Unsupervised Domain Adaptation for Depth Prediction from Images

Unsupervised Domain Adaptation for Depth Prediction from Images

Alessio Tonioni*, *Student Member, IEEE,* Matteo Poggi*, *Member, IEEE,*
Stefano Mattoccia, *Member, IEEE,* and Luigi Di Stefano, *Member, IEEE*

Abstract—State-of-the-art approaches to infer dense depth measurements from images rely on CNNs trained end-to-end on a vast amount of data. However, these approaches suffer a drastic drop in accuracy when dealing with environments much different in appearance and/or context from those observed at training time. This domain shift issue is usually addressed by fine-tuning on smaller sets of images from the target domain annotated with depth labels. Unfortunately, relying on such supervised labeling is seldom feasible in most practical settings. Therefore, we propose an unsupervised domain adaptation technique which does not require groundtruth labels. Our method relies only on image pairs and leverages on classical stereo algorithms to produce disparity measurements alongside with confidence estimators to assess upon their reliability. We propose to fine-tune both *depth-from-stereo* as well as *depth-from-mono* architectures by a novel confidence-guided loss function that handles the measured disparities as noisy labels weighted according to the estimated confidence. Extensive experimental results based on standard datasets and evaluation protocols prove that our technique can address effectively the domain shift issue with both stereo and monocular depth prediction architectures and outperforms other state-of-the-art unsupervised loss functions that may be alternatively deployed to pursue domain adaptation.

Index Terms—Deep learning, depth estimation, unsupervised learning, self-supervised learning, domain adaptation

1 Introduction

Depth sensing plays a central role in many computer vision applications. Indeed, the availability of 3D data can boost the effectiveness of solutions to tasks as relevant as autonomous or assisted driving, SLAM, robot navigation and guidance, and many others. Active 3D sensors exhibit well-known drawbacks that may limit their practical usability: LiDAR, *e.g.*, is cumbersome, expensive and provides only sparse measurements, while structured light features a limited working range and is mainly suited to indoor environments. On the other hand, passive techniques enabling to infer depth from images are suitable to most scenarios due to their low cost and easiness of deployment. Among these, binocular stereo [1] represents one of the most popular choices and a very active research topic since several decades. Depth-from-stereo relies on finding the displacement (disparity) between corresponding pixels in two horizontally-aligned frames, which, in turn, enables depth estimation via triangulation. Although stereo has been tackled for years by hand-engineered algorithms, deep learning approaches have recently proved to be effective and yield superior accuracy. The advent of deep learning in stereo initially concerned replacing key steps within traditionally handcrafted pipelines. Afterward, the whole process was addressed by deep architectures trained end-to-end to regress depths (disparities) from image pairs. These approaches represent nowadays the undisputed state-of-the-art provided that a vast amount of stereo pairs endowed with groundtruth depth labels are available for training. Purposely, the training procedure for end-to-end stereo architectures relies on an initial optimization based on a large synthetic dataset [2] followed by fine-tuning on, possibly many, image pairs with groundtruth sourced from the target domain. As a matter of fact, the popular KITTI benchmarks [3], [4] witness the supremacy of deep stereo architectures [5], [6], while this is quite less evident in the Middlebury benchmark [7], where traditional, hand-crafted algorithms [8], [9] still keep the top rankings on the leaderboards due to the smaller amount of images available for training. Deep learning did also dramatically boost development and performance of depth-from-mono architectures, which can predict depth from just one image and, thus, be potentially deployed on the far broader range of devices equipped with a single camera.

Nonetheless, with both stereo and monocular setups, deep architectures aimed at predicting depth from images are severely affected by the *domain shift* issue, which hinders effectiveness when performing inference on images significantly diverse from those deployed throughout the training process. This can be observed, for instance, when moving between indoor and outdoor environments, from synthetic to real data or between different outdoor/indoor environments. As already pointed out, in the standard training procedure this issue is addressed by fine-tuning on labeled images from the target domain. However, suitable labeled data are available only for a few benchmark datasets, *e.g.* KITTI, whilst in most practical settings acquiring images annotated by depth labels would require the deployment of expensive sensors (*e.g.*, LiDAR) alongside with careful calibration. As this procedure is cumbersome and costly, collecting and labeling enough images to pursue fine-tuning in the target domain may easily turn out unfeasible. Thus,

- **joint first authorship*
- *A. Tonioni, M. Poggi, S. Mattoccia and L. Di Stefano are with the Department of Computer Science and Engineering, University of Bologna, Italy, IT.*
  {alessio.tonioni,m.poggi,stefano.mattoccia,luigi.distefano } @unibo.it

although all state-of-the-art approaches for depth/disparity estimation from images rely on deep CNNs, the domain shift issue prevents widespread adoption of these architectures in practical settings.

To address the above issue, in this paper we propose an unsupervised technique which allows for fine-tuning end-to-end architectures aimed at depth prediction without the need for groundtruth labels from the target domain. We argue that classical stereo matching algorithms rely on domain-agnostic computations that can deliver disparity/depth measurements in any working environment seamlessly. Although these measurements are prone to errors due to the known sub-optimality of stereo algorithms, we posit that they may be deployed as noisy labels to pursue fine-tuning of depth prediction architectures. Indeed, state-of-the-art estimators can reliably assess the confidence of disparity/depth predictions. Thus, we propose a novel learning framework based on a confidence-guided loss function which allows for fine-tuning depth prediction models by weighting the disparity/depth measurements provided by a stereo algorithm according to the estimated confidence. As a result, our approach can perform adaptation by solely feeding the model with synchronized stereo images from the target domain, *i.e.* without requiring cumbersome and expensive depth annotations.

2 Related work

Deep stereo. Since the early works on stereo, classical algorithms [1] comprise several sequential steps dealing with initial matching cost computation, local aggregation, disparity optimization and refinement. The first attempt to plug deep learning into a well established stereo pipeline was aimed at replacing matching cost computation [10], [11], [12], while disparity optimization [13], [14] and refinement [15] have been addressed more recently. Although these works proved the superiority of learning-based methods in the addressed steps, in most cases traditional optimization strategies, such as Semi Global Matching (SGM) [16], were needed to reach top accuracy. The shift toward end-to-end architectures started with DispNet, a seminal work by Mayer et al. [2]. Unlike previous proposals that process small image patches to compute similarity scores [10], [11], [12], DispNetC relies on a much larger receptive field, extracts features jointly from the two input images and computes correlations to predict the final disparities. This approach, however, mandates a significant amount of labeled training samples such that the few hundreds of images available in KITTI [3], [4] turn out definitely insufficient. To tackle this issue, a large synthetic dataset [2] was created and deployed for training, with KITTI images used to address the domain shift issue arising when running the network on real imagery. Although DispNetC did not reach the top rank on KITTI, it inspired other end-to-end models [5], [17], [18] which, in turn, were able to achieve state-of-the-art performance. Along a similar research line, some authors deploy 3D convolutions to exploit geometry and context [6], [19], [20], [21]. Despite the different architectural details, these techniques follow the same synthetic-to-real training schedule as originally proposed for DispNet. Differently, Zhout et. al. [22] described an iterative procedure based on the left-right check to train a deep stereo network from scratch without the need of groundtruth disparity labels. Finally, Zhang et al. [23] proposed a novel loss function formulation to enable depth estimation without supervision within an active stereo acquisition setup.

Confidence measures for stereo. Confidence measures were extensively reviewed at first by Hu and Mordohai [24] and more recently by Poggi et al. [25], who considered approaches leveraging on machine-learning. These are mainly based either on random forests [26], [27], [28], [29] or CNNs [13], [30], [31], [32]. While most of the former methods usually combine different cues available from the intermediate cost volume calculated by classical stereo algorithms [16], [33], [34], the latter can deploy just disparity maps and image cues, which renders it amenable also to depth estimation frameworks, such as end-to-end CNNs, that do not explicitly provide a cost volume. Moreover, CNN-based confidence estimators have been recently shown to exhibit better outlier detection performance [25]. [35] proposed an effective deep learning approach to improve confidence measures by exploiting local consistency while [36] a method to ameliorate random forest-based approaches for confidence fusion [27], [28], [29]. Shaked and Wolf [37] embedded confidence estimation within a deep stereo network while other works looked deeper into the learning process of confidence measures, either by studying features augmentation [38] or designing self-supervised techniques to train on static video sequences [39] or stereo pairs [40]. Finally, Poggi et al. [41] evaluated simplified confidence measures for embedded systems.

Depth-from-mono. Deep learning dramatically boosted the results attainable by a monocular depth prediction setup. While the vast majority of works addressed the depth-from-mono problem through supervised learning [42], [43], [44], [45], [46], [47], [48], [49], [50], an exciting recent trend concerns self-supervising the model by casting training as an image reconstruction problem. This formulation is earning increasing attention due to the potential to train depth prediction networks without hard to source depth labels. Self-supervised depth-from-mono methods can be broadly classified into monocular and stereo. With the former approach [51], [52], [53], [54] images are acquired by an unconstrained moving camera and the estimated depth is used to reconstruct views across the different frames through camera-to-world projection and vice-versa. Thus, the network has to estimate also the unknown camera pose between frames and the computation tends to fail when moving objects are present in the scene. The latter category requires a calibrated stereo setup to carry out the training phase [55], [56], [57], [58], [59]. As, in this case, the relative pose between the two cameras is known, the network has only to estimate the depth (actually, disparity) that minimizes the reprojection error between the two views. Thus, on one hand, this strategy can handle seamlessly moving objects, on the other it constraints data collection. Networks trained according to a stereo setup yield usually more accurate depth estimations. Moreover, this approach can be extended to three views [60] to compensate for the occlusions inherited by the binocular setup. Finally, we mention the joint use of these two supervision strategy [58] and the semi-supervised frameworks proposed in [61],

[62] that combined sparse groundtruth labels with stereo supervision.

