Meta-confidence estimation for stereo matching

Seungryong Kim*[1], Matteo Poggi*[2], Sunok Kim[3], Kwanghoon Sohn[4], Stefano Mattoccia[2]

***Abstract*— We propose a novel framework to estimate the confidence of a disparity map taking into account, for the first time, the uncertainty affecting the confidence estimation process itself. Conversely to other tasks such as disparity estimation, the uncertainty of confidence directly hints that the confidence should be increased if initially low, but with high uncertainty, decreased otherwise. By modelling such a cue in the form of a second-level confidence, or meta-confidence, our solution allows for finding incorrect predictions inferred by confidence estimator and for learning a correction for them. Our strategy is suited for any state-of-the-art method known in literature, either implemented using random forest classifiers or deep neural networks. Especially, for deep neural networks-based models, we present a multi-headed confidence estimator followed by an uncertainty network, so as to predict mean confidence and meta-confidence within a single network without the cost of lower accuracy, a known limitation in literature for uncertainty estimation. Experimental results on a variety of stereo algorithms and confidence estimation models prove that the modeled meta-confidence is meaningful of the reliability of the estimated confidence and allows for refining it.**

I. Introduction

Estimating depth from images often is the first step allowing autonomous agents and robots to understand the surrounding environment, and stereo matching [1], [2] is one of the most popular approaches to achieve this goal. It allows to retrieve the depth of any 3D point when captured by two synchronized and calibrated cameras, specifically by looking at the horizontal displacement (*disparity*) occurring between its pixel coordinates on the two images. A vast literature of algorithms exists [1], aimed at solving the correspondence problem between pixels on the *reference* and the *target* views, which are conventionally the left and right images respectively. More recently, it was enriched by the advent of deep learning [2] and a variety of solutions either mixing neural networks and hand-designed strategies [3] or deploying end-to-end neural networks [4], [5], [6].

Concurrent to the race for depth accuracy, confidence estimation [7], [8] has been developed as a parallel research track in the stereo matching literature, allowing for both improvement to known stereo algorithms [9], [10], [11], [12], [13] as well as for additional, higher-level applications [14], [15], [16], [17]. Indeed, being aware of failures of the depth perception pipeline represents an appealing capability for an autonomous system. As for stereo algorithms, machine learning and recent deep learning brought the focus from hand-crafted confidence measures [7] to data-driven strategies [8].

* Joint first authorship
[1] Korea University [2] University of Bologna
[3] Korea Aerospace University [4] Yonsei University

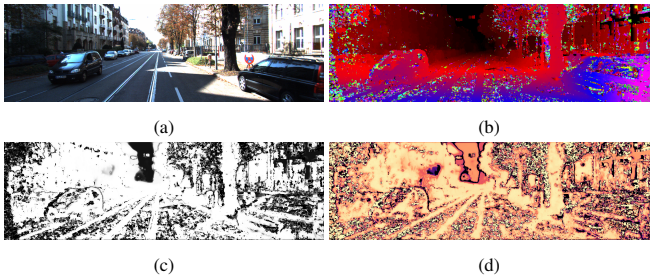


Fig. 1. **Confidence and meta-confidence estimation.** Given (a) a reference image and (b) its estimated disparity (*e.g.*, by MCCNN-fst-CBCA [3]), we estimate (c) confidence by means of a neural network or a random forest together with (d) meta-confidence encoding confidence reliability. Bright colors in (c,d) encode high confidence/meta-confidence.

Although achieving outstanding performance, learned models and their confidence predictions are intrinsically affected by some *uncertainty* [18], either linked to observed data or the model itself. This uncertainty might explain wrong confidence predictions and thus, potentially, could help to correct them. Indeed, conversely to other tasks such as disparity estimation itself for instance, for which knowing the low confidence of a pixel does not give hints about its correct disparity, by knowing that a confidence score has high uncertainty would directly hint that in a manner score should be increased if initially low, decreased otherwise.

We can see such uncertainty as a second-level confidence or *meta-confidence*. Since ignored in the literature until now, in this paper we study this aspect, we show how existing deep learning models for confidence estimation can be extended to take into account the meta-confidence and to exploit this information to predict more reliable confidence score, thus improving capability of an autonomous system to detect failures of the stereo matching module. Existing strategies to model the uncertainty of deep neural networks [18], [19], [20], however, are ineffective for less complex tasks such as confidence estimation. To improve the performance, we present a multi-headed confidence estimator followed by an uncertainty network, so as to predict mean confidence and meta-confidence within a single network without the cost of lower accuracy, a known limitation in literature [18].

We also show how existing random forest strategies already allow to retrieve this information for free and how, similarly to the case of deep networks, it can be used to refine the predicted confidence. In Figure 1, we show an example of confidence and meta-confidence, this latter encoding the complementary of the uncertainty over the first.

The main contributions can be resumed as follows:

- We consider for the first time the problem of modelling meta-confidence in state-of-the-art confidence estimation frameworks for stereo.
- We show how, if properly modeled, meta-confidence is meaningful to detect incorrect confidence predictions and how it can be used to refine and correct them.
- We show how existing methods based on random forests already expose meta-confidence in a similar manner and how it can be used to refine their predictions as well.

II. Related work

We briefly review the literature relevant to our work.

Confidence measures for stereo matching. Several confidence measures have been proposed in the years, reviewed and evaluated for the first time in [7]. With the advent of machine learning first, followed by deep learning, the confidence estimation task shifted towards data-driven approaches [21] thanks to the availability of stereo datasets annotated with ground-truth disparity labels. As a consequence, confidence measures can now be classified into two broad categories [8], [22], respectively *hand-made* and *learned* measures, with the latter being consistently more effective at distinguishing good from bad disparities compared to previous strategies.

