Self-Supervised Monocular Depth Estimation by Digging into Uncertainty Quantification

Yuan-Zhen Li[†] (李远珍), Sheng-Jie Zheng[†] (郑圣杰), Zi-Xin Tan (谭梓欣), Tuo Cao (曹拓), Fei Luo[*] (罗飞), and Chun-Xia Xiao[*] (肖春霞), *Member, CCF, IEEE*

School of Computer Science, Wuhan University, Wuhan 430072, China

E-mail: {yuanzhen, 2020282110184, 2019302060051, maplect, luofei, cxxiao}@whu.edu.cn

Received July 15, 2018 [Month Day, Year]; accepted October 14, 2018 [Month Day, Year].

Abstract Based on well-designed network architectures and objective functions, self-supervised monocular depth estimation has made great progress. However, lacking a specific mechanism to make the network learn more about the regions containing moving objects or occlusion scenarios, existing depth estimation methods likely produce poor results for them. Therefore, we propose an uncertainty quantification method to improve the performance of existing depth estimation networks without changing their architectures. Our uncertainty quantification method consists of uncertainty measurement, the learning guidance by uncertainty, and the ultimate adaptive determination. Firstly, with Snapshot and Siam learning strategies, we measure the uncertainty degree by calculating the variance of pre-converged epochs or twins during training. Secondly, we use the uncertainty to guide the network to strengthen learning about those regions with more uncertainty. Finally, we use the uncertainty to adaptively produce the final depth estimation results with a balance of accuracy and robustness. To demonstrate the effectiveness of our uncertainty quantification method, we apply it to two state-of-the-art models, Monodepth2 and Hints. Experimental results show that our method has improved the depth estimation performance in seven evaluation metrics compared with two baseline models and exceeded the existing uncertainty method.

Keywords self-supervised, monocular depth estimation, uncertainty quantification, variance

1 Introduction

Depth estimation [1] is a fundamental task in computer graphics and computer vision, which can be used in text-to-image [2], 6D pose estimation [3], and scene reconstruction [4–6], etc. Depth estimation from a single RGB image is an ill-posed problem. However, with the development of deep learning, monocular depth estimation has become a possibility. The deep network learns the relationship between the spatial distance and image features with large datasets. Compared with fully supervised learning, self-supervised monocular depth estimation only needs stereo image pairs or monocular video to supervise, which is a significant advantage.

To improve the performance of the self-supervised depth estimation network, novel loss functions [7–9] and network architectures [10–13] have been proposed. The pre-processing or post-processing [14] is also considered to increase data usage. However, these techniques only partially solve self-supervised monocular depth estimation defects. On the one hand, a specific technique to improve certain depth estimation performance always requires an application prerequisite. For example, the semantic information used to sharpen the object bound-

Regular Paper

Special Section of CVM 2023

This work was supported by the National Natural Science Foundation of China under Grant No. 61972298 and the Xinjiang Bingtuan Science and Technology Program of China under Grant No. 2019BC008.

[*]Corresponding Author: Chun-Xia Xiao and Fei Luo.

[†]Yuan-Zhen Li and Sheng-Jie Zheng are joint first authors.

ary in the depth map is limited by the number of known objects. On the other hand, self-supervised training is an under-constraint task due to needing more optimization objectives to restrict the factors such as weak textures, moving objects, varying illumination, and occlusion. Solely depending on improving neural network architectures is hard to solve above issues.

Uncertainty quantification is an effective strategy to improve the accuracy of the depth estimation network. There are some methods [15–19] about uncertainty, but these methods still have several weaknesses. First, these methods are based on ground truth depth to obtain uncertainty. Second, these methods can not completely solve the under-constraint problem. Third, these methods do not explicitly deal with the learning difficulty of uncertainty and uncertainty regions in the training process.

This paper proposes an uncertainty quantification method to learn self-supervised monocular depth estimation. Our idea is based on the observation that uncertainty is caused by under-constraint and manifested as unstable prediction among consecutive training epochs. Thus, we propose to estimate uncertainty regions based on the variance of consecutive epoch results and guide the network to learn them. Our uncertainty quantification method consists of uncertainty measurement, guidance, and post-processing. Based on our simple but effective method, uncertainty regions can be detected and better learned (see Fig.1).

Our contributions can be summarized as follows:

- We propose to use consecutive training epochs or a Siamese network to measure the uncertainty of the estimated depth. The estimated uncertainty mask is used to guide the depth network learning.
- We propose ensemble-based uncertainty post-processing to adaptively produce final depth results with a balance of accuracy and robustness.
- Our uncertainty quantification method does not add additional modules, which could avoid substantially modifying the baseline model.

The rest of the paper is organized as below. Section 2 reviews the related work. Section 3 describes our method. Section 4 reports our experimental details and results. Finally, Section 5 concludes our work.

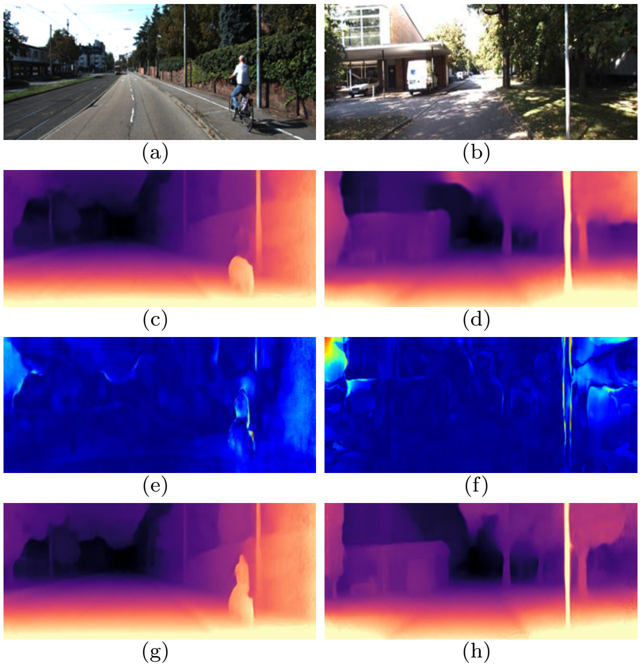


Fig.1. Two examples of our method. (a) RGB input. (b) RGB input. (c) Depth estimated by the baseline model Monodepth2 [8]. (d) Depth estimated depth by the baseline model Hints [14]. (e) Uncertainty mask of depth. (f) Uncertainty mask of depth. (g) Depth estimated by our method (Snapshot). (h) Depth estimated by our method (Siam).

2 Related Work

2.1 Self-supervised Monocular Depth Estimation

Garg *et al.* [7] established the cornerstone of self-supervised monocular depth estimation, and the photometric reconstruction loss is the core loss function. Godard *et al.* [20] proposed a depth estimation network named Monodepth. Monodepth predicts left-right disparities to enforce consistency between the disparities produced relative to the left and right images.

Zhou *et al.* [21] first used the monocular video to train the depth estimation network by jointly learning the depth and relative pose. Godard *et al.* [8] proposed the three innovations in Monodepth2. First, they designed a minimum photometric reconstruction loss to address the problem of occluded pixels. Second, they designed an auto-masking loss to ignore training pixels that violate relative camera motion assumptions. Finally, they up-sampled the predicted multi-resolution depth maps to the input resolution and computed all losses to reduce texture-copy artifacts. Bian *et al.* [9] proposed a geometry consistency loss to penalize the inconsistency of predicted depths between adjacent views and a self-discovered mask to automatically localize moving objects that violate the underlying static scene assumption and cause noisy signals during training.

Some methods proposes to use multi-task training strategies to improve depth estimation accuracy. Yin and Shi [22] proposed to use a multi-task learning network GeoNet for monocular depth, optical flow, and ego-motion estimation. Zou *et al.* [23] proposed DF-Net to solve the same three objectives. GeoNet uses the deep network to estimate the residual flow, but DF-Net uses the deep network to estimate the optical flow. Klingner *et al.* [10] used a semantic segmentation network to detect moving objects, preventing photometric reconstruction from contaminating.

2.2 Uncertainty in Depth Estimation

Machine learning treats under-constraint as an uncertainty problem [24]. Liu *et al.* [25] made a systematical discussion on uncertainty in depth estimation. Song *et al.* [26] divided the uncertainty of neural networks into two categories: random uncertainty and model uncertainty. Random uncertainty is from sensor and motion noise, which may cause inaccurate observation data. Model uncertainty mainly refers to the uncertainty of model parameters [26].

