The second applies an edge aware smoothness constraint to estimated disparities d^{cl}, d^{lc}, d^{rc} and d^{cr} as described in [12]. Finally, the consistency-check term includes left-right losses between pairs d^{cl}, d^{lc}.

$$\mathcal{L}_{lcr} = \mathcal{L}_{lr}(d^{cl}, d^{lc}) + \mathcal{L}_{lr}(d^{cr}, d^{rc}) \quad (2)$$

For a detailed description of $\mathcal{L}_{ap}, \mathcal{L}_{ds}$ and $\mathcal{L}_{lr}$ please refer to [12] or our supplementary material.

Thus, according to the interleaved training schedule described in Section 3.2, we optimize 3Net splitting the function 1 into two sub-losses $\mathcal{L}_{p_1}, \mathcal{L}_{p_2}$ deployed in the two different phases:

$$\begin{aligned}\mathcal{L}_{p_1} =& \beta_{ap}(\mathcal{L}_{ap}(\tilde{I}^{cl}, I^c) + \mathcal{L}_{ap}(\tilde{I}^{l}, I^l)) \\ &+ \beta_{ds}(\mathcal{L}_{ds}(d^{cl}, I^c) + \mathcal{L}_{ds}(d^{lc}, I^l)) \\ &+ \beta_{lcr}(\mathcal{L}_{lr}(d^{cl}, d^{lc})\end{aligned} \quad (3)$$

$$\begin{aligned}\mathcal{L}_{p_2} =& \beta_{ap}(\mathcal{L}_{ap}(\tilde{I}^{cr}, I^c) + \mathcal{L}_{ap}(\tilde{I}^{r}, I^r)) \\ &+ \beta_{ds}(\mathcal{L}_{ds}(d^{cr}, I^c) + \mathcal{L}_{ds}(d^{rc}, I^l)) \\ &+ \beta_{lcr}(\mathcal{L}_{lr}(d^{cr}, d^{lr})\end{aligned} \quad (4)$$

We also evaluated an additional loss term $\mathcal{L}_{cc} = |d^{cl} - d^{cr}|$ to enforce consistency between depth representation centered on I^c, being the baseline equal on both directions. However, this term propagates occlusions artifacts between the two depth maps leading to worse results. We point out that despite the interleaved training protocol outlined, in any phase the outcome of 3Net always consists of four depth maps d^{cl}, d^{lc}, d^{rc} and d^{cr}. Of course, this happens at testing/inference time as well, when 3Net is fed with a single image. Considering that decoders outputs depth maps at four scales, all losses are computed for each of them as in [12].

4.3. Training protocol

We assume as baseline the framework proposed by Godard et al. [12] using a binocular setup for unsupervised training. For a fair comparison, we train our models following the same guidelines reported in [12]. In particular, we use CityScapes [3] (CS) and KITTI raw sequences [11] datasets for training, this latter sub-sampled according to two training splits of data [12, 6] to be able to compare our results with any recent works in this field using unsupervised learning. We refer to these two subsets as KITTI split (K) and Eigen split (E) [6]. The three training sets count respectively about 23k, 29k and 22.6k stereo pairs. As pointed out by first works using image reconstruction losses [12, 42], training on different datasets helps the network to achieve higher-quality results. Therefore, to better assess the performance of each method, we report experimental results training the networks on K or E. Moreover, we also report results training on CityScapes and then fine tuning on K or E (respectively, referred to as CS+K and CS+E in the tables). Consistently with [12], we run 50 epochs of training on each single dataset using a batch size of 8 and input resized to 256×512. We use Adam optimizer [19] with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\varepsilon = 10^{-8}$, setting an initial learning rate of 10^{-4} halved after 30 and 40 epochs. We maintain the same hyperparameters configuration for β_{ap}, β_{ds} and β_{lrc} defined in [12] and the same data augmentation procedure as well.

