Learning monocular depth estimation infusing traditional stereo knowledge

Fabio Tosi, Filippo Aleotti, Matteo Poggi, Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{fabio.tosi5, filippo.aleotti2, m.poggi, stefano.mattoccia }@unibo.it

Abstract

Depth estimation from a single image represents a fascinating, yet challenging problem with countless applications. Recent works proved that this task could be learned without direct supervision from ground truth labels leveraging image synthesis on sequences or stereo pairs. Focusing on this second case, in this paper we leverage stereo matching in order to improve monocular depth estimation. To this aim we propose monoResMatch, a novel deep architecture designed to infer depth from a single input image by synthesizing features from a different point of view, horizontally aligned with the input image, performing stereo matching between the two cues. In contrast to previous works sharing this rationale, our network is the first trained end-to-end from scratch. Moreover, we show how obtaining proxy ground truth annotation through traditional stereo algorithms, such as Semi-Global Matching, enables more accurate monocular depth estimation still countering the need for expensive depth labels by keeping a self-supervised approach. Exhaustive experimental results prove how the synergy between i) the proposed monoResMatch architecture and ii) proxy-supervision attains state-of-the-art for self-supervised monocular depth estimation. The code is publicly available at https://github.com/fabiotosi92/monoResMatch-Tensorflow.

1. Introduction

Inferring accurate depth information of a sensed scene is paramount for several applications such as autonomous driving, augmented reality and robotics. Although technologies such as LiDAR and time-of-flight are quite popular, obtaining depth from images is often the preferred choice. Compared to other sensors, those based on standard cameras potentially have several advantages: they are inexpensive, have a higher resolution and are suited for almost any environment. In this field, stereo is the preferred choice to infer *disparity* (*i.e.*, the inverse of depth) from two or more images sensing the same area from different points of view and *Semi-Global Matching* (SGM) [15] is a popular, yet effective algorithm to accomplish this task. However, inferring depth from a single image is particularly attractive because it does not require a stereo rig and overcomes some intrinsic limitations of a binocular setup (*e.g.*, occlusions). On the other hand, it is an extremely challenging task due to the ill-posed nature of the problem. Nonetheless, deep learning enabled to achieve outstanding results for this task [7], although the gap with state-of-the-art stereo solutions is still huge [3, 24]. Self-supervised learning paradigms for monocular depth estimation [11, 63, 32, 44, 40, 58] became very popular to overcome the need for costly ground truth annotations, usually obtained employing expensive active sensors and human post-processing [10, 35, 52]. Following this strategy, Convolutional Neural Networks (CNNs) can be trained to tackle depth estimation as an image synthesis task from stereo pairs or monocular sequences [11, 63]. For this purpose, using stereo pairs rather than monocular sequences as supervision turned out to be more effective according to the literature. Although the former strategy is more constrained since a stereo setup is necessary for training, it does neither require to infer relative pose between adjacent frames in a sequence nor to segment moving objects in the scene. Moreover, a stereo setup does not require

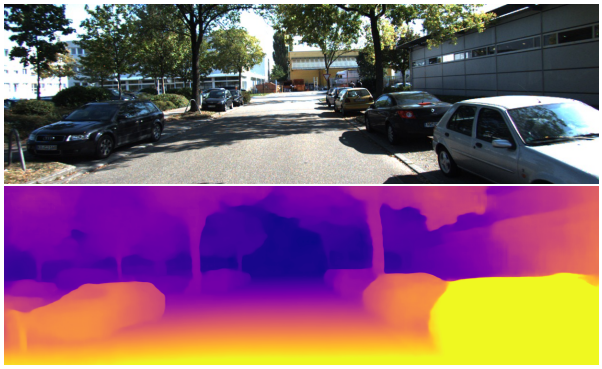


Figure 1. Overview of the proposed depth-from-mono solution. Input image from KITTI dataset (top). Estimated depth map by our monoResMatch (bottom).

camera motion, conversely to a monocular setup, to provide meaningful supervision. Other means for self-supervision consist into distilling *proxy* labels in place of more expensive annotations for various tasks [49, 51, 28, 33, 20, 13].

In this paper, we propose monocular Residual Matching (shorten, monoResMatch), a novel end-to-end architecture trained to estimate depth from a monocular image leveraging a virtual stereo setup. In the first stage, we map input image into a features space, then we use such representation to estimate a first depth outcome and consequently synthesize features aligned with a *virtual* right image. Finally, the last refinement module performs stereo matching between the real and synthesized representations. Differently from other frameworks following a similar rationale [30] that combines heterogeneous networks for synthesis [55] and stereo [34], we use a single architecture trained in end-to-end fashion yielding a notable accuracy improvement compared to the existing solutions. Moreover, we leverage traditional knowledge from stereo to obtain accurate proxy labels in order to improve monocular depth estimation supervised by stereo pairs. We will show that, despite the presence of outliers in the produced labels, training according to this paradigm results in superior accuracy compared to image warping approaches for self-supervision. Experimental results on the KITTI raw dataset [9] will show that the synergy between the two aforementioned key components of our pipeline enables to achieve state-of-the-art results compared to other self-supervised frameworks for monocular depth estimation not requiring any ground truth annotation. Figure 1 shows an overview of our framework, depicting an input frame and the outcome of monoResMatch.

2. Related Work

In this section, we review the literature relevant to our work concerned with stereo/monocular depth estimation and proxy label distillation.