In [63] we highlighted the issues and challenges set forth by the deployment of deep stereo architectures across multiple domains due to the lack of labeled data to perform fine-tuning. Accordingly, we proposed to adapt a deep stereo network to a new domain without any supervision by a novel loss function that leverages on a confidence estimator in order to detect reliable measurements among the disparities provided by a classical stereo algorithm. Later, Pang et al. [64] addressed the same topic and proposed to achieve adaptation of a deep stereo network by combining the disparity maps computed at multiple resolutions within an iterative optimization procedure.

This paper extends the early ideas and findings presented in [63]. In particular, while in [63] we considered only deep stereo, we provide here a general formulation to addresses both depth-from-stereo as well as depth-from-mono. Besides, we present a more comprehensive collection of quantitative and comparative experimental results. As for depth-from-stereo, thanks to the vast amount of depth labels released recently [65], starting with DispNetC [2] pre-trained on synthetic data we show adaptation results on the KITTI raw dataset [66], which includes more than 40k images. As for depth-from-mono, we consider the deep architecture recently proposed by Godard et al. [56] and perform domain adaptation from the CityScapes dataset [67] toward KITTI.

3 Domain Adaptation for Depth Sensing

This section describes our domain adaptation framework, which is suited to both deep stereo as well as monocular depth estimation networks. To adapt a pre-trained model facing a new environment, we first acquire stereo pairs from the target domain. Then, we deploy a classical (*i.e.*, not learning-based) stereo algorithm to generate dense depth measurements together with a state-of-the-art confidence measure to estimate the reliability of the depth values calculated by the stereo algorithm.

A key observation behind our method is that classical stereo algorithms, although affected by well-known shortcomings such as occlusions, poorly-textured regions, and repetitive patterns, are substantially agnostic to the specific target environment and thus behave similarly across different scenarios. More importantly, they fail in the same predictable way, thereby enabling confidence measures to achieve remarkably good accuracy in detecting mistakes regardless of the sensed environment [25].

Based on the above observations, we advocate deploying the depths delivered by a classical stereo algorithm as noisy labels endowed with reliability estimations in order to fine-tune a network aimed at depth prediction. This is achieved through a novel per-pixel regression loss wherein the error between each model prediction and the corresponding depth measurement provided by the stereo algorithm is weighted according to the reliability estimated by the confidence measure, with higher weights associated to more reliable depth measurements. Thereby, the learning process is guided by the high-confidence depth measurements, *i.e.* those labels that appear to be more reliable, while the errors due to the shortcomings of the stereo algorithm have a negligible impact.

Thus, given a pre-trained depth estimation network, either stereo or monocular, and a set of stereo pairs, $(I^l, I^r) \in \mathcal{I}$, acquired from the target domain, for each pair we compute a dense disparity map, $D \in \mathcal{D}$, by means of a classical stereo algorithm, $f : (\mathcal{I}, \mathcal{I}) \rightarrow \mathcal{D}$, such as, *e.g.*, SGM [16] or AD-CENSUS [33]. Moreover, for each disparity map, D, we estimate a pixel-wise degree of reliability according to a confidence measure, $c : \mathcal{D} \rightarrow \mathcal{C}$. The resulting confidence map, $C \in \mathcal{C}$, encodes the reliability of the disparity calculated at each pixel as a score ranging from 0 (*not reliable*) to 1 (*reliable*).

We run f and c on each stereo pair available from the target domain so as to produce the training set deployed to perform fine-tuning of the pre-trained depth estimation network. Therefore, each sample, (S_i), in the training set is a tuple of four elements:

$$S_i = (I_i^l, I_i^r, D_i, C_i) = (I_i^l, I_i^r, f(I_i^l, I_i^r), c(f(I_i^l, I_i^r))) \quad (1)$$

Given the depth estimation network (either stereo or monocular), which takes input images and outputs per pixel disparities, we fine tune it toward the target domain by minimizing a loss function, L, consisting of three terms: a *confidence guided loss* (L_c), a *smoothing loss* (L_s) and an *image reconstruction loss* (L_r):

$$L = L_c + \lambda_1 \cdot L_s + \lambda_2 \cdot L_r \quad (2)$$

with λ_1, λ_2 hyper-parameters to weight the contribution of the associated loss terms. All the three components of our loss can be applied seamlessly to deep learning models aimed either at depth-from-stereo or depth-from-mono (in the latter case one just need to convert disparities into depths). The structure of the three terms in Equation 2 is detailed in the next sub-sections, while in Sec. 4 we present model ablation experiments aimed at assessing their individual contribution to performance.

3.1 Confidence Guided Loss

The inspiration for the L_c term in the loss function of Equation 2 comes from the observation that deep models can be successfully fine-tuned to new environments even by deploying only a few sparse groundtruth annotations. This is vouched by the performance achievable on the KITTI datasets [3], [4], [66], where only a subset of pixels carries depth annotations (roughly $\frac{1}{3}$ of the image). The common strategy to account for the missing values consists simply in setting the loss function to 0 at those locations, thereby providing the network with meaningful gradients only at a subset of the spatial locations. Indeed, even in these sub-optimal settings, networks are able to adapt and ameliorate accuracy remarkably well. We build on these observations and leverage on the confidence measure, c, to obtain sparse and reliable depth labels from the noisy output D of the stereo algorithm. With reference to Equation 1, denoting as $\tilde{D}$ the output predicted by the model at the current training iteration, we compute L_c as

$$L_c = \frac{1}{|P_v|} \sum_{p \in \mathcal{P}_v} \mathcal{E}(p) \quad (3)$$

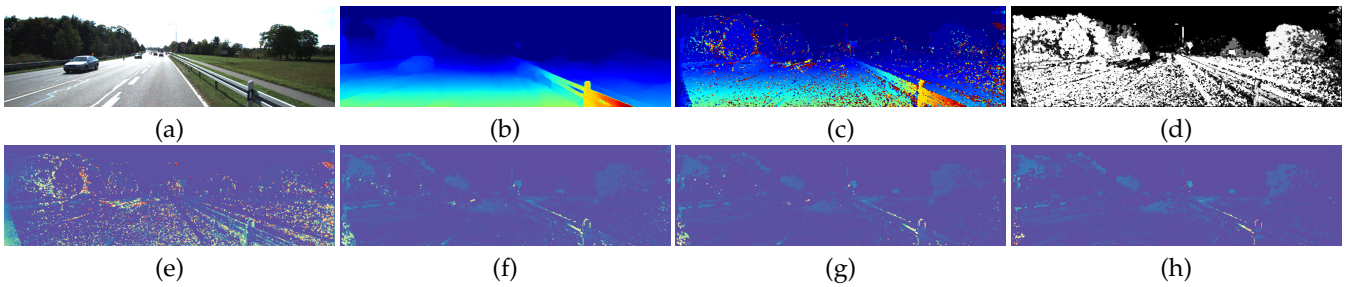


Fig. 1. Visualization of our confidence guided loss: (a) left frame I^l; (b) Disparity map, $\tilde{D}$, predicted by the model; (c) Disparity map, D, estimated by a stereo algorithm; (d) Confidence map, C, on D; (e) L1 regression errors between (b) and (c), (f-h) same L1 errors weighted by C with $\tau = 0.00$ (f), $\tau = 0.50$ (g) and $\tau = 0.99$ (h). (e-h) Hotter colors encode larger differences.

$$\mathcal{E}(p) = C(p) \cdot |\tilde{D}(p) - D(p)| \tag{4}$$

$$\mathcal{P}_v = \{p \in \mathcal{P} : C(p) > \tau\} \tag{5}$$

where $\mathcal{P}$ is the set of all spatial locations on the image and $\tau \in [0, 1]$ a hyper-parameter that controls the sparseness and reliability of the disparity measurements provided by f that are deployed to update the model. A higher value of τ will mask out more mistakes in D though permitting less spatial locations to contribute to model update. Hence, points belonging to $\mathcal{P}_v$ define a set of sparse labels that, assuming the availability of a perfect confidence measure, may be used as if they were *groundtruth annotations*, *e.g.* akin to the LiDAR measurements deployed in the KITTI dataset. Yet, confidence measures are not perfect and often show some degree of uncertainty in the score assigned to disparity measurements. Thus, we weight the contribution at location p by $C(p) \in [0, 1]$, *i.e.* as much as the depth measurement, $D(p)$, can be trusted according to the confidence estimation, $C(p)$. We point out that, re-weighting the loss function in the presence of noisy labels has been successfully exploited in supervised classification [68], [69]. Our formulation deploys a similar idea for a dense regression problem. Yet, we leverage on an external and highly accurate strategy to detect noise in the labels (*i.e.*, the confidence measure) and mask out those labels which, according to the adopted strategy, are very likely wrong, *i.e.*, $\{D(p) : p \notin \mathcal{P}_v\}$. In Sec. 4.1.1 we will show how both masking and re-weighting are crucial components to maximize performance in the presence of noisy depth labels.

