Guided Stereo Matching

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

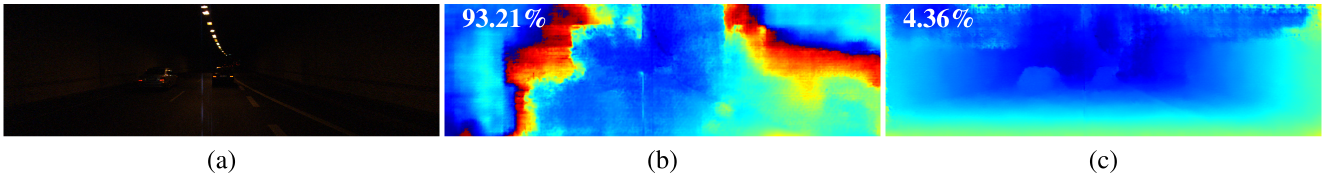

Figure 1. **Guided stereo matching.** (a) Challenging, reference image from KITTI 2015 [20] and disparity maps estimated by (b) iResNet [14] trained on synthetic data [19], or (c) guided by sparse depth measurements (5% density). Error rate (> 3) superimposed on each map.

Abstract

*Stereo is a prominent technique to infer dense depth maps from images, and deep learning further pushed forward the state-of-the-art, making end-to-end architectures unrivaled when enough data is available for training. However, deep networks suffer from significant drops in accuracy when dealing with new environments. Therefore, in this paper, we introduce Guided Stereo Matching, a novel paradigm leveraging a small amount of sparse, yet reliable depth measurements retrieved from an external source enabling to ameliorate this weakness. The additional sparse cues required by our method can be obtained with any strategy (*e.g.*, a LiDAR) and used to enhance features linked to corresponding disparity hypotheses. Our formulation is general and fully differentiable, thus enabling to exploit the additional sparse inputs in pre-trained deep stereo networks as well as for training a new instance from scratch. Extensive experiments on three standard datasets and two state-of-the-art deep architectures show that even with a small set of sparse input cues, i) the proposed paradigm enables significant improvements to pre-trained networks. Moreover, ii) training from scratch notably increases accuracy and robustness to domain shifts. Finally, iii) it is suited and effective even with traditional stereo algorithms such as SGM.*

1. Introduction

Obtaining dense and accurate depth estimation is pivotal to effectively address higher level tasks in computer vision such as autonomous driving, 3D reconstruction, and robotics. It can be carried out either employing active sensors, such as LiDAR, or from images acquired by standard cameras. The former class of devices suffers from some limitations, depending on the technology deployed to infer depth. For instance, sensors based on structured light have limited working range and are ineffective in outdoor environments, while LiDARs, although very popular and accurate, provide only sparse depth measurements and may have shortcomings when dealing with reflective surfaces. On the other hand, passive sensors based on standard cameras potentially allow obtaining dense depth estimation in any environment and application scenario. Stereo [28] relies on two (or more) rectified images to compute the disparity by matching corresponding pixels along the horizontal epipolar line, thus enabling to infer depth via triangulation. The most recent trend in stereo consists in training end-to-end Convolutional Neural Networks (CNNs) on a large amount of (synthetic) stereo pairs [19] to directly infer a dense disparity map. However, deep stereo architectures suffer when *shifting domain*, for example moving from synthetic data used for the initial training [19] to the real target imagery. Therefore, deep networks are fine-tuned in the target environment to ameliorate domain shift issues. Nonetheless, standard benchmarks used to assess the accuracy of stereo [7, 20, 27, 29] give some interesting insights concerning such paradigm. While it is unrivaled when a massive amount of data is available for fine-tuning, as is the case of KITTI datasets [7, 20], approaches mixing learning and traditional pipelines [35] are still competitive with deep networks when not enough data is available, as particularly evident with Middlebury v3 [27] and ETH3D [29] datasets.

*Joint first authorship.

In this paper, we propose to leverage a small set of sparse depth measurements to obtain, with deep stereo networks, dense and accurate estimations in any environment. It is worth pointing out that our proposal is different from depth fusion strategies (*e.g.*, [17, 21, 5, 1]) aimed at combining the output of active sensors and stereo algorithms such as Semi-Global Matching [10]. Indeed, such methods mostly aim at selecting the most reliable depth measurements from the multiple available using appropriate frameworks whereas our proposal has an entirely different goal. In particular, given a deep network and a small set (*e.g.*, less than 5% of the whole image points) of accurate depth measurements obtained by any means: can we improve the overall accuracy of the network without retraining? Can we reduce domain shift issues? Do we get better results training the network from scratch to exploit sparse measurements? To address these goals, we propose a novel technique acting at feature level and deployable with any state-of-the-art deep stereo network. Our strategy enhances the features corresponding to disparity hypotheses provided by the sparse inputs maintaining the stereo reasoning capability of the original deep network. It is versatile, being suited to boost the accuracy of pre-trained models as well as to train a new instance from scratch to achieve even better results. Moreover, it can also be applied to improve the accuracy of conventional stereo algorithms like SGM. In all cases, our technique adds a negligible computational overhead to the original method. It is worth noting that active sensors, especially LiDAR-based, and standard cameras are both available as standard equipment in most autonomous driving setups. Moreover, since the cost of LiDARs is dropping and solid-state devices are already available [26], sparse and accurate depth measurement seems not to be restricted to a specific application domain. Thus, independently of the technology deployed to infer sparse depth data, to the best of our knowledge this paper proposes the first successful attempt to leverage an external depth source to boost the accuracy of state-of-the-art deep stereo networks. We report extensive experiments conducted with two top-performing architectures with source code available, PSMNet by Chang et al. [3] and iResNet by Liang et al. [14], and standard datasets KITTI[7, 20], Middlebury v3 [27] and ETH3D [29]. The outcome of such evaluation supports the three following main claims of this work:

- Given sparse ($<$ 5% density) depth inputs, applying our method to pre-trained models always boosts its accuracy, either when the network is trained on synthetic data only or fine-tuned on the target environments.
- Training from scratch a network guided by sparse inputs dramatically increases its generalization capacity, significantly reducing the gap due to domain shifts (*e.g.*, when moving from synthetic to real imagery).
- The proposed strategy can be applied seamlessly even to conventional stereo algorithms such as SGM.