		Proposed method			Lower is better		Higher is better			
Method	Train set	Abs Rel	Sq Rel	RMSE	RMSE log	D1-all	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$	Forwards
Godard et al. [12]	K	0.124	1.388	6.125	0.217	30.272	0.841	0.936	0.975	×1
3Net	K	0.119	1.201	5.888	0.208	31.851	0.844	0.941	0.978	×1
Godard et al. [12] + pp	K	0.117	1.177	5.804	0.206	29.945	0.848	0.943	0.977	×2
3Net + pp	K	0.114	1.088	5.756	0.203	31.141	0.848	0.944	0.979	×2
Godard et al. [12]	CS+K	0.104	1.070	5.417	0.188	25.523	0.875	0.956	0.983	×1
3Net	CS+K	0.101	0.954	5.211	0.181	24.632	0.875	0.958	0.985	×1
Godard et al. [12] + pp	CS+K	0.100	0.934	5.141	0.178	25.077	0.878	**0.961**	**0.986**	×2
3Net + pp	CS+K	**0.097**	**0.893**	**5.079**	**0.176**	**23.867**	**0.881**	**0.961**	**0.986**	×2

Table 1. Comparison between 3Net and [12], both using VGG as encoder, on KITTI 2015 training dataset [29].

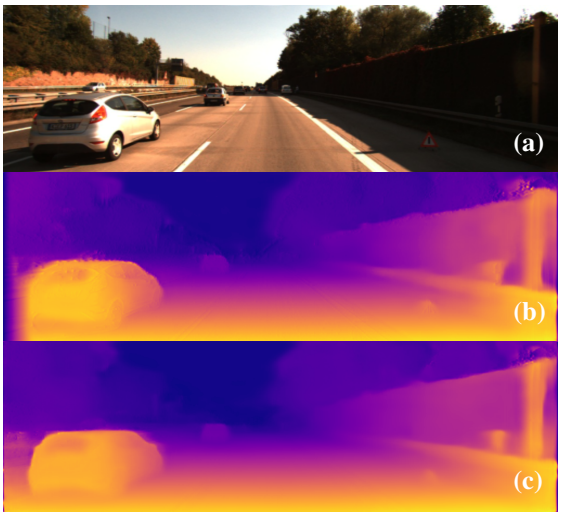

Figure 4. Depth maps predicted from input image (a) by Godard et al. [12] (b) and 3Net (c) running a single forward pass.

5. Experimental results

In this section, we assess the performance of our 3Net framework with respect to state-of-the-art. In all our tests, we report 7 main metrics measuring the average depth error (Abs Rel, Sq Rel, RMSE and RMSE log, the lower the better) and three accuracy scores ($\delta < 1.25, \delta < 1.25^2$ and $\delta < 1.25^3$, the higher the better), First, we report experiments on the K split assuming Godard et al. [12] as baseline. Then, we exhaustively compare 3Net with top performing unsupervised frameworks for depth-from-mono estimation, highlighting how our proposal is state-of-the-art. It is worth stressing that the proposed interleaved training procedure of 3Net, described in Section 3.2, allows for a fair comparison with any other method included in our evaluation being all trained exactly on the same (binocular) datasets. Finally, we report qualitative results concerning the trinocular representation learned by 3Net.

5.1. KITTI split

Table 1 reports experimental results on the KITTI 2015 stereo dataset [29]. The evaluation was carried out on 200 stereo pairs with available high quality ground-truth disparity annotations. Additionally, being the outputs of [12] and 3Net disparity maps, in our evaluation we include the D1-all score representing the percentage of pixels having a disparity error larger than 3.

We compare the raw output d_c of our network with the map predicted by Godard et al. with and without post-processing [12] (namely "+pp" in the table) running, respectively, a single or two forwards of the network. Moreover, since 3Net can benefit from the same refinement technique by running two predictions on I^c and its horizontally flipped version, we also provide results for our network applying the same post-processing. Therefore, we estimate post-processed d^{cl} and d^{cr} before combining them into d^c. Anyway, we report for clarity in the last column of the table, the number of forwards required by each entry.

As reported on the first two rows of Table 1, training the networks on KITTI data only, our method outperforms the competitor on all metrics except D1-all when running a single forward and it performs very similar to the post-processed version of [12] reported in the third row of the table. Rows 3 and 4 highlight that, performing two forwards and post-processing, 3Net + pp outperforms Godard et al. + pp again on all metrics except D1-all.