The bottom row of Fig. 1 shows a graphical visualization of the errors that our L_c loss term tries to minimize. On (e) we report the errors that will be minimized trying to directly regress the noisy depth labels of (c) given the model prediction on (b); on (f-g-h), instead, the errors minimized by applying L_c with different τ values ($0, 0.5$ and 0.99 respectively). By tuning τ we can control the number of pixels, and therefore labels, taking part in the network adaptation process. Clearly, leveraging on more labels comes at the cost of injecting more noise in the process, which, in turn, may harm adaptation, even if their contribution will be attenuated by C, *e.g.* compare (f) to (e) where the only difference is the scaling of errors by $C(p)$ in (f). In (h) we can appreciate how even with $\tau = 0.99$ the amount of pixels considered during the optimization process is still quite high. We refer the reader to [63] for a detailed analysis of the quantity and quality of the labels used in the optimization process for different values of τ.

3.2 Self-filtering Outliers

In our previous work, [63], a properly hand-tuned τ proved to be effective. However, as τ represents a hyper-parameter of the method, an appealing alternative would consist in learning it alongside with the model adaptation process. To this aim, we define τ as a learnable parameter in our framework and update its value by gradient descent anytime the confidence guided loss described in Sec. 3.1 is optimized. Unfortunately, as τ determines the number of pixels on which such loss is computed, with this learning strategy its value would rapidly converge to 1, *i.e.* so as to mask out all pixels in order to obtain a loss as small as zero. To avoid such a behavior, we reformulate Equation 3 as

$$L_c = \frac{1}{|P_v|} \sum_{p \in \mathcal{P}_v} \mathcal{E}(p) - \log(1 - \tau) \tag{6}$$

The additional logarithmic penalty discourages τ from being equal to 1, thereby avoiding complete masking out of all pixels. In the experimental results, we will show how learning τ performs almost equivalently to the use of a hand-tuned threshold obtained by validation on ground-truth data. The latter, however, would turn out quite a less practical approach in those scenarios for which our adaptation technique is designed. In our evaluation, we will report two main experiments by formulating τ as i) a learnable variable or ii) the output of a shallow neural network, referred to as τNet, applied to the reference image and consisting of three 3×3 Conv layers with 64 filters followed by a global average pooling operation. With this second approach, we allow τ to be a function of the current image content rather than a fixed threshold for the whole dataset.

3.3 Smoothing Loss

As L_c produces error signals to improve disparity prediction only at the subset of sparse image locations P_v, similarly to [70] we use an additional loss term, L_s, to propagate model update signals across neighboring spatial locations. In particular, L_s tends to penalize large gradients in the predicted disparity map ($\partial \tilde{D}$) while taking into account the presence of gradients in pixel intensities (∂I):

$$L_s = \frac{1}{|P|} \sum_{p \in \mathcal{P}} \partial_x \tilde{D}(p) \cdot e^{-||\partial_x I(p)||} + \partial_y \tilde{D}(p) \cdot e^{-||\partial_y I(p)||} \tag{7}$$

Thus, based on the consideration that depth discontinuities are likely to occur in correspondence of image edges, L_s constrains the predicted disparity map, $\tilde{D}$, to be smooth everywhere but at image edges. To efficiently compute gradients along x and y we use convolutions with 3×3 Sobel filter.

3.4 Image Reconstruction Loss

To further compensate for the sparse model update information yielded by L_c, we include in the loss function a pixel-wise *image reconstruction* term, denoted as L_r in Equation 2. Inclusion of this term in our loss has been inspired by [56], which has shown how deploying image re-projection between stereo frames can deliver a form of self-supervision to train a depth-from-mono network. Hence, given a stereo pair, I^l can be reconstructed from I^r according to the current disparity prediction $\tilde{D}$ by employing a bilinear sampler in order to render the process locally differentiable. Denoted as $\tilde{I}^l$ the re-projection of I^r according to $\tilde{D}$, we define the image reconstruction loss, L_r, as a weighted combination of the $L1$ norm and the single scale SSIM [71]:

$$L_r = \frac{1}{|P|} \sum_{p \in P} \alpha \frac{1 - SSIM(I^l(p), \tilde{I}^l(p))}{2} + (1-\alpha)|I^l(p) - \tilde{I}^l(p)| \tag{8}$$

Similarly to [56], we use a simplified SSIM based on a 3×3 block filter and set $\alpha = 0.85$ throughout all our experiments.

4 Experimental results

In this section, we present a large corpus of experiments aimed at assessing the effectiveness of our proposed unsupervised domain adaptation framework. As already mentioned, although in the initial proposal [63] our approach was concerned with deep stereo models only, in this paper we present a general formulation to adapt any architecture trained to predict dense depth maps provided that stereo pairs are available at training time. Therefore, we address two main settings: i) adaptation of a deep stereo network and ii) adaptation of a depth-from-mono network. As for the former, we carry out extensive experiments according to the protocol proposed in our previous work [63]. At that time, experiments were limited to KITTI 2012 and 2015, whilst in this paper, we can consider the whole KITTI raw dataset [66], which includes about 40K images, thanks to the groundtruth labels released recently in the official website [65]. As for the latter evaluation scenario, we follow the standard protocol from the literature of self-supervised monocular depth estimation [56], which consists in splitting the KITTI raw data into train and test, as proposed by Eigen et al. [44].

To deploy the confidence guided loss described in Sec. 3.1, in our evaluation we consider two classical stereo algorithms: AD-CENSUS (shortened AD) [33] and Semi-Global Matching (shortened *SGM*) [16] and leverage the implementations of [72]. We have selected these two popular algorithms because they show quite different behaviors. While AD tends to generate prediction errors in the form of small spikes in the disparity maps, the errors generated by SGM can often cause over-smoothing. Effectiveness with both types of error patterns may help testify the general validity of our proposal. Besides, while SGM may turn out remarkably accurate, AD is notoriously significantly more prone to errors, which, in our framework, leads to fewer disparity measurements used at training time to compute L_c due to fewer pixels belonging to $\mathcal{P}_v$. To measure the confidence of the disparity measurements coming from the stereo algorithms, we rely on CCNN [30] as it can yield state-of-the-art performance and does require just the disparity map as input. Thanks to the latter trait, CCNN can be applied to any stereo system, even in case one has no access to the source code of the algorithm or is willing to employ an off-the-shelf external device. As CCNN consists of a network trained to classify each disparity pixel as reliable or not according to a small support region, it needs to be trained before deployment. To avoid reliance on expensive depth annotations, we used the original authors' implementation[1] and trained two variants of the network - one for AD and the other for SGM - on synthetic images taken from the SceneFlow dataset [2]. More precisely, we took six random stereo pairs from the Driving portion of the dataset (0040, 0265 forward from 15mm focal length set and 0075 forward, 0099, 0122, 0260 backward from 35mm set) and trained CCNN for 14 epochs, as suggested in [30].

All the code developed is available to ease development of applications relying on depth sensing using deep learning models.[2]

4.1 Deep Stereo

Our first experimental scenario is about the adaptation of a depth-from-stereo network to a new environment. The common training procedure for this kind of models consists of first training on the large synthetic FlyingThings3D dataset [2] and then fine-tuning on the target environment. In these settings, our proposal brings in the advantage of enabling fine-tuning without reliance on depth annotations from the target environment, which would be costly or even prohibitive to collect. For all our tests we have used the DispNet-Corr1D [2] architecture, from now on shortened as DispNetC. Following the authors' guidelines [2], we have trained a re-implementation of DispNetC on FlyingThings3D by the standard supervised L1 regression loss. Then, we have used these pre-trained weights as initialization for all the tests discussed hereinafter.

For our experiments we rely on the KITTI RAW [66] dataset, which features $\sim 43K$ images with depth labels [65] converted into disparities by known camera parameters. Images are taken from stereo video sequences concerning four diverse environments, namely *Road*, *Residential*, *Campus* and *City*, containing 5674, 28067, 1149 and 8027 frames, respectively. Although all images come from driving scenarios, each environment shows peculiar traits that would lead a deep stereo model to gross errors without suitable fine-tuning. For example, *City* and *Residential* often depict road surrounded by buildings, while *Road* mostly concerns highways and country roads where the most common objects are cars and vegetation. Using this data and extend-

1. https://github.com/fabiotosi92/CCNN-Tensorflow
2. https://github.com/CVLAB-Unibo/Unsupervised_Depth_Adaptation

	Hyper parameters			Target Domain		Similar Domains	
Test	τ	λ_1	λ_2	bad3	MAE	bad3	MAE
(a) AD [33]	✗	✗	✗	32.03	19.60	32.03	19.60
(b) No Adaptation	✗	✗	✗	10.86	1.73	10.86	1.73
(c) Regression	✗	✗	✗	11.73	2.49	12.23	2.47
(d) Weighted	0	0	0	3.66	1.03	4.57	1.12
(e) Masked	0.8	0	0	3.17	1.02	3.97	1.09
(f) Masked+Smoothness	0.8	0.1	0	3.17	0.98	3.78	1.05
(g) Masked+Reprojection	0.8	0	0.1	3.03	0.98	3.70	1.05
(h) Complete Adaptation	0.8	0.1	0.1	**2.96**	**0.96**	**3.66**	**1.04**
(i) Learned Adaptation	learned	0.1	0.1	3.15	1.01	3.84	1.08
(j) τ*Net Adaptation*	learned	0.1	0.1	3.15	0.99	3.83	1.07

TABLE 1
Ablation study on the effectiveness of the different components of our *Adaptation* loss using AD as noisy labels estimator. Results computed on the KITTI RAW dataset using a 4-fold cross validation schema, best results highlighted in bold.