In Figure 1 we can notice how on a very challenging stereo pair from KITTI 2015 [20] (a) a state-of-the-art deep stereo network trained on synthetic data produces inaccurate disparity maps (b), while guiding it with our method deploying only 5% of sparse depth data allows for much more accurate results (c) despite the domain shift.

2. Related work

Stereo has a long history in computer vision and Scharstein and Szeliski [28] classified conventional algorithms into two main broad categories, namely *local* and *global* approaches, according to the different steps carried out: i) cost computation, ii) cost aggregation, iii) disparity optimization/computation and iv) disparity refinement. While local algorithms are typically fast, they are ineffective in the presence of low-texture regions. On the other hand, global algorithms perform better at the cost of higher complexity. Hirschmuller's SGM [10] is often the favorite trade-off between the two worlds and for this reason the preferred choice for most practical applications. Early attempts to leverage machine learning for stereo aimed at exploiting learning-based confidence measures [25] to detect outliers or improve disparity accuracy [34, 23, 24]. Some works belonging to the latter class modified the *cost volume*, an intermediate representation of the matching relationships between pixels in the two images, by replacing winning matching costs [34] or modulating their distribution [23] guided by confidence estimation.

The spread of deep learning hit stereo matching as well. Early works focused on a single step of traditional stereo pipelines, for example learning a matching function by means of CNNs [44, 4, 16], improving optimization carried out by SGM [30, 31] or refining disparity maps [8, 2]. Later, the availability of synthetic data [19] enabled to train end-to-end architectures for disparity estimation embodying all the steps mentioned above. In the last year, a large number of frameworks appeared, reaching higher and higher accuracy on KITTI benchmarks [7, 20]. All of them can be broadly categorized into two main classes according to how they represent matching relations between pixels along the epipolar line, similarly to what cost volume computation does for traditional stereo algorithms.

The first class consists of networks computing correlation scores between features belonging to the left and right frames. The outcome are feature maps, linked to disparity hypotheses, concatenated along the channel dimension. This volume is processed through 2D convolutions, usually by encoder-decoder architectures. DispNetC by Mayer et al. [19] was the first end-to-end network proposed in the literature suggesting this paradigm. More recent architectures

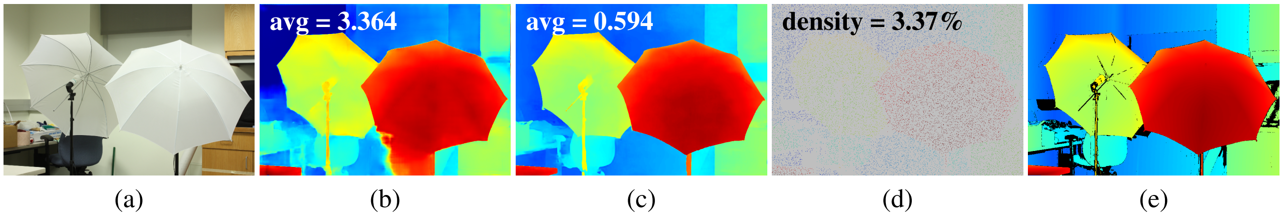

Figure 2. **Example of improved generalization.** (a) Reference image from Middlebury [27], disparity maps obtained by (b) iResNet [14] trained on SceneFlow synthetic dataset [19], (c) iResNet trained on SceneFlow for guided stereo, (d) visually enhanced sparse depth measurements taken from (e) ground-truth depth. We stress the fact that (b) and (c) are obtained training on synthetic images only.

such as CLR [22], iResNet [14], DispNet3 [11] were built on top of DispNetC. In addition, other frameworks such as EdgeStereo [32] and SegStereo [42] jointly tackled stereo with other tasks, respectively edge detection and semantic segmentation.

The second class consists of frameworks building 3D cost volumes (actually, 4D considering the feature dimension) obtained by concatenation [12] or difference [13] between left and right features. Such data structure is processed through 3D convolutions, and the final disparity map is the result of a differentiable *winner-takes-all* (WTA) strategy. GC-Net by Kendall et al. [12] was the first work to follow this strategy and the first end-to-end architecture reaching the top of the KITTI leaderboard. Following architectures built upon GC-Net improved accuracy, adding specific aggregation modules [15] and spatial pyramidal pooling [3], or efficiency, by designing a tiny model [13].

Despite the different strategies adopted, both classes somehow encode the representation of corresponding points in a data structure similar to the cost volume of conventional hand-crafted stereo algorithms. Therefore, with deep stereo networks and conventional algorithm, we will act on such data structure to guide disparity estimation with sparse, yet accurate depth measurements.