Random Uncertainty. Choi *et al.* [18] proposed a model consisting of a monocular depth network, a confidence network, and a threshold network. They distilled the training dataset with the confidence and threshold networks to supervise the monocular depth network. Shen *et al.* [27] supposed that the noise obeyed the Gaussian distribution in the training dataset. They used a two-stage teacher-student framework to estimate the uncertainty.

Model Uncertainty. Asai *et al.* [15] formulated regression with uncertainty estimation as a multi-task learning problem and designed a separate multi-task loss to optimize the depth and uncertainty, respectively. Mertan *et al.* [16] treated the relative depth estimation problem as maximum likelihood estimation. They assumed that the depth followed a normal distribution and used a neural network to learn the mean and variance distribution parameters. The mean represents the depth, and the variance indicates the uncertainty. Teixeira *et al.* [17] constructed a confidence network and a depth network. The estimated confidence is used to filter out unreliable depth.

Poggi *et al.* [19] summarized the uncertainty quantification of the depth estimation. Their work analyzed three uncertainty categories: empirical, predictive, and Bayesian. Predictive and Bayesian categories need extra uncertainty estimation models. However, integrating them into the baseline model is inconvenient. Empirical estimation could work independently with the baseline model, which is suitable for single-value objective optimization by increasing the diversity of iteration solutions.

The most difference between the method of Poggi *et al.* [19] and our method is that we make use of consecutive training epochs or a Siamese network to identify uncertainty and convert it into a spatial mask over the training image to guide network learning, rather than increasing the diversity by making an ensemble of different solutions.

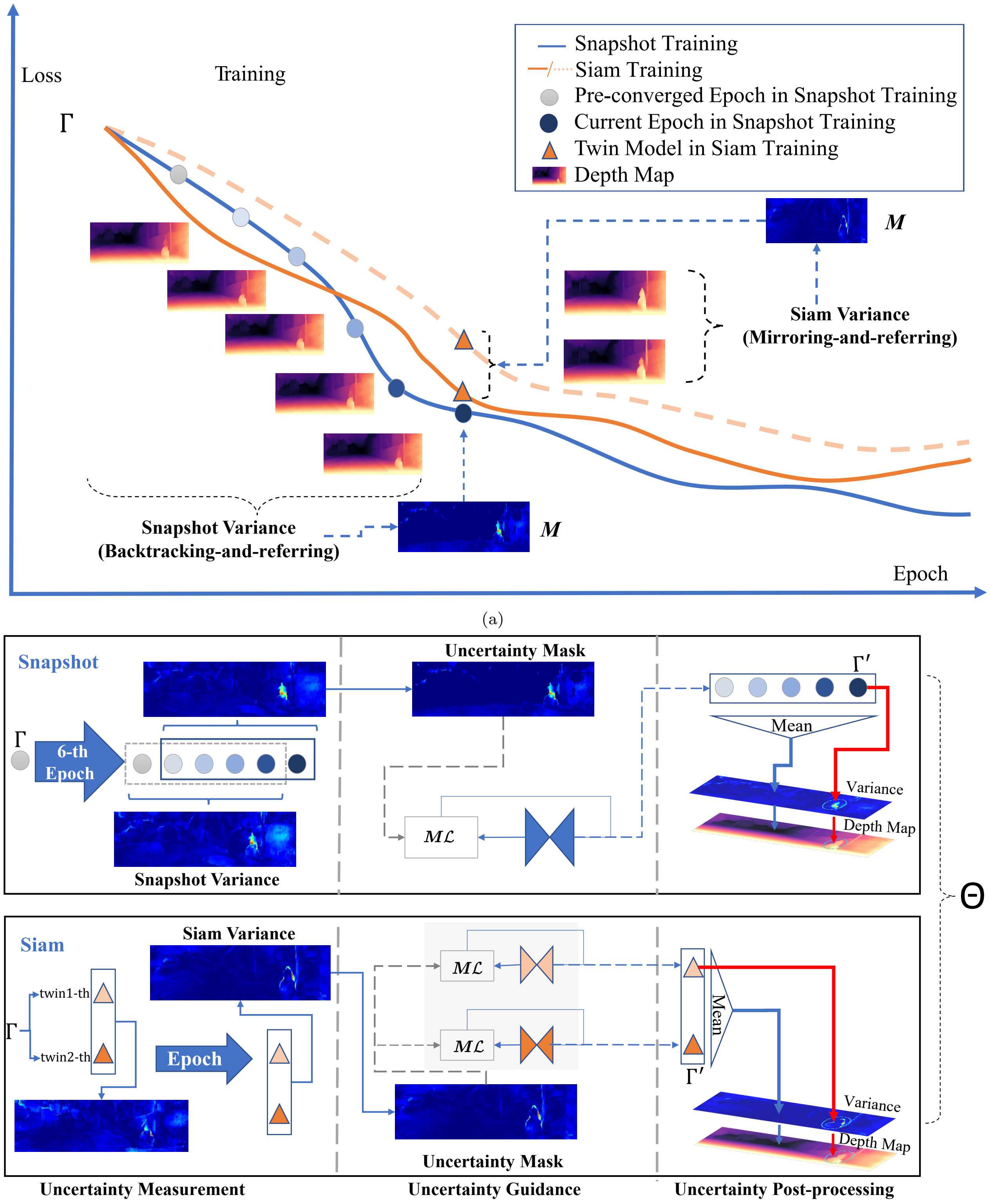
3 Depth Estimation with Uncertainty Quantification

We propose an uncertainty quantification method to train self-supervised monocular depth estimation. Our goal function can be expressed as follows:

$$F(\Gamma, \boldsymbol{M}),$$

where Γ is the baseline model, $\boldsymbol{M}$ is the uncertainty mask constructed by the uncertainty information over all pixels of the depth map to identify uncertainty positions and measure the uncertainty degree.

We use Snapshot [28] and Siam [29] to realize our method (see Fig.2(a)). The uncertainty quantification


(a)

(b)

Fig. 2. Overview of the uncertainty quantification method. (a) Two empirical uncertainty quantification approaches, Snapshot and Siam, in the training process. (b) Uncertainty quantification method consisting of three steps: uncertainty measurement, uncertainty guidance, and uncertainty post-processing. Symbol Γ is the baseline model, $\boldsymbol{M}$ is the uncertainty mask, and $\mathcal{L}$ denotes the loss function.

consists of uncertainty measurement, uncertainty guidance, and uncertainty post-processing (see Fig.2(b)).

3.1 Snapshot and Siam

Snapshot. Snapshot is a learning strategy to ensemble multiple solutions to solve the single-value optimization question [28], promoting the diversity of models by aggressively cycling the learning rate used during a single training. We find that neighboring epochs can exhibit well constraint and under-constraint parts with inconsistent results. Thus, we choose pre-converged epochs as members to distinguish certainty and uncertainty pixels.

Siam. Siamese network (Siam) consists of two identical sub-networks [29] called twins. We use Siam to run two streams of training, where the twins in each epoch are used to compare and distinguish certainty pixels and uncertainty pixels.

We consider Snapshot and Siam to suit for different conditions. Snapshot calculates the horizontal variance of the baseline model based on the difference of consecutive epochs, which can provide more meaningful uncertainty information for the relatively weak baseline model. Siam calculates the uncertainty by measuring the vertical variance between the two sub-networks, which can provide more useful uncertainty information for models with relatively stronger performance.

3.2 Uncertainty Measurement

Snapshot collects consecutive epochs in back-forward order from the current epoch and calculates uncertainty (see Fig.3). During the training process, the variance of depth maps from different epochs is used to calculate the uncertainty information:

$$\boldsymbol{U} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{D}_i - \overline{\boldsymbol{D}}_{\text{Snapshot}})^2, \qquad (1)$$

where N is the number of closely adjacent pre-converge models, $\boldsymbol{D}_i$ is the estimated depth of model Γ at the i-th epoch and $\overline{\boldsymbol{D}}_{\text{Snapshot}}$ is the average of depth maps:

$$\overline{\boldsymbol{D}}_{Snapshot} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{D}_i.$$

We need to determine two factors. One is how many pre-converged epochs are needed, and the other is which are chosen. We search one small interval for one empirically optimal value for the first one. For the second one, we reasonably use consecutive pre-converged epochs just before the current epoch because certainty parts benefiting from the well-constraint should keep stable outputs in closely adjacent epochs.