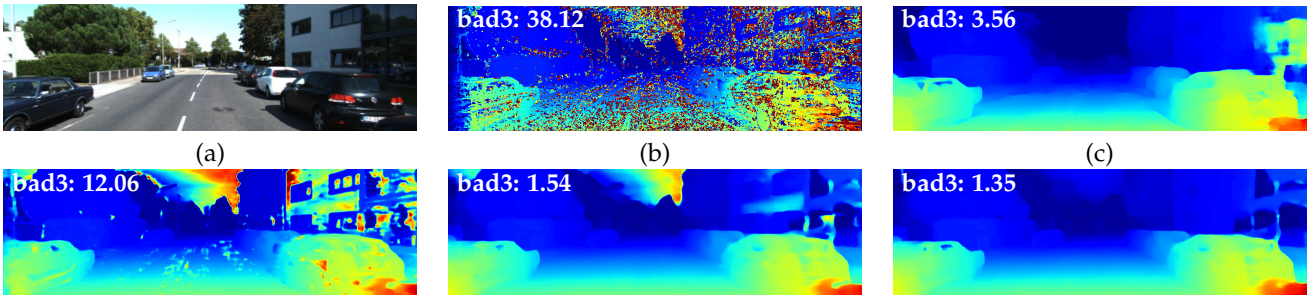


Fig. 2. Ablation experiments: fine-tuning DispNetC to new domains using AD [33]. (a) input image from KITTI, (b) disparities estimated by AD, (c) results without fine-tuning, (d) fine-tuning by AD only (*Regression*), (e) fine-tuning by weighting the loss through the confidence estimator (*Weighted*) and (f) our complete *Adaptation* method.

ing the protocol introduced in [63], we wish to measure both *target domain* performance, *i.e.*, how the network performs on the target domain upon unsupervised adaptation without access to any groundtruth information, as well as *similar domains* performance, *i.e.*, how the network adapted unsupervisedly generalizes to unseen images from similar domains. To analyze both behaviours, we have alternatively used one of the environments as the training set to perform fine-tuning, then tested the resulting model on all the four environments. In fact, this allows for assessing *target domain performance* by testing on the environment used for unsupervised fine-tuning and *similar domains performance* by testing on the other three. Since the environments amenable to perform fine-tuning are four, we can carry out 4-fold cross-validation in order to average performance figures. Hence, for each fold we average performance figures within an environment (*i.e.*, across all of its frames), obtaining, thereby, four sets of measurements. Then, we compute *target domain* performance by averaging the scores dealing with the four training sets in the corresponding four folds and *similar domains* performance by averaging across the other twelve scores.

As for the per-frame performance figures, we compute both the Mean Average Error (MAE) and the percentage of pixels with disparity error larger than 3 (bad3) as suggested in [3], [4]. Due to image formats being different across the KITTI RAW dataset, we extract a central crop of size 320×1216 from each frame, which matches to the downsampling factor of DispNetC and allows for validating almost all pixels with respect to the available groundtruth disparities.

4.1.1 Ablation Study

Our previous work [63] presented a detailed study on the impact of the hyper-parameters of the method for the depth-from-stereo networks, a similar discussion for depth-from-mono networks is reported in Sec. 4.2.2. Here, instead, we perform a more comprehensive ablation study aimed at answering the following questions: i) Can we simply use D as noisy groundtruth without deploying C? ii) Is masking by τ really needed or could we just use C as a per-pixel weighting in L_c? iii) How important is the contribution of the additional loss terms L_s, L_r? iv) How is performance affected by the use of a learnable τ?

To answer the above questions, we set AD as stereo algorithm, CCNN as confidence measure and run a set of experiments according to the cross validation protocol described in Sec. 4.1. The resulting performance figures are reported in Tab. 1 as follows. Starting from the top row: (a) AD, *i.e.* the stereo algorithm providing us with the noisy labels, (b) DispNetC trained only on synthetic data (*i.e.* the initial weights used for all the subsequent fine tuning), (c) DispNetC fine tuned to directly regress AD without deploying a confidence measure (*i.e.*, minimization of the error plotted in Fig. 1-(e)), (d) DispNetC fine tuned to minimize L_c with $\tau = 0$ (*i.e.*, minimization fo the error plotted in Fig. 1-(f) without explicit masking), (e-h) training to minimize different combinations of L_c, L_s and L_r with a fixed $\tau = 0.8$, (i) training with a learnable τ parameter or (j) by inferring it for each image with τNet. The values for λ_1, λ_2 and τ (when fixed) are obtained by preliminary cross-validation with a methodology similar to that described in our previous work [63]. Since rows (a) and (b) do not

need any kind of fine tuning on KITTI we report the same performances for both *target* and *similar domains*.

To answer the question (i), we can compare results between rows (c) and (b). As expected, fine-tuning the network to directly regress the noisy measurements produced by AD is not a valid optimization strategy as it worsens the initial network performance both in the *target domain* as well as in *similar domains*. Interestingly, the network structure seems to behave as a regularizer and does not overfit too much to the noise in the labels, as testified by the huge performance gap between rows (c) and (a). To answer the question (ii), we can compare line (e) and (d), where the only difference is the value of τ. The presence of $\tau = 0.8$ in (e) helps improving performance by about 0.5% bad3 while obtaining comparable performance in MAE. These results testify how masking out disparity measurements that are likely wrong yields better performance even though it increases the sparsity of the gradient signal actually deployed to update the model. A possible explanation for the close performance gap between (d) and (e) may be ascribed to the confidence maps produced by CCNN being highly bi-modal, with the vast majority of pixels carrying confidence scores equal to either 0 or 1. Therefore, even without applying a fixed threshold, many completely mistaken labels will see their contribution masked out during loss computation. To answer the question (iii) we can compare the performance reported in the last four rows. Adding L_s in the optimization process does not improve *target domain* performance but slightly helps in *similar domains*, as clearly observable by comparing rows (f) and (e). The introduction of L_r, instead, seems more effective and results in improvement across all metrics, as shown by rows (g) and (e). Once again, larger improvements are obtained in case of unseen images from *similar domains*. Furthermore, it is worth pointing out how our complete *Adaptation* loss yields the best results, as vouched by the performance figures reported in row (h). Finally, to answer question (iv), we can compare rows (i) and (j) to row (h). Letting τ be a learnable parameter (i) may ease the overall training process by avoiding manual tuning or grid-search to find the optimal threshold while yielding only a slight performance decrease, *i.e.*$+0.19\%$ and $+0.18\%$ bad3 in target and similar domains, respectively. Deploying the shallow τNet (j) to predict different thresholds places in between the two, showing improvements over learning a single τ but still not reaching the performance obtained through manual cross-validation.

Fig. 2 shows qualitative results related to the ablation study proposed in this subsection. The top row depicts the reference image (a), the noisy disparities provided by AD (b) and the prediction produced by DispNetC trained only on synthetic data (c). The bottom row, instead, reports three different predictions obtained by the three adaptation approaches referred to as *Regression* (d), *Weighted* (e) and *Complete* (f) in Tab. 1. By comparing (f) to (d) and (e) we can clearly verify that our adaptation scheme can successfully mask out all the noise in the labels and learn only from good disparities. Moreover, we can perceive the effectiveness of our adaptation approach by comparing (f) to (c), for example by observing how it can significantly reduce the errors caused by the reflective surface on the right portion of the image, without at the same time introducing many artifacts, as unfortunately does happen in (c) and (d).

	Target Domain		Similar Domains	
Loss	bad3	MAE	bad3	MAE
(a) No Adaptation	10.86	1.73	10.86	1.73
(b) GT Tuned (K12/15)	5.04	1.28	5.04	1.28
(c) Godard et. al. [56]	4.01	1.07	4.20	1.09
(d) Yinda et. al. [23]	3.59	1.00	5.15	1.14
(e) Tonioni et. al. [63]-AD	3.10	0.97	3.80	1.05
(f) *Masked-AD+Smooth.*	3.17	0.98	3.78	1.05
(g) Tonioni et. al. [63]-SGM	2.73	0.93	3.71	1.09
(h) *Masked-SGM+Smooth.*	2.79	1.01	3.63	1.09
(i) *Adaptation-AD* (τ=0.8)	2.96	0.96	3.66	1.04
(j) *Learned Adaptation-AD*	3.15	1.01	3.88	1.08
(k) *τNet-AD*	3.15	0.99	3.83	1.07
(l) *Adaptation-SGM* (τ=0.9)	**2.58**	**0.91**	3.39	**1.01**
(m) *Learned Adaptation-SGM*	2.84	0.99	3.75	1.07
(n) *τNet-SGM*	2.71	0.97	3.54	1.05
(o) *Adaptation-AD-SGM*	2.61	0.92	**3.37**	**1.01**
(p) *Learned Adaptation-AD-SGM*	2.77	0.99	3.54	1.07
(q) *τNet*-AD-SGM	2.79	0.97	3.67	1.07

TABLE 2
Results obtained performing fine tuning of a pre-trained DispNetC network using different unsupervised strategy. All results are computed on the KITTI raw dataset using a 4-fold cross validation schema, best results highlighted in bold, our proposals in italic.

4.1.2 Comparison to other self-supervised losses

We compare our proposal to other loss functions known in the literature that may be employed in order to fine-tune a deep stereo network without supervision. In particular, we consider two losses that, akin to ours, rely only on stereo frames to achieve a form of self-supervision: the appearance based re-projection and smoothness loss by Godard et al. [56] and the local constraint normalization with window-based optimization loss of [23]. As the underlying principles and mechanisms are quite straightforward to reproduce, we have re-implemented the two losses following the authors' guidelines. Thus, we apply these alternative losses together with variants of our proposal, relying either on AD or SGM or both stereo algorithms, in order to fine-tune DispNetC upon pre-training on synthetic data. As an additional comparison, we also report results obtained by our previous loss formulation [63] with both stereo algorithms. When using AD together with SGM, we fuse the disparity maps according to the corresponding confidences. For each pixel, we keep the disparity value with higher confidence among the two predictions. Then we obtain the corresponding confidence map as the pixel-wise max between those associated with the two algorithms. Finally, we consider all variants of our method: with a fixed $\tau = 0.9$ (*Adaptation*), a learned τ (*Learned Adaptation*) or the output of τNet (*τNet Adaptation*).

