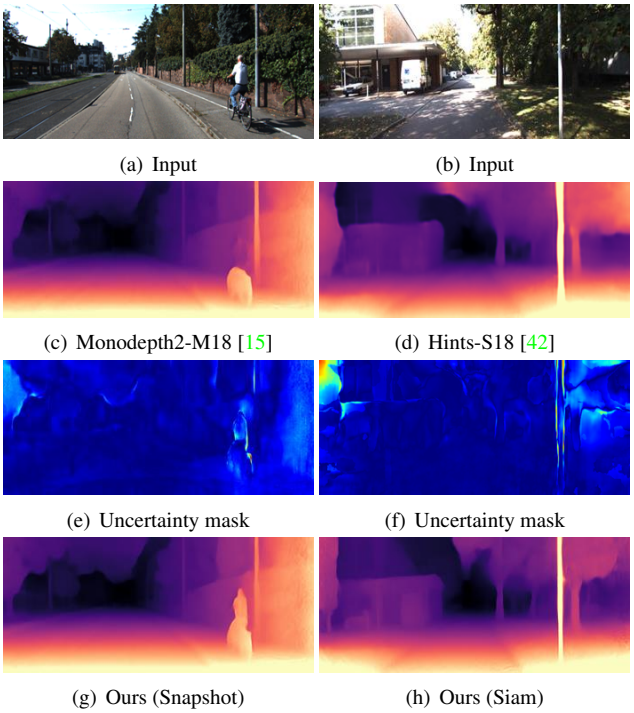
Self-Supervised Monocular Depth Estimation by Digging into Uncertainty Quantification

Yuanzhen Li, Shengjie Zheng, Zixin Tan, Tuo Cao, Fei Luo*, Chunxia Xiao*
School of Computer Science, Wuhan University
Wuhan, Hubei, China
{yuanzhen, maplect, luofei, cxxiao}@whu.edu.cn

Abstract

Based on well-defined objective functions and well-designed network architectures, self-supervised monocular depth estimation has greatly advanced the development of depth estimation. However, due to exceptions such as moving objects and occlusion, accurate depth inference in real scenes still needs to be within the scope of current explicit optimization restrictions. Therefore, the under-constraint issue is a common problem for existing methods, which reflects the uncertainty of the estimated depth map. Consequently, we dig into uncertainty quantification, which includes how to measure uncertainty and promote learning performance with uncertainty. Concretely, with Snapshot and Siam learning, we focus on the learning difficulty difference between certainty and uncertainty and measure the uncertainty degree by calculating the variance of pre-converged epochs or twins in training. Then we leverage the uncertainty to guide the network model to strengthen learning on uncertainty regions in the scene. Finally, we propose uncertainty post-processing, adaptively producing final depth with the balance of accuracy and robustness. We choose Monodepth2 and Hints as baseline models, carry out the comprehensive comparison and ablation study experiments to verify the validity of our uncertainty quantification method.

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



(a) Input (b) Input (c) Monodepth2-M18 [15] (d) Hints-S18 [42] (e) Uncertainty mask (f) Uncertainty mask (g) Ours (Snapshot) (h) Ours (Siam)

Figure 1: Compared with the baseline methods on the Eigen split test dataset [7]. (a-b) input images, (c-d) baseline results: Monodepth2 [15] and Hints [42], (e-f) uncertainty mask of depth, and (g-h) our results.

1. Introduction

Depth estimation is a fundamental task in computer graphics and computer vision, which can be used SLAM [4], autonomous driving [19], scene reconstruction [8–10], etc. Besides the industrial distance range devices like Light Detection and Ranging (LiDAR) and Time of Flight (ToF), incorporating consumer-grade cameras and machine learning-based methods can estimate the relative depth of a scene. Estimating depth from a single RGB image is an ill-posed problem. However, machine learning methods, including the popular deep learning, let monocular depth estimation become a possibility in the application, which learn the relationship between the spatial distance and RGB features from a large dataset. Unlike supervised methods, self-supervised methods do not need costly ground truth depth.

To improve self-supervised monocular depth estimation, some methods [3,12,15] introduce novel loss function items to optimize new objectives. Other methods [16, 21, 24, 34] modify the network architectures or add functional modules to focus on the special depth estimation effect. The pre-

processing or post-processing [22,42,43] is also considered to argue data usage. However, these techniques cannot solve some self-supervised monocular depth estimation defects. The reasons lie in the following aspects.

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 boundary in the depth map is limited by the number of known objects. On the other hand, self-supervised based estimation is an under-constraint task due to needing more adequate optimization objectives to restrict the factors such as low texture, motion objects, varying illumination, occlusion, and so on. Solely depending on improving neural network architecture is hard to solve the above issues.