Stereo depth estimation. Most conventional dense stereo algorithms rely on some or all the well-known four steps thoroughly described in [46]. In this field, SGM [15] stood out for the excellent trade-off between accuracy and efficiency thus becoming very popular. Žbontar and LeCun [61] were the first to apply deep learning to stereo vision replacing the conventional matching costs calculation with a siamese CNN network trained to predict the similarity between patches. Luo *et al*. [29] cast the correspondence problem as a multi-class classification task, obtaining better results. Mayer *et al*. [34] backed away from the previous approaches and proposed an end-to-end trainable network called *DispNetC* able to infer disparity directly from images. While DispNetC applies a 1-D correlation to mimic the cost volume, GCNet by Kendall *et al*. [17] exploited 3-D convolutions over a 4-D volume to obtain matching costs and finally applied a differentiable version of *argmin* to select the best disparity along this volume. Other works followed these two main strategies, building more complex architectures starting from DispNetC [37, 25, 57, 47] or GCNet [3, 26, 18] respectively. The domain shift issue affecting these architectures (*e.g*. synthetic to real) has been addressed in either offline [49] or online [50] fashion, or greatly reduced by guiding them with external depth measurements (*e.g*. Lidar) [42].

Monocular depth estimation. Before the deep learning era, some works tackled depth-from-mono with MRF [45] or boosted classifiers [22]. However, with the increasing availability of ground truth depth data, supervised approaches based on CNNs [23, 27, 56, 7] rapidly outperformed previous techniques. An attractive trend concerns the possibility of learning depth-from-mono in a self-supervised manner, avoiding the need for expensive ground truth depth labels that are replaced by multiple views of the sensed scene. Then, supervision signals can be obtained by image synthesis according to the estimated depth, camera pose or both. In general, acquiring images from a stereo camera enables a more effective training than using a single, moving camera, since the pose between frames known. Concerning stereo supervision, Garg *et al*. [8] first followed this approach, while Godard *et al*. [11] introduced spatial transform network [16] and a left-right consistency loss. Other methods improved efficiency [40], deploying a pyramidal architecture, and accuracy by simulating a trinocular setup [44] or including joint semantic segmentation [60]. In [38], a strategy was proposed to reduce further the energy efficiency of [40] leveraging fixed-point quantization. The semi-supervised framework by Kuznietsov *et al*. [21] combined stereo supervision with sparse LiDAR measurements. The work by Zhou *et al*. [63] represents the first attempt to supervise a depth-from-mono framework with single camera sequences. This approach was improved including additional cues such as point-cloud alignment [32], differentiable DVO [53] and multi-task learning [64]. Zhan *et al*. [62] combined the two supervision approaches outlined so far deploying stereo sequences. Another class of methods [2, 1, 5] applied a generative adversarial paradigm to the monocular scenario.

Finally, relevant to our work is Single View Stereo matching (SVS) [30], processing a single image to obtain a second synthetic view using Deep3D [55] and then computing a disparity map between the two using DispNetC [34]. However, these two architectures are trained independently. Moreover, DispNetC is supervised with ground truth labels from synthetic [34] and real domains [35]. Differently, the framework we are going to introduce requires no ground truth at all and is elegantly trained in an end-to-end manner, outperforming SVS by a notable margin.

Proxy labels distillation. Since for most tasks ground truth labels are difficult and expensive to source, some

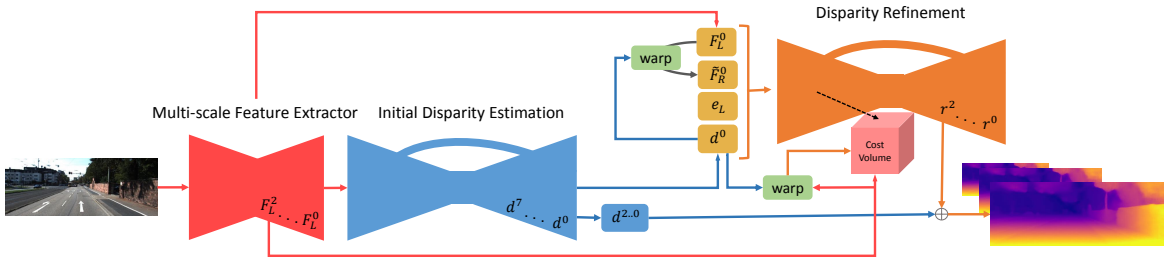


Figure 2. Illustration of our *monoResMatch* architecture. Given one input image, the multi-scale feature extractor (in red) generates high-level representations in the first stage. The initial disparity estimator (in blue) yields multi-scale disparity maps aligned with the left and right frames of a stereo pair. The disparity refinement module (in orange) is in charge of refining the initial left disparity relying on features computed in the first stage, disparities generated in the second stage, matching costs between high-dimensional features F_L^0 extracted from input and synthetic $\tilde{F}_R^0$ from a *virtual* right viewpoint, together with absolute error e_L between F_L^0 and back-warped $\tilde{F}_R^0$ (see Section 3.3).

works recently enquired about the possibility to replace them with easier to obtain proxy labels. Tonioni *et al*. [49] proposed to adapt deep stereo networks to unseen environments leveraging traditional stereo algorithms and confidence measures [43], Tosi *et al*. [51] learned confidence estimation selecting positive and negative matches by means of traditional confidence measures, Makansi *et al*. [33] and Liu *et al*. [28] generated proxy labels for training optical flow networks using conventional methods. Specifically relevant to monocular depth estimation are the works proposed by Yang *et al*. [58], using stereo visual odometry to train monocular depth estimation, by Klodt and Vedaldi [20], leveraging structure from motion algorithms and by Guo *et al*. [13], obtaining labels from a deep network trained with supervision to infer disparity maps from stereo pairs.