Again, we follow the same 4-fold cross validation protocol as in Sec. 4. Results are reported in Tab. 2 alongside with the performance of the pre-trained DispNetC model (No Adaptation) and those attainable by fine-tuning the pre-trained model by the LIDAR groundtruth available for the 400 frames of the KITTI2012 [3] and KITTI2015 [4] training sets (GT Tuned), *i.e.* according to the standard training methodology adopted in the vast majority of works dealing with deep stereo. For the sake of fair comparison, all methods are evaluated based only on the disparity map predicted for the left frames of the stereo pairs and can not leverage additional external networks besides DispNetC

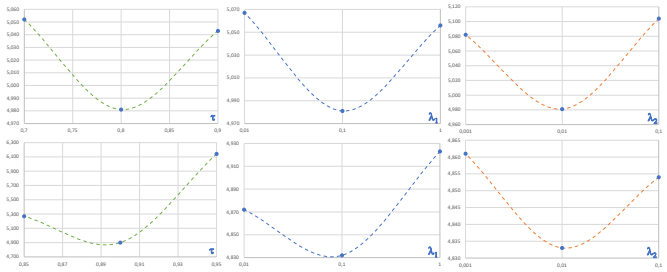


Fig. 3. Hyper-parameters study for unsupervised adaptation for *monodepth* [56], VGG model. Top: AD algorithm, bottom: SGM. From left to right, RMSE achieved after 5 epochs of adaptation by varying respectively τ, λ_1 and λ_2. Points are interpolated for visualisation purposes.

(*i.e.*, as for [23] we do not deploy also an external Invalidation Network).

Tab. 2 shows that our proposal outperforms other approaches both in the *target domain* as well as *similar domain* experiments. In particular, *Adaptation-SGM* (*row k*) delivers the best performance on the target domain, with gain as large as $\sim 1\%$ in the bad3 metric with respect to the closest competitor known in literature beside our previous work, *i.e.*, Yinda *et al.* at *row d*. The improvement is less substantial in the MAE figure, though our proposal still consistently outperforms alternative approaches. We also point out how our original proposal [63] (*rows e* and *d*) already outperforms competitors, which suggests the key component in our technique to be the confidence-guided loss. Yet, the novel *Adaptation* scheme proposed in this paper further ameliorates performance significantly. Moreover, from *row e* to *row h* we compare the impact of the original smoothness proposed in [63] to the edge-aware term introduced in this paper. In particular, while the former performs better on the target domain, the latter achieves lower errors when moving to similar domains, *e.g.* -0.08 on bad3 when comparing *row h* to *row g*. By comparing *Adaptation-AD* (*row i*) to *Adaptation-SGM* (*row l*) we can verify how a more accurate stereo algorithm (SGM vs AD) yields better performance. This can be ascribed to less noise in the disparities leading to a larger number of pixels scoring confidence $> \tau$ which, in turn, is conducive to denser and more accurate pseudo-groundtruths. Using both stereo algorithms (*Adaptation-AD-SGM – row o*) yields comparable performance to *Adaptation-SGM* in both scenarios, with the best absolute perfomance in *similar domains* and second best in *target domain*. This behaviour might be explained considering that the errors of AD are not usually complementary to those of SGM due to the vast majority of pixels with low confidence with SGM corresponding to equally low confidence pixels with AD. Therefore, the fusion of the two algorithms does not add many new useful labels that our method may use, leading to a marginal improvement on similar domains compared to SGM alone (-0.02% bad3). Comparing the performance of methods with fixed τ (*i.e.*, *Adaptation – row i* and *row l*) to those with τ as a learnable variable (*i.e.*, *Learned Adaptation – rows j, m, p*) we can see how the self-filtering strategy can ease the training process with a negligible loss in performance ($+0.2\%$ bad3 and $+0.06$ MAE), further reduced by estimating τ with a shallow network (*i.e.*, τ*Net – rows k, n, q*).

Finally, it is interesting to compare the performance achievable by fine-tuning without supervision on many data (*rows* from *e* to *p*) to those achievable by fine-tuning with supervision on few similar data (*i.e.*, *GT Tuned - row b*). The large performance margin in favour to most of unsupervised approaches indicates that training on much more data with a sub-optimal objective turns out not only easier and cheaper but also beneficial to performance with respect to training on few, perfectly annotated samples (*e.g.*, -1.65% bad3 and -0.27 MAE by comparing *Adaptation-SGM* to *GT Tuned*).

4.2 Depth-from-Mono

To investigate the application of our approach to depth prediction from a single image, we run experiments based on the popular depth-from-mono system developed by Godard et al. [56]. This choice is driven by two main factors i) despite a large number of works in this field [51], [52], [53], [54], it still represents one of the most effective solutions for unsupervised depth-from-mono and ii) the image reconstruction loss proposed by Godard et al. represent the main competitor to our approach. Thus the comparison to [56] turns out the ideal test bench for our proposal.

The network proposed in [56], referred to here as *monodepth*, consists in a DispNet-like architecture featuring a backbone encoder followed by a decoder to restore the original input resolution and predict the final depth map. In [56], both VGG [73] and ResNet50 [74] were tested as encoders. The output is provided as disparity (*e.g.*, inverse depth), and used at training time to warp the stereo images. This also eases the use of our unsupervised adaptation technique, that could be deployed anyway also in case of architectures directly predicting depth by simply converting our disparity labels based on known camera parameters. Moreover, in [56] a post-processing step is proposed to deal with occlusions and artifacts inherited from stereo supervision, by producing both normal and flipped depth maps and combining them. We will run experiments with and without this optional step, referred to as '+pp'.

We start from the TensorFlow codebase provided by the authors of [56], adding our proposal therein and running experiments within the same framework to ensure perfectly fair test conditions.

4.2.1 Evaluation protocol

We follow exactly the same protocol as reported in [56]. In particular, the KITTI raw dataset [66] is split into a training set and an evaluation set according to the guidelines by Eigen et al. [44]. Unlike the adopted stereo evaluation protocol [65], raw LiDAR measurements are usually assumed as groundtruth in the depth-from-mono literature despite their being sparse and noisy. Nonetheless, we adhere to the standard depth-from-mono evaluation protocol to ensure consistency with existing literature and enable a fair comparison with respect to [56].

Several works in this field [51], [54], [56] deploy pre-training on the CityScapes dataset [67] before fine-tuning on the KITTI training split [44], [66]. Indeed, training only on KITTI leads to inferior accuracy due to the fewer training images, whilst training only on CityScapes let the networks

				Lower is better		Higher is better		
Supervision	Encoder	Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard et al. [56]	VGG	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Masked-AD	VGG	0.119	0.989	4.981	0.207	0.859	0.950	0.977
Adaptation-AD	VGG	**0.118**	**0.976**	5.009	0.206	0.859	0.949	0.977
Learned Adaptation-AD	VGG	0.120	1.020	5.265	0.217	0.849	0.943	0.974
τNet Adaptation-AD	VGG	0.119	0.976	5.096	0.213	0.854	0.946	0.974
Masked-SGM	VGG	0.123	1.055	4.900	0.208	0.860	0.951	0.977
Adaptation-SGM	VGG	0.119	0.977	**4.833**	0.205	0.864	**0.952**	**0.978**
Learned Adaptation-SGM	VGG	**0.118**	1.015	5.166	0.213	0.854	0.947	0.975
τNet Adaptation-SGM	VGG	0.126	1.213	5.113	0.214	0.859	0.953	0.976
Masked-AD-SGM	VGG	0.122	1.049	4.975	0.207	0.857	0.950	0.976
Adaptation-AD-SGM	VGG	0.120	1.031	4.976	**0.204**	**0.865**	**0.952**	**0.978**
Learned Adaptation-AD-SGM	VGG	0.124	1.089	5.100	0.213	0.857	0.948	0.975
τNet Adaptation-AD-SGM	VGG	0.122	1.034	5.077	0.210	0.857	0.949	0.975
Godard et al. [56]	VGG+pp	0.118	0.923	5.015	0.210	0.854	0.947	0.976
Masked-AD	VGG+pp	**0.111**	0.871	4.852	0.199	0.858	0.952	0.980
Adaptation-AD	VGG+pp	**0.111**	0.865	4.901	0.200	0.859	0.950	0.979
Learned Adaptation-AD	VGG+pp	0.117	0.909	5.065	0.213	0.846	0.944	0.976
τNet Adaptation-AD	VGG+pp	0.111	0.872	4.974	0.215	0.853	0.948	0.978
Masked-SGM	VGG+pp	0.112	0.848	4.766	0.197	0.859	0.953	**0.981**
Adaptation-SGM	VGG+pp	**0.111**	**0.840**	**4.744**	**0.197**	**0.862**	**0.954**	0.980
Learned Adaptation-SGM	VGG+pp	0.114	0.890	4.961	0.207	0.853	0.948	0.978
τNet Adaptation-SGM	VGG+pp	0.113	0.922	4.904	0.199	0.858	0.953	0.980
Masked-AD-SGM	VGG+pp	0.114	0.915	4.909	0.199	0.859	0.953	0.980
Adaptation-AD-SGM	VGG+pp	**0.111**	0.902	4.863	0.199	**0.862**	**0.954**	**0.981**
Learned Adaptation-AD-SGM	VGG+pp	0.113	0.903	4.902	0.201	0.858	0.952	0.979
τNet Adaptation-AD-SGM	VGG+pp	0.112	0.892	4.913	0.200	0.859	0.952	0.979
Godard et al. [56]	ResNet50+pp	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Masked-AD	ResNet50+pp	**0.109**	0.867	4.810	0.197	0.866	0.953	0.979
Adaptation-AD	ResNet50+pp	**0.109**	0.867	4.852	0.196	0.866	0.954	0.978
Learned Adaptation-AD	ResNet50+pp	0.110	0.864	4.953	0.195	0.858	0.948	0.976
τNet Adaptation-AD	ResNet50+pp	0.109	0.863	4.927	0.204	0.858	0.948	0.976
Masked-SGM	ResNet50+pp	**0.109**	0.837	4.703	0.194	0.867	0.955	0.980
Adaptation-SGM	ResNet50+pp	**0.109**	**0.831**	**4.681**	**0.193**	0.867	**0.956**	**0.981**
Learned Adaptation-SGM	ResNet50+pp	0.111	0.880	4.820	0.196	0.864	0.954	0.980
τNet Adaptation-SGM	ResNet50+pp	0.109	0.858	4.794	0.196	0.865	0.954	0.979
Masked-AD-SGM	ResNet50+pp	0.110	0.866	4.775	0.195	0.867	0.955	0.980
Adaptation-AD-SGM	ResNet50+pp	0.110	0.891	4.809	0.196	**0.868**	**0.956**	**0.981**
Learned Adaptation-AD-SGM	ResNet50+pp	0.110	0.879	4.838	0.198	0.864	0.953	0.979
τNet Adaptation-AD-SGM	ResNet50+pp	0.110	0.872	4.837	0.198	0.863	0.953	0.979