3. Guided stereo matching

Given sparse, yet precise depth information collected from an external source, such as a LiDAR or any other means, our principal goal is to exploit such cues to assist state-of-the-art deep learning frameworks for stereo matching. For this purpose, we introduce a *feature enhancement* technique, acting directly on the intermediate features processed inside a CNN by peaking those directly related to the depth value suggested by the external measurement. This can be done by precisely acting where an equivalent representation of the cost volume is available. The primary goal of such an approach is to further increase the reliability of the already highly accurate disparity map produced by CNNs. Moreover, we also aim at reducing the issues introduced by domain shifts. By feeding sparse depth measurements into a deep network during training, we also expect that it can learn to exploit such information together with image content, compensating for domain shift if such measurements are fed to the network when moving to a completely different domain (*e.g.*, from synthetic to real imagery). Our experiments will highlight how, following this strategy, given an extremely sparse distribution of values, we drastically improve the generalization capacity of a CNN. Figure 2 shows how deploying a 3.36% density of sparse depth inputs is enough to reduce the average error of iResNet from 3.364 to 0.594.

3.1. Feature enhancement

Traditional stereo algorithms collect into a cost volume the relationship between potentially corresponding pixels across the two images in a stereo pair, either encoding similarity or dissimilarity functions. The idea we propose consists in opportunely acting on such representation, still encoded within modern CNNs employing correlation or concatenation between features from the two images, favoring those disparities suggested by the sparse inputs. Networks following the first strategy [19, 14, 42, 32, 22] use a *correlation layer* to compute similarity scores that are higher for pixels that are most likely to match, while networks based on the second strategy rely upon a 3D volume of concatenated features. The cost volume of a conventional stereo algorithm has dimension $H \times W \times D$, with $H \times W$ being the resolution of the input stereo pair and D the maximum disparity displacement considered, while representative state-of-the-art deep stereo networks rely on data structures of dimension, respectively, $H \times W \times (2D+1)$ [19] and $H \times W \times D \times 2F$ [12], being F the number of features from a single image. Given sparse depth measurements z, we can easily convert them into disparities d by knowing the focal length f and baseline b of the setup used to acquire stereo pairs, as $d = b \cdot f \cdot \frac{1}{z}$.

With the availability of sparse external data in the disparity domain, we can exploit them to peak the correlation scores or the features activation related to the hypotheses suggested by these sparse hints and to dampen the remaining ones. For example, given a disparity value of k, we will enhance the k*-th* channel output of a correlation layer or the k*-th* slice of a 4D volume. For our purposes, we introduce two new inputs, both of size $H \times W$: a (sparse) matrix

G, conveying the externally provided disparity values, and a binary mask V, specifying which elements of G are valid (*i.e.*, if $v_{ij} = 1$). For each pixel with coordinates (i, j) in the reference image such that $v_{ij} = 1$ we alter features as discussed before, based on the known disparity value g_{ij}. On the other hand, every point with $v_{ij} = 0$ is left untouched. Thus, we rely on the ability of the deep network to reason about stereo and jointly leverage the additional information conveyed by sparse inputs.

In literature, some techniques were proposed to modify the cost volume of traditional stereo algorithms leveraging prior knowledge such as per-pixel confidence scores [25]. A simple, yet effective approach for this purpose consists in hard-replacing matching costs (features, in our case). In [34], matching costs of winning disparities were set to the minimum value and the remaining ones to the maximum, only for those pixels having high confidence score before optimization. The equivalent solution in our domain would consist in zeroing each element corresponding to a disparity d such that $g_{ij} \neq d$. However, this strategy has severe limitations: it is not well-suited for CNNs, either when injected into a pre-trained network – a large number of zero values would aggressively alter its behavior – or when plugged during the training from scratch of a new model – this would cause gradients to not be back-propagated on top of features where the zeros have been inserted. Moreover, no default behavior is defined in case of sub-pixel input disparities, unless they are rounded at the cost of a loss in precision.

Conversely, we suggest to modulate using a Gaussian function centred on g_{ij}, so that the single correlation score or $2F$ features corresponding to the disparity $d = g_{ij}$ are multiplied by the peak of the function, while any other element is progressively multiplied by lower factors, until being dampened the farther they are from g_{ij}. Specifically, our modulating function will be

$$k \cdot e^{-\frac{(d-g_{ij})^2}{2c^2}} \tag{1}$$

where c determines the width of the Gaussian, while k represents its maximum magnitude and should be greater than or equal to 1. Thus, to obtain a new feature volume $\mathcal{G}$ by multiplying the whole correlation or 3D volume $\mathcal{F}$ regardless of the value of v_{ij}, we apply

$$\mathcal{G} = \left(1 - v_{ij} + v_{ij} \cdot k \cdot e^{-\frac{(d-g_{ij})^2}{2c^2}}\right) \cdot \mathcal{F} \tag{2}$$

making the weight factor equal to 1 when $v_{ij} = 0$. An example of the effect of our modulation is given in Figure 3 (left).

3.2. Applications of guided stereo

We will now highlight some notable applications of our technique, that will be exhaustively discussed in the experimental result section.

Pre-trained deep stereo networks. The proposed Gaussian enhancement acts smoothly, yet effectively, on the features already learned by a deep network. Opportunely tuning the hyper-parameters k and c, we will prove that our method allows improving the accuracy of pre-trained state-of-the-art networks.

Training from scratch deep stereo networks. Compared to a brutal zero-product approach, the dampening mechanism introduced by the Gaussian function still allows gradient to flow, making this technique suited for deployment inside a CNN even at training time, so that it can learn from scratch how to better exploit the additional cues. Specifically, the gradients of $\mathcal{G}$ with respect to weights $\mathcal{W}$ will be computed as follows:

$$\frac{\partial \mathcal{G}}{\partial \mathcal{W}} = \left(1 - v_{ij} + v_{ij} \cdot k \cdot e^{-\frac{(d-g_{ij})^2}{2c^2}}\right) \cdot \frac{\partial \mathcal{F}}{\partial \mathcal{W}} \tag{3}$$

Thus, training from scratch a deep network leveraging the sparse input data is possible with our technique.