3. Monocular Residual Matching

In this section, we describe in detail the proposed *monocular Residual Matching* (monoResMatch) architecture designed to infer accurate and dense depth estimation in a self-supervised manner from a single image. Figure 2 recaps the three key components of our network. First, a multi-scale feature extractor takes as input a single raw image and computes deep learnable representations at different scales from quarter resolution F_L^2 to full-resolution F_L^0 in order to toughen the network to ambiguities in photometric appearance. Second, deep high-dimensional features at input image resolution are processed to estimate, through an hourglass structure with skip-connections, multi-scale inverse depth (*i.e*., disparity) maps aligned with the input and a *virtual* right view learned during training. By doing so, our network learns to emulate a binocular setup, thus allowing further processing in the stereo domain [30]. Third, a disparity refinement stage estimates residual corrections to the initial disparity. In particular, we use deep features from the first stage and back-warped features of the *virtual* right image to construct a cost volume that stores the stereo matching costs using a correlation layer [34].

Our entire architecture is trained from scratch in an end-to-end manner, while SVS [30] by training its two main components, Deep3D [55] and DispNetC [34], on image synthesis and disparity estimation tasks separately (with the latter requiring additional, supervised depth labels from synthetic imagery [34]).

Extensive experimental results will prove that monoResMatch enables much more accurate estimations compared to SVS and other state-of-the-art approaches.

3.1. Multi-scale feature extractor

Inspired by [25], given one input image I we generate deep representations using layers of convolutional filters. In particular, the first 2-stride layer convolves I with 64 learnable filters of size 7×7 followed by a second 2-stride convolutional layer composed of 128 filters with kernel size 4×4. Two deconvolutional blocks, with stride 2 and 4, are deployed to upsample features from lower-spatial resolution to full input resolution producing 32 features maps each. A 1×1 convolutional layer with stride 1 further processes upsampled representations.

3.2. Initial Disparity Estimation

Given the features extracted by the first module, this component is in charge of estimating an initial disparity map. In particular, an encoder-decoder architecture inspired by DispNet processes deep features at quarter resolution from the multi-scale feature extractor (*i.e*., *conv2*) and outputs disparity maps at different scales, specifically from $\frac{1}{128}$ to full-resolution. Each down-sampling module, composed of two convolutional blocks with stride 2 and 1 each, produces a growing number of extracted features, respectively 64, 128, 256, 512, 1024, and each convolutional layer uses 3×3 kernels followed by ReLU non-linearities. Differently from DispNet, which computes matching costs in the early part of this stage using features from the left and right images of a stereo pair, our architecture lacks such neces-

sary information required to compute a cost volume since it processes a single input image. Thus, no 1-D correlation layer can be imposed to encode geometrical constraints in this stage of our network. Then, upsampling modules are deployed to enrich feature representations through skip-connections and to extract two disparity maps, aligned respectively with the input frame and a *virtual* viewpoint on its right as in [11]. This process is carried out at each scale using 1-stride convolutional layers with kernel size 3×3.

3.3. Disparity Refinement

Given an initial estimate of the disparity at each scale obtained in the second part of the network, often characterized by errors at depth discontinuities and occluded regions, this stage predicts corresponding multi-scale residual signals [14] by a few stacked nonlinear layers that are then used to compute the final left-view aligned disparity map. This strategy allows us to simplify the end-to-end learning process of the entire network. Moreover, motivated by [30], we believe that geometrical constraints can play a central role in boosting the final depth accuracy. For this reason, we embed matching costs in feature space computed employing a horizontal correlation layer, typically deployed in deep stereo algorithms. To this end, we rely on the right-view disparity map computed previously to generate right-view features $\tilde{F}_R^0$ from the left ones F_L^0 using a differentiable bilinear sampler [16]. The network is also fed with error e_L, *i.e.* the absolute difference between left and *virtual* right features at input resolution, with the latter back-warped at the same coordinates of the former, as in [24].

We point out once more that, differently from [30], our architecture produces both a synthetic right view, *i.e.* its features representation, and computes the final disparity map following stereo rationale. This makes monoResMatch a single end-to-end architecture, effectively performing stereo out of a single input view rather than the combination of two models (*i.e.*, Deep3D [55] and DispNetC [34] for the two tasks outlined) trained independently as in [31]. Moreover, exhaustive experiments will highlight the superior accuracy achieved by our fully self-supervised, end-to-end approach.

3.4. Training Loss

In order to train our multi-stage architecture, we define the total loss as a sum of two main contributions, a $\mathcal{L}_{init}$ term from the initial disparity estimation module and a $\mathcal{L}_{ref}$ term from the disparity refinement stage. Following [12], we embrace the idea to up-sample the predicted low-resolution disparity maps to the full input resolution and then compute the corresponding signals. This simple strategy is designed to force the inverse depth estimation to reproduce the same objective at each scale, thus leading to much better outcomes. In particular, we obtain the final training loss as:

$$\mathcal{L}_{total} = \sum_{s=1}^{n_i} \mathcal{L}_{init} + \sum_{s=1}^{n_r} \mathcal{L}_{ref} \tag{1}$$

where s indicates the output resolution, n_i and n_r the numbers of considered scales during loss computation, while $\mathcal{L}_{init}$ and $\mathcal{L}_{ref}$ are formalised as:

$$\begin{aligned}\mathcal{L}_{init} =& \alpha_{ap}(\mathcal{L}_{ap}^l + \mathcal{L}_{ap}^r) + \alpha_{ds}(\mathcal{L}_{ds}^l + \mathcal{L}_{ds}^r) \\ &+ \alpha_{ps}(\mathcal{L}_{ps}^l + \mathcal{L}_{ps}^r)\end{aligned} \tag{2}$$

$$\mathcal{L}_{ref} = \alpha_{ap}\mathcal{L}_{ap}^l + \alpha_{ds}\mathcal{L}_{ds}^l + \alpha_{ps}\mathcal{L}_{ps}^l \tag{3}$$

where $\mathcal{L}_{ap}$ is an image reconstruction loss, $\mathcal{L}_{ds}$ is a smoothness term and $\mathcal{L}_{ps}$ is a proxy-supervised loss. Each term contains both the left and right components for the initial disparity estimator, and the left components only for the refinement stage.

Image reconstruction loss. A linear combination of l_1 loss and *structural similarity measure* (SSIM) [54] encodes the quality of the reconstructed image $\tilde{I}$ with respect to the original image I:

$$\mathcal{L}_{ap} = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - SSIM(I_{ij}, \hat{I}_{ij})}{2} + (1-\alpha)|I_{ij} - \hat{I}_{ij}| \tag{4}$$

Following [11], we set $\alpha = 0.85$ and use a SSIM with 3×3 block filter.

Disparity smoothness loss. This cost encourages the predicted disparity to be locally smooth. Disparity gradients are weighted by an edge-aware term from image domain:

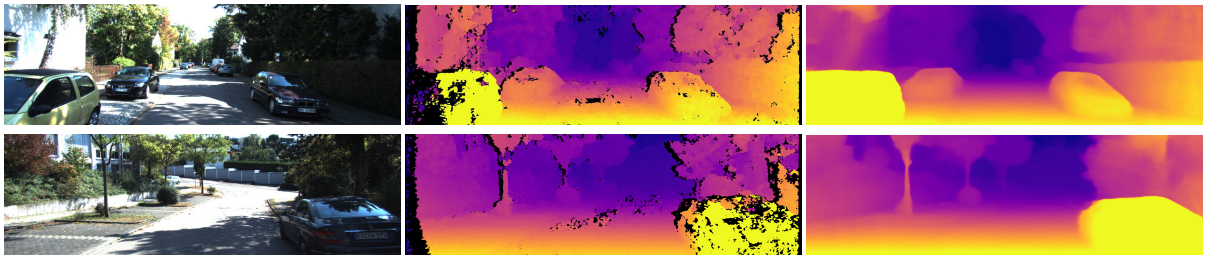
$$\mathcal{L}_{ds} = \frac{1}{N}\sum_{i,j} |\partial_x d_{ij}| e^{-|\partial_x I_{ij}|} + |\partial_y d_{ij}| e^{-|\partial_y I_{ij}|} \tag{5}$$

Proxy-supervised loss. Given the proxy disparity maps obtained by a conventional stereo algorithm, detailed in Section 4, we coach the network using reverse Huber (berHu) loss [36]:

$$\mathcal{L}_{ps} = \frac{1}{N}\sum_{i,j} berHu(d_{ij}, d_{ij}^{st}, c) \tag{6}$$

$$berHu(d_{ij}, d_{ij}^{st}, c) = \begin{cases} |d_{ij} - d_{ij}^{st}| & \text{if } |d_{ij} - d_{ij}^{st}| \leq c \\ \frac{|d_{ij} - d_{ij}^{st}|^2 - c^2}{2c} & \text{otherwise} \end{cases} \tag{7}$$

where d_{ij} and d_{ij}^{st} are, respectively, the predicted disparity and the proxy annotation for pixel at the coordinates i, j of the image, while c is adaptively set as $\alpha \max_{i,j} |d_{ij} - d_{ij}^{st}|$, with $\alpha = 0.2$.



(a) (b) (c)

Figure 3. Examples of proxy labels computed by SGM. Given the source image (a), the network exploits the SGM supervision filtered with left-right consistency check (b) in order to train monoResMatch to estimate the final disparity map (c). No post-processing from [11] is performed on (c) in this example.

4. Proxy labels distillation

To generate accurate proxy labels, we use the popular SGM algorithm [15], a fast yet effective solution to infer depth from a rectified stereo pair without training. In our implementation, initial matching costs are computed for each pixel p and disparity hypothesis d applying a 9×7 census transform and computing Hamming distance on pixel strings. Then, scanline optimization along eight different paths refines the initial cost volume as follows:

$$E(p,d) = C(p,d) + \min_{j>1}[C(p',d), C(p',d\pm1) + P1, \\ C(p',d\pm q) + P2] - \min_{k<D_{max}}(C(p',k)) \tag{8}$$

being $C(p,d)$ the matching cost for pixel p and hypothesis d, P_1 and P_2 two smoothness penalties, discouraging disparity gaps between p and previous pixel p' along the scanline path. The final disparity map D is obtained applying a winner-takes-all strategy to each pixel of the reference image. Although SGM generates quite accurate disparity labels, outliers may affect the training of a depth model negatively, as noticed by Tonioni *et al*. [49]. They applied a learned confidence measure [41] to filter out erroneous labels when computing the loss. Differently, we run a non-learning based left-right consistency check to detect outliers. Purposely, by extracting both disparity maps D^L and D^R with SGM, respectively for the left and right images, we apply the following criteria to invalidate (*i.e*., set to -1) pixels having different disparities across the two maps:

$$D(p) = \begin{cases} D(p) & \text{if } |D^L(p) - D^R(p - D^L(p))| \leq \varepsilon \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

The left-right consistency check is a simple strategy that removes many wrong disparity assignments, mostly near depth discontinuities, without needing any training that would be required by [49]. Therefore, our proxy labels generation process does not rely at all on ground truth depth labels. Figure 3 shows an example of distilled labels (b), where black pixels correspond to outliers filtered out by left-right consistency. Although some of them persist, we can notice how they do not affect the final prediction by the trained network and how our proposal can recover accurate disparity values in occluded regions on the left side of the image (c).