TABLE 3
Experimental results on the KITTI dataset [66] on the data split proposed by Eigen et al. [44]. On even conditions, the proposed adaptation scheme outperforms the supervision by Godard et al. [56].

predicts depth maps of reasonable visual quality but totally wrong in terms of the actual depth values. This scenario, thus, points out again how a domain shift severely affects the accuracy of depth-from-images networks, *i.e.* exactly the issue we aim to address by the general domain adaptation framework proposed in this paper. Therefore, to assess the effectiveness of our proposal also in depth-from-mono settings, we will start from models pre-trained on CityScapes in order to adapt them to KITTI. In particular, relying on the very same models pre-trained on CityScapes we compare the results attained on the KITTI test split by performing fine-tuning on the KITTI train split by either our approach or the reconstruction loss proposed in [56]. As for our method, we use the same stereo algorithms (AD and SGM), confidence measure (CCNN) and hyper-parameter settings as in depth-from-stereo experiments. Coherently to [56], we used the Adam optimizer and found that, while our competitor needs to run 50 epochs of training on KITTI, our method reaches convergence after only 5 epochs with a fixed learning rate of 0.001, thus resulting in faster and, as we shall see in the next section, more effective adaptation.

4.2.2 Results on KITTI

We discuss here the outcomes of our experiments on the KITTI RAW dataset [66]. In particular, we report the standard error metrics, *i.e.* Absolute Relative error (Abs Rel), Square Relative error (Sq Rel), Root Mean Square Error (RMSE), logarithmic RMSE and the δ accuracy score computed as:

$$\delta = \tilde{D}_{i,j}\% : \max(\frac{\tilde{D}_{i,j}}{D_{i,j}}, \frac{\tilde{D}_{i,j}}{\tilde{D}_{i,j}}) < th \tag{9}$$

Hyper-parameters τ, λ_1 and λ_2 were manually tuned to obtain the best accuracy. Figure 3 reports how the RMSE metric behaves by varying each of the three parameters while adapting the VGG model on either AD (top) or SGM (bottom). We found configurations $\tau = 0.8, \lambda_1 = 0.1, \lambda_2 = 0.01$ and $\tau = 0.9, \lambda_1 = 0.1, \lambda_2 = 0.01$ to be the best for AD and SGM, respectively.

Table 3 reports a detailed comparison between the self-supervised loss proposed in [56] and our proposal in the aforementioned configurations *Masked*, *Adaptation*, *Learned* and *τNet Adaptation*, all applied to the same *monodepth* model pre-trained on CityScapes by the authors [56]. From

				Lower is better		Higher is better		
Configuration	Encoder	Abs Rel	Sq Rel	RMSE	RMSE log	δ <1.25	$\delta < 1.25^2$	$\delta < 1.25^3$
Regression-AD	VGG+pp	0.209	2.121	7.788	0.402	0.639	0.818	0.900
Weighted-AD	VGG+pp	0.124	1.010	5.446	0.236	0.825	0.932	0.968
Masked-AD	VGG+pp	0.111	0.871	4.852	0.199	0.858	0.952	0.980
Regression-SGM	VGG+pp	0.136	1.697	5.540	0.220	0.848	0.942	0.973
Weighted-SGM	VGG+pp	0.117	0.983	4.987	0.202	0.857	0.951	0.979
Masked-SGM	VGG+pp	0.112	0.848	4.766	0.197	0.859	0.953	0.981
Regression-AD	ResNet50+pp	0.230	3.240	8.361	0.418	0.624	0.806	0.893
Weighted-AD	ResNet50+pp	0.120	0.952	5.288	0.225	0.836	0.937	0.971
Masked-AD	ResNet50+pp	0.109	0.867	4.810	0.197	0.866	0.953	0.979
Regression-SGM	ResNet50+pp	0.129	1.456	5.385	0.214	0.854	0.943	0.973
Weighted-SGM	ResNet50+pp	0.115	0.966	4.925	0.199	0.863	0.952	0.979
Masked-SGM	ResNet50+pp	0.109	0.837	4.703	0.194	0.867	0.955	0.980

TABLE 4
Ablation experiments on the KITTI dataset [66] on the data split proposed by Eigen et al. [44].

top to bottom, we show the results dealing with VGG, VGG using post-processing step (+pp) and ResNet50 +pp models. The best metrics across the different configurations on a single model are higlighted in bold.

Starting from the basic VGG on top, we can observe that adapting by either AD, SGM or both combined with the Masked configuration alone leads to better performance with respect to using the image reconstruction loss proposed in [56]. In general, adapting by SGM yields superior results, outperforming the model based on AD in nearly all metrics. Applying our full adaptation scheme yields further improvements in almost all metrics with respect to the results achieved by the confidence guided loss alone. Contextually, we point out that combining AD and SGM achieves similar performance as observed for stereo experiments, leading to the best $\delta < 1.25^2$ and $\delta < 1.25^3$ together with *Adaptation-SGM* and achieving alone the best $\delta < 1.25$ score. Moreover, the *Learned Adaptation* scheme always achieves slightly worse results compared to a hand-tuned threshold τ, with *τNet Adaptation* placing in between the two alternatives. Nonetheless, all adaptation proposals turn out more accurate than the loss by Godard *et al.* [56].

This finding is confirmed when applying the post-processing step (i.e., VGG+pp), as our adaptation approach outperforms [56] under all evaluation metrics. Moreover, VGG+pp networks optimized by variants of our technique can deliver better results than using a ResNet+pp network trained according to the image reconstruction loss of [56], despite the large difference in complexity between the two networks (VGG features about 31 millions learnable parameters, ResNet50 about 57 millions). In this case, Adaptation-SGM consistently achieves the best results on most metrics, except for $\delta < 1.25^3$ where *Masked-SGM* and *Adaptation-AD-SGM* slightly outperforms it. Again, learning τ, by either the *Learned* or *τNet* strategy, leads to better results than Godard *et al.* on most metrics, although slightly reducing the effectiveness of our adaptation scheme.

Moving to ResNet50+pp model, the margin turns out even higher. We highlight once more how all the variants of our technique consistently outperforms Godard *et al.* on almost all cases. Similarly to VGG+pp, the lowest error metrics are achieved by Adaptation-SGM, while the highest $\delta < 1.25^2$ and $\delta < 1.25^3$ are sourced by both Adaptation-SGM and Adaptation-AD-SGM, being finally $\delta < 1.25$ better for the latter strategy thanks to the combination of the two stereo algorithms. Finally, determining τ by either the *Learned* or *τNet* strategy yields, again, to minor drops in almost all metrics. Thus, it may represent a practical alternative to explicit hand-tuning of τ.