Conventional stereo matching algorithms. These methods, based on hand-crafted pipelines, can also take advantage of our proposal by leveraging sparse depth cues to improve their accuracy. Sometimes they do not use a similarity measure (*e.g.*, zero mean normalized cross-correlation) to encode the matching cost, for which the same strategy described so far applies, but cost volumes are built using a dissimilarity measure between pixels (*e.g.*, sum of absolute/squared differences or Hamming distance [43]). In both cases, the winning disparity is assigned employing a WTA strategy. When deploying a dissimilarity measure, costs corresponding to disparities close to g_{ij} should be reduced, while the others amplified. We can easily adapt Gaussian enhancement by choosing a modulating function that is the difference between a constant k and a Gaussian function with the same height, obtaining an enhanced volume $\mathcal{G}$ from initial costs $\mathcal{F}$ as

$$\mathcal{G} = \left[1 - v_{ij} + v_{ij} \cdot k \cdot \left(1 - e^{-\frac{(d-g_{ij})^2}{2c^2}}\right)\right] \cdot \mathcal{F} \tag{4}$$

Figure 3 (right) shows the effect of this formulation.

4. Experimental Results

In this section, we report exhaustive experiments proving the effectiveness of the Guided Stereo Matching paradigm showing that the proposed feature enhancement strategy always improves the accuracy of pre-trained or newly trained networks significantly. Moreover, when training the networks from scratch, our proposal increases the ability to generalize to new environments, thus enabling to better tackle domain shifts. Demo source code is available at https://github.com/mattpoggi/guided-stereo.

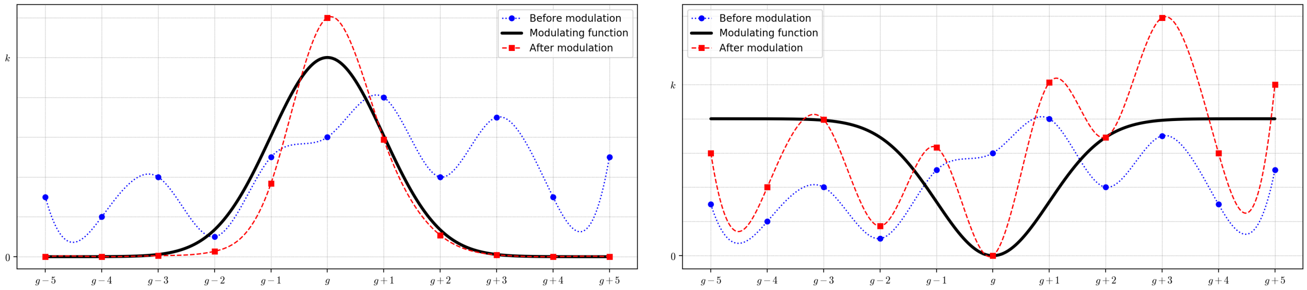

Figure 3. **Application of the proposed feature enhancement**. In blue, features $\mathcal{F}$ for pixel i, j in proximity of $d = g_{ij}$, in black the modulating function, in red enhanced features $\mathcal{G}$ for $v_{ij} = 1$, applied to correlation features (left) or dissimilarity functions (right).

	iResNet [14]			PSMNet [3]		
c	k=1	k=10	k=100	k=1	k=10	k=100
0.1	2.054	1.881	2.377	4.711	4.391	4.326
1	1.885	**1.338**	6.857	4.540	**3.900**	4.286
10	1.624	1.664	32.329	4.539	3.925	9.951

Table 1. **Tuning of Gaussian hyper-parameters k and c.** Experiments with iResNet (left) and PSMNet (right) trained on synthetic data and tested on KITTI 2015 (average errors without modulation: 1.863 and 4.716)

4.1. Training and validation protocols

We implemented our framework in PyTorch. For our experiments, we chose two state-of-the-art models representative of the two categories described so far and whose source code is available, respectively iResNet [14] for correlation-based architectures and PSMNet [3] for 3D CNNs. Both networks were pre-trained on synthetic data [19] following authors' instructions: 10 epochs for PSMNet [3] and 650k iterations for iResNet [14]. The only difference was the batch size of 3 used for PSMNet since it is the largest fitting in a single Titan Xp GPU used for this research. The proposed guided versions of these networks were trained accordingly following the same protocol. Fine-tuning on realistic datasets was carried out following the guidelines from the original works when available. In particular, both papers reported results and a detailed training protocol for KITTI datasets [3, 14], while training details are not provided for Middlebury [27] and ETH3D [29], despite results are present on both benchmarks. The following sections will report accurate details about each training protocol deployed in our experiments. To tune k and c, we ran a preliminary experiment applying our modulation on iResNet and PSMNet models trained on synthetic data and tested on KITTI 2015 training set [20]. Table 1 shows how the average error varies with different values of k and c. According to this outcome, for both networks we will fix k and c, respectively, to 10 and 1 for all the following experiments.

To simulate the availability of sparse depth cues, we randomly sample pixels from the ground-truth disparity maps for both training and testing. For this reason, all the evaluation will be carried out on the training splits available from KITTI, Middlebury and ETH3D datasets. Finally, we point out that the KITTI benchmarks include a depth completion evaluation. However, it aims at assessing the performance of monocular camera systems coupled with an active sensor (*i.e.*, LiDAR) and thus the benchmark does not provide the stereo pairs required for our purposes.