5. Experimental results

In this section, we describe the datasets, implementation details and then present exhaustive evaluations of monoResMatch on various training/testing configurations, showing that our proposal consistently outperforms self-supervised state-of-the-art approaches. As standard in this field, we assess the performance of monocular depth estimation techniques following the protocol by Eigen *et al*. [6], extracting data from the KITTI [9] dataset, using sparse LiDAR measurements as ground truth for evaluation. Additionally, we also perform an exhaustive ablation study proving that proxy supervision from SGM algorithm and effective architectural choices enable our strategy to improve predicted depth map accuracy by a large margin.

5.1. Datasets

For all our experiments we compute standard monocular metrics [6, 11]: *Abs rel*, *Sq rel*, *RMSE* and *RMSE log* represent error measures while $\delta < \zeta$ the percentage of predictions whose maximum between ratio and inverse ratio with respect to the ground truth is lower than a threshold ζ. Two main datasets are involved in our evaluation, that are KITTI [9] and CityScapes [4].

KITTI. The KITTI stereo dataset [9] is a collection of rectified stereo pairs made up of 61 scenes (containing about 42,382 stereo frames) mainly concerned with driving scenarios. Predominant image size is 1242×375 pixels. A LiDAR device, mounted and calibrated in proximity to the left camera, was deployed to measure depth information.

Following other works [6, 11], we divided the overall dataset into two subsets, composed respectively of 29 and

Method	Supervision		Train set	Lower is better				Higher is better		
	Image	SGM		Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard *et al.* [11] ResNet50	✓		K	0.128	1.038	5.355	0.223	0.833	0.939	0.972
Poggi *et al.* [44] ResNet50	✓		K	0.126	0.961	5.205	0.220	0.835	0.941	0.974
monoResMatch	✓		K	0.116	0.986	5.098	0.214	0.847	0.939	0.972
monoResMatch	✓	✓	K	**0.111**	**0.867**	**4.714**	**0.199**	**0.864**	**0.954**	**0.979**
Godard *et al.* [11] ResNet50	✓		CS,K	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Poggi *et al.* [44] ResNet50	✓		CS,K	0.111	0.849	4.822	0.202	0.865	0.952	0.978
Godard *et al.* [11] ResNet50	✓	✓	CS,K	0.110	0.822	4.675	0.199	0.862	0.953	0.980
monoResMatch (no-refinement)	✓	✓	CS,K	0.107	0.781	4.588	0.195	0.869	0.957	0.980
monoResMatch (no-corr)	✓	✓	CS,K	0.104	0.766	4.553	0.192	0.875	0.958	0.980
monoResMatch (no-pp)	✓	✓	CS,K	0.098	0.711	4.433	0.189	0.888	0.960	0.980
monoResMatch	✓	✓	CS,K	**0.096**	**0.673**	**4.351**	**0.184**	**0.890**	**0.961**	**0.981**

Table 1. Ablation studies on the Eigen split [6], with maximum depth set to 80m. All networks run post-processing as in [11] unless otherwise specified.

32 scenes. We used 697 frames belonging to the first group for testing purposes and 22600 more taken from the second for training. We refer to these subsets as *Eigen split*.

CityScapes. The CityScapes dataset [4] contains stereo pairs concerning about 50 cities in Germany taken from a moving vehicle in various weather conditions. It consists of 22,973 stereo pairs with a shape of 2048×1024 pixels. Since most of the images include the hood of the car, mostly reflective and thus leading to wrong estimates, we discarded the lower 20% of the frame before applying the random crop during training [11].

5.2. Implementation details

Following the standard protocol in this field, we used CityScapes followed by KITTI for training. We refer to these two training sets as Cityscapes (CS) and Eigen KITTI split (K) from now on. We implemented our architecture using the TensorFlow framework, counting approximately 42.5 millions of parameters, summing variables from the multi-scale feature extractor (0.51 M), the initial disparity stage (41.4 M) and the refinement module (0.6 M). In the experiments, we pre-trained monoResMatch on CS running about 150k iteration using a batch size of 6 and random crops of size 512×256 on 1024×512 resized images from the original resolution. We used Adam optimizer [19] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We set the initial learning rate to 10^{-4}, manually halved after 100k and 120k steps, then continuing until convergence. After the first pre-initialisation procedure, we perform fine-tuning of the overall architecture on 22600 KITTI raw images from K. Specifically, we run 300k steps using a batch size of 6 and extracting random crops of size 640×192 from resized images at 1280×384 resolution. At this stage, we employed a learning rate of 10^{-4}, halved after 180k and 240k iterations. We fixed the hyper-parameters of the different loss components to $\alpha_{ap} = 1, \alpha_{ds} = 0.1$ and $\alpha_{ps} = 1$, while $n_i = 4$ and $n_r = 3$. As in [11], data augmentation procedure has been applied to both images from CS and K at training, in order to increase the robustness of the network. At test time, we post-process disparity as in [11, 44, 58]. Nevertheless, we preliminary highlight that, differently from the strategies mentioned above, effects such as disparity ramps on the left border are effectively solved by simply picking random crops on proxy disparity maps generated by SGM, as clearly visible in Figure 3 (c).