Uncertainty quantification is an effective strategy to improve the accuracy of the depth estimation model. There are some methods [2,6,29,32,38] discuss uncertainty, but several weaknesses still exist. First, these methods are based on ground truth depth to obtain uncertainty. Second, the proposed functional modules or models must solve the under-constraint problem. Third, they do not significantly distinguish between the learning difficulty of certainty and uncertainty regions in the training scenes.

In this paper, we propose an uncertainty quantification strategy 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 (Fig. 2). Thus, we propose to base the variance of consecutive epoch results, estimate the uncertainty regions and guide the network to learn them. Our uncertainty quantification consists of uncertainty measurement, uncertainty guidance, and uncertainty post-processing. Based on our simple but effective uncertainty quantification method, the regions associated with uncertainty in a scene can be detected and better learned (Fig. 1).

Our contributions can be summarized as follows:

• We propose to use consecutive training epochs or the Siam network to measure the uncertainty of the depth map. Then, the estimation uncertainty mask is used to guide the depth network model learning.

• We propose ensemble-based uncertainty post-processing, adaptively producing final depth results with a balance of accuracy and robustness.

• The proposed uncertainty quantification method does not add additional modules, which could avoid substantially modifying the baseline model. It can be conveniently and effectively generalized to other self-supervised monocular depth estimation methods.

2. Related Work

Self-supervised monocular depth estimation Garg *et al.* [12] established the cornerstone of self-supervised monocular depth estimation. The photometric reconstruction loss is the core loss function of self-supervised monocular depth estimation. This loss measures the difference between a reference image and the depth-guided re-projection of other views into that reference viewpoint. Monodepth [14] inputs a left image into a depth network and predicts left-right disparities to enforce consistency between the disparities produced relative to both the left and right images.

Zhou *et al.* [48] first proposed by using monocular video to train the monocular depth estimation model. Monodepth2 [15] makes the following three innovations. First, they proposed a minimum photometric re-projection loss to address the problem of occluded pixels. Then, they designed an auto-masking loss to ignore training pixels that violate relative camera motion assumptions. Finally, they upsampled the predicted depth maps to the input resolution and computed all losses to reduce texture-copy artifacts. SC-Depth [3] proposes geometry consistency loss that penalizes 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 propose using multi-task training strategies to improve the accuracy of depth estimate. GeoNet [46] and DF-Net [49] propose a jointly learning framework for monocular depth, optical flow, and ego-motion estimation from videos. The three components are coupled by the nature of 3D scene geometry, jointly learned by the framework in an end-to-end manner. They used the predicted depth and optical flow to mask motion objects during training. Klingner *et al.* [21] used the learned semantic information to eliminate the influence of moving objects on computing photometric re-projection loss.

Some methods use semi-supervised ways to train the model. Jamie *et al.* [42] used the classical disparity map estimation algorithm SGM [17] from a rectified stereo image pair to provide the depth hints for the network. Klingner *et al.* [21] added a semantic segmentation network to detect moving objects, which aimed to prevent moving objects from contaminating the photometric reconstruction. SD-SSMDE [30] presents a novel self-distillation based monocular depth estimation learning framework. First, train a self-supervised high-resolution depth estimation model as pseudo depth labels. It was then based on the pseudo-depth labels to train the depth estimation network. According to the photometric reconstruction principle, most existing methods strongly depend on image features. As image depth estimation is a pixel-level estimation task, the fact that the scene is only observed from a single view can lead to inconsistent or missing information in the generated virtual

views. Furthermore, factors like low texture, motion objects, occlusion, and poor illumination are beyond the photometric reconstruction or other relevant optimization restrictions and will cause estimation ambiguity in these pixel regions. In theory, data-driven deep learning can alleviate under-constraint influence, but it needs to identify these uncertain regions and learn more.

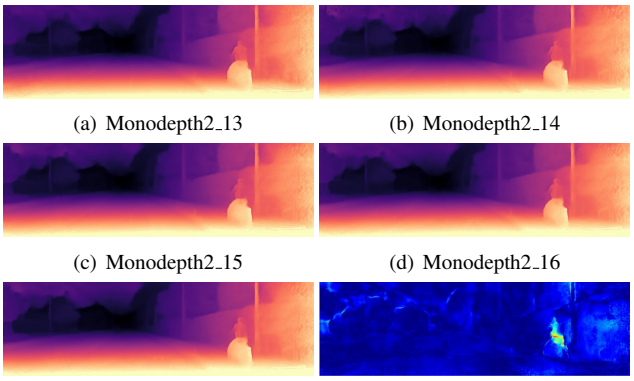
Uncertainty in depth estimation. The machine learning community usually treats under-constraint as an uncertainty problem [1]. Liu *et al.* [25] proposed a systematical discussion on uncertainty in depth estimation. Song *et al.* [37] proposed that the uncertainty of neural networks is generally divided into random and model uncertainty. Random uncertainty is from sensor noise, and motion noise may cause the observation data to be inaccurate. Model uncertainty is from the model parameters and model structures.

