Meta-confidence estimation for stereo matching

Seunryong Kim\textsuperscript{1}, Matteo Poggi\textsuperscript{2}, Sunok Kim\textsuperscript{3}, Kwanghoon Sohn\textsuperscript{4}, Stefano Mattoccia\textsuperscript{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 \cite{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 \cite{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 \cite{2} and a variety of solutions either mixing neural networks and hand-designed strategies \cite{3} or deploying end-to-end neural networks \cite{4, 5, 6}.

 Concurrent to the race for depth accuracy, confidence estimation \cite{7, 8} has been developed as a parallel research track in the stereo matching literature, allowing for both improvement to known stereo algorithms \cite{9, 10, 11, 12, 13} as well as for additional, higher-level applications \cite{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 handcrafted confidence measures \cite{7} to data-driven strategies \cite{8}.

 Although achieving outstanding performance, learned models and their confidence predictions are intrinsically affected by some uncertainty \cite{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 \cite{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 \cite{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
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], bootstrapped 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 modeling 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 maximum 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

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

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

\[
L_{\text{lim}} = \frac{D(\mu(c), c^*)}{\sigma(c)} + \log \sigma(c),
\]  

where \( D(\mu(c), c^*) \) 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 \( \Phi_u \), 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 \( 3 \times 3 \) Convolution-BatchNorm-ReLu producing 64 feature maps, followed by \( 1 \times 1 \) 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 \( T_n \), with \( n \in [1, N] \), each one casting an individual vote \( T_n(v) \) for any given feature vector \( v \). 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

\[
\mu(v) = \frac{1}{N} \sum_{n=1}^{N} T_n(v).
\]  

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

\[
\sigma(v) = \frac{1}{N} \sum_{n=1}^{N} (T_n(v) - \mu(v))^2.
\]
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 \( \Phi \), 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 \( f = \psi(\mu(c), \sigma(c); \Phi) \). To train the network, a refinement loss \( \mathcal{L}_{ref} \) is computed as \( \mathcal{L}_{ref} = D(f, c') \). 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 \( \tau \). 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 \( \epsilon \) on the whole disparity map:

\[
AUC_{opt} = \int_{1-\epsilon}^{\infty} \frac{p - (1-\epsilon)}{p} dp = \epsilon + (1-\epsilon) \ln(1-\epsilon). \tag{5}
\]

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 \( \Delta_{AUC} \) metric proposed in [27] as

\[
\Delta_{AUC} = \frac{AUC - AUC_f}{AUC - AUC_{opt}} \tag{6}
\]

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, \( \mathcal{L}_{lim} \) and \( \mathcal{L}_{ref} \). 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.

**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 \( \tau \) 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.
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 $\Delta_{\text{AUC}}$ (%) achieved by these latter.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Census-CBCA</td>
<td>3.50</td>
<td>3.05</td>
<td>7.90</td>
<td>5.90</td>
<td>3.04</td>
<td>7.81</td>
<td>5.86</td>
<td>3.02</td>
<td>7.78</td>
</tr>
<tr>
<td>Census-SGM</td>
<td>3.45</td>
<td>3.00</td>
<td>7.54</td>
<td>5.90</td>
<td>3.04</td>
<td>7.38</td>
<td>5.86</td>
<td>3.02</td>
<td>7.27</td>
</tr>
<tr>
<td>MCCNN-fst-CBCA</td>
<td>3.62</td>
<td>3.05</td>
<td>7.72</td>
<td>5.84</td>
<td>3.00</td>
<td>7.51</td>
<td>5.80</td>
<td>2.98</td>
<td>7.30</td>
</tr>
<tr>
<td>MCCNN-fst-SGM</td>
<td>3.57</td>
<td>3.00</td>
<td>7.65</td>
<td>5.80</td>
<td>2.98</td>
<td>7.48</td>
<td>5.76</td>
<td>2.96</td>
<td>7.36</td>
</tr>
</tbody>
</table>

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 $L_{\text{lim}}$ 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.

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 $\Delta_{\text{AUC}}$ scores highlighting in green when our formulation yields to improvements (positive $\Delta_{\text{AUC}}$), in red otherwise (negative $\Delta_{\text{AUC}}$).

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
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.

![Graph showing relationship between meta-confidence and confidence](image)

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 $\Delta_{AUC}$ 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 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-2022-00-01819, ICT Creative Consilience program), and National Research Foundation of Korea (NRF-2021R1C1C1006897, NRF-2021R1C1C2005202).
REFERENCES