4.2. Evaluation on KITTI

At first, we assess the performance of our proposal on the KITTI 2015 dataset [20]. Table 2 collects results obtained with iResNet and PSMNet trained and tested in different configurations. For each experiment we highlight the imagery used during training, respectively the SceneFlow dataset alone [19] or the KITTI 2012 [7] used for fine-tuning ("*-ft*"). Moreover, we report results applying our feature enhancement to pre-trained networks (*i.e.*, at test time only, "*-gd*") and training the networks from scratch ("*-gd-tr*"). For each experiment, we report the error rate as the percentage of pixels having a disparity error larger than a threshold, varying between 2 and 5, as well as the mean average error on all pixels with available ground-truth. To obtain sparse measurements, we randomly sample pixels with a density, computed on the whole image, of 5% on SceneFlow [19]. On KITTI, we keep a density of 15%, then remove unlabelled pixels to obtain again 5% with respect to the lower portion of the images with available ground-truth (i.e., a 220 × 1240 pixel grid).

From Table 2 we can notice how both baseline architectures (row 1 and 7) yield large errors when trained on SceneFlow dataset only. In particular, PSMNet seems to suffer the domain shift more than correlation-based technique iResNet. By applying the proposed feature enhancement to both networks, we can ameliorate all metrics sensibly, obtaining first improvements to network generalization capability. In particular, by looking at the >3 error rate, usually taken as the reference metric in KITTI, we have an absolute reduction of about 3.8 and 4.3 % respectively for iResNet-*gd* and PSMNet-*gd* compared to the baseline networks. In this case, we point out once more that we are modifying only the features of a pre-trained network, by just altering what the following layers are used to process. Nonetheless, our

Model	Training Datasets		Guide		Error rate (%)				avg.
	SceneFlow	KITTI 12	Train	Test	>2	>3	>4	>5	(px)
iResNet [14]	✓				21.157	11.959	7.881	5.744	1.863
iResNet-*gd*	✓			✓	15.146	8.208	5.348	3.881	1.431
iResNet-*gd-tr*	✓		✓	✓	7.266	3.663	2.388	1.754	0.904
iResNet-*ft* [14]	✓	✓			9.795	4.452	2.730	1.938	1.049
iResNet-*ft-gd*	✓	✓		✓	7.695	3.812	2.524	1.891	0.994
iResNet-*ft-gd-tr*	✓	✓	✓	✓	**4.577**	**2.239**	**1.476**	**1.099**	**0.735**
PSMNet [3]	✓				39.505	27.435	20.844	16.725	4.716
PSMNet-*gd*	✓			✓	33.386	23.125	17.598	14.101	3.900
PSMNet-*gd-tr*	✓		✓	✓	12.310	3.896	2.239	1.608	1.395
PSMNet-*ft* [3]	✓	✓			6.341	3.122	2.181	1.752	1.200
PSMNet-*ft-gd*	✓	✓		✓	5.707	3.098	2.266	1.842	1.092
PSMNet-*ft-gd-tr*	✓	✓	✓	✓	**2.738**	**1.829**	**1.513**	**1.338**	**0.763**

Table 2. **Experimental results on KITTI 2015 dataset [20].** Tag "*-gd*" refers to guiding the network only at test time, "*-tr*" to training the model to leverage guide, "*-ft*" refers to fine-tuning performed on KITTI 2012 [7].

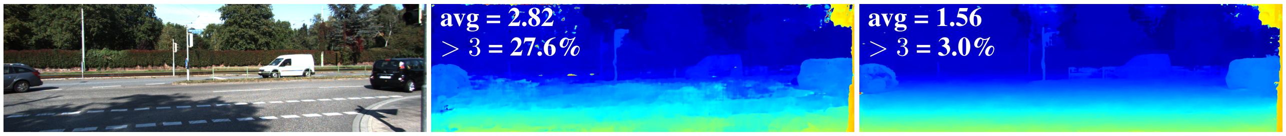

Figure 4. **Comparison between variants of PSMNet [3].** From top to bottom, reference image from 000022 pair (KITTI 2015 [20]), disparity maps obtained by PSMNet [3] and PSMNet-*gd-tr*, both trained on synthetic images only.

proposal preserves the learned behavior of the baseline architecture increasing at the same time its overall accuracy.

When training the networks from scratch to process sparse inputs with our technique, iResNet-*gd-tr* and PSMNet-*gd-tr* achieve a notable drop regarding error rate and average error compared to the baseline models. Both reach degrees of accuracy comparable to those of the original network fine-tuned on KITTI 2012, iResNet-*ft* and PSMNet-*ft* without actually being trained on such realistic imagery, by simply exploiting a small amount of depth samples (about 5%) through our technique. Moreover, we can also apply the feature enhancement paradigm to fine-tuned models. From Table 2 we can notice again how our technique applied to the fine-tuned models still improves their accuracy. Nonetheless, fine-tuning the networks pre-trained to exploit feature enhancement leads to the best results across all configurations, with an absolute decrease of about 2.2 and 1.3% compared, respectively, to the already low error rate of iResNet-*ft* and PSMNet-*ft*. Finally, Figure 4 shows a comparison between the outputs of different PSMNet variants, highlighting the superior generalization capacity of PSMNet-*gd-tr* compared to baseline model.