Random uncertainty. Choi *et al.* [6] proposed a model with a monocular depth network, confidence network, and threshold network. They distilled the data set with the confidence and threshold networks to supervise the monocular depth network. Shen *et al.* [36] supposed that the noise in the data set obeys the Gaussian distribution, used the Teacher-Student model to distill the data set, and modeled its uncertainty.

Model uncertainty. Unlike previous Bayesian-based approaches, Kendall *et al.* [2] proposed a deep learning model that estimated depth and confidence values. This multi-task approach separately formulates depth and uncertainty estimation to balance depth regression and uncertainty estimation. Mertan *et al.* [29] statistically developed the relative depth estimation problem as a maximum likelihood estimation. They assumed the pixel 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.* [38] constructed two confidence depth completion networks and a loss network, which conducted depth completion and confidence estimation with an image-guided approach. The confidence map filters out unreliable depth estimation to obtain a more accurate result.

Poggi *et al.* [32] was the first to summarize the uncertainty quantification of depth estimation comprehensively. This work analyzed three uncertainty categories estimation strategies, including empirical estimation, predictive estimation, and Bayesian estimation. Among these three types, both predictive estimation and Bayesian estimation need the extra uncertainty estimation model. 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.

Our method is an empirical estimation. The difference between the empirical estimation [32] and our method is that we use consecutive epochs or Siamese network to estimate uncertainty and convert it into a spatial mask, guiding network model learning and ensemble learning post-processing work.



(a) Monodepth2_13 (b) Monodepth2_14 (c) Monodepth2_15 (d) Monodepth2_16 (e) Monodepth2_17 (f) Uncertainty mask

Figure 2: Five depth outputs from consecutive pre-converged epochs from 13 to 17 based on the baseline model (Monodepth2-M50) and the uncertainty mask.

3. Method

This paper proposes an uncertainty quantification strategy (UQ) to train self-supervised monocular depth estimation. Our goal function can be expressed as follows:

$$\mathrm{UQ}(\Gamma, \mathrm{M}), \tag{1}$$

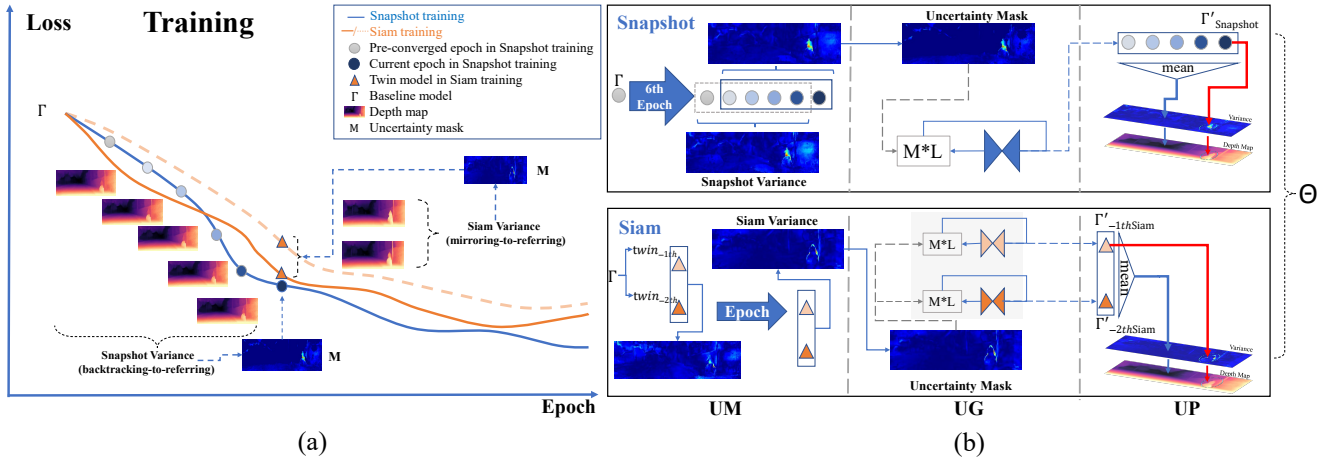
where Γ is the baseline model, M is an uncertainty mask constructed by the uncertainty information over all pixels of the depth map to identify uncertainty positions and measure uncertainty degree.

The overview of the proposed uncertainty quantification is illustrated in Fig. 3. We use the consecutive Snapshot and Siam models to implement our uncertainty quantification strategy (Fig. 3(a)). The uncertainty quantification consists of uncertainty measurement, uncertainty guidance, and uncertainty post-processing (Fig. 3(b)). In the subsequent sections, we will introduce the technical details.