4.2.3 Ablation experiments

Similarly to the stereo settings previously addressed in Table 1, we report here an ablation study aimed at establishing the relative importance of the key ingredients deployed in our framework. Table 4 collects the results obtained in this evaluation. We comment about four main experiments, dealing with running our method with both AD and SGM in order to adapt VGG and ResNet50. The post-processing step is enabled in all tests, thereby solving most issues near occlusions and left border and highlighting how the full confidence-guided loss ameliorates results in many regions of the images where post-processing cannot operate. Three setups are considered in descending order in the Table for each of the four experiments: i) adaptation by minimization of the L1 loss with respect to the disparity maps estimated by the stereo algorithm (AD or SGM) "as is" (*Regression*) ii) adaptation by weighting the L1 loss with per-pixel confidence scores (*Weighted*) iii) full confidence-guided loss using threshold τ (*Masked*). We turn off additional terms to focus on the different key factors of the confidence-guided loss. In all experiments, we can notice how using the disparity labels alone leads to poor results, in particular when adapting the model by the AD algorithm, which is much more prone to outliers. This further highlights how, in our framework, deploying the confidence measure is crucial to avoid the impact of the wrong disparities possibly computed by the stereo algorithms. Formulating the confidence-guided loss as a simple weighting between confidence scores and loss signals reduces the impact of the outliers, but does not completely removes it as they can still contribute to the entire loss function with a lower weight and thus may lead, as reported, to worse performance. To better perceive this effect, Fig. 4 shows some qualitative results obtained by the three ablated configurations reported in the Table. In particular, we point out how on (c) the results from the original model trained on different environments look good qualitatively, but the range of the predicted depth values is totally wrong (Abs Rel of 0.620). We can observe how ablated configurations of our technique (d-e) do yield gradual improvements, whereas the full adaptation scheme

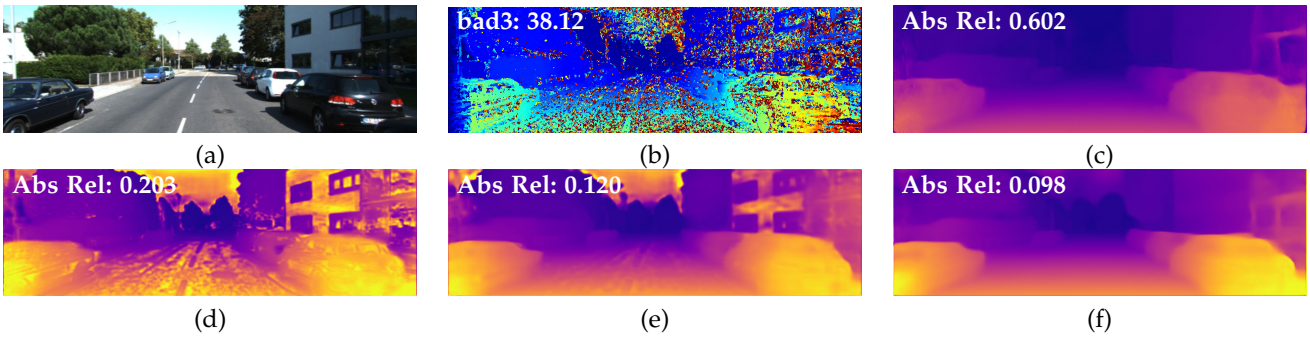


Fig. 4. Ablation experiments: adaptation of *monodepth* (VGG encoder) using AD algorithm. a) input image from KITTI b) result from AD algorithm c) result before adaptation d) adapting with stereo algorithm only e) using confidence to weight the loss function f) running full adaptation.

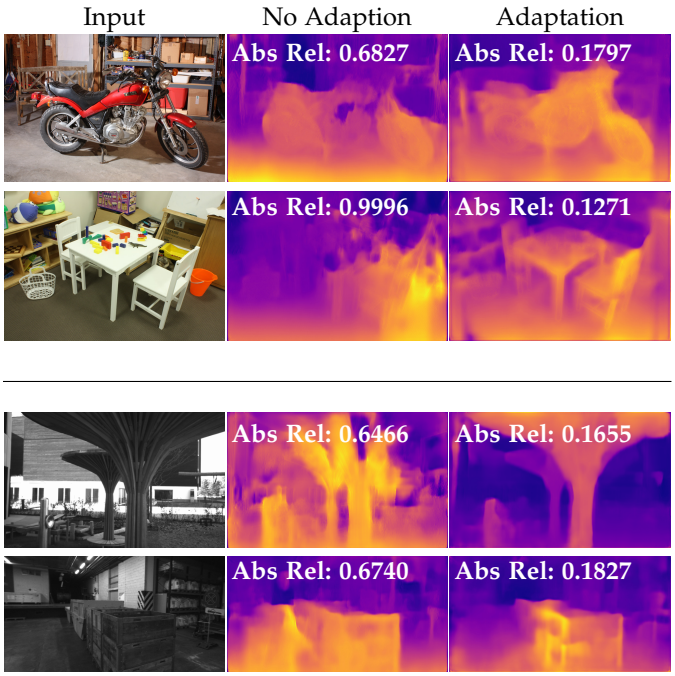


Fig. 5. Adaptation results for depth-from-mono on Middlebury v3 [7] (top) ETH3D dataset [75] (bottom). From left to right: input (left) image, depth maps from network before adaptation and after fine tuning with our adaptation technique. The absolute error rate is overimposed on each depth map.

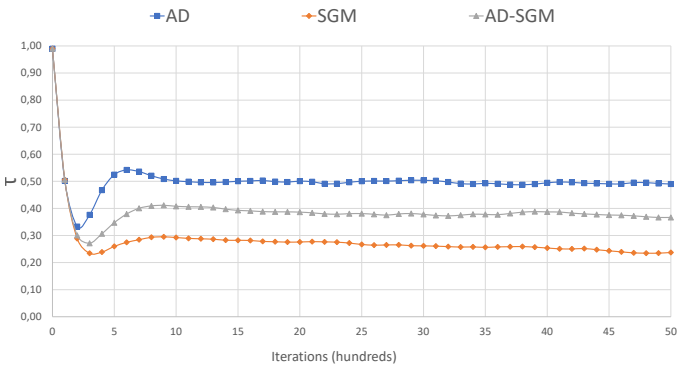


Fig. 6. Learned values of τ across three different training using different stereo algorithms and CCNN as confidence measure.

(f) greatly ameliorates the quality of the estimated depth maps, *i.e.* so as to bring the error down to 0.098 Abs Rel.

4.3 Analysis of τ convergence

To get insights on which values are automatically selected for τ using the learned adaptation scheme presented in Sec. 3.2, we plot in Fig. 6 the value of the variable across 5000 training iterations using either AD, SGM or the mixed stereo dataset and directly optimizing τ as a learnable parameter. In all the three runs, τ was initialized to 0.99 and then updated by gradient descent along with the other parameters. Similar behaviours are observed adapting both stereo and mono models therefore we report only the former.

The plot shows how across the three runs the value of τ starts to stabilize around 1000 iterations after an initial drop and subsequent rebound in the first 500. This occurs when the disparity loss surpasses the penalty term after several outliers have been included, thus preventing τ to decrease further. Overall the behaviour of τ resembles a *curriculum learning* [76] schedule. At the beginning a high τ value filters out most low-confidence pixels while keeping only high confidence ones, *i.e.* an easier regression task to learn. Then, τ starts decreasing, thereby considering more pixels, as well as noise, in the loss estimation process, *i.e.* the optimization task for the network becomes harder. In the end, the value of τ stabilizes to a reasonable threshold for both the considered stereo algorithms, with AD ending up to a higher value due to its higher amount of outliers. Consistently, the learned τ for AD-SGM is higher than that of SGM alone, suggesting how the fusion strategy introduce errors from AD within the SGM predictions. Concerning τNet we observed empirically that the predicted values of τ, on average, exhibit a similar behaviour with a slightly higher variance due to τNet being a function of the current input and not a global threshold.

Compared to the fixed τ, both learning strategy produce lower threshold, thus introducing more outliers during adaptation. Nevertheless, as hand-tuning by cross-validation is unlikely to happen in a real scenario without any available groundtruth, learning τ by the proposed techniques represents an effective strategy.

4.4 Qualitative Results

Finally we show some qualitative results, concerning both stereo and depth-from-mono networks, on the Middlebury v3 [7] and ETH3D [75] datasets. Fig. 5 shows examples of depth maps obtained by *monodepth* pre-trained on

CityScapes [67] before and after adaptation by our technique. The overall quality of the maps is greatly improved by the adaptation step, which is also vouched by the drastic drop of the absolute error reported in the Figure. We show similar results for DispNetC on Fig. 7: the column labeled as *No Adaptation* concerns predictions obtained by the model pre-trained on FlyingThings3D while the *Adaptation* column deals with the results obtained after fine-tuning by our unsupervised adaptation approach. Results indicate clearly how our proposal can successfully correct the prediction range and drastically reduce the percentage of wrong pixels.

Additional qualitative results are provided as supplementary material, in form of video sequences.

5 Conclusion

In this work, we have presented an effective methodology to fine-tune depth regressors based on CNNs towards brand-new environments by relying only on image pairs from the target domain. Through an extensive experimental evaluation, we have discussed the effectiveness of the different components of our method as well as proved its superior performance in comparison to popular alternatives dealing with both depth-from-stereo and depth-from-mono.

Our experiments suggest that combining naively noisy labels obtained from two very different stereo algorithms does not improve performance. Recent works like [77], however, have shown how combining different disparity estimations while taking into account the associated confidence maps can result in more reliable predictions. We plan to include in our framework a similar procedure in order to obtain more reliable disparity measurements from multiple, noisy stereo algorithms. Moreover, throughout this work, we have considered an offline adaptation phase aimed at ameliorating a successive online inference phase. Yet, one may conjecture a further extension of this concept whereby the two phases get fused together so as to adapt the depth prediction model online to ever-changing environments as soon as new images are gathered. By doing so, one may achieve better accuracy as well as realize a dynamic inference process capable of seamless adaption to unforeseen scenarios, like, *e.g.*, bad weather conditions in autonomous driving, which, nowadays, are hardly dealt with by both hand-crafted and learning-based methods aimed at estimating depth from images. Along this path, we would also point out the potential for improving the accuracy of the confidence scores assigned to disparity labels in an online manner, *e.g.*, by *self-paced* learning techniques or estimating confidence scores by the disparity regressor itself like in [78]. Eventually, the ideas and experiments proposed in this paper concern *adaptation* of a pre-trained CNN model to new settings. However, we believe that it would be worth investigating whether and how our unsupervised learning framework may be deployed to train a depth prediction model from scratch without supervision.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Thanks to Filippo Aleotti for its help with experiments on monocular depth estimation.