Previous works in literature [42, 12, 41, 36, 40, 27] proved that transfer learning from CityScape dataset [3] to KITTI is beneficial and leads to more accurate depth estimation, thus we follow this guideline training on CS+K as well. The last four rows of Table 1 compare both frameworks with and without post-processing. Without post-processing, we can notice how pre-training on CityScapes dataset allows 3Net to outperform [12] on all metrics including D1-all. In the last two rows, applying post-processing to the output of both models, 3Net outperforms the competitor on all metrics tying on $\delta < 1.25^2$ and $\delta < 1.25^3$. Figure 4 qualitatively shows depth maps predicted by [12] (b) and 3Net (c) without applying any post-processing to better perceive the improvements lead by our framework.

Summarizing, experiments on the KITTI split highlighted how enforcing the trinocular assumption is more effective than leveraging a conventional stereo paradigm for training. Moreover, these results prove that stereo pairs can

			Proposed method		Lower is better		Higher is better		
Method	Supervision	Train set	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Kumar et al. [5] (photo. + adv.)	Temporal	E	0.211	1.980	6.154	0.264	0.732	0.898	0.959
Zhou et al. [42]	Temporal	E	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou et al. [42] updated [40]	Temporal	E	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [27]	Temporal	E	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Yin et al. [40] GeoNet	Temporal	E	0.164	1.303	6.090	0.247	0.765	0.919	0.968
Yin et al. [40] GeoNet ResNet50	Temporal	E	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Wang et al. [36]	Temporal	E	0.151	1.257	5.583	0.228	0.810	0.936	**0.974**
Poggi et al. [32] PyD-Net (200)	Stereo	E	0.153	1.363	6.030	0.252	0.789	0.918	0.963
Godard et al. [12]	Stereo	E	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Zhan et al. [41]	Stereo+Temp.	E	0.144	1.391	5.869	0.241	0.803	0.928	0.969
3Net	Stereo	E	0.142	1.207	5.702	0.240	0.809	0.928	0.967
Godard et al. [12] ResNet50	Stereo	E	0.133	1.142	5.533	0.230	0.830	0.936	0.970
3Net ResNet50	Stereo	E	0.129	0.996	5.281	0.223	0.831	0.939	**0.974**
Godard et al. [12] ResNet50 + pp	Stereo	E	0.128	1.038	5.355	0.223	0.833	0.939	0.972
3Net ResNet50 + pp	Stereo	E	**0.126**	**0.961**	**5.205**	**0.220**	**0.835**	**0.941**	**0.974**
Zhou et al. [42]	Temporal	CS+E	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al. [27]	Temporal	CS+E	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yin et al. [40] GeoNet ResNet50	Temporal	CS+E	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Wang et al. [36]	Temporal	CS+E	0.148	1.187	5.496	0.226	0.812	0.938	0.975
Poggi et al. [32] PyD-Net (200)	Stereo	CS+E	0.146	1.291	5.907	0.245	0.801	0.926	0.967
Godard et al. [12]	Stereo	CS+E	0.124	1.076	5.311	0.219	0.847	0.942	0.973
3Net	Stereo	CS+E	0.117	0.905	4.982	0.210	0.856	0.948	0.976
Godard et al. [12] ResNet50	Stereo	CS+E	0.121	1.037	5.212	0.216	0.854	0.944	0.973
3Net ResNet50	Stereo	CS+E	0.113	0.885	4.898	0.204	0.862	0.950	0.977
Godard et al. [12] ResNet50 + pp	Stereo	CS+E	0.114	0.898	4.935	0.206	0.861	0.949	0.976
3Net ResNet50 + pp	Stereo	CS+E	**0.111**	**0.849**	**4.822**	**0.202**	**0.865**	**0.952**	**0.978**

Table 2. Evaluation on the KITTI dataset [11] using the split of Eigen et al. [6], with maximum depth set to 80m. Results concerned with state-of-the-art techniques for unsupervised monocular depth estimation leveraging video sequences (Temporal), binocular stereo pairs (Stereo) and both cues (Stereo+Temp.).

be used in a smarter way following our interleaving strategy.