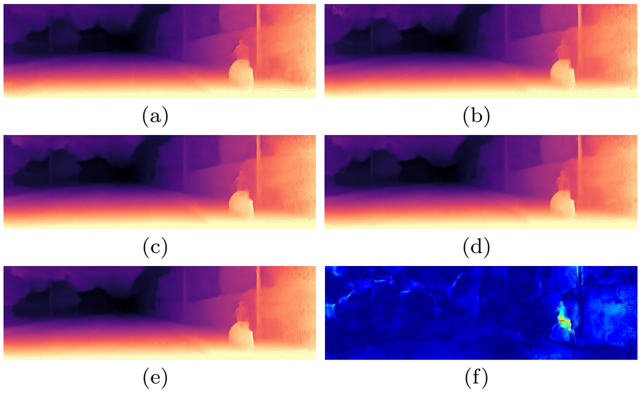


Fig.3. Five depth maps from consecutive pre-converged epochs based on the baseline model Monodepth2-M50 and the corresponding uncertainty mask. (a) Depth estimated from the 13th epoch. (b) Depth estimated from the 14th epoch. (c) Depth estimated from the 15th epoch. (d) Depth estimated from the 16th epoch. (e) Depth estimated from the 17th epoch. (f) Uncertainty mask.

Siam calculates the uncertainty in a mirroring-and-referring way, where the twin networks act like a mirror for each other to refer to and calculate uncertainty. Siam runs relatively independent streams. At the same epoch, the depth results from twins would compare and calculate the uncertainty:

$$\boldsymbol{U} = \frac{1}{2}\sum_{i=1}^{2}(\boldsymbol{D}_i - \overline{\boldsymbol{D}}_{\text{Siam}})^2,$$

where $\overline{\boldsymbol{D}}_{\text{Siam}}$ is the average of depth maps:

$$\overline{\boldsymbol{D}}_{\text{Siam}} = \frac{1}{2}\sum_{i=1}^{2}\boldsymbol{D}_i.$$

There is one factor for Siam to determine which epoch starts to estimate the uncertainty. We differently get an empirically optimal value.

3.3 Uncertainty Guidance

Here, we use the uncertainty measurement $\boldsymbol{U}$ to explicitly and spatially guide the learning of the network. We use the mean value of the uncertainty $\boldsymbol{U}$ as the threshold u, imposing the uncertainty on pixels differently:

$$u = \frac{1}{|\boldsymbol{I}|}\sum_{k\in\boldsymbol{I}}\boldsymbol{U}(k), \tag{2}$$

where $\boldsymbol{U}(k)$ is the uncertainty value at each pixel k in the image space $\boldsymbol{I}$, and $|\boldsymbol{I}|$ is the total amount of pixels in input image $\boldsymbol{I}$.

If the uncertainty value $\boldsymbol{U}(k)$ is smaller than the threshold u, we think it has not been influenced by uncertainty and should only have the definite well-constraint loss part. Conversely, the total loss can add uncertainty when the uncertainty value $\boldsymbol{U}(k)$ is greater than the threshold u. The uncertainty mask $\boldsymbol{M}$ is:

$$\boldsymbol{M}(k) = \begin{cases} 1, & \boldsymbol{U}(k) \leq u, \\ 1+\lambda\boldsymbol{U}(k), & \boldsymbol{U}(k) > u, \end{cases} \tag{3}$$

where λ is an empirical parameter to control the weight given to the uncertain pixel.

After considering the uncertainty guidance, the new loss function can be expressed as $\mathcal{L}_{\text{new}} = \boldsymbol{M}\mathcal{L}$, and $\mathcal{L}$ is the loss function of the baseline model Γ. Fig.4 demonstrates two uncertainty guidance examples on Snapshot and Siam. Uncertainty guidance can persistently concentrate on masking the rich uncertainty regions, and their area shrinks when the learning advances.

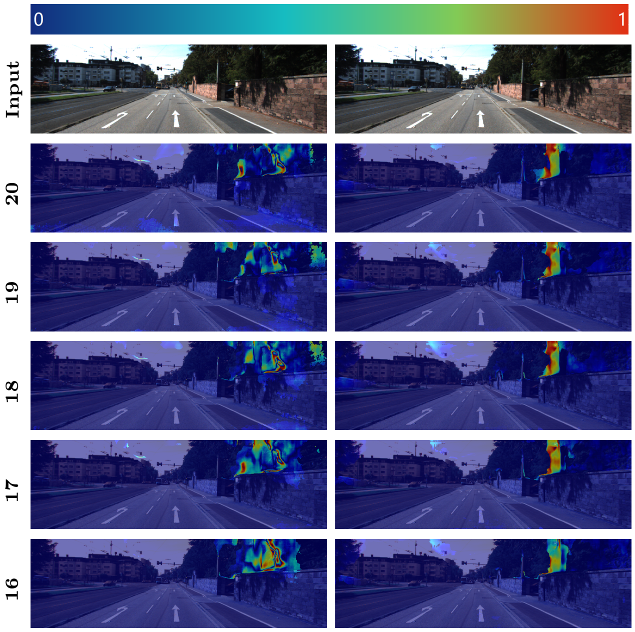
3.4 Uncertainty Post-processing

When the training process terminates, the trained model cannot completely reach the desired optimal point. It is possible to be a bit under-fit or over-fit. If the training termination is beyond the optimal point, it may cause texture copy or other artifacts. Therefore, we use the averaging result as the final output. If the last epoch in Snapshot or the better one in twin models is quite near but not reaches the optimal point, we choose the last epoch as the final output for Snapshot and the better one in twin models as the final output for Siam because it would be most close to the optimal point.

According to the principle above, the final depth after uncertainty post-processing $\boldsymbol{D}_{\text{up}}$ is determined based on the uncertainty information:

$$\boldsymbol{D}_{\text{up}}(k) = \begin{cases} \overline{\boldsymbol{D}}(k), & \boldsymbol{U}(k) \leq u, \\ \boldsymbol{D}_{\Gamma'}(k), & \boldsymbol{U}(k) > u, \end{cases}$$

where $\overline{\boldsymbol{D}}$ denotes $\overline{\boldsymbol{D}}_{\text{Snapshot}}$ or $\overline{\boldsymbol{D}}_{\text{Siam}}$, $\boldsymbol{D}_{\Gamma'}$ is the depth map of the last epoch model Γ' in Snapshot or the better twine model Γ' in the Siam, and u is the threshold in (2).



(a) (b)

Fig.4. Five uncertainty masks from the 20th epoch back to the 16th epoch. (a) Monodepth2+Snapshot-M50. (b) Hints+Siam-MS50.

3.5 Baseline Models

We choose Monodepth2 [8] and Hints [14] as the baseline model Γ to validate the proposed uncertainty quantification method, respectively. Monodepth2 and Hints are the two frequently used methods and have well-organized source codes, which could guarantee the fairness of evaluation. We do not modify the parameters and structures of the two baseline models but only impose uncertainty on their loss functions.

Monodepth2. Referring to [20, 30], the photometric reconstruction loss function $\mathcal{L}_{\mathrm{p}}$ is as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{p}}(\boldsymbol{I}_t, \boldsymbol{I}_{t'\to t}) = \frac{\alpha}{2}(1 - SSIM(\boldsymbol{I}_t, \boldsymbol{I}_{t'\to t}))\\ +(1-\alpha)\left\|\boldsymbol{I}_t - \boldsymbol{I}_{t'\to t}\right\|_1,\end{aligned}$$

where $\alpha = 0.85$ and $SSIM()$ denotes structure similarity index measure which is computed over a 3×3 pixel window [31]. The re-projected image $\boldsymbol{I}_{t'\to t}$ is generated by operation:

$$\boldsymbol{I}_{t'\to t} = \boldsymbol{I}_{t'}\langle proj(\boldsymbol{D}_t, \boldsymbol{T}_{t\to t'}, \boldsymbol{K})\rangle,$$

where $\langle . \rangle$ is the sampling operator, $\boldsymbol{T}_{t\to t'}$ is the camera relative pose, and $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$ is the camera intrinsic parameter matrix. Operation $proj()$ gets the 2D coordinates of the projected depths $\boldsymbol{D}_t$ in image $\boldsymbol{I}_{t'}$:

$$proj(\boldsymbol{D}_t, \boldsymbol{T}_{t\to t'}, \boldsymbol{K}) = \boldsymbol{K}\boldsymbol{T}_{t\to t'}\boldsymbol{D}_t(p_t)\boldsymbol{K}^{-1}p_t,$$

where p_t is a pixel coordinate.