3.1. Snapshot and Siam

Poggi *et al.* [32] summarised the three types of uncertainty: predictive estimation, Bayesian estimation, and empirical quantification. The predictive estimation and Bayesian estimation need the extra uncertainty estimate models. We follow the way of empirical quantification to avoid modifying the prototype of the given depth estimation model and further explore its potential from the uncertainty perspective. We use Snapshot and Siam training approaches.

Snapshot. Snapshot is a method to ensemble multiple solutions to solve the single-value optimization question [18].



(a) (b)

Figure 3: Overview of the proposed uncertainty quantification method. (a) Two empirical uncertainty quantification approaches are Snapshort and Siam in the training process. (b) The uncertainty quantification strategy contains three steps: uncertainty measurement (UM), uncertainty guidance (UG), and uncertainty post-processing (UP). Symbol L is loss function.

Snapshot aims to promote the diversity of models by aggressively cycling the learning rate used during a single training. Choose N snapshots from a single training by leveraging cyclic learning rate schedules to obtain C pre-converged epochs. At each training iteration, the learning rate λ_t is derived from the following equation on parameters of the initial learning rate λ_0, the total number of steps T and cycles C:

$$\lambda_t = \frac{\lambda_0}{2}(\cos(\frac{\pi \cdot \mathrm{mod}(t-1, \lceil \frac{T}{C} \rceil)}{\lceil \frac{T}{C} \rceil}) + 1). \qquad (2)$$

In our work, we choose consecutive pre-converged epochs as members to distinguish certainty pixels and uncertainty pixels. We propose this strategy because the neighboring epochs have similar prediction abilities for well-constraint parts; inconsistent output results where the under-constraint is hard to understand.

Siam. Siamese Network is a siamese neural network (Siam). 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. Our Siam has the same network structure, initialization parameters, and training process. In the training process of Siam, the variance between two sub-networks is used to evaluate the uncertainty, and the uncertainty information is used to guide the network learning. We use the two sub-networks currently being trained as two checkpoint models to generate the uncertainty map during each iteration. Then the uncertainty mask is computed and used to guide the training process of the network. Thus, the loss caused by pixels in regions with high uncertainty is endowed with increased weight.

The pipelines of our Snapshot and Siam are presented in Fig. 3(b). When the training procedure starts the Snapshot and Siam, uncertainty measurement (UM) and uncertainty guidance (UG) are implemented along the epochs iteratively. Uncertainty measurement would identify uncertainty, which is used to construct a 2D mask with a threshold in uncertainty guidance. This mask distinguishes the certainty pixels and uncertainty pixels and is imposed on the loss function of the baseline model with different weights. Once the training procedure reaches convergence, uncertainty post-processing (UP) would take effect to produce depth for each pixel, which adaptively chooses from the last epoch or an ensemble mean based on the uncertainty mask.

3.2. Uncertainty Measurement

The first step of our proposed uncertainty quantification approach is to measure uncertainty. We use the endogenous variance of the models to estimate the uncertainty of the depth map with the self-supervised learning method. During the training process, we select Snapshot or Siam model, of which the variance is used to calculate the uncertainty information of the depth map.

Snapshot calculates the uncertainty in backtracking to referring mode, which collects consecutive epochs in backforward order from the current epoch to compare them and calculate uncertainty. There are two factors for Snapshot to determine. One is how many pre-converged epochs are needed, and the other is which are chosen. For the first one, we could search one small interval to find one empirically optimal value. For the second one, we reasonably using consecutive pre-converged epochs just before the current epoch because certainty parts benefiting from the well-constraint should keep stable outputs in closely adjacent e-

pochs. This stability could decrease its interference with the identification of uncertainty. The visual demonstration is illustrated in Fig. 2.

Siam calculates the uncertainty in a mirroring-to-referring mode, where the twins 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. There is one factor for Siam to determine which epoch starts to estimate the uncertainty. We still use a search strategy to get an empirically optimal value.

When Snapshot and Siam start the uncertainty measurement, we use the endogenous variance of the models to estimate the uncertainty U:

$$U = \mathrm{UM}(D) = \begin{cases} \frac{\sum_{i=1}^{\mathrm{N}}(D_i - \overline{D})^2}{\mathrm{N}}, & \text{Snapshot} \\ \frac{\sum_{i=1}^{2}(D_i - \overline{D})^2}{2}, & \text{Siam} \end{cases} \tag{3}$$

where N is closely adjacent pre-converge models number, D_i is the estimated depth map of model Γ at the i epoch. $\overline{D}$ is the average of depth maps on training strategy Snapshot and Siam, respectively.