Proxy supervision is obtained through SGM implementation from [48], which allows us to quickly generate disparity maps aligned with the left and right images for both CS and K. We process such outputs using left-right consistency check in order to reduce the numbers of outliers, as discussed in Section 4 using an ϵ of 1. We assess the accuracy of our proxy generator on 200 high-quality disparity maps from KITTI 2015 training dataset [35], measuring 96.1% of pixels having disparity error smaller than 3. Compared to Tonioni *et al.* [49], we register a negligible drop in accuracy from 99.6% reported in their paper. However, we do not rely on any learning-based confidence estimator as they do [41], so we maintain label distillation detached from the need for ground truth as well. Since SGM runs over images at full resolution while monoResMatch inputs are resized to 1280×384 before extracting crops, we enforce a scaling factor to SGM disparities given by $\frac{1280}{W}$, where W is the original image *width*. Consequently, the depth map estimated by monoResMatch must be properly multiplied by $\frac{W}{1280}$ at test time. The architecture is trained end-to-end on a single Titan XP GPU without any stage-wise procedure and infers depth maps in 0.16s per frame at test time, processing images at KITTI resolution (*i.e.*, about 1280×384 to be compatible with monoResMatch downsampling factors).

5.3. Ablation study

In this section we examine the impact of i) proxy-supervision from SGM and ii) the different components of monoResMatch. The outcomes of these experiments, con-

					Lower is better		Higher is better		
Method	Supervision	Train set	Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Zou *et al*. [64]	Seq	CS,K	0.146	1.182	5.215	0.213	0.818	0.943	0.978
Mahjourian *et al*. [32]	Seq	CS,K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yin *et al*. [59] GeoNet ResNet50	Seq	CS,K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Wang *et al*. [53]	Seq	CS,K	0.148	1.187	5.496	0.226	0.812	0.938	0.975
Poggi *et al*. [40] PyD-Net (200)	Stereo	CS,K	0.146	1.291	5.907	0.245	0.801	0.926	0.967
Godard *et al*. [11] ResNet50	Stereo	CS,K	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Poggi *et al*. [44] 3Net ResNet50	Stereo	CS,K	0.111	0.849	4.822	0.202	0.865	0.952	0.978
Pilzer *et al*. [39] (Teacher)	Stereo	CS,K	0.098	0.831	4.656	0.202	0.882	0.948	0.973
Yang *et al*. [58]	Seq+Stereo	K_o, K_r, K_o	0.097	0.734	4.442	0.187	0.888	0.958	0.980
monoResMatch	Stereo	CS,K	**0.096**	**0.673**	**4.351**	**0.184**	**0.890**	**0.961**	**0.981**

Table 2. Quantitative evaluation on the test set of KITTI dataset [9] using the split of Eigen *et al*. [6], maximum depth: 80m. Last four entries include post-processing [11]. K_o, K_r, K_o are splits from K, defined in [58]. Best results are shown in bold.

ducted on the Eigen split, are collected in Table 1.

Proxy-supervised loss analysis. We train *monodepth* framework by Godard *et al*. [11] from scratch adding our proxy-loss, then we compare the obtained model with the original one, as well as with the more effective strategy used by 3Net [44]. We can observe that proxy-loss enables a more accurate *monodepth* model (row 3) compared to [11], moreover it also outperforms virtual trinocular supervision proposed in [44], attaining better metrics with respect of both, but $\delta < 1.25$ for 3Net. Specifically, by recalling Figure 3, the proxy distillation couples well with a cropping strategy, solving well-known issues for stereo supervision such as disparity ramps on the left border. We refer to supplementary material for additional qualitative examples.

Component analysis. Still referring to Table 1, we evaluate different configurations of our framework by ablating the key modules peculiar to our architecture. First, we train monoResMatch on K without proxy supervision (row 3) to highlight that our architecture already outperforms [11] (row 1). Training on CS+K with proxy labels, we can notice how without any refinement module (*no-refinement*), our framework already outperforms the proxy-supervised ResNet50 model of Godard *et al*. [11]. Adding the disparity refinement component without encoding any matching relationship (*no-corr*) enables small improvements, becoming much larger on most metrics when a correlation layer is introduced (*no-pp*) to process real and synthesized features as to resemble stereo matching. Finally, post-processing as in [11] (row 11) still ameliorates all scores, although the larger contribution is given by the correlation-based refinement module, as perceived by comparing *no-refinement* and *no-pp* entries. Finally, by comparing rows 4 and 11 we can also perceive the impact given by CS pretraining on our full model.

5.4. Comparison with self-supervised frameworks

Having studied in detail the contribution of both monoResMatch architecture and proxy supervision, we compare our framework with state-of-the-art self-supervised approaches for monocular depth estimation. Table 2 collects results obtained evaluating different models on the aforementioned Eigen split [6]. In this evaluation, we consider only competitors trained without *any* supervision from ground truth labels (*e.g*., synthetic datasets [34]) involved in *any* phase of the training process [30, 13]. We refer to methods using monocular supervision (*Seq*), binocular (*Stereo*) or both (*Seq+Stereo*). Most methods are trained on CS and K, except Yang *et al*. [58] that leverages on different sub-splits of K. From the table, we can notice that monoResMatch outperforms all of them significantly.