To encourage neighboring pixels to have similar depths, an edge-aware depth smoothness loss $\mathcal{L}_{\mathrm{s}}$ weighted by image gradients is used to improve the predictions around object boundaries:

$$\mathcal{L}_{\mathrm{s}} = |\partial_{\mathrm{x}}\boldsymbol{D}_t^{\mathrm{mn}}|\mathrm{e}^{-\|\partial_{\mathrm{x}}\boldsymbol{I}_t\|} + |\partial_{\mathrm{y}}\boldsymbol{D}_t^{\mathrm{mn}}|\mathrm{e}^{-\|\partial_{\mathrm{y}}\boldsymbol{I}_t\|},$$

where ∂_{x} and ∂_{y} are gradient operations on x-axis and y-axis respectively, and $\boldsymbol{D}_t^{\mathrm{mn}}$ is the mean-normalized inverse depth.

The final loss is computed as the weighted sum of image photometric reconstruction loss $\mathcal{L}_{\mathrm{p}}$ and smoothness loss $\mathcal{L}_{\mathrm{s}}$:

$$\mathcal{L} = \mathcal{L}_{\mathrm{p}} + \mu\mathcal{L}_{\mathrm{s}},$$

where $\mu = 0.01$ is the weighting for the smoothness term.

Hints. Watson *et al.* [14] introduced depth hint to help the network escape from local minima and to guide it toward a better overall solution. A stereo matching algorithm [32] is used to get depth hint $\widetilde{\boldsymbol{D}}_t$, creating a second synthesized view $\widetilde{\boldsymbol{I}}_{t'\to t}$. They had conditions to determine whether or not to apply a supervised loss $\widetilde{\boldsymbol{D}}_t$ as the ground truth on pixel k:

$$\mathcal{L} = \begin{cases}\mathcal{L}_{\mathrm{p}}(\boldsymbol{D}(k)) + \mathcal{L}_{\mathrm{s}}^{\log L_1}(\boldsymbol{D}(k), \widetilde{\boldsymbol{D}}(k)), & \text{if } \upsilon,\\ \mathcal{L}_{\mathrm{p}}(\boldsymbol{D}(k)), & \text{otherwise,}\end{cases}$$

where $\mathcal{L}_{\mathrm{s}}^{\log L_1}(\boldsymbol{D}(k), \widetilde{\boldsymbol{D}}(k)) = \log(1+\|\boldsymbol{I}_t - \boldsymbol{I}_{t'\to t}\|_1)$, and υ denotes $\mathcal{L}_{\mathrm{p}}(\boldsymbol{D}(k)) < \mathcal{L}_{\mathrm{p}}(\widetilde{\boldsymbol{D}}(k))$.

4 Experiments

In experiments, we train the proposed uncertainty quantification method on the KITTI dataset [33] and evaluate it using the Eigen split test frames [34]. The program is implemented with Pytorch and GPU: NVIDIA GeForce GTX 2080Ti×2. In our experiments, M denotes that the training is on monocular videos, S denotes that the training is on stereo pairs, and M-S denotes that the training is on calibrated binocular videos. In all tables, the symbol ↓ means that the smaller the value is, the better the performance is. The symbol ↑ means that the bigger the value is, the better the performance is. The best result in each category is written in **bold**.

4.1 Evaluation Metrics

Depth Metrics. We use seven metrics [34] to evaluate the estimated depth results. The four error metrics measure the difference between predicted depth $\boldsymbol{D}$ and ground-truth depth $\boldsymbol{D}_{\mathrm{gt}}$: the absolute relative error (Abs Rel), the squared relative error (Sq Rel), the root mean square error (RMSE), and the logarithmic root mean square error (RMSE log). The three accuracy metrics give the fraction δ of predicted depth inside an image whose ratio and inverse ratio with the ground truth are below the thresholds $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$. For the first four metrics, the lower, the better. For the last three metrics, the higher, the better (see Table 1).

Table 1. Depth Metrics

Metric	Definition
Abs Rel	$\frac{1}{\lvert\boldsymbol{D}\rvert}\sum\limits_{k\in\boldsymbol{I}}\frac{\lvert\boldsymbol{D}(k)-\boldsymbol{D}_{\mathrm{gt}}(k)\rvert}{\boldsymbol{D}_{\mathrm{gt}}(k)}$
Sq Rel	$\frac{1}{\lvert\boldsymbol{D}\rvert}\sum\limits_{k\in\boldsymbol{I}}\frac{\lvert\boldsymbol{D}(k)-\boldsymbol{D}_{\mathrm{gt}}(k)\rvert^2}{\hat{\boldsymbol{D}}(k)}$
RMSE	$\sqrt{\frac{1}{\lvert\boldsymbol{D}\rvert}\sum\limits_{k\in\boldsymbol{I}}\lvert\boldsymbol{D}(k)-\boldsymbol{D}_{\mathrm{gt}}(k)\rvert^2}$
RMSE log	$\sqrt{\frac{1}{\lvert\boldsymbol{D}\rvert}\sum\limits_{k\in\boldsymbol{I}}\lvert\log\boldsymbol{D}(k)-\log\boldsymbol{D}_{\mathrm{gt}}(k)\rvert^2}$
Accuracy	% of $\boldsymbol{D}(k)$ s.t. $\delta = \max(\frac{\boldsymbol{D}(k)}{\boldsymbol{D}_{\mathrm{gt}}(\mathrm{k})}, \frac{\boldsymbol{D}_{\mathrm{gt}}(k)}{\boldsymbol{D}(k)}) <$ threshold

Note: $\boldsymbol{D}(k)$ is the predicted depth at each pixel k in the image space $\boldsymbol{I}$, $\boldsymbol{D}_{\mathrm{gt}}(k)$ is the corresponding ground truth depth, $|\boldsymbol{D}|$ is the total amount of pixels in input image. Three different thresholds $1.25, 1.25^2, 1.25^3$ are used in the accuracy metrics, respectively.

Uncertainty metrics. We use two metrics of the area under the sparsification error (Ause) (4) and the area under the random gain (Aurg) (5) to evaluate how significant the model uncertainties are [19]:

$$Ause(\boldsymbol{U}, \boldsymbol{D}) = \epsilon(\boldsymbol{D}) - \epsilon(\boldsymbol{D}_{\mathrm{u}}), \quad (4)$$

$$Aurg(\boldsymbol{U}, \boldsymbol{D}) = E_1(rand, \boldsymbol{D}) - E_1(\boldsymbol{U}, \boldsymbol{D}), \quad (5)$$

where $\epsilon(.)$ is the depth map error metric and $\boldsymbol{D}_{\mathrm{u}}$ is the depth map for the 2% pixels with the highest uncertainty. Abs Rel, RMSE or $\delta \geq 1.25$ (since $\delta < 1.25$ defines an accuracy score) is used as ϵ. Ause quantifies how close the estimation is to the ideal sparsification uncertainty (the lower, the better). Aurg quantifies how better it is compared with no modeling (the higher, the better).

4.2 Parameter Setting

We do several important many experiments to determine the empirical parameter λ in (3), the starting epoch N of Siam, the number of closely adjacent pre-converge models N in (1) of the Snapshot, and the threshold of uncertainty mask $\boldsymbol{M}$. We approximately enumerate multiple λ values to determine a recommended setting for the subsequent experiments. As shown in Table 2, we set $\lambda = 1$ for all experiments to reduce the computation cost. As shown in Table 3, we set the starting epoch of the Siam at the 1st epoch. As shown in Table 4, we set parameter $N = 5$ and start the uncertainty guidance of the Snapshot at the 6th epoch. In (2), the mean and other possible options like fractional mean and median can be used as the threshold to determine the uncertainty mask. Table 5 and Fig.5 show that the mean value as the threshold achieves the best performance.