3.3. Uncertainty Guidance

Here, we use the uncertainty measurement result to guide the training of the network model Γ. Unlike previous uncertainty quantification methods, we make the uncertainty information explicitly and spatially guide the learning of the model. We use the mean of the uncertainty U as the threshold $\overline{U}$, imposing the uncertainty on pixels differentially:

$$\overline{U} = \frac{1}{\|\Omega\|}\sum_{\mathrm{k}\in\Omega} U(\mathrm{k}), \tag{4}$$

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

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

$$\mathrm{M} = \mathrm{UG}(U) = \begin{cases} 1, & U(\mathrm{k}) \leq \overline{U} \\ 1 + \lambda U(\mathrm{k}), & U(\mathrm{k}) > \overline{U} \end{cases} \tag{5}$$

where λ is an empirical parameter to control how much weight is given to the uncertainty pixel.

Supposing L is the loss function of the baseline model Γ, which would function on each pixel for the input image. After considering the uncertainty guidance, the new loss function L_{new} can be expressed as:

$$\mathrm{L}_{new} = \mathrm{M} * \mathrm{L}. \tag{6}$$

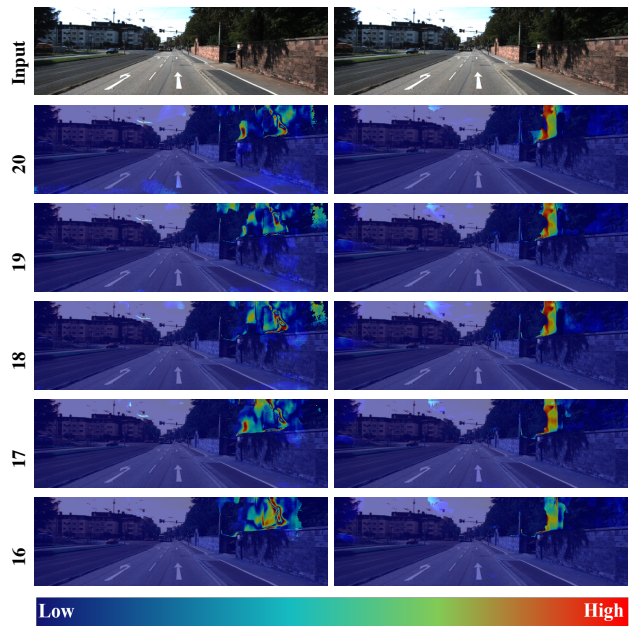


Figure 4: The uncertainty mask from the current 20 epoch back to the 16 epoch. Left: Monodepth2-Snapshot-M50, right: Hints-Siam-MS50.

Fig. 4 demonstrates two uncertainty guidance examples of Snapshot and Siam. Uncertainty guidance can persistently concentrate on masking the rich uncertainty regions, and meanwhile, their area shrinks when the learning advances.

3.4. Uncertainty Post-processing

The third step for our uncertainty quantification is uncertainty-based ensemble learning post-processing work. The last model lies close to the desired optimal point when the training terminates. Lying over the optimal point may cause texture copy or other artifacts. If lying quite near but not reaching the optimal point, it is better to choose the last epoch as the final output, as it would be one most close to the ideal optimal point. The final estimator $\check{D}$ is conditionally determined based on the ensemble of Snapshot or the twins of Siam:

$$\check{D}(\mathrm{k}) = \mathrm{UP}(\overline{D}(\mathrm{k}), D_{\Gamma'}(\mathrm{k})) = \begin{cases} \overline{D}(\mathrm{k}), & U(\mathrm{k}) \leq \overline{U} \\ D_{\Gamma'}(\mathrm{k}), & U(\mathrm{k}) > \overline{U} \end{cases} \tag{7}$$

where Γ' denotes the model at the last epoch in Snapshot or the superior one of the Siam twins in the last epoch, $D_{\Gamma'}$ is the depth map of Γ', and $\overline{D}(\mathrm{k})$ is the average of depth maps on training strategy Snapshot or Siam. This step differs slightly from Eq. (5). When the uncertainty of pixel from Γ' is below $\overline{U}$, it means that this pixel has been learned well; use $\overline{D}$ as final depth $\check{D}$. When the uncertainty of pixel is greater than the $\overline{U}$, the output is from $D_{\Gamma'}$.

3.5. Baseline Models

We choose Monodetph2 [15] and Hints [42] as the baseline model Γ to validate the proposed uncertainty quantification method, respectively. We do not modify the parameters or structures of the baseline models but only impose uncertainty on loss functions. Monodepth2 and Hints are the two frequently used methods and have well-organized source codes, which could guarantee the fairness of evaluation.