4.3. Evaluation on Middlebury

We also evaluated our proposal on Middlebury v3 [27], since this dataset is notoriously more challenging for end-to-end architectures because of the very few images available for fine-tuning and the more heterogeneous scenes framed compared to KITTI. Table 3 collects the outcome of these experiments. We use the same notation adopted for KITTI experiments. All numbers are obtained processing the *additional* split of images at quarter resolution, since higher-resolution stereo pairs do not fit into the memory of a single Titan Xp GPU. Fine-tuning is carried out on the *training* split. We compute error rates with thresholds 0.5, 1, 2 and 4 as usually reported on the online benchmark. Sparse inputs are randomly sampled with a density of 5% from ground-truth data. We can notice how applying feature enhancement on both pre-trained models or training new instances from scratch gradually reduces the errors as observed on KITTI experiments. Interestingly, we point out that while this trend is consistent for iResNet-*gd* and iResNet-*gd-tr*, a different behavior occurs for PSMNet-*gd-tr*. In particular, we can notice a huge reduction of the error rate setting the threshold to > 2 and > 4. On the other hand, the percentage of pixels with lower disparity errors > 0.5 and > 1 gets much higher. Thus, with PSMNet, the architecture trained with guiding inputs seems to correct most erroneous patterns at the cost of increasing the number of small errors. Nonetheless, the average error always decreases significantly.

Regarding fine-tuning, we ran about 300 epochs for each baseline architecture to obtain the best results. After this phase, we can observe minor improvements for iResNet, while PSMNet improves its accuracy by a large margin. Minor, but consistent improvements are yielded by iResNet-*ft-gd* and PSMNet-*ft-gd*. Finally, we fine-tuned iResNet-*ft-gd-tr* and PSMNet-*ft-gd-tr* for about 30 epochs, sufficient to reach the best performance. Again, compared to all other configurations, major improvements are always yielded by

Model	Training Datasets		Guide		Error rate (%)				avg.
	SceneFlow	trainingQ	Train	Test	>0.5	>1	>2	>4	(px)
iResNet [14]	✓				69.967	50.893	30.742	16.019	2.816
iResNet-*gd*	✓			✓	62.581	40.831	22.154	10.889	2.211
iResNet-*gd-tr*	✓		✓	✓	44.385	25.555	12.505	5.776	1.470
iResNet-*ft* [14]	✓	✓			69.526	49.027	28.178	14.126	2.682
iResNet-*ft-gd*	✓	✓		✓	60.979	36.255	19.558	10.136	2.130
iResNet-*ft-gd-tr*	✓	✓	✓	✓	**31.526**	**17.045**	**8.316**	**4.307**	**0.930**
PSMNet [3]	✓				54.717	33.603	20.239	13.304	5.332
PSMNet-*gd*	✓			✓	53.090	31.416	18.619	12.588	4.921
PSMNet-*gd-tr*	✓		✓	✓	83.433	54.147	7.472	3.208	1.732
PSMNet-*ft* [3]	✓	✓			45.523	25.993	15.203	8.884	1.964
PSMNet-*ft-gd*	✓	✓		✓	44.004	25.151	14.337	8.676	1.894
PSMNet-*ft-gd-tr*	✓	✓	✓	✓	**32.715**	**15.724**	**6.937**	**3.756**	**1.348**

Table 3. **Experimental results on Middlebury v3 [27].** "-*gd*" refers to guiding the network only at test time, "-*tr*" to training the model to leverage guide, "-*ft*" refers to fine-tuning performed on *trainingQ* split.

Model	Training Datasets		Guide		Error rate (%)				avg.
	SceneFlow	ETH3D	Train	Test	>0.5	>1	>2	>4	(px)
iResNet [14]	✓				57.011	36.944	20.380	12.453	5.120
iResNet-*gd*	✓			✓	50.361	29.767	16.495	10.293	2.717
iResNet-*gd-tr*	✓		✓	✓	26.815	10.673	3.555	1.312	0.537
iResNet-*ft* [14]	✓	✓			48.360	26.212	11.865	4.678	0.997
iResNet-*ft-gd*	✓	✓		✓	47.539	22.639	8.153	2.445	0.820
iResNet-*ft-gd-tr*	✓	✓	✓	✓	**23.433**	**8.694**	**2.803**	**0.876**	**0.443**
PSMNet [3]	✓				45.522	23.936	12.550	7.811	5.078
PSMNet-*gd*	✓			✓	43.667	21.140	10.773	7.081	4.739
PSMNet-*gd-tr*	✓		✓	✓	96.976	71.970	2.730	0.512	1.266
PSMNet-*ft* [3]	✓	✓			28.560	11.895	4.272	1.560	0.560
PSMNet-*ft-gd*	✓	✓		✓	25.707	10.095	3.084	1.123	0.505
PSMNet-*ft-gd-tr*	✓	✓	✓	✓	**17.865**	**4.195**	**1.360**	**0.817**	**0.406**

Table 4. **Experimental results on ETH3D dataset [29].** "-*gd*" refers to guiding the network only at test time, "-*tr*" to training the model to leverage guide, "-*ft*" refers to fine-tuning performed on the training split (see text).

guiding both networks.

4.4. Evaluation on ETH3D

Finally, we assess the performance of our method on the ETH3D dataset [29]. In this case we split the training dataset, using images from *delivery_area_1l*, *delivery_area_1s*, *electro_1l*, *electro_1s*, *facade_1s*, *forest_1s*, *playground_1l*, *playground_1s*, *terrace_1s*, *terrains_1l*, *terrains_1s* for fine-tuning and the remaining ones for testing. For -*ft* models, we perform the same number of training epochs as for Middlebury dataset, and Table 4 collects results for the same configurations considered before. We can notice behaviors similar to what reported in previous experiments. By guiding iResNet we achieve major improvements, nearly halving the average error, while the gain is less evident for PSMNet, although beneficial on all metrics. Training iResNet-*gd-tr* and PSMNet-*gd-tr* leads to the same outcome noticed during our experiments on Middlebury. In particular, PSMNet-*gd-tr* still decimates the average error and percentage of errors greater than 2 and 4 at the cost of a large amount of pixels with error greater than 0.5 and 1. With this dataset, fine-tuning the baseline models enables to significantly increase the accuracy of both, in particular decimating the average error from beyond 5 pixels to less than 1. Nevertheless, our technique is useful even in this case, enabling minor yet consistent improvements when used at test time only and significant boosts for iResNet-*ft-gd-tr* and PSMNet-*ft-gd-tr*, improving all metrics by a large margin.