Table 2. The Enumeration Experiment of the Hyperparameter λ in the Uncertainty Mask Calculation

λ	Monodepth2+Snapshot-M50			Hints+Siam-MS50		
	Abs Rel↓	RMSE↓	$\delta < 1.25$↑	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
0.6	0.110	4.574	0.881	**0.101**	4.561	0.881
0.8	0.110	4.599	0.882	0.102	**4.539**	0.881
1.0	0.109	4.551	0.885	0.102	4.546	0.880
1.2	**0.108**	**4.542**	0.884	0.102	4.563	**0.882**
1.4	0.110	4.580	**0.886**	0.102	4.572	**0.882**

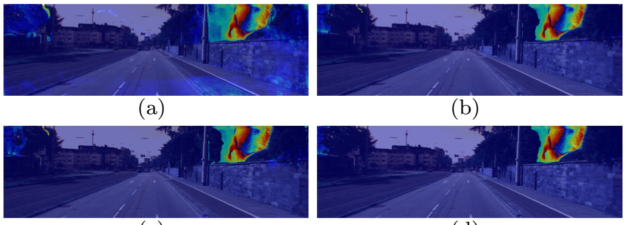


Fig. 5. Uncertainty masks from the uncertainty map $\boldsymbol{U}$ based on different thresholds on the baseline model: Monodepth2-M50. (a) The median of the uncertainty map $\boldsymbol{U}$ as the threshold. (b) The mean of the uncertainty map $\boldsymbol{U}$ as the threshold. (c) 0.8 mean of the uncertainty map $\boldsymbol{U}$ as the threshold. (d) 1.2 mean of the uncertainty map $\boldsymbol{U}$ as the threshold.

Table 3. The Enumeration Experiments of the Starting Epoch N in Siam

Model	N	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
Hints+Siam-MS50	1	**0.102**	**4.546**	0.880
Hints+Siam-MS50	3	**0.102**	4.572	**0.881**
Hints+Siam-MS50	5	**0.102**	4.568	**0.881**

Table 4. The Number of Closely Adjacent Pre-converge Models N of Snapshot

Model	N	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
Monodepth2+Snapshot-M50	3	0.110	4.593	0.883
Monodepth2+Snapshot-M50	5	**0.109**	4.551	**0.885**
Monodepth2+Snapshot-M50	7	0.146	**5.366**	0.802

Table 5. Verification Experiments of Different Thresholds in the Uncertainty Mask Calculation in (3)

Mask	Monodepth2+Snapshot-M50			Hints+Siam-MS50		
	Abs Rel↓	RMSE↓	$\delta < 1.25$↑	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
mean	**0.109**	**4.551**	**0.885**	**0.102**	**4.546**	0.880
0.8 mean	0.115	4.670	0.871	**0.102**	**4.546**	**0.882**
1.2 mean	0.115	4.702	0.873	**0.102**	4.555	0.881
median	0.111	4.584	0.882	0.103	4.584	0.878

Table 6. Uncertainty Evaluation

Method	Abs Rel		RMSE		$\delta \geq 1.25$	
	Ause↓	Aurg↑	Ause↓	Aurg↑	Ause↓	Aurg↑
[19]	0.069	0.005	3.733	0.258	0.101	0.008
Ours1	0.054	0.018	3.316	0.557	0.071	0.035
Ours2	**0.043**	**0.027**	**3.071**	**0.860**	**0.057**	**0.051**

Note: Ours1: Monodepth2+Snapshot-M50 and Ours2: Hint+Siam-MS50.

4.3 Performance Evaluation

We evaluate the accuracy of the depth and how significant the model uncertainties are. A total of six conditions of Monodepth2, Monodepth2+Snapshot, Monodepth2+Siam, Hints, Hints+Snapshot, and Hints+Siam are taken into evaluation. We use three

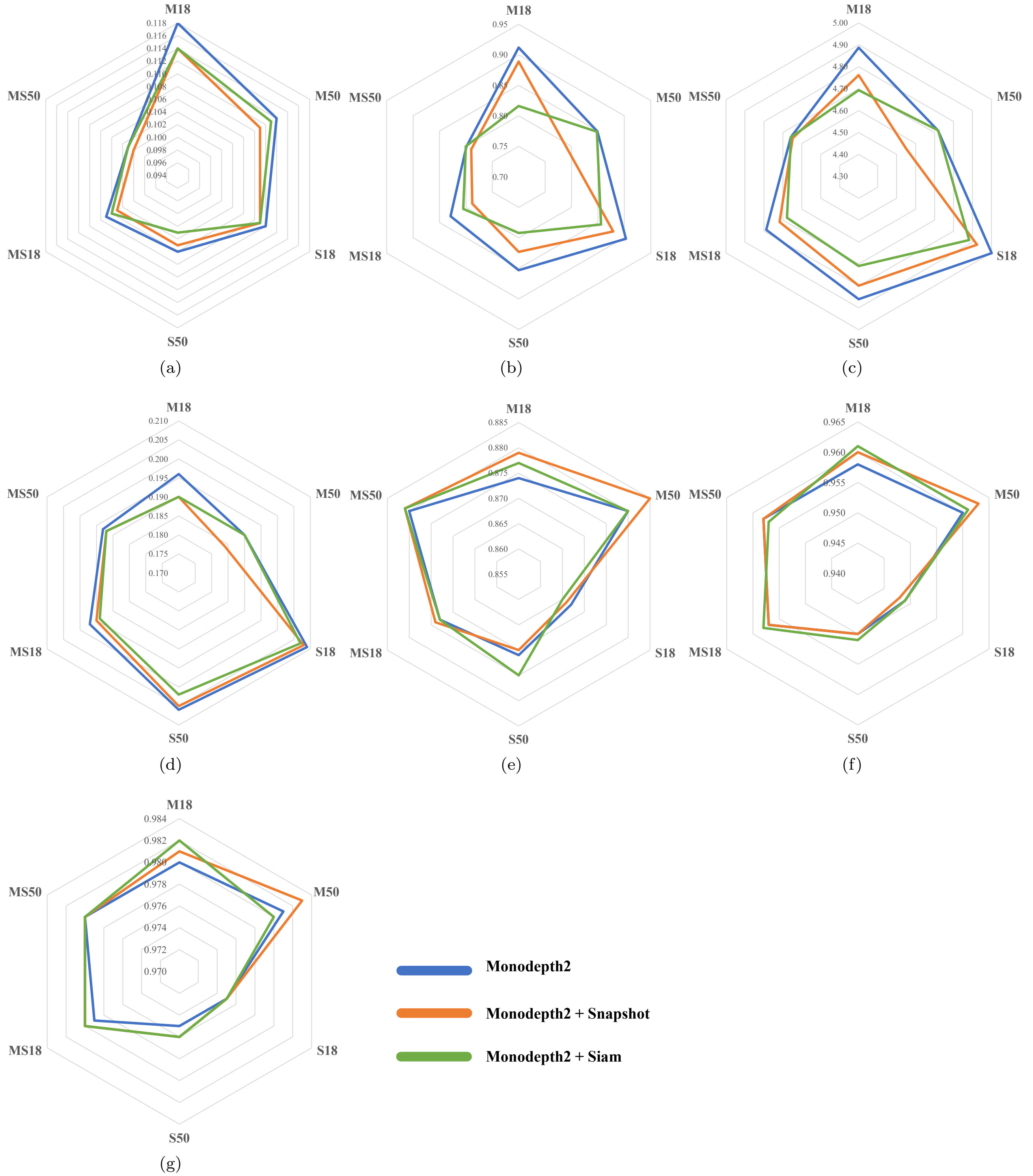
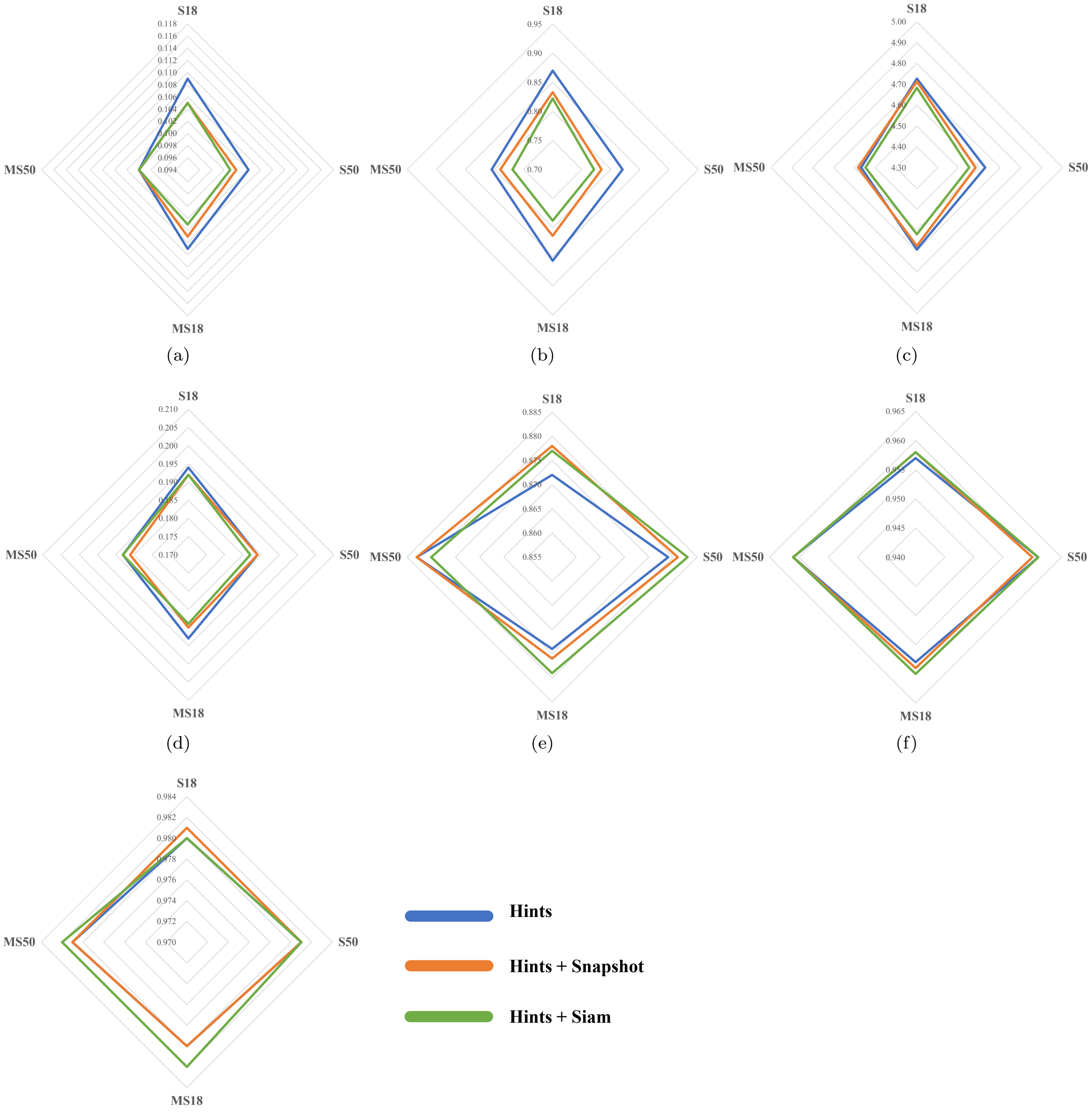