Monodepth2. Self-supervised monocular depth estimation usually uses photometric re-projection loss at training time. Monodepth2 [15] predicts the dense depth map D_t by minimizing the photometric re-projection loss L_p:

$$\mathrm{L}_p = pe(I_t, I_{t'\to t}), \tag{8}$$

The re-projected image $I_{t'\to t}$:

$$I_{t'\to t} = I_{t'}\langle proj(D_t, T_{t\to t'}, K)\rangle, \tag{9}$$

where $\langle . \rangle$ is the sampling operator; $K \in \mathbb{R}^{3\times 3}$ is the camera intrinsic parameter matrix is identical for all images. $proj()$ returns the resulting 2D coordinates of the projected depths D_t in $I_{t'}$:

$$proj(D_t, T_{t\to t'}, K) = KT_{t\to t'}D_t(p_t)K^{-1}p_t, \tag{10}$$

where p_t denotes a pixel. Referring to [14,47], Monodepth2 uses L_1 and SSIM [41] as the photometric loss function pe:

$$\begin{aligned} pe(I_a, I_b) = &\frac{\alpha}{2}(1 - \mathrm{SSIM}(I_a, I_b)) \\ &+(1-\alpha)\left\|I_a - I_b\right\|_1, \end{aligned} \tag{11}$$

where $\alpha = 0.85$, SSIM() is computed over a 3x3 pixel window.

To encourage neighboring pixels to have similar depths, use an edge-aware depth smoothness loss L_s weighted by image gradients to improve the predictions around object boundaries. The edge-aware smoothness L_s:

$$\mathrm{L}_s = |\partial_x D_t^*|e^{-|\partial_x I_t|} + |\partial_y D_t^*|e^{-|\partial_y I_t|}, \tag{12}$$

where ∂_x, ∂_y are gradient operation on x, y-axis, $D_t^* = D_t/\overline{D_t}$ is the mean-normalized inverse depth.

The final loss is computed as the weighted sum of image photometric re-projection loss L_p and smoothness loss L_s:

$$\mathrm{L} = \mathrm{L}_p + \mu \mathrm{L}_s, \tag{13}$$

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

In stereo training (S), $I_{t'}$ is the second view in the stereo image pair to I_t. When relative poses are not known in advance for training (M), a pose estimation network predicts the relative pose $T_{t\to t'}$. In mixed training (MS), $I_{t'}$ includes the temporally adjacent frames and the stereo view.

Table 1: Depth metrics. $D(\mathrm{k})$ is the predicted depth at each pixel k in the image space Ω, $\hat{D}(\mathrm{k})$ is the corresponding ground truth depth, $\|\Omega\|$ is the total amount of pixels in input image I. Three different thresholds, $\mathrm{a1} = 1.25$, $\mathrm{a2} = 1.25^2$, $\mathrm{a3} = 1.25^3$, are used in the accuracy metric.

Metric	Definition
Abs Rel:	$\frac{1}{\|\Omega\|}\sum_{\mathrm{k}\in\Omega}\frac{\lvert D(\mathrm{k})-\hat{D}(\mathrm{k})\rvert}{\hat{D}(\mathrm{k})}$
Sq Rel:	$\frac{1}{\|\Omega\|}\sum_{\mathrm{k}\in\Omega}\frac{\lvert D(\mathrm{k})-\hat{D}(\mathrm{k})\rvert^2}{\hat{D}(\mathrm{k})}$
RMSE:	$\sqrt{\frac{1}{\|\Omega\|}\sum_{\mathrm{k}\in\Omega}\lvert D(\mathrm{k})-\hat{D}(\mathrm{k})\rvert^2}$
RMSE log:	$\sqrt{\frac{1}{\|\Omega\|}\sum_{\mathrm{k}\in\Omega}\lvert \log D(\mathrm{k})-\log\hat{D}(\mathrm{k})\rvert^2}$
Accuracy (ai):	% of $D(\mathrm{k})$ s.t. $\delta = \max(\frac{D(\mathrm{k})}{\hat{D}(\mathrm{k})}, \frac{\hat{D}(\mathrm{k})}{D(\mathrm{k})}) < \mathrm{ai}$

Hints. Hints [42] introduces stereo matching algorithm SGM [17] to get depth hints $\widetilde{D}_t$, and then uses $\widetilde{D}_t$ to create a second synthesized view with the following formula:

$$\widetilde{I}_{t'\to t} = I_{t'}\langle proj(\widetilde{D}_t, I_{t\to t'}, K)\rangle, \tag{14}$$

This extra synthesized view can be seen as semi-supervised information. Hints has conditions to determine whether or not to apply a supervised loss $\widetilde{D}_t$ as ground truth on a per-pixel k basis:

$$\mathrm{L} = \begin{cases} \mathrm{L}_p(D(\mathrm{k})) + \mathrm{L}_s^{\log L_1}(D(\mathrm{k}), \widetilde{D}(\mathrm{k})), & \text{if } \upsilon \\ \mathrm{L}_p(D(\mathrm{k})), & \text{else} \end{cases} \tag{15}$$

where $\mathrm{L}_s^{\log L_1}(D(\mathrm{k}), \widetilde{D}(\mathrm{k})) = \log(1 + |D(\mathrm{k}) - \widetilde{D}(\mathrm{k})|)$, and $\upsilon = \mathrm{L}_p(D(\mathrm{k})) < \mathrm{L}_p(\widetilde{D}(\mathrm{k}))$. They introduced depth hints as a practical approach to help escape from local minima and to guide the network toward a better overall solution.