5.2. Eigen split

Table 2 reports evaluation with the split of data of Eigen et al. [6], made of 697 images and relative depth measurements acquired with a Velodyne sensor. The table collects results concerning most recent works addressing unsupervised monocular depth estimation. For each method, we indicate the kind of supervision it leverages on: monocular sequences (*Temporal*), stereo pairs (*Stereo*) or stereo sequences (*Stereo+Temp.*). We report results either training on E only or on CS+E, allowing to compare our scores with state-of-the-art approaches. We point out that all methods, including our proposal, are trained exactly on the same images and all of them *see* the same scenes[1]. On top, we report results for models trained on the Eigen split of data. For GeoNet [40], Godard et al. [12] and our method we report results for both VGG and ResNet50 encoders. We can notice that, in general, methods trained using stereo data usually outperform those trained on monocular video sequences, as evident from recent literature [42, 12, 41, 36, 40, 27]. Zhan et al. [41] leveraging temporally adjacent stereo frames outperform, on most metrics, [12]. Nevertheless, 3Net achieves better scores except for $\delta < 1.25^3$ still without exploiting temporal supervision. This proves that a smarter deployment of binocular training samples, i.e. by applying our interleaved training to fulfill trinocular hypothesis, is an effective alternative to sequence supervision. It is worth to note that Wang et al. [36] obtain better scores on most metrics (RMSE, RMSE log and δ metrics) w.r.t. [12] and 3Net with the VGG encoder. However, by switching to the ResNet50 encoder, Godard et al. [12] overtakes most recent works that use *Time* supervision [36, 40] with and without post-processing. Systematically, 3Net always outmatches [12] and consequently all its competitors. In particular, we point out that 3Net ResNet50 without post-processing already achieves some better scores compared to Godard et al. ResNet50 + pp performing, respectively, a single and a double forward.

On the bottom of Table 2, we resume results achieved by models trained on CS+E. We observe the same trend highlighted in the previous experiments, being [12] and our proposal the most effective solutions for this task thanks to stereo supervision. In equal conditions, i.e. same encoder and number of forwards, 3Net always outperforms the framework of Godard et al. exploiting the trinocular assumption. Moreover, the proposed technique leads to major improvements such that 3Net VGG outperforms ResNet50 model by Godard et al. [12] (rows 20^{th} and 21^{st}), 3Net ResNet50 without post-processing achieves more accurate

[1]Zhan et al. [41] report scores training on E only or after pre-training on NYU dataset [30]. For fairness we report the first setup only.

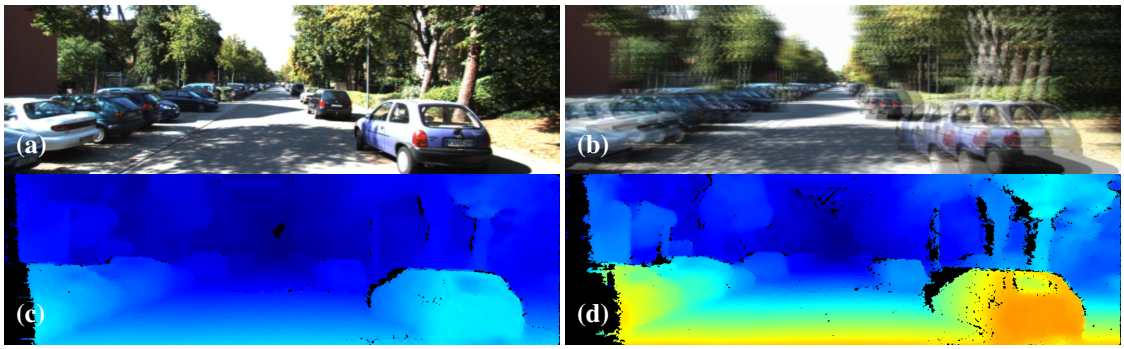

Figure 5. Qualitative example of learned trinocular setup. Given a single, input image (a), 3Net can generate two additional points of view, shown superimposed to the real frame in (b). Running traditional stereo algorithms [14], assuming the left-most generated frame as reference, allows to obtain disparity maps with narrow (c) and wide (d) baseline (encoded with colormap jet). We point out that the center frame (a) is the only real image. More qualitative examples in the supplementary material.

results than the best configuration ResNet50 + pp [12] (rows 22^{nd} and 23^{rd}) and finally 3Net ResNet50 + pp outmatches all known frameworks for unsupervised depth-from-mono estimation. These facts clearly highlight that our proposal is state-of-the-art.

It is important to underline that the availability of a *real* trinocular rig would most probably allow training a more accurate model, given the larger amount of images w.r.t. a binocular stereo rig. The interleaved training proposed in this paper allows to overcome the lack of trinocular training samples using binocular pairs and also allows for a more fair comparison with other techniques leveraging this latter configuration only. This fact proves that the effectiveness of our strategy is due to the rationale behind it and not driven by a more extensive availability of data.