Fig.6. Quantitative performance evaluation of Snapshot and Siam on the baseline model Monodepth2 [8] with seven depth metrics. One radar chart illustrates one metric and an axis of the radar chart represents one training paradigm (M, S, or MS). (a) Abs Rel. (b) Sq Rel. (c) RMSE. (d) RMSE log. (e) $\delta < 1.25$. (f) $\delta < 1.25^2$. (g) $\delta < 1.25^3$.



(a) (b) (c)

(d) (e) (f)

(g)

Fig. 7. Quantitative performance evaluation of Snapshot and Siam on the baseline model Hints [14] with seven depth metrics. One radar chart illustrates one metric and an axis of the radar chart represents one training paradigm (M, S, or MS). (a) Abs Rel. (b) Sq Rel. (c) RMSE. (d) RMSE log. (e) $\delta < 1.25$. (f) $\delta < 1.25^2$. (g) $\delta < 1.25^3$.

Table 7. Ablation Study on the Baseline Model Monodepth2

Backbone [8]	Train	UG	UP	Snapshot			Siam		
				Abs Rel↓	RMSE↓	$\delta < 1.25$↑	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
Monodepth2-18	M	✕	✕	0.118	4.887	0.874	0.118	4.887	0.874
Monodepth2-18	M	✓	✕	0.116	4.807	0.876	0.115	4.784	0.876
Monodepth2-18	M	✓	$\overline{\boldsymbol{D}}$	0.115	4.807	0.874	0.116	4.784	0.873
Monodepth2-18	M	✓	✓	**0.114**	**4.762**	**0.879**	**0.114**	**4.693**	**0.877**
Monodepth2-50	M	✕	✕	0.112	4.718	0.880	0.112	4.718	**0.880**
Monodepth2-50	M	✓	✕	**0.109**	4.556	**0.885**	**0.111**	4.714	0.879
Monodepth2-50	M	✓	$\overline{\boldsymbol{D}}$	0.110	4.560	0.881	**0.111**	4.716	0.878
Monodepth2-50	M	✓	✓	**0.109**	**4.551**	**0.885**	**0.111**	**4.712**	**0.880**
Monodepth2-18	S	✕	✕	0.110	5.001	**0.867**	0.110	5.001	**0.867**
Monodepth2-18	S	✓	✕	0.110	4.950	0.864	0.109	4.921	0.865
Monodepth2-18	S	✓	$\overline{\boldsymbol{D}}$	**0.109**	4.957	0.866	**0.108**	4.890	0.866
Monodepth2-18	S	✓	✓	**0.109**	**4.924**	0.866	0.109	**4.882**	0.865
Monodepth2-50	S	✕	✕	0.106	4.861	**0.871**	0.106	4.861	0.871
Monodepth2-50	S	✓	✕	**0.105**	4.816	0.870	0.105	4.803	0.872
Monodepth2-50	S	✓	$\overline{\boldsymbol{D}}$	**0.105**	4.833	0.868	0.104	4.780	0.874
Monodepth2-50	S	✓	✓	**0.105**	**4.799**	0.870	**0.103**	**4.709**	**0.875**
Monodepth2-18	MS	✕	✕	0.107	4.788	0.873	0.107	4.788	**0.873**
Monodepth2-18	MS	✓	✕	0.106	4.725	0.873	**0.106**	4.714	0.871
Monodepth2-18	MS	✓	$\overline{\boldsymbol{D}}$	0.108	4.723	0.871	0.107	4.715	0.872
Monodepth2-18	MS	✓	✓	**0.105**	**4.717**	**0.874**	**0.106**	**4.678**	**0.873**
Monodepth2-50	MS	✕	✕	0.103	4.658	0.880	**0.103**	4.658	0.880
Monodepth2-50	MS	✓	✕	**0.102**	4.650	0.880	0.104	4.650	**0.881**
Monodepth2-50	MS	✓	$\overline{\boldsymbol{D}}$	0.103	4.651	0.880	0.104	4.651	0.880
Monodepth2-50	MS	✓	✓	**0.102**	**4.648**	**0.881**	**0.103**	**4.649**	**0.881**

Table 8. Ablation Study on the Baseline Model Hints

Backbone [14]	Train	UG	UP	Snapshot			Siam		
				Abs Rel↓	RMSE↓	$\delta < 1.25$↑	Abs Rel↓	RMSE↓	$\delta < 1.25$↑
Hints-18	S	✕	✕	0.109	4.812	0.872	0.109	4.812	0.872
Hints-18	S	✓	✕	0.107	4.742	0.876	0.107	4.747	0.875
Hints-18	S	✓	$\overline{\boldsymbol{D}}$	0.106	4.763	0.874	0.106	4.748	0.874
Hints-18	S	✓	✓	**0.105**	**4.714**	**0.878**	**0.105**	**4.683**	**0.877**
Hints-50	S	✕	✕	0.104	4.677	0.879	0.104	4.677	0.879
Hints-50	S	✓	✕	0.103	4.604	0.879	0.102	4.581	0.881
Hints-50	S	✓	$\overline{\boldsymbol{D}}$	0.104	4.613	0.879	0.102	4.576	0.882
Hints-50	S	✓	✓	**0.102**	**4.582**	**0.881**	**0.101**	**4.551**	**0.883**
Hints-18	MS	✕	✕	0.107	4.780	0.874	0.107	4.780	0.874
Hints-18	MS	✓	✕	0.105	4.726	0.875	0.105	4.654	0.877
Hints-18	MS	✓	$\overline{\boldsymbol{D}}$	**0.104**	4.727	**0.876**	0.107	4.649	0.876
Hints-18	MS	✓	✓	0.105	**4.676**	**0.876**	**0.103**	**4.620**	**0.879**
Hints-50	MS	✕	✕	**0.102**	4.629	**0.883**	**0.102**	4.629	**0.883**
Hints-50	MS	✓	✕	**0.102**	4.602	0.882	0.103	4.599	0.879
Hints-50	MS	✓	$\overline{\boldsymbol{D}}$	0.103	4.602	0.881	0.104	4.599	0.881
Hints-50	MS	✓	✓	**0.102**	**4.582**	**0.883**	**0.102**	**4.546**	0.880

training paradigms: stereo pairs, monocular videos, and calibrated binocular videos. We use the convolutional neural networks ResNet18 and ResNet50 to extract the image features. Because the stereo matching algorithm needs a stereo pairs image, Hints can not train on monocular videos.