4. Experiments

In the experiments, we validate the proposed uncertainty quantification method on the KITTI dataset [13] and evaluate it using the Eigen split [7]. The program is implemented with Pytorch and runs on a server with the following configuration: CPU: Intel(R) Xeon(R) Silver 4114 CPU@2.20GHz*2; RAM: 192G and GPU: NVIDIA GeForce GTX 2080Ti*2.

4.1. Evaluation metrics

Depth metrics. We use seven metrics [7] to evaluate the depth estimation model. The four error metrics measure the difference between predicted depth D and ground-truth depth $\hat{D}$: 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

Table 2: The empirical parameter λ in Eq. (5), the starting epoch N of Siam, the parameter N in Eq. (3): closely adjacent pre-converge models number.

(a) λ: Blue: Monodepth2-Snapshot-M50. Purple: Hints-Siam-MS50.

λ	Abs Rel↓	RMSE↓	a1 ↑	Abs Rel↓	RMSE↓	a1 ↑
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**

(b) Siam: Starting Epoch N.

Model	N	Abs Rel↓	RMSE↓	a1↑
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**

(c) Snapshot: closely adjacent pre-converge models number N.

Model	N	Abs Rel↓	RMSE↓	a1↑
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 3: Validation experiments on threshold $\overline{U}$. Blue: Monodepth2-Snapshot-M50. Purple: Hints-Siam-MS50.

Mask	Abs Rel↓	RMSE↓	a1↑	Abs Rel↓	RMSE↓	a1↑
$\overline{U}$	**0.109**	**4.551**	**0.885**	**0.102**	**4.546**	0.880
0.8* $\overline{U}$	0.115	4.670	0.871	**0.102**	**4.546**	**0.882**
1.2* $\overline{U}$	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

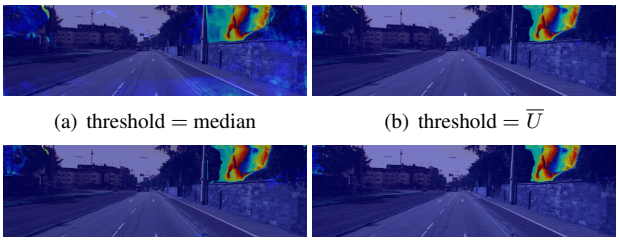
depth inside an image whose ratio and inverse ratio with the ground truth is below the thresholds $\mathrm{a1} = 1.25$, $\mathrm{a2} = 1.25^2$, and $\mathrm{a3} = 1.25^3$. The smaller the first four metrics are, the better the results are, while the bigger the last three metrics are, the better the results are. Table 1 presents the above detailed equations.

Uncertainty metrics. We use two metrics of the area under the sparsification error (Ause) and the area under the random gain (Aurg) [32] to evaluate how significant our modeled uncertainties are:

$$\mathrm{Ause}(U, D) = \epsilon(D) - \epsilon(D_U), \tag{16}$$

$$\mathrm{Aurg}(U, D) = \mathrm{Ause}(rand, D) - \mathrm{Ause}(U, D), \tag{17}$$

where ϵ is the depth map error metric, D_U is the depth map for the 2% pixels with the highest uncertainty. Here, Abs Rel, RMSE or $\delta \geq 1.25$ (since $\delta < 1.25$ defines an accuracy score) be used as ϵ. Ause (the lower, the better) quantifies how close the estimate is to the ideal sparsification uncertainty. Aurg (the higher, the better) quantifies how better it is compared to no modeling.



(a) threshold = median (b) threshold = $\overline{U}$

(c) threshold = $0.8\overline{U}$ (d) threshold = $1.2\overline{U}$

Figure 5: Uncertainty masks from the uncertainty map based on different thresholds, the baseline model is the Monodepth2-M50. The best result is threshold = $\overline{U}$.

Table 4: Quantitative evaluation on uncertainty measurement. [32]: Monodepth2-Snap+Self-M; Ours1: Monodepth2-Snapshot-M50; Ours2: Hint-Siam-MS50.