To compete with methods exploiting supervision from dense synthetic ground truth [34], we run additional experiments using very few annotated samples from KITTI as in [31, 13], for a more fair comparison. Table 3 collects the outcome of these experiments according to different degrees of supervision, in particular using accurate ground truth labels from the KITTI 2015 training split (200-acrt) or different amounts of samples from K with LiDAR measurements, respectively 100, 200, 500 and 700 as proposed in [31, 13], running only 5k iterations for each configuration. We point out that monoResMatch, on direct comparisons to methods trained with the same amount of labeled images, consistently achieves better scores, with rare exceptions. Moreover, we highlight in red for each metric the best score among all the considered configurations, figuring out that monoResMatch trained with 200-acrt plus 500 samples from K attains the best accuracy on all metrics. This fact points out the high effectiveness of the proposed architecture, able to outperform state-of-the-art techniques [30, 13] trained with much more supervised data (*i.e*., more than 30k stereo pairs from [34] and pre-trained weights from ImageNet). Leveraging on the traditional SGM algorithm instead of a deep stereo network as in [13] for proxy-supervision ensures a faster and easier to handle training procedure.

5.5. Performance on single view stereo estimation

Finally, we further compare monoResMatch directly with Single View Stereo (SVS) by Luo *et al*. [30], being both driven by the same rationale. We fine-tuned monoRes-

Method	Supervision					Lower is better				Higher is better		
	200-acrt	100	200	500	700	Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Luo *et al.* [30]	✓					0.101	0.673	4.425	**0.176**	-	-	-
monoResMatch	✓					**0.089**	**0.575**	**4.186**	0.181	0.897	0.964	0.982
Luo *et al.* [30]	✓		✓			0.100	0.670	4.437	0.192	0.882	0.958	0.979
monoResMatch	✓		✓			**0.096**	**0.573**	**3.950**	**0.168**	**0.897**	**0.968**	**0.987**
Luo *et al.* [30]	✓			✓		0.094	0.635	4.275	0.179	0.889	0.964	0.984
monoResMatch	✓			✓		**0.093**	**0.567**	**3.914**	**0.165**	**0.901**	**0.969**	**0.987**
Luo *et al.* [30]	✓				✓	**0.094**	0.626	4.252	0.177	0.891	0.965	0.984
monoResMatch	✓				✓	0.095	**0.567**	**3.942**	**0.166**	**0.899**	**0.969**	**0.987**
Guo *et al.* [13]		✓				**0.096**	0.641	4.095	**0.168**	0.892	0.967	0.986
monoResMatch		✓				0.098	**0.597**	**3.973**	0.169	**0.895**	**0.968**	**0.987**

Table 3. Experimental results on the Eigen split [6], maximum depth: 80m. Comparison between methods supervised by few annotated samples. Best results in direct comparisons are shown in bold, best overall scores are in red, consistently attained by monoResMatch.

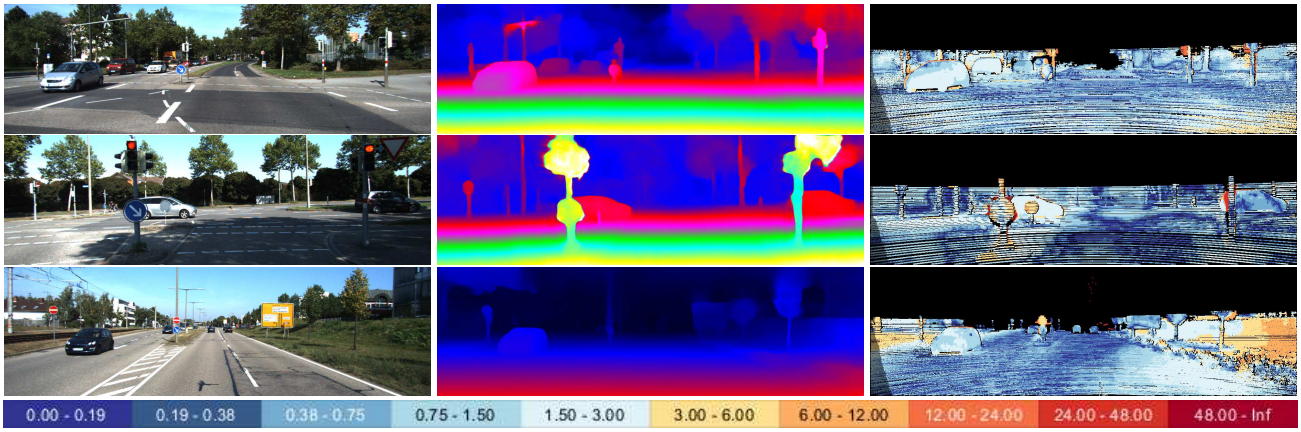


Figure 4. Stereo evaluation of our depth-from-mono framework. From left to right the input image, the predicted depth and the errors with respect to ground truth. The last line reports the color code used to display the seriousness of the shortcomings (same of [35])

Method	D1-bg	D1-fg	D1-all
monodepth [11]	27.00	28.24	27.21
OCV-BM	24.29	30.13	25.27
SVS [30]	25.18	20.77	24.44
monoResMatch	**22.10**	**19.81**	**21.72**

Table 4. Quantitative results on the test set of the KITTI 2015 Stereo Benchmark [35]. Percentage of pixels having error larger than 3 or 5% of the ground truth. Best results are shown in bold.