The first attempts to design learned confidence measures consist into combining several hand-made cues (or measures) as input to a random forest classifier [21], being the input features extracted from the cost volume [23], [9], [10], [24] or the disparity map [11], [12]. Then, deep learning solutions have been proposed [25], [26], [27], [28], [29] deploying small CNNs processing patches from the disparity map and reference image, then moving to larger receptive fields [30] and accessing the cost volume as well [31], [32], [33], [34].

In parallel, some works explored the possibility of training state-of-the-art confidence estimators without the need for ground-truth disparity labels, for instance by collecting stereo videos in static environments [35], by distilling proxy labels from a pool of hand-made measures [36] or from few supervisory signals extracted from the disparity map and input stereo pair [37] allowing for online adaptation.

Uncertainty estimation in deep networks. Finding a way to estimate the uncertainty over the predictions of neural networks has a long history as well [38], [39], [40]. The main goal of most strategies is to regress a *distribution* in place of a single output. This can be approximated in an empirical manner, by sampling a finite number of weights configuration for a given network [41], [42], [43], [44], [45], and then computing mean and variance of the predictions. Another strategy consists into learning output distributions in a predictive manner, for instance by modelling a Laplacian or Gaussian output distribution [18]. The two strategies model two different type of uncertainties, *epistemic* and *aleatoric*, connected respectively to the model itself or to different input data. Recent works combined the aforementioned strategies for several tasks, such as optical flow [46], self-supervised monocular depth estimation [20] or joint depth, semantic and instance segmentation estimation [47]. In particular, Ilg *et al.*[46] proposed a multi-hypotheses model, predicting multiple outputs and their uncertainties fed to a refinement network inferring a final, more accurate result.

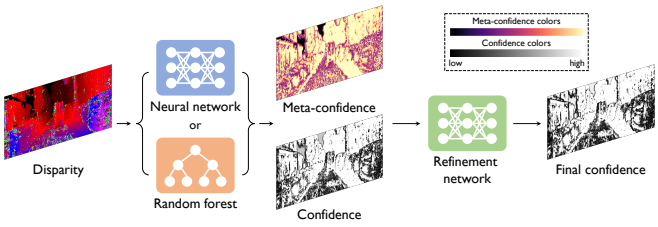


Fig. 2. **Overview of meta-confidence estimation framework.** Given a disparity map, we estimate its confidence with a neural network or random forest and wrong confidence prediction by means of a meta-confidence map. We then predict more reliable confidence with a refinement network.

Being the performance of state-of-the-art confidence estimators still sub-optimal, inspired by the literature reviewed so far we focus on modelling their uncertainty, in the form of meta-confidence, to further shrink the gap with optimal confidence prediction.

III. Meta-confidence framework

In this section, we introduce our framework based on meta-confidence estimation, used to find wrong confidence predictions estimated by deep neural networks or random forest classifiers and then correct them. Figure 2 gives an overview of our pipeline: while estimating the confidence for an input disparity map by means of a learned model, we also infer the uncertainty over such estimated confidence, *i.e.*, its meta-confidence, and then refine it by means of an additional neural network.

A. Confidence estimation

Estimating the confidence of each pixel in a disparity map $d = \mathcal{S}(l, r)$, output of a stereo algorithm $\mathcal{S}$ processing two images (l, r), consists into assigning a score encoding the reliability of such disparity hypothesis, meaning the higher the more reliable. In most cases, the disparity is estimated by computation and optimization of a cost-volume C, storing matching costs $C(l, r, i)$ between each pixel in l and a number of candidates on r shifted by i along the horizontal scanline, with $i \in [0, d_{\max}]$. For each pixel, the disparity is assigned by selecting the minimum cost $d = \text{argmin}_i\, C(l, r, i)$. To estimate a confidence map c, a generic function ψ taking as input (a subset of) the cost volume C, the disparity map d and the input images (l, r) is defined as $c = \psi(l, r, d, C)$. According to recent literature, a specific ψ function can be *learned* from data and implemented in the form of a neural network or a random forest [8]. In such a case, the inferred confidence c will be function of (a subset of) the aforementioned cues, as well as the set of parameters Φ_c of the network or forest such that $c = \psi(l, r, d, C; \Phi_c)$. This solution proved to be particularly effective with respect to hand-made functions [8]. Nevertheless, we argue that any estimator trained to estimate confidence of its prediction is

intrinsically affected by some *uncertainty*, that can be caused either by the current observation or the learned model itself.

B. Meta-confidence as model uncertainty

Estimating the uncertainty of confidence offers the possibility to correct the wrong confidence predictions. In general, the confidence is learned by regression or classification models [8], and thus there may exist numerous techniques that estimate the uncertainty of the models themselves, as exemplified in [46], [20] for other tasks such as optical flow and monocular depth estimation. In our case, we aim at estimating the uncertainty concerning the predicted confidence. We refer to this quantity as meta-confidence, since it plays the role of *confidence measure for a confidence measure.* In this section, we show it can be simultaneously learned with the state-of-the-art deep networks-based models and directly found for free in existing random forest strategies.

Deep networks – multiple hypotheses. According to the literature, several strategies exist to model the uncertainty of deep neural networks [18], [46], [20]. Most of them follow two main approaches, respectively *empirical* or *predictive*. The *empirical* approach is perhaps the simplest, usually implemented by training multiple different models independently, for instance by means of dropout [44], boostrapped ensembles [43] or SGDR [45], such that the mean and the variance of the distribution can be approximated with empirical mean and variance of individual model's predictions [43], [48], [49], but this approach is computationally infeasible due to multiple forward sampling or ensembles at test time. According to the *predictive* approach, we can train a predictive model to output the parameters of a parametric model of the distribution [18], which can be optimized by minimizing their negative log-likelihood. Although effective at modelling uncertainty, this often comes at the price of a lower accuracy of the network predictions [18], [19].