4.5. Evaluation with SGM

To prove the effectiveness of our proposal even with conventional stereo matching algorithms, we evaluated it with SGM [10]. For this purpose, we used the code provided by [33], testing it on all datasets considered so far. As pointed out in Section 3.2, feature enhancement can be opportunely revised as described in Equation 4 to deal with dissimilarity measures. We apply this formulation before starting scanline optimizations. As for any previous experiments, we sample sparse inputs as described in Section 4, obtaining an average density below 5%. Table 5 reports the comparison between SGM and its *cost enhanced* counterpart using sparse input cues (SGM-*gd*) on KITTI 2012 [7] (top) and KITTI 2015 [20] (bottom). With both datasets we can notice dramatic improvements in all metrics. In particular, the

Alg.	Error rate (%)				avg.
	>2	>3	>4	>5	(px)
SGM [10]	11.845	8.553	7.109	6.261	2.740
SGM-*gd*	**5.657**	**4.601**	**4.162**	**3.892**	**2.153**
SGM [10]	15.049	8.843	6.725	5.645	2.226
SGM-*gd*	**6.753**	**4.294**	**3.625**	**3.282**	**1.680**

Table 5. **Experimental results on KITTI.** Comparison between raw [10] and guided SGM on KITTI 2012 (top) and 2015 (bottom).

Alg.	Error rate (%)				avg.
	>0.5	>1	>2	>4	(px)
SGM [10]	62.428	32.849	20.620	15.786	4.018
SGM-*gd*	**56.882**	**24.608**	**12.655**	**9.909**	**2.975**
SGM [10]	64.264	31.966	18.741	14.675	4.978
SGM-*gd*	**59.596**	**24.856**	**11.307**	**8.960**	**3.815**
SGM [10]	58.994	27.356	10.685	5.632	1.433
SGM-*gd*	**54.051**	**20.156**	**4.169**	**2.459**	**1.032**

Table 6. **Experimental results on Middlebury v3 and ETH3D datasets.** Comparison between raw [10] and guided SGM on *training* (top), *additional* (middle) [27] and ETH3D [29] (bottom).

amount of outliers > 2 is more than halved and reduced in absolute by about 4, 3 and 2% for higher error bounds. Table 6 reports experiments on Middlebury *training* (top) and *additional* (bottom) splits [27], as well as on the entire ETH3D training set [29]. Experiments on Middlebury are carried out at quarter resolution for uniformity with previous experiments with deep networks reported in Section 4.3. The error rate is reduced by about 5.5% for > 0.5 on the three experiments, by about 7.5% for > 1, nearly halved on Middlebury and reduced by a factor 2.5 on ETH3D for > 2, and by 6, 6 and 3% for > 4. Finally, average errors are reduced by 1.1 on Middlebury and 1.4 on ETH3D.

The evaluation with SGM highlights how our technique can be regarded as a general purpose strategy enabling notable improvements in different contexts, ranging from state-of-the-art deep learning frameworks to traditional stereo algorithms.

4.6. Experiments with Lidar measurements

Finally, we evaluate the proposed paradigm using as guide the raw and noisy measurements from a Velodyne sensor [40], to underline the practical deployability of the proposed solution further. Table 7 reports experiment from sequence 2011_09_26_0011 of the KITTI raw dataset [6]. We compare our framework with fusion strategies proposed by Martins *et al.* [18] and Marin *et al.* [17], combining outputs by the stereo networks respectively with monocular estimates (using the network by Guo *et al.* [9]) and Lidar, reporting the ideal result as in [17]. Ground-truth labels for evaluation are provided by [39]. Our proposal consistently outperforms fusion approaches by a large margin, evaluating on all pixels (All) as well as excluding those with Lidar (NoG) to stress that the improvement yielded by our method is not limited to pixels with associated Lidar measurement in contrast to fusion techniques [17].

Model / Alg.	<2%		avg.	
	All	NoG	All	NoG
iResNet [14]	18.42	18.37	1.28	1.28
iResNet+Martins *et al.* [18]	18.14	18.09	1.26	1.26
iResNet+Marin *et al.* (opt.)	15.20	18.37	1.07	1.28
iResNet-*gd*	11.12	10.99	1.04	1.03
iResNet-*gd-tr*	**5.38**	**5.27**	**0.77**	**0.77**
iResNet-*ft* [14]	5.29	5.30	0.81	0.81
iResNet-*ft*+Martins *et al.* [18]	5.26	5.28	0.80	0.80
iResNet-*ft*+Marin *et al.* [17] (opt.)	4.48	5.30	0.67	0.81
iResNet-*ft-gd*	3.14	3.13	0.64	0.64
iResNet-*ft-gd-tr*	**1.91**	**1.88**	**0.55**	**0.55**
PSMNet [3]	38.60	38.86	2.36	2.37
PSMNet+Martins *et al.* [18]	38.32	38.58	2.33	2.34
PSMNet+Marin *et al.* [17] (opt.)	34.85	38.86	1.99	2.17
PSMNet-*gd*	33.47	33.74	2.07	2.08
PSMNet-*gd-tr*	**21.57**	**21.30**	**1.60**	**1.59**
PSMNet-ft [3]	1.71	1.73	0.72	0.72
PSMNet-*ft*+Martins *et al.* [18]	1.82	1.83	0.72	0.72
PSMNet-*ft*+Marin *et al.* [17] (opt.)	1.52	1.73	0.66	0.72
PSMNet-*ft-gd*	1.13	1.15	0.60	0.61
PSMNet-*ft-gd-tr*	**0.67**	**0.67**	**0.47**	**0.47**
SGM [10]	9.42	9.54	1.24	1.24
SGM+Martins *et al.* [18]	9.41	9.53	1.24	1.24
SGM+Marin *et al.* [17] (opt.)	8.15	9.54	1.14	1.24
SGM-*gd*	**2.79**	**3.03**	**0.99**	**0.99**