Match on the KITTI 2015 training set as in Table 3 and submitted to the online stereo benchmark [35] as performed in [31]. Table 4 compares monoResMatch with SVS and other techniques evaluated in [31], respectively monodepth [11] and OpenCV Block-Matching (OCV-BM). D1 scores represent the percentages of pixels having a disparity error larger than 3 or 5% of the ground truth value on different portions of the image, respectively background (bg), foreground (fg) or its entirety (all). We can observe from the table a margin larger than 3% on D1-bg and near to 1% for D1-fg, resumed in a total reduction of 2.72%. This outcome supports once more the superiority of monoResMatch, although SVS is trained on many, synthetic images with ground truth [34]. Finally, Figure 4 depicts qualitative examples retrieved from the KITTI online benchmark.

6. Conclusions

In this paper, we proposed monoResMatch, a novel framework for monocular depth estimation. It combines i) pondered design choices to tackle depth-from-mono in analogy to stereo matching, thanks to a correlation-based refinement module and ii) a more robust self-supervised training leveraging on proxy ground truth labels generated through a traditional (*i.e.* non-learning based) algorithm such as SGM. In contrast to state-of-the-art models [30, 13, 58], our architecture is elegantly trained in an end-to-end manner. Through exhaustive experiments, we prove that plugging proxy-supervision at training time leads to more accurate networks and, coupling this strategy with monoResMatch architecture, is state-of-the-art for self-supervised monocular depth estimation.

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

References

[1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *15th European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[2] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 18, page 1, 2018. 2

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[4] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 6

[5] Arun CS Kumar, Suchendra M. Bhandarkar, and Prasad Mukta. Monocular depth prediction using generative adversarial networks. In *1st International Workshop on Deep Learning for Visual SLAM, (CVPR)*, 2018. 2

[6] David Eigen, Christian Puhrsch, and Rob 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. 5, 6, 7, 8

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[8] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 2

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 5, 7

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 1

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 2, 4, 5, 6, 7, 8

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260, 2018.*, 2018. 4

[13] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018. 2, 3, 7, 8

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[15] Heiko 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. 1, 2, 5

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2, 4

[17] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[18] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *15th European Conference on Computer Vision (ECCV 2018)*, 2018. 2

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 3

[21] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[22] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014. 2

[23] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 2

[24] Zhengfa Liang, Yiliu Feng, YGHLW Chen, and LQLZJ Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018. 1, 4

[25] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3

[26] Yu Lidong, Wang Yucheng, Yuwei Wu, and Yunde Jia. Deep stereo matching with explicit cost aggregation sub-architecture. In *32th AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[27] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian 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

[28] Pengpeng Liu, Irwin King, Michael Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation, 2019. 2, 3

[29] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016. 2

[30] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 155–163, 2018. 2, 3, 4, 7, 8

[31] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4, 7, 8

[32] Reza Mahjourian, Martin Wicke, and Anelia 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. 1, 2, 7

[33] Osama Makansi, Eddy Ilg, and Thomas Brox. Fusionnet and augmentedflownet: Selective proxy ground truth for training on unlabeled images. *arXiv preprint arXiv:1808.06389*, 2018. 2, 3

[34] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3, 4, 7, 8

[35] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 6, 8

[36] Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007. 4

[37] Jiahao Pang, Wenxiu Sun, Jimmy SJ. Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 2

[38] Valentino Peluso, Antonio Cipolletta, Andrea Calimera, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms. In *Design Automation and Test in Europe (DATE)*, 2019. 2

[39] Andrea Pilzer, Stephane Lathuiliere, Nicu Sebe, and Elisa Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[40] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *IEEE/JRS Conference on Intelligent Robots and Systems (IROS)*, 2018. 1, 2, 7

[41] Matteo Poggi and Stefano Mattoccia. Learning from scratch a confidence measure. In *Proceedings of the 27th British Conference on Machine Vision, BMVC*, 2016. 5, 6

[42] Matteo Poggi, Davide Pallotti, Fabio Tosi, and Stefano Mattoccia. Guided stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[43] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Quantitative evaluation of confidence measures in a machine learning world. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[44] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *6th International Conference on 3D Vision (3DV)*, 2018. 1, 2, 6, 7

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

[46] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 2

[47] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2018. 2

[48] Robert Spangenberg, Tobias Langner, Sven Adfeldt, and Raúl Rojas. Large scale semi-global matching on the cpu. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 195–201. IEEE, 2014. 6

[49] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3, 5, 6

[50] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[51] Fabio Tosi, Matteo Poggi, Alessio Tonioni, Luigi Di Stefano, and Stefano Mattoccia. Learning confidence measures in the wild. In *28th British Machine Vision Conference (BMVC 2017)*, September 2017. 2, 3

[52] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 1

[53] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7

[54] Zhou 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. 4

[55] Junyuan Xie, Ross Girshick, and Ali 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, 4

[56] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural

fields for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[57] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *15th European Conference on Computer Vision (ECCV)*, 2018. 2

[58] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision*, pages 835–852. Springer, 2018. 1, 3, 6, 7, 8

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

[60] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantic for semi-supervised monocular depth estimation. In *14th Asian Conference on Computer Vision (ACCV)*, 2018. 2

[61] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 2

[62] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[63] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 1, 2

[64] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, pages 38–55. Springer, 2018. 2, 7