To alleviate these limitations, we propose an alternative way, which benefits from the advantages of two solutions discussed above. We formulate a *multi-headed* network for confidence estimation that yields multiple hypotheses in a single network. Specifically, we reformulate the confidence estimator to generate N multiple hypotheses, $\{c_1, ..., c_N\}$. To train them, existing multiple hypotheses frameworks [46], [49] optimize each hypothesis separately, while the final loss is obtained through a per-pixel minimum across all hypotheses, called winner-takes-all (WTA). But we observed that it is ineffective in our case, because the multiple hypotheses converge to the same prediction for less complex tasks such as confidence estimation and thus do not allow to model a distribution. To overcome this, we instead estimate the mean $\mu(c)$ of multiple hypotheses to jointly train them, defined as

$$\mu(c) = \frac{1}{N} \sum_{i=1}^{N} c_i. \tag{1}$$

The confidence estimation network is then trained by negative log-likelihood minimization [18], resulting in the following loss function

$$\mathcal{L}_{\text{llm}} = \frac{D\left(\mu(c), c^*\right)}{\sigma(c)} + \log \sigma(c), \tag{2}$$

where $D\left(\mu(c), c^*\right)$ is a distance function between $\mu(c)$ and ground-truth confidence c^*, which can be mean squared error (MSE) loss or binary cross entropy (BCE) loss, which is also widely used as a loss function in existing confidence measures [26], [30], [31]. The logarithmic term discourages infinite predictions for uncertainty, that would easily drive the loss toward zero. Regarding numerical stability [18], [20], the network is trained to estimated the log-variance in order to avoid zero values at the denominator. This loss function encourages the networks to make the confidence $\mu(c)$ close to the ground-truth c^* while $\sigma(c)$ can be considered as uncertainty of $\mu(c)$. We will see in our experiments that estimating $\mu(c)$ rather than a single confidence output leads to better results.

The uncertainty $\sigma(c)$ can be directly computed as a variance of multiple hypotheses, as in existing *empirical* approach [43], [48], [49], but we argue that it can be more accurately predicted by an additional neural network, called *uncertainty* network, of parameters Φ_u, that takes multiple hypotheses as input and output $\sigma(c)$. To summarize, our networks predict confidence $\mu(c)$ and its uncertainty $\sigma(c)$ simultaneously such that $\{\mu(c), \sigma(c)\} = \psi(l, r, d, C; \Phi_c, \Phi_u)$. The additional network consists of two sequential convolution modules, where each convolution module follows the architecture 3×3 Convolution-BatchNorm-ReLu producing 64 feature maps, followed by 1×1 Convolution-Sigmoid. For multiple hypotheses as input, we apply the softmax normalization to handle a scale change.

In addition, for multiple hypotheses, we should choose the size N. In the experiments, we set the size $N = 8$, considering the trade-off between complexity and accuracy.

Random forests – empirical variance. Since the first attempts to learn a confidence measure were built upon random forests [50], we also show how to obtain meta-confidence from these models. In particular, this can be carried out for free according to their definition.

A random forest consists into a number N of independent decision trees $\mathcal{T}_n$, with $n \in [1, N]$, each one casting an individual vote $\mathcal{T}_n(\nu)$ for any given feature vector ν. Once trained, the forest acts as an ensemble of trees and thus the final prediction is obtained, in case of a regression problem as ours, as the mean of the predictions of the trees

$$\mu(\nu) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{T}_n(\nu). \tag{3}$$

Similarly to what done for bootstrapped ensembles of neural networks [46], [20], we can compute the empirical variance over the trees predictions

$$\sigma(\nu) = \frac{1}{N} \sum_{n=1}^{N} \left(\mathcal{T}_n(\nu) - \mu(\nu)\right)^2. \tag{4}$$

We will see how this uncertainty, although obtained in an empirical manner, can be seamlessly used in our framework to refine the initial confidence.

C. Confidence refinement with meta-confidence

Meta-confidence directly hints that confidence should be increased if initially low, but with high uncertainty, decreased otherwise. Similarly to [27], we formulate a module as a deep neural network, called *refinement* network, of parameters Φ_r to enforce a local consistency of confidence as well as to refine uncertain confidences. The networks consist of three sequential convolution modules, where each convolution module follows the architecture 3×3 Convolution-BatchNorm-ReLu producing 64 feature maps, followed by 1×1 Convolution-Sigmoid. We train this network to generate a final confidence f as $f = \psi(\mu(c), \sigma(c); \Phi_r)$. To train the network, a refinement loss $\mathcal{L}_{\text{ref}}$ is computed as $\mathcal{L}_{\text{ref}} = D(f, c^*)$. We will show how, without taking the uncertainty in input as in [27], the performance gain achieved by f with respect to c may be marginal.

IV. Experimental results

In this section, we report exhaustive evaluations to assess the effectiveness of meta-confidence estimation.