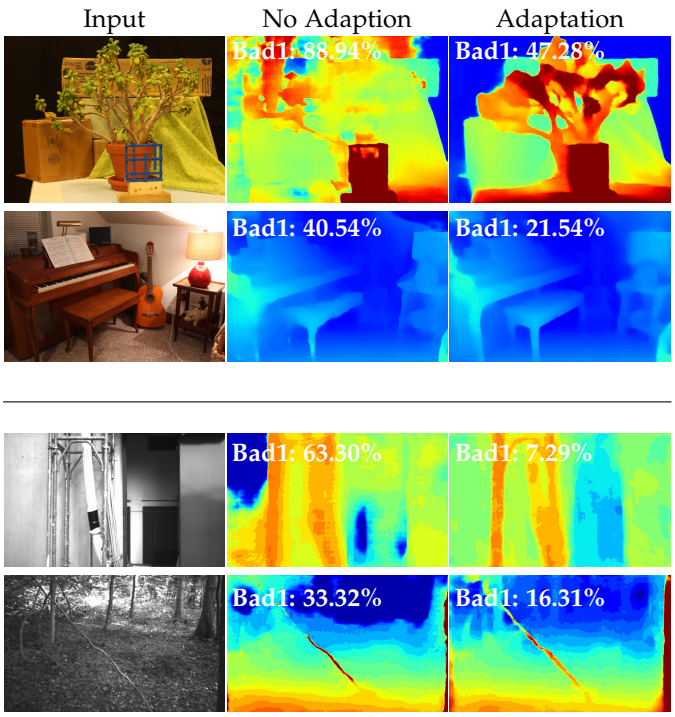


Fig. 7. Adaptation results for DispNetC on Middlebury v3 [7] (top) ETH3D dataset [75] (bottom). From left to right input (left) image, disparity maps predicted from network before any adaptation and after fine tuning with our adaptation technique. The bad1 error is overimposed on each map.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

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

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3354–3361.

[4] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," *arXiv preprint arXiv:1812.06264*, 2018.

[6] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *arXiv preprint arXiv:1810.02695*, 2018.

[7] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth." in *GCPR*, ser. Lecture Notes in Computer Science, X. Jiang, J. Hornegger, and R. Koch, Eds., vol. 8753. Springer, 2014, pp. 31–42.

[8] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, "Continuous 3D Label Stereo Matching using Local Expansion Moves," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 11, pp. 2725–2739, 2018.

[9] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang, "3d cost aggregation with multiple minimum spanning trees for stereo matching," *Applied optics*, vol. 56, no. 12, pp. 3411–3420, 2017.

[10] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592–1599.

[11] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs,"

in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.

[13] A. Seki and M. Pollefeys, "Patch based confidence prediction for dense disparity map," in *British Machine Vision Conference (BMVC)*, 2016.

[14] ——, "Sgm-nets: Semi-global matching with neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] 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*, vol. 2. IEEE, 2005, pp. 807–814.

[17] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. 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.

[18] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] L. Yu, Y. Wang, Y. Wu, and Y. Jia, "Deep stereo matching with explicit cost aggregation sub-architecture," in *AAAI Conference on Artificial Intelligence*, 2018.

[21] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] C. Zhou, H. Zhang, X. Shen, and J. Jia, "Unsupervised learning of stereo matching," in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 2, no. 8, 2017.

[23] Y. Zhang, S. Khamis, C. Rhemann, J. Valentin, A. Kowdle, V. Tankovich, M. Schoenberg, S. Izadi, T. Funkhouser, and S. Fanello, "Activestereonet: End-to-end self-supervised learning for active stereo systems," in *15th European Conference on Computer Vision (ECCV)*, September 2018.

[24] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pp. 2121–2133, 2012.

[25] M. Poggi, F. Tosi, and S. Mattoccia, "Quantitative evaluation of confidence measures in a machine learning world," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[26] R. Haeusler, R. Nair, and D. Kondermann, "Ensemble learning for confidence measures in stereo vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 305–312.

[27] A. Spyropoulos, N. Komodakis, and P. Mordohai, "Learning to detect ground control points for improving the accuracy of stereo matching." in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1621–1628.

[28] M. G. Park and K. J. Yoon, "Leveraging stereo matching with learning-based confidence measures," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] M. Poggi and S. Mattoccia, "Learning a general-purpose confidence measure based on o(1) features and a smarter aggregation strategy for semi global matching," in *Proceedings of the 4th International Conference on 3D Vision, 3DV*, 2016.

[30] ——, "Learning from scratch a confidence measure," in *Proceedings of the 27th British Conference on Machine Vision, BMVC*, 2016.

[31] Z. Fu and M. Ardabilian, "Learning confidence measures by multi-modal convolutional neural networks." in *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.

[32] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia, "Beyond local reasoning for stereo confidence estimation with deep learning," in *15th European Conference on Computer Vision (ECCV)*, September 2018.

[33] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, ser. ECCV '94. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994, pp. 151–158.

[34] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, no. 1-32, p. 2, 2016.

[35] M. Poggi and S. Mattoccia, "Learning to predict stereo reliability enforcing local consistency of confidence maps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] M. Poggi, F. Tosi, and S. Mattoccia, "Even more confident predictions with deep machine-learning," in *12th IEEE Embedded Vision Workshop (EVW2017) held in conjunction with IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] S. Kim, D. Min, S. Kim, and K. Sohn, "Feature augmentation for learning confidence measure in stereo matching," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 6019–6033, 2017.

[39] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, "Using self-contradiction to learn confidence measures in stereo vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4067–4076.

[40] F. Tosi, M. Poggi, A. Tonioni, L. Di Stefano, and S. Mattoccia, "Learning confidence measures in the wild," in *28th British Machine Vision Conference (BMVC 2017)*, September 2017.

[41] M. Poggi, F. Tosi, and S. Mattoccia, "Efficient confidence measures for embedded stereo," in *19th International Conference on Image Analysis and Processing (ICIAP 2017)*, September 2017.

[42] 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*, vol. 31, no. 5, pp. 824–840, 2009.

[43] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 89–96.

[44] 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*, 2014, pp. 2366–2374.

[45] 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*, vol. 38, no. 10, pp. 2024–2039, 2016.

[46] 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*. IEEE, 2016, pp. 239–248.

[47] 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*, 2015, pp. 1119–1127.

[48] 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)*, vol. 5, 2017.

[49] 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.

[50] 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.

[51] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, no. 6, 2017, p. 7.

[52] 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.

[53] 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.

[54] 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.

[55] 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*. Springer, 2016, pp. 740–756.

[56] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, no. 6, 2017, p. 7.

[57] 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.

[58] 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.

[59] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *15th European Conference on Computer Vision (ECCV) Workshops*, 2018.

[60] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *6th International Conference on 3D Vision (3DV)*, 2018.

[61] 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.

[62] 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.

[63] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[64] J. Pang, W. Sun, C. Yang, J. Ren, R. Xiao, J. Zeng, and L. Lin, "Zoom and learn: Generalizing deep stereo matching to novel domains," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[65] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017.

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

[67] 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*, 2016, pp. 3213–3223.

[68] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Procedings of the International Conference on Machine Learning (ICML)*, 2018.

[69] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Regularizing very deep neural networks on corrupted labels," in *Procedings of the International Conference on Machine Learning (ICML)*, 2018.

[70] P. Heise, S. Klose, B. Jensen, and A. Knoll, "Pm-huber: Patchmatch with huber regularization for stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2360–2367.

[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[72] R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large scale semi-global matching on the cpu," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 195–201.

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

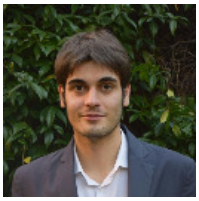
[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[75] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[76] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[77] K. Batsos, C. Cai, and P. Mordohai, "Cbmv: A coalesced bidirectional matching volume for disparity estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[78] M. Klodt and A. Vedaldi, "Supervising the new with the old: learning sfm from sfm," in *The European Conference on Computer Vision (ECCV)*, September 2018.



Alessio Tonioni Received his PhD degree in Computer Science and Engineering from University of Bologna in 2019. Currently, he is a Post-doc researcher at Department of Computer Science and Engineering, University of Bologna. His research interest concerns machine learning for depth estimation and object detection.

Matteo Poggi received his PhD degree in Computer Science and Engineering from University of Bologna in 2018. Currently, he is a Post-doc researcher at Department of Computer Science and Engineering, University of Bologna.

Stefano Mattoccia received a Ph.D. degree in Computer Science Engineering from the University of Bologna in 2002. Currently he is an associate professor at the Department of Computer Science and Engineering of the University of Bologna. His research interest is mainly focused on computer vision, depth perception from images, deep learning and embedded computer vision. In these fields, he has authored about 100 scientific publications/patents.

Luigi Di Stefano received the PhD degree in electronic engineering and computer science from the University of Bologna in 1994. He is currently a full professor at the Department of Computer Science and Engineering, University of Bologna, where he founded and leads the Computer Vision Laboratory (CVLab). His research interests include image processing, computer vision and machine/deep learning. He is the author of more than 150 papers and several patents. He has been scientific consultant for major companies in the fields of computer vision and machine learning. He is a member of the IEEE Computer Society and the IAPR-IC.