6. View synthesis

Finally, we show through qualitative images some outcomes of 3Net obtainable exploiting the embodied trinocular assumption.[2]. A peculiar experiment allowed by our framework consists of generating three horizontally aligned views from a single input image. This feature is possible thanks to estimated d^{lc} and d^{rc}, used to warp the input image towards two new viewpoints, respectively, on the left and the right. In other words, given I^c at test time we compute $\tilde{I}^l$ and $\tilde{I}^r$, producing a trinocular setup of horizontally aligned images. Figure 5 shows an example of a single frame (a) taken from the KITTI dataset and how our network generates the three views superimposed in (b). These views effectively enable to realize a multi-baseline setup. Thus we can run any stereo algorithm between the possible pairs. For instance, we run the Semi-Global Matching algorithm (SGM) [14] between $\tilde{I}^l$ and I^c, showing the results in Figure 5 (c), then we run SGM between $\tilde{I}^l$ and $\tilde{I}^r$ obtaining the disparity map shown in (d). The two disparity maps assume the same frame as the reference image ($\tilde{I}^l$) and two different targets, according to two different *narrow* and *wide* virtual baseline. The shorter baseline is learned from the KITTI acquisition rig while the longer one is inherited by our trinocular assumption although actually, it does not exist at all in the training set. This fact can be perceived by looking at the different disparity ranges encoded, with colormap jet, on (c) and (d). This feature of our network paves the way to exciting novel developments. For instance, a conceivable application would consist in the synthesis of *augmented* stereo pairs to train CNNs for disparity inference [28, 12] or to improve recent techniques such as single view stereo [25].

[2]A video is available at http://youtu.be/uMA5YWJME4M

7. Conclusions

In this paper, we proposed a novel methodology for unsupervised training of a depth-from-mono CNN. By enforcing a trinocular assumption, we overcome some limitations due to binocular stereo images used as supervision and obtain a more accurate depth estimation with our 3Net architecture. Although three horizontally aligned views are seldom available, we proposed an interleaved training protocol allowing to leverage on traditional binocular datasets. This latter technique also ensures for a fair comparison w.r.t. all previous works and allows us to prove that 3Net outperforms all unsupervised techniques known in the literature, establishing itself as state-of-the-art. Moreover, 3Net learns a trinocular representation of the world, making it suitable for image synthesis purposes and other interesting future developments.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

References

[1] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[2] N. Ayache and F. Lustman. Trinocular stereo vision for robotics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(1):73–85, Jan. 1991. 3

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 5, 6

[4] A. CS Kumar, S. M. Bhandarkar, and P. Mukta. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *1st International Workshop on Deep Learning for Visual SLAM, (CVPR)*, 2018. 2

[5] A. CS Kumar, S. M. Bhandarkar, and P. Mukta. Monocular depth prediction using generative adversarial networks. In *1st International Workshop on Deep Learning for Visual SLAM, (CVPR)*, 2018. 3, 7

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2, 5, 7

[7] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016. 2

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[9] R. Garcia, J. Batlle, and J. Salvi. A new approach to pose detection using a trinocular stereovision system. *Real-Time Imaging*, 8(2):73–93, Apr. 2002. 3

[10] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 3

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 5, 7

[12] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 2, 3, 4, 5, 6, 7, 8

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[14] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. 8

[15] D. Honegger, T. Sattler, and M. Pollefeys. Embedded real-time multi-baseline stereo. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017. 3

[16] M. Ito and A. Ishii. Three-view stereo analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):524–532, 1986. 3

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[18] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 3

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[21] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014. 2

[22] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 2, 5

[23] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 2

[24] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2016. 2, 5

[25] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin. Single view stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[26] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 3, 4

[27] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7

[28] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 8

[29] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 6

[30] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7

[31] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on pattern analysis and machine intelligence*, 15(4):353–363, 1993. 3

[32] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *IEEE/JRS Conference on Intelligent Robots and Systems (IROS)*, 2018. 3, 7

[33] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009. 2

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[35] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2017. 2

[36] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, Apr. 2004. 5

[38] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016. 2, 3

[39] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[40] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7

[41] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 6, 7

[42] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 3, 4, 5, 6, 7