A. Evaluation protocol

Confidence evaluation. Analysis of the Area Under Curve (AUC) [7], [8], [30], [31] represents the standard approach to evaluate confidence estimators. Pixels in a given disparity map are sorted in decreasing order of confidence and gradually sampled (*e.g.*, 5% each time). For each sample, the error rate is computed as the percentage of pixels having absolute error larger than τ. A ROC curve is obtained by plotting the error rate at any sampling, whose AUC quantitatively assesses the confidence effectiveness (the lower, the better). Optimal AUC [7] is obtained as a function of the error rate ε on the whole disparity map:

$$\text{AUC}_{opt} = \int_{1-\varepsilon}^{\varepsilon} \frac{p - (1-\varepsilon)}{p} dp = \varepsilon + (1-\varepsilon)\ln(1-\varepsilon). \quad (5)$$

As in [22], we report AUCs $\times 10^2$ to improve readability.

Refined confidence evaluation. To measure the improvement yielded by our framework based on meta-confidence, we compute the Δ_{AUC} metric proposed in [27] as

$$\Delta_{\text{AUC}} = \frac{\text{AUC} - \text{AUC}_f}{\text{AUC} - \text{AUC}_{opt}} \quad (6)$$

with AUC and AUC_f, respectively, AUC by the baseline and meta-confidence framework averaged over the entire dataset. We report this score as percentage (%).

B. Implementation details

Confidence networks. Our meta-confidence estimation framework can be incorporated into any confidence estimator involving deep neural networks-based models. We consider five architectures as backbone networks: CCNN [26], ConfNet [30], LGC [30], UCN [33], and LAF [31] because they are clearly the state-of-the-art in literature [8] and the source code is fully available. As in [37], we modified ConfNet to replace deconvolutions with a bilinear upsampling followed by convolutions and process a disparity only. We trained those backbone networks and proposed *uncertainty* and *refinement* networks in an end-to-end manner, with the proposed loss functions, $\mathcal{L}_{\text{llm}}$ and $\mathcal{L}_{\text{ref}}$. For CCNN, we used batches of 128 patches, for ConfNet, UCN, and LAF, batches of 4, 128×256 crops on inputs, and for LGC, batches of 128 patches. We trained all networks with ADAM optimizer and a constant learning rate of 0.003.

			ConfNet [30]				LAF [31]			
Configuration			Census		MCCNN-fst		Census		MCCNN-fst	
Hyp.	Uncert.	Ref.	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM
1	✗	✗	4.28	1.38	2.84	0.94	4.26	1.59	2.85	1.10
N	✗	✗	4.26	1.36	2.80	0.91	4.23	1.60	2.82	1.03
N	var.	✗	4.30	1.41	2.83	0.95	4.32	1.61	2.89	1.08
N/2	net.	✗	4.08	1.38	2.79	0.93	4.20	1.54	2.79	1.01
N	net.	✗	4.10	1.35	2.76	0.90	4.11	1.50	2.76	0.99
2N	net.	✗	4.08	1.36	2.71	0.91	4.09	1.51	2.73	0.98
N	✗	✓	4.16	1.31	2.78	0.89	4.11	1.49	2.81	1.00
N	net.	✓	**4.03**	**1.28**	**2.71**	**0.86**	**4.06**	**1.34**	**2.64**	**0.87**
Optimal			3.40	0.74	2.16	0.44	3.40	0.74	2.16	0.44

TABLE I. **Ablation study on the proposed meta-confidence estimation and refinement.** We report AUC scores for ConfNet [30] and LAF [31] networks within meta-confidence framework trained on KITTI 2012 (20 images) and tested on KITTI 2015.

	ConfNet [30]				LAF [31]			
	Census		MCCNN-fst		Census		MCCNN-fst	
	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM
No uncertainty	4.28	1.38	2.84	0.94	4.26	1.59	2.85	1.10
Snapshots [48]	4.26	1.39	2.86	0.97	4.30	1.58	2.87	1.12
Predictive [18]	4.30	1.37	2.88	1.01	4.24	1.55	2.86	1.08
Multi-head [46]	4.28	1.37	2.83	0.99	4.27	1.57	2.84	0.99
Ours	**4.03**	**1.28**	**2.71**	**0.86**	**4.06**	**1.34**	**2.64**	**0.87**
Optimal	3.40	0.74	2.16	0.44	3.40	0.74	2.16	0.44

TABLE II. **Comparison with existing methods to model uncertainty.** We report AUC scores for ConfNet [30] and LAF [31] implementing uncertainty modeling using known methods and our framework, trained on KITTI 2012 (20 images) and tested on KITTI 2015.

Random forest frameworks. For experiments involving random forest models, we follow the guidelines from the literature [9], [8], [11], [10], [12], setting the number of trees to 10, with a maximum depth of 25 nodes each. The code is implemented in C++ using OpenCV library, this latter opportunely modified to extract empirical variance. We consider four forest-based methods among those in the literature: ENS [21], GCP [23], [51], LEV50 [10] and O2 [12]. We trained the *refinement* networks using the same settings described for confidence networks.

Datasets. Following the most recent literature [37], we consider four standard datasets in our experiments: KITTI 2012 [52], KITTI 2015 [53], Middlebury 2014 at quarter resolution and ETH3D [54], setting τ respectively to 3, 3, 1 and 1 when computing the error rates. We train the confidence estimators on the first 20 images from KITTI 2012 [8], evaluating on the remaining 174 images and on the 200 available from KITTI 2015. Finally, we study the generalization over unseen content of the models trained on KITTI 2012 by testing on the Middlebury 2014 and ETH3D datasets, counting respectively 15 and 27 stereo images.