Depth Evaluation. Fig.6 and Fig.7 illustrate the accuracy of the estimated depth on the two baseline models Monodepth2 and Hints, respectively. Snapshot and Siam have improved the accuracy of the two baseline models on all training paradigms and ResNet18/50.

Uncertainty Evaluation. We quantitatively evaluate the model uncertainty using the two uncertainty metrics. Monodepth2+Snap+Self-M50 is the best model in the existing uncertainty method [19]. Monodepth2+Snapshot-M50 and Hint+Siam-MS50 are the optimal models in our results. Table 6 summarizes the quantitative evaluation results on the two uncertainty metrics. We can see that our results are better than the method [19].

Table 9. Quantitative Comparison with Existing Methods

Method	Year	Train	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Zhou *et al.* [21]	2017	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
GeoNet [22]	2018	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DF-Net [23]	2018	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Ranjan *et al.* [35]	2019	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
Struct2depth [36]	2019	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [8]⋆ ‡	2019	M	0.118	0.912	4.887	0.196	0.874	0.958	0.980
Klingner *et al.* [10]	2020	M	0.117	0.907	4.844	0.196	0.875	0.958	0.980
PackNet [11]	2020	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Johnston *et al.* [37]	2020	M	**0.106**	0.861	4.699	0.185	**0.899**	0.962	0.982
Poggi *et al.* [19]⊘	2020	M	0.112	0.838	4.691	0.186	0.881	0.961	0.983
Bian *et al.* [9]	2021	M	0.126	0.920	5.245	0.208	0.840	0.949	0.979
SD-SSMDE [38]	2022	M	0.108	**0.751**	**4.485**	**0.180**	0.885	**0.964**	**0.984**
Ours1 ((Monodepth2+Snapshot-50))	-	M	0.109	0.792	4.551	0.184	0.885	0.963	0.983
Garg *et al.* [7]	2016	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth R50 [20]	2017	S	0.133	1.142	5.533	0.230	0.830	0.936	0.970
StrAT [39]	2018	S	0.128	1.019	5.403	0.227	0.827	0.935	0.971
Poggi *et al.* [40]	2018	S	0.129	0.996	5.281	0.223	0.831	0.939	0.974
Monodepth2 [8]*	2019	S	0.110	0.903	5.001	0.209	0.867	0.949	0.975
Hints [14]* ‡	2019	S	0.109	0.870	4.812	0.194	0.872	0.957	0.980
Poggi *et al.* [19]⊘	2020	S	0.108	0.835	4.856	0.202	0.865	0.951	0.977
Wavelet Decomposition [12]	2021	S	0.105	0.813	4.625	0.191	0.879	0.959	**0.981**
Ours(Hints+Siam-50)	-	S	**0.101**	**0.771**	**4.551**	**0.187**	**0.883**	**0.961**	**0.981**
Hints [14]* ‡	2019	MS	0.107	0.857	4.780	0.193	0.874	0.958	0.980
Monodepth2 [8]*	2019	MS	0.107	0.829	4.788	0.197	0.873	0.957	0.979
Poggi *et al.* [19]⊘	2020	MS	0.104	0.783	4.654	0.190	0.876	0.958	0.981
Ours(Hints+Siam-50)	-	MS	**0.102**	**0.769**	**4.546**	**0.188**	**0.880**	**0.961**	**0.982**

Note: ⋆ denotes the model is retrained, ‡ denotes the baseline method, and ⊘ denotes the uncertainty method [19].

Ablation Study. We conduct an ablation study to validate the effectiveness of the uncertainty guidance and post-processing. We switch on and off uncertainty guidance and post-processing in all three training paradigms on baseline models Monodepth2 and Hints. Table 7 and Table 8 present the complete result, respectively, where UG denotes uncertainty guidance, UP denotes uncertainty post-processing, ✓denotes ”turn on the corresponding step”, and ✕ denotes ”turn off the corresponding step”. We can see that uncertainty guidance and uncertainty post-processing can improve the accuracy of the depth estimation network, respectively. The best result is to use both uncertainty guidance and uncertainty post-processing.

Comparison with Existing Methods. We make comprehensive comparisons with the current representative methods. Table 9 presents the quantitative results of the estimated depth. For the baseline model Monodepth2, Monodepth2+Snapshot-M50 achieves the best result. For the baseline model Hints, Hints+Siam-MS50 achieves the best result. Our proposed uncertainty quantification improves the accuracy of the two baseline models. Compared with the uncertainty work [19], our results are superior. Fig.8 demonstrates a group of resulted depth maps. The objects have complete structures and sharp boundaries in our depth maps.

5 Conclusion

This paper proposes a novel uncertainty quantification method to train a self-supervised monocular depth estimation network. The uncertainty quantification method contains uncertainty measurement, guidance, and post-processing. Experimental results on the KITTI dataset show that our approach can improve the depth estimation accuracy of seven evaluation metrics and exceeds the existing uncertainty method in two uncertainty metrics. We also demonstrated the effectiveness of the proposed uncertainty guidance and post-processing. They can improve the accuracy of the depth estimation network. The uncertainty quantification can be conveniently generalized to other deep-learning work.

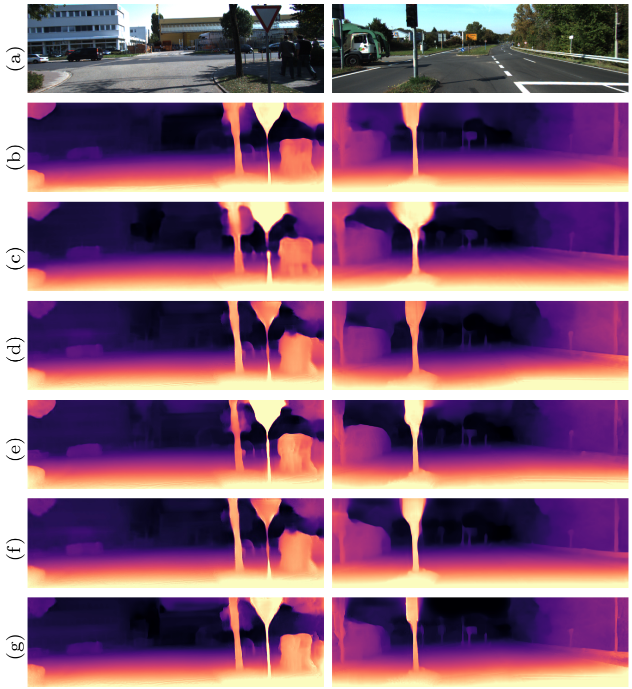


Fig.8. Qualitative comparison with existing methods. (a) RGB input. (b) Depth estimated by Monodepth2 [8]. (c) Depth estimated by Hints [14]. (d) Depth estimated by PackNet [11]. (e) Depth estimated by Klingner *et al.* [10]. (f) Depth estimated by Poggi *et al.* [19]. (g) Depth estimated by our method.

References

[1] Zhao X R, Wang X, Chen Q C. Temporally consistent depth map prediction using deep convolutional neural network and spatial-temporal conditional random field. *Journal of Computer Science and Technology*, 2017, 32(3): 443-456. DOI: 10.1007/s11390-017-1735-x.