Method	Abs Rel		RMSE		$\delta \geq 1.25$	
	Ause↓	Aurg↑	Ause↓	Aurg↑	Ause↓	Aurg↑
[32]	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**

4.2. Parameter Setting

We use many experiments to determine the empirical parameter λ in Eq. (5), the starting epoch of Siam, the parameter N in Eq. (3), and the threshold of uncertainty mask M. We approximately enumerate multiple λ values to determine a recommended setting for the subsequent experiments.

In Table 2(a), we display 0.8 to 1.2 results, which is an optimal interval. We set $\lambda = 1$ for all experiments to reduce the computation cost. As shown in Table 2(b), we set the starting epoch of the Siam at the 1 epoch. As shown in Table 2(c), we set the parameter $N = 5$ in Eq. (3) and start the uncertainty guidance of the Snapshot at 6 epoch. In Eq. (4), the mean $\overline{U}$ and other possible options like fractional mean and median, we have validated that $\overline{U}$ is a more proper one to work as the threshold. In Table 3 and Fig. 5, we can see that the threshold $\overline{U}$ achieves better performance.

4.3. Performance Evaluation

Here, we evaluate the proposed uncertainty estimation strategy on the two baseline models: Monodepth2 and Hints. We evaluate the depth accuracy and modeled uncertainties. Total six conditions of Monodepth2, Monodepth2-Snapshot, Monodepth2-Siam, Hints, Hints-Snapshot, and Hints-Siam are taken into evaluation, which is carried out by varying the training paradigms (stereo pairs, monodepth, stereo and monodepth), and CNN modules (ResNet18, ResNet50). Because the SGM algorithm in Hints needs a stereo pair image, which can not train on M.

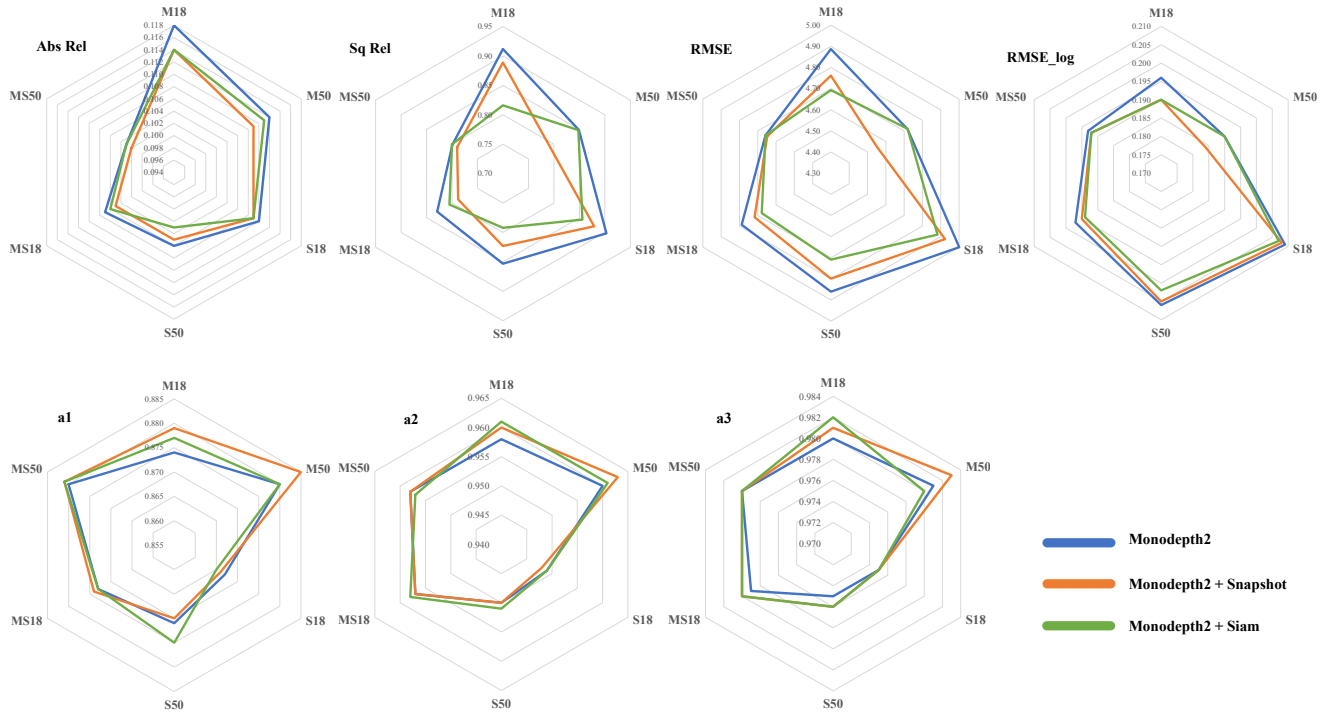