	KITTI 2012				KITTI 2015				Middlebury 2014				ETH3D			
	Census		MCCNN-fst		Census		MCCNN-fst		Census		MCCNN-fst		Census		MCCNN-fst	
Model	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM
CCNN [26]	5.56	1.71	3.30	1.71	4.32	1.72	3.29	1.96	10.52	11.54	9.09	8.94	9.02	7.26	15.96	3.99
CCNN-μ	5.55	1.66	2.84	1.38	4.32	1.73	2.91	1.38	10.31	11.59	8.18	7.13	10.16	7.27	13.23	3.96
CCNN-f	5.33	1.61	2.74	1.04	4.13	1.66	2.80	1.04	9.09	10.57	7.81	6.20	8.91	7.11	13.11	3.41
Δ_{AUC}(%)	27.38	10.87	58.94	45.89	20.65	6.12	43.36	60.53	27.45	13.92	36.99	45.67	2.22	2.93	50.44	22.48
ConfNet [30]	5.27	1.40	2.73	0.56	4.28	1.38	2.84	0.94	10.70	9.88	8.08	6.03	10.42	4.20	13.44	3.47
ConfNet-μ	5.22	1.35	2.70	0.51	4.11	1.30	2.77	0.89	10.73	9.32	7.84	6.53	10.12	3.88	13.17	3.64
ConfNet-f	5.11	1.30	2.64	0.46	4.03	1.28	2.71	0.86	10.45	9.34	7.80	6.50	10.07	3.74	13.08	3.44
Δ_{AUC}(%)	29.09	16.39	23.68	32.26	28.41	15.63	19.12	16.00	4.64	10.17	11.43	-15.21	5.51	22.33	11.50	1.46
LGC [30]	5.13	1.36	2.65	0.55	4.08	1.35	2.74	0.90	10.44	9.73	7.79	5.87	11.85	4.73	13.33	3.39
LGC-μ	5.12	1.36	2.65	0.43	4.06	1.29	2.73	0.85	10.88	9.12	7.81	5.87	9.42	4.34	13.36	3.41
LGC-f	5.10	1.34	2.62	0.41	4.02	1.28	2.70	0.84	10.12	9.01	7.75	5.79	9.24	4.25	13.22	3.33
Δ_{AUC}(%)	7.32	3.51	10.00	46.67	8.82	11.48	6.90	13.04	6.24	13.95	1.85	2.73	33.55	18.53	3.64	1.41
UCN [33]	5.41	1.36	2.79	0.63	4.23	1.53	2.95	1.21	9.16	9.51	8.48	5.86	6.64	5.78	12.86	3.89
UCN-μ	5.36	1.37	2.73	0.64	4.19	1.50	2.91	1.20	9.06	8.67	8.49	5.81	6.30	5.14	12.77	3.58
UCN-f	5.24	1.23	2.69	0.57	4.06	1.39	2.80	1.07	8.83	8.53	8.31	5.13	6.08	4.80	12.62	3.12
Δ_{AUC}(%)	24.64	22.81	22.72	15.80	20.48	17.72	18.99	18.18	8.57	19.84	5.96	25.00	27.80	26.92	9.41	31.04
LAF [31]	5.31	1.37	2.71	0.61	4.26	1.59	2.85	1.10	9.26	8.94	8.00	5.37	8.06	5.21	13.34	3.98
LAF-μ	5.21	1.36	2.69	0.59	4.20	1.57	2.84	1.08	8.88	8.99	8.02	5.38	7.53	4.94	13.11	3.99
LAF-f	5.17	1.33	2.63	0.54	4.06	1.34	2.64	0.87	8.67	8.12	7.91	5.21	7.28	4.67	12.98	3.93
Δ_{AUC}(%)	23.73	6.90	22.22	19.44	23.26	29.41	30.43	34.85	14.94	18.76	3.80	6.58	19.55	17.59	11.88	1.95
Optimal	4.72	0.79	2.35	0.25	3.40	0.74	2.16	0.44	5.31	4.57	5.63	2.94	4.07	2.14	10.31	1.41

(a) (b)

TABLE III. **Results on KITTI 2012 and 2015 (a), Middlebury and ETH3D (b) – deep networks.** We report AUCs for deep learning methods, their variants with refinement network and the Δ_{AUC}(%) achieved by these latter.

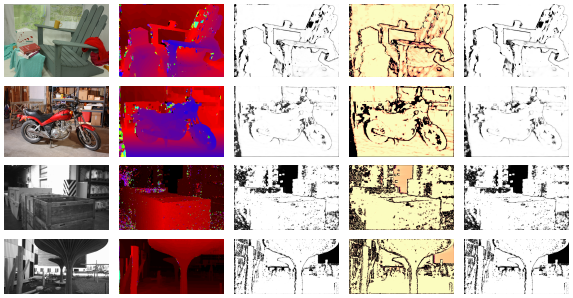


Fig. 3. **Qualitative results on Middlebury 2014 and ETH3D datasets.** From left: reference image, disparity by (from top) Census-CBCA, Census-SGM, MCCNN-fst-CBCA and MCCNN-fst-SGM, confidence, meta-confidence, and final confidence by UCN [33].

Stereo algorithms. Confidence estimators are traditionally evaluated over a variety of stereo algorithms [8], [30], [31], in order to assess their effectiveness when dealing either with noisy or accurate disparity maps. A standard benchmark in this field [31], [22] uses four stereo matchers implemented by Zbontar and LeCun [3]: Census-CBCA, Census-SGM, MCCNN-fst-CBCA and MCCNN-fst-SGM.

C. Ablation study

We now study the impact of the different components of our meta-confidence framework and compare it with existing strategies to model uncertainty [48], [18], [46]. To this aim, on the KITTI 2012 dataset [52], we trained two confidence estimation networks, namely ConfNet [30] and LAF [31], and evaluated several variants on the KITTI 2015 dataset.