[2] Fang F, Luo F, Zhang H P, Zhou H J, Chow A L, Xiao C X. A comprehensive pipeline for complex text-to-image synthesis. *Journal of Computer Science and Technology*, 2020, 35(3), 522-537. DOI: 10.1007/s11390-020-0305-9.

[3] Cao T, Luo F, Fu Y, Zhang W, Zheng S, Xiao C. DGECN: A depth-guided edge convolutional network for end-to-end 6D pose estimation. In *Proc. the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 3773-3782. DOI: 10.1109/CVPR52688.2022.00376.

[4] Fu Y, Yan Q, Liao J, Xiao C. Joint texture and geometry optimization for rgb-d reconstruction. In *Proc. the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, June 2020, pp.5949-5958. DOI: 10.1109/CVPR42600.2020.00599.

[5] Fu Y, Yan Q, Yang L, Liao J, Xiao C. Texture mapping for 3d reconstruction with rgb-d sensor. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp.4645-4653. DOI: 10.1109/CVPR.2018.00488.

[6] Fu Y, Yan Q, Liao J, Zhou H, Tang J, Xiao C. Seamless texture optimization for RGB-D reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2023, 29(3): 1845-1859. DOI: 10.1109/TVCG.2021.3134105.

[7] Garg R, Vijay Kumar B G, Carneiro G, Reid I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proc. the 14th European Conference on Computer Vision*, October 2016, pp.740-756. DOI: 10.1007/978-3-319-46484-8_45.

[8] Godard C, Aodha O M, Firman M, Brostow G. Digging into self-supervised monocular depth estimation. In *Proc. the 2019 IEEE/CVF International Conference on Computer Vision*, October 2019, pp.3828-3838. DOI: 10.1109/ICCV.2019.00393.

[9] Bian J W, Zhan H, Wang N, Li Z, Zhang L, Shen C, Cheng M M, Reid I. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision*, 2021, 129(9): 2548-2564. DIO: 10.1007/s11263-021-01484-6.

[10] Klingner M, Termöhlen J A, Mikolajczyk J, Fingscheidt T. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Proc. the 16th European Conference on Computer Vision*, November 2020, pp.582-600. DOI: 10.1007/978-3-030-58565-5_35.

[11] Guizilini V, Ambrus R, Pillai S, Raventos A, Gaidon A. 3d packing for self-supervised monocular depth estimation. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp.2485-2494. DOI: 10.1109/CVPR42600.2020.00256.

[12] Ramamonjisoa M, Firman M, Watson J, Lepetit V, Turmukhambetov D. Single image depth prediction with wavelet decomposition. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021, pp.11084-11093. DOI: 10.1109/CVPR46437.2021.01094.

[13] Li Y, Luo F, Xiao C. Self-supervised coarse-to-fine monocular depth estimation using a lightweight attention module. *Computational Visual Media*, 2022, 8(4): 631-647. DOI: 10.1007/s41095-022-0279-3.

[14] Watson J, Firman M, Brostow G, Turmukhambetov D. Self-supervised monocular depth hints. In *Proc. the 2019 IEEE/CVF International Conference on Computer Vision*, October 2019, pp.2162-2171. DOI: 10.1109/ICCV.2019.00225.

[15] Asai A, Ikami D, Aizawa K. Multi-task learning based on separable formulation of depth estimation and its uncertainty. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, June 2019, pp.21-24.

[16] Mertan A, Sahin Y H, Duff D J, Unal G. A new distributional ranking loss with uncertainty: Illustrated in relative

depth estimation. In *Proc. the 2020 International Conference on 3D Vision*, November 2020, pp.1079-1088. DOI: 10.1109/3DV50981.2020.00118.

[17] Teixeira L, Oswald M R, Pollefeys M, Chli M. Aerial single-view depth completion with image-guided uncertainty estimation. *IEEE Robotics and Automation Letters*, 2020, 5(2): 1055-1062. DOI: 10.1109/LRA.2020.2967296.

[18] Choi H, Lee H, Kim S, Kim S, Kim S, Sohn K, Min D. Adaptive confidence thresholding for monocular depth estimation. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 12788-12798. DOI: 10.1109/ICCV48922.2021.01257.

[19] Poggi M, Aleotti F, Tosi F, Mattoccia S. On the uncertainty of self-supervised monocular depth estimation. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp.3224-3234. DOI: 10.1109/CVPR42600.2020.00329.

[20] Godard C, Aodha O M, Brostow G J. Unsupervised monocular depth estimation with left-right consistency. In *Proc. the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2017, pp.6602-6611. DOI: 10.1109/CVPR.2017.699.

[21] Zhou T, Brown M, Snavely N, Lowe D G. Unsupervised learning of depth and ego-motion from video. In *Proc. the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2017, pp.6612-6619. DOI: 10.1109/CVPR.2017.700.

[22] Yin Z, Shi J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proc. the 2018 IEEE/CVF conference on Computer Vision and Pattern Recognition*, June 2018, pp.1983-1992. DOI: 10.1109/CVPR.2018.00212.

[23] Zou Y, Luo Z, Huang J B. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proc. the 15th European Conference on Computer Vision*, September 2018, pp.38-55. DOI: 10.1007/978-3-030-01228-1_3.

[24] Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR, Makarenkov V, Nahavandi S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021, 76: 243-297. DOI: 10.1016/j.inffus.2021.05.008.

[25] Liu C, Gu J, Kim K, Narasimhan S G, Kautz J. Neural rgb→d sensing: Depth and uncertainty from a video camera. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019, pp.10978-10987. DOI: 10.1109/CVPR.2019.01124.

[26] Song C, Qi C, Song S, Xiao F. Unsupervised monocular depth estimation method based on uncertainty analysis and retinex algorithm. *Sensors*, 2020, 20(18):5389, 1-11. DOI: 10.3390/s20185389.

[27] Shen Y, Zhang Z, Sabuncu M R, Sun L. Real-time uncertainty estimation in computer vision via uncertainty-aware distribution distillation. In *Proc. the 2021 IEEE Winter Conference on Applications of Computer Vision*, January 2021, pp.707-716. DOI: 10.1109/WACV48630.2021.00075.

[28] Huang G, Li Y, Pleiss G, Liu Z, Hopcroft J E, Weinberger K Q. Snapshot ensembles: Train 1, get m for free. arXiv:1704.00109, 2017. https://doi.org/10.48550/arXiv.1704.00109, April 2023.

[29] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition. In *Proc. the International Conference on Machine Learning Workshop*, July 2015.

[30] Zhao H, Gallo O, Frosio I, Kautz J. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 2017, 3(1): 47-57. DOI: 10.1109/TCI.2016.2644865.

[31] Wang Z, Bovik A C, Sheikh H R, Simoncelli E P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004, 13(4): 600-612. DOI: 10.1109/TIP.2003.819861.

[32] Hirschmüller H. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005, pp.807-814. DOI: 10.1109/CVPR.2005.56.

[33] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. the 2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2012, pp.3354-3361. DOI: 10.1109/CVPR.2012.6248074.

[34] Eigen D, Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. the 2015 IEEE/CVF International Conference on Computer Vision*, December 2015, pp.2650-2658. DOI: 10.1109/ICCV.2015.304.

[35] Ranjan A, Jampani V, Balles L, Kim K, Sun D, Wulff J, Black M J. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019, pp.12232-12241. DOI: 10.1109/CVPR.2019.01252.

[36] Casser V, Pirk S, Mahjourian R, Angelova A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proc. the AAAI Conference on Artificial Intelligence*, July 2019, pp.8001-8008. DOI: 10.1609/aaai.v33i01.33018001.

[37] Johnston A, Carneiro G. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proc. the 2020 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, June 2020, pp. 4755-4764. DOI: 10.1109/CVPR42600.2020.00481.

[38] Petrovai A, Nedevschi S. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *Proc. the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp.1568-1578. DOI: 10.1109/CVPR52688.2022.00163.

[39] Mehta I, Sakurikar P, Narayanan P J. Structured adversarial training for unsupervised monocular depth estimation. In *Proc. the 2018 International Conference on 3D Vision*, September 2018, pp.314-323. DOI: 10.1109/3DV.2018.00044.

[40] Poggi M, Tosi F, Mattoccia S. Learning monocular depth estimation with unsupervised trinocular assumptions. In *Proc. the 2018 International Conference on 3D Vision*, September 2018, pp. 324-333. DOI: 10.1109/3DV.2018.00045.