Table 7. Experiments on KITTI Velodyne, seq. *2011_09_26_0011*.

5. Conclusions

In this paper, we proposed Guided Stereo Matching, a novel paradigm to boost state-of-the-art deep architectures trained for dense disparity inference using as additional input cue a small set of sparse depth measurements provided by an external source. By enhancing the features that encode matching relationships between pixels across left and right images, we can improve the accuracy and robustness to domain shifts. Our feature enhancement strategy can be used seamlessly with pre-trained models, yielding significant accuracy improvements. More importantly, thanks to its fully-differentiable nature, it can even be used to train new instances of a CNN from scratch, in order to fully exploit the input guide and thus to remarkably improve overall accuracy and robustness to domain shifts of deep networks. Finally, our proposal can be deployed even with conventional stereo matching algorithms such as SGM, yielding significant improvements as well. The focus of future work will be on devising strategies to guide our method without relying on active sensors. For instance, selecting reliable depth labels leveraging confidence measures [25] – since this strategy proved to be successful for self-supervised adaptation [36, 37] and training learning-based confidence measures [38] – or from the output of a visual stereo odometry systems [41].

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

References

[1] Gianluca Agresti, Ludovico Minto, Giulio Marin, and Pietro Zanuttigh. Deep learning for confidence information in stereo and tof data fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Workshops)*, pages 697–705, 2017. 2

[2] Konstantinos Batsos and Philipos Mordohai. Recresnet: A recurrent residual cnn architecture for disparity map enhancement. In *In International Conference on 3D Vision (3DV)*, 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. 2, 3, 5, 6, 7, 8

[4] Zhuoyuan Chen, Xun Sun, Liang Wang, Yinan Yu, and Chang Huang. A deep visual correspondence embedding model for stereo matching costs. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[5] Carlo Dal Mutto, Pietro Zanuttigh, and Guido Maria Cortelazzo. Probabilistic tof and stereo data fusion based on mixed pixels measurement models. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2260–2272, 2015. 2

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 8

[7] 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, 2, 5, 6, 7

[8] Spyros Gidaris and Nikos Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[9] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018. 8

[10] 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. 2, 7, 8

[11] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for optical flow, disparity, or scene flow estimation. In *15th European Conference on Computer Vision (ECCV)*, 2018. 3

[12] 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. 3

[13] 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. 3

[14] 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. 1, 2, 3, 5, 6, 7, 8

[15] Yu Lidong, Wang Yucheng, Yuwei Wu, and Yunde Jia. Deep stereo matching with explicit cost aggregation subarchitecture. In *32th AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3

[16] 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

[17] Giulio Marin, Pietro Zanuttigh, and Stefano Mattoccia. Reliable fusion of tof and stereo depth driven by confidence measures. In *14th European Conference on Computer Vision (ECCV 2016)*, pages 386–401, 2016. 2, 8

[18] Diogo Martins, Kevin Van Hecke, and Guido De Croon. Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 849–856. IEEE, 2018. 8

[19] 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. 1, 2, 3, 5

[20] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 5, 6, 7

[21] Rahul Nair, Kai Ruhl, Frank Lenzen, Stephan Meister, Henrik Schäfer, Christoph S Garbe, Martin Eisemann, Marcus Magnor, and Daniel Kondermann. A survey on time-of-flight stereo fusion. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 105–127. Springer, 2013. 2

[22] 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. 3

[23] Min Gyu Park and Kuk Jin Yoon. Leveraging stereo matching with learning-based confidence measures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[24] Matteo Poggi and Stefano 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. 2

[25] 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. 2, 4, 8

[26] Quanergy System, Inc. S3-8: The world's first affordable solid state lidar sensor. https://quanergy.com/s3/. Accessed: 2018-11-12. 2

[27] Daniel Scharstein, Heiko Hirschmller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *GCPR*, volume 8753 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2014. 1, 2, 3, 5, 6, 7, 8

[28] 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. 1, 2

[29] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5, 7, 8

[30] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016. 2

[31] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[32] 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. 3

[33] Robert Spangenberg, Tobias Langner, Sven Adfeldt, and Ral Rojas. Large scale semi-global matching on the cpu. In *IV*, 2014. 7

[34] Aristotle Spyropoulos, Nikos Komodakis, and Philippos 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)*, pages 1621–1628. IEEE, 2014. 2, 4

[35] Tatsunori Taniai, Yasuyuki Matsushita, Yoichi Sato, and Takeshi Naemura. Continuous 3d label stereo matching using local expansion moves. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2725–2739, 2018. 1

[36] 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. 8

[37] 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. 8

[38] 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. 8

[39] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 8

[40] Velodyne LIDAR, Inc. Velodyne lidar. https://velodynelidar.com/. Accessed: 2019-04-3. 8

[41] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 8

[42] 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. 3

[43] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, ECCV '94, pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. 4

[44] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015. 2