In Table I we collect ablation experiments. We first analyze the results of confidence estimation networks reformulated as multi-headed networks, showing improvements over the single output ones. Considering the trade-off between accuracy and complexity, we set $N = 8$, similarly to [46], [20]. When training the networks with $\mathcal{L}_{\text{llm}}$ and uncertainty as variance (var.), the performance is even degraded. Instead, by employing the uncertainty network in place of variance (net.), the gain becomes apparent. The trade-off in modeling uncertainty at the cost of lower accuracy is known in literature [5]. However, we have found that *learning* the uncertainty by means of a dedicated network processing the N hypotheses is not affected by this issue, overcoming a known limitation in literature. In addition, adding more hypotheses ($2N$) does not improve the results significantly. Moreover, the refinement networks slightly improve the performance even without modeling uncertainty (a configuration that is equivalent to [27]), but marginally. By considering multiple hypotheses, uncertainty and refinement networks, our framework achieves the best performance.

	KITTI 2012				KITTI 2015				Middlebury 2014				ETH3D			
	Census		MCCNN-fst		Census		MCCNN-fst		Census		MCCNN-fst		Census		MCCNN-fst	
Model	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM	CBCA	SGM
ENS [21]	6.62	1.99	3.53	0.82	5.60	2.18	3.76	1.65	11.15	12.82	9.58	7.94	8.20	7.82	14.48	5.39
ENS-f	6.16	1.83	3.33	0.77	5.09	1.93	3.48	1.64	10.57	12.51	8.49	7.85	7.74	7.61	13.96	5.27
Δ_{AUC}(%)	24.31	13.19	17.24	8.93	22.89	17.02	17.76	0.83	9.93	3.72	27.68	1.67	10.97	3.67	12.53	2.85
GCP [23], [51]	6.37	2.17	3.32	1.00	5.29	2.40	3.44	1.91	11.18	13.12	9.86	7.53	8.54	6.34	15.69	5.61
GCP-f	5.71	1.78	2.96	0.76	4.64	1.84	3.08	1.60	11.14	12.20	8.96	6.96	7.96	5.63	14.68	5.00
Δ_{AUC}(%)	39.97	28.12	37.51	31.27	34.44	33.48	28.41	20.83	0.68	10.81	21.32	12.52	13.00	16.73	18.86	14.59
LEV50 [10]	5.74	1.54	2.96	0.66	4.51	1.69	3.02	1.14	11.57	12.42	8.81	6.70	7.93	7.00	13.69	3.79
LEV50-f	5.28	1.38	2.69	0.62	4.12	1.50	2.78	1.04	10.09	10.87	7.88	6.00	7.04	6.22	12.64	3.39
Δ_{AUC}(%)	45.11	20.43	45.05	10.35	34.95	19.04	27.51	14.38	23.60	19.68	29.17	18.48	23.11	16.01	31.21	16.93
O2 [12]	5.81	1.55	2.93	0.72	4.53	1.66	2.97	1.07	11.06	10.81	8.09	6.24	9.28	8.08	14.91	5.36
O2-f	5.41	1.51	2.73	0.70	4.25	1.61	2.81	1.05	10.81	10.71	7.69	6.37	8.57	7.29	15.28	5.22
Δ_{AUC}(%)	36.85	4.07	34.43	4.56	24.63	5.16	19.59	2.82	12.24	1.57	16.53	-4.03	13.72	13.24	-7.82	3.74
Optimal	4.72	0.79	2.35	0.25	3.40	0.74	2.16	0.44	5.31	4.57	5.63	2.94	4.07	2.14	10.31	1.41

(a) (b)

TABLE IV. **Results on KITTI 2012 and 2015 (a), Middlebury and ETH3D (b) – random forests.** We report AUCs for random forest methods, their variants with refinement network and the Δ_{AUC}(%) achieved by these latter.

Table II shows a comparison between our full framework with existing methods [48], [18], [46] modelling uncertainty. We can notice how these latter rarely improve the performance of the baseline networks not modelling uncertainty at all (first row), while our framework consistently outperforms both baseline and competitors, resulting superior when it comes to improve the performance of a confidence estimator.

The computational overhead of our full framework with respect to baseline estimators is negligible – *i.e.* runtime and parameters just increase by 3.65% and 9.36% in LAF [31].

D. Meta-confidence framework evaluation

In this section, we compare the performance of state-of-the-art confidence estimators achieved by their original formulation and by our refined variants modeling meta-confidence. Specifically, we report Δ_{AUC} scores highlighting in green when our formulation yields to improvements (positive Δ_{AUC}), in red otherwise (negative Δ_{AUC}).

Deep networks. We report the results achieved by state-of-the-art confidence networks [26], [30], [30], [33], [31] trained with the meta-confidence estimation and refinement. Table III (a) shows results on KITTI 2012 and KITTI 2015 datasets. We can notice that each of them outperforms the performance of the corresponding baseline. In addition, Table III (b) demonstrates results on Middlebury 2014 and ETH3D datasets, where in general, the confidence estimator struggles against domain discrepancy and uncertain predictions may be dramatically increased. Meta-confidence gives our method a hint about the regions, and can correct them. Figure 3 shows some qualitative examples on Middlebury and ETH3D.

Random forests. Moreover, we report results achieved by random forests frameworks [21], [51], [10], [12] and refined by our formulation exploiting empirical uncertainty. Table IV

Model	K12	K15	Mid	ETH
CCNN	3.29	3.93	23.34	7.02
CCNN-f	2.94	3.44	21.68	6.10
Δ_{AUC}(%)	12.87	15.91	9.19	15.26
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
ConfNet	5.46	5.21	22.31	4.69
ConfNet-f	5.10	4.81	19.01	4.14
Δ_{AUC}(%)	7.36	9.17	19.38	14.86
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
LGC	3.53	4.61	22.42	6.31
LGC-f	3.01	4.22	19.07	5.62
Δ_{AUC}(%)	17.57	10.37	19.54	12.97
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
UCN	2.62	3.18	23.31	12.23
UCN-f	2.21	2.97	21.09	9.88
Δ_{AUC}(%)	20.00	9.01	12.31	20.91
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
LAF	1.70	2.58	18.19	7.40
LAF-f	1.51	2.21	16.52	6.31
Δ_{AUC}(%)	16.81	21.39	12.94	17.00
Optimal	0.57	0.85	5.28	0.99

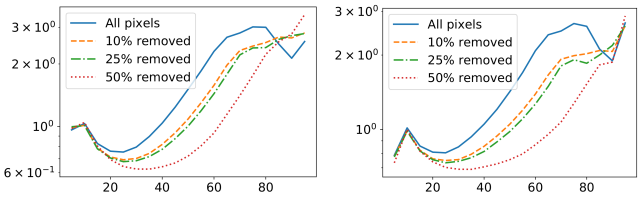
Model	K12	K15	Mid	ETH
ENS	3.18	4.52	18.28	5.58
ENS-f	2.66	3.81	18.84	5.69
Δ_{AUC}(%)	20.09	19.18	-4.27	-2.25
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
GCP	4.19	5.39	21.00	5.24
GCP-f	2.86	4.28	18.82	4.30
Δ_{AUC}(%)	36.72	24.50	13.92	20.94
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
LEV50	1.99	2.87	18.47	4.23
LEV50-f	1.79	2.59	17.49	3.62
Δ_{AUC}(%)	15.08	12.56	7.40	17.44
Optimal	0.57	0.85	5.28	0.99

Model	K12	K15	Mid	ETH
O2	2.76	3.67	22.03	7.10
O2-f	2.56	3.54	21.43	6.25
Δ_{AUC}(%)	9.09	4.72	3.58	13.32
Optimal	0.57	0.85	5.28	0.99

TABLE V. **Results on GANet disparity maps.** We report AUCs for the nine estimators, their variants with refinement network and the $\Delta_{AUC}(\%)$ achieved by these latter.



(a) LEV50 [10] (b) O2 [30]

Fig. 4. **Relationship between meta-confidence and confidence.** We plot the difference between ROC and optimum curves (y axis) in log scale, when sampling different percentages of pixels (x axis) according to meta-confidence.

(a) collects results on KITTI 2012 and KITTI 2015 datasets, showing consistent improvements for any method across the four stereo matchers. We point out how the gain in terms of Δ_{AUC} is, in general, higher for CBCA matchers while it is lower, yet consistent, for SGM algorithms.

Considering different domains, Table IV (b) confirms that our framework also consistently improves the performance on completely unseen content, with very few exceptions.

Performance with modern stereo network. Finally, as in [22], we prove that our framework is effective also at improving confidences estimated for a deep stereo network such as GANet [55]. Table V collects results for the nine confidence estimators, again trained on KITTI 2012 and tested on all the datasets considered so far, confirming that our meta-confidence framework consistently yields improvements, except for ENS on Middlebury and ETH3D.

E. Analysis

Finally, we show qualitatively how effective the meta-confidence is at explaining wrong confidence predictions prior to refinement. To this aim, given a disparity map we plot a curve result of the difference between the ROC obtained by confidence sampling and the optimum curve. We repeat this after removing a percentage of the pixels having the lowest meta-confidence, in an iterative way similar to the ROC computation procedure. In Fig. 4 we plot, for different confidence measures, the difference between ROC and optimal curves on the entire disparity map (blue) and on sparse maps after removing 10% (yellow), 25% (green) or 50% (red) least meta-confident pixels, measured on the entire

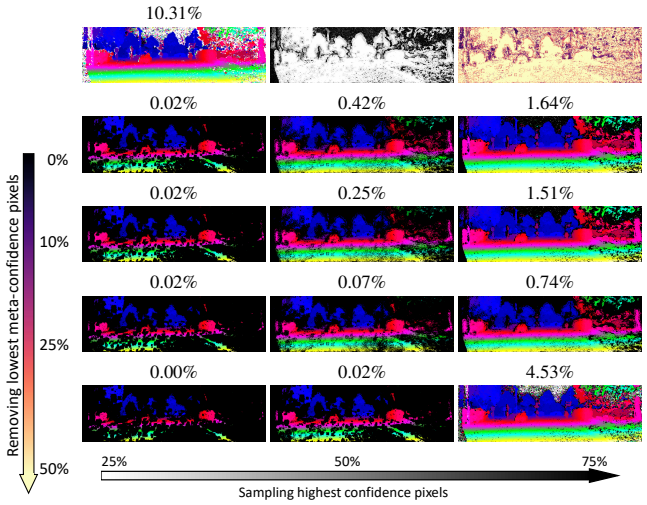


Fig. 5. **Effects of sparsification according to meta-confidence (LEV50).** By removing pixels with lowest meta-confidence (top to bottom), correct pixels selection according to confidence (left to right) is most of the times improved.

KITTI 2015 dataset, using MCCNN-fst-CBCA algorithm. We can notice how the curves after sampling are, most of the times, lower than the blue one, hinting a lower gap with respect to the optimum and suggesting that meta-confident is a meaningful hint of the confidence errors.

Accordingly, we can exploit confidence and meta-confidence in an orthogonal manner to select reliable pixels. Fig. 5 shows how, by removing the least meta-confident pixels and selecting the most confident one, we are able to select a subset of pixels containing no outliers (bottom left), whereas confidence alone cannot. Sampling accuracy becomes less consistent when sampling higher percentages of most confident pixels (> 75%). More qualitative examples are reported in the **supplementary video**.

V. Conclusion

In this paper we proposed, for the first time, to take into account the uncertainty of the confidence of a disparity map, as a second-level confidence or meta-confidence. We have shown how existing deep learning models for confidence estimation can be extended to learn the meta-confidence and to exploit it to predict more reliable and accurate confidence score. It has been also shown how existing random forest strategies already allow to retrieve this information for free and how it can be used to refine the confidence. Experimental results on a variety of stereo algorithms and confidence estimators, including state-of-the-art deep learning models and random forest-based ones, proved that our meta-confidence framework is effective in finding incorrect confidence prediction and correcting it.

Acknowledgements. The work was supported by the MSIT, Korea (IITP-2022-2020-0-01819, ICT Creative Consilience program), and National Research Foundation of Korea (NRF-2021R1C1C1006897, NRF-2021R1C1C2005202).

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] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and stereo: a survey," *arXiv preprint arXiv:2004.08566*, 2020.

[3] J. Žbontar 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.

[4] 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.

[5] 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.

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

[7] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," vol. 34, no. 11, pp. 2121–2133, 2012.

[8] M. Poggi, F. Tosi, and S. Mattoccia, "Quantitative evaluation of confidence measures in a machine learning world," in *ICCV*, 2017, pp. 5228–5237.

[9] M.-G. Park and K.-J. Yoon, "Leveraging stereo matching with learning-based confidence measures," in *CVPR*, 2015, pp. 101–109.

[10] ——, "Learning and selecting confidence measures for robust stereo matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1397–1411, 2018.

[11] 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." IEEE, 2016, pp. 509–518.

[12] M. Poggi, F. Tosi, and S. Mattoccia, "Learning a confidence measure in the disparity domain from o (1) features," *Computer Vision and Image Understanding*, vol. 193, p. 102905, 2020.

[13] J. L. Schonberger, S. N. Sinha, and M. Pollefeys, "Learning to fuse proposals from multiple scanline optimizations in semi-global matching," in *ECCV*, 2018, pp. 739–755.

[14] G. Marin, P. Zanuttigh, and S. Mattoccia, "Reliable fusion of tof and stereo depth driven by confidence measures," in *European Conference on Computer Vision*. Springer, 2016, pp. 386–401.

[15] M. Poggi, G. Agresti, F. Tosi, P. Zanuttigh, and S. Mattoccia, "Confidence estimation for tof and stereo sensors and its application to depth data fusion," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1411–1421, 2020.

[16] 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.

[17] ——, "Unsupervised domain adaptation for depth prediction from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[18] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[19] E. Ilg, T. Saikia, M. Keuper, and T. 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.

[20] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] 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.

[22] M. Poggi, S. Kim, F. Tosi, S. Kim, F. Aleotti, D. Min, K. Sohn, and S. Mattoccia, "On the confidence of stereo matching in a deep-learning era: a quantitative evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

[24] 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.

[25] A. Seki and M. Pollefeys, "Patch based confidence prediction for dense disparity map." in *BMVC*, vol. 2, no. 3, 2016, p. 4.

[26] M. Poggi and S. Mattoccia, "Learning from scratch a confidence measure." in *BMVC*, 2016.

[27] ——, "Learning to predict stereo reliability enforcing local consistency of confidence maps," in *CVPR*, 2017, pp. 2452–2461.

[28] M. Poggi, F. Tosi, and S. Mattoccia, "Even more confident predictions with deep machine-learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 76–84.

[29] Z. Fu and M. A. Fard, "Learning confidence measures by multi-modal convolutional neural networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1321–1330.

[30] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia, "Beyond local reasoning for stereo confidence estimation with deep learning," in *ECCV*, 2018, pp. 319–334.

[31] S. Kim, S. Kim, D. Min, and K. Sohn, "Laf-net: Locally adaptive fusion networks for stereo confidence estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[32] M. S. K. Gul, M. Bätz, and J. Keinert, "Pixel-wise confidences for stereo disparities using recurrent neural networks," in *BMVC*, 2019.

[33] S. Kim, D. Min, S. Kim, and K. Sohn, "Unified confidence estimation networks for robust stereo matching," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1299–1313, 2019.

[34] ——, "Adversarial confidence estimation networks for robust stereo matching," *IEEE Transactions on Image Processing*, 2020.

[35] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, "Using self-contradiction to learn confidence measures in stereo vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] F. Tosi, M. Poggi, A. Tonioni, L. Di Stefano, and S. Mattoccia, "Learning confidence measures in the wild," in *BMVC*, Sept. 2017.

[37] M. Poggi, F. Aleotti, F. Tosi, G. Zaccaroni, and S. Mattoccia, "Self-adapting confidence estimation for stereo," in *European Conference on Computer Vision (ECCV)*, 2020.

[38] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[39] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *International conference on machine learning*, 2014, pp. 1683–1691.

[40] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 681–688.

[41] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[42] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[43] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[44] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[45] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[46] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 652–667.

[47] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weight losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.

[48] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," in *ICLR*, 2017.

[49] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager, "Learning in an uncertain world: Representing ambiguity through multiple hypotheses," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3591–3600.

[50] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[51] A. Spyropoulos and P. Mordohai, "Correctness prediction, accuracy improvement and generalization of stereo matching using supervised learning," *International Journal of Computer Vision*, vol. 118, no. 3, pp. 300–318, 2016.

[52] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *CVPR*, 2012.

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

[54] T. Schops, J. L. Schonberger, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3260–3269.

[55] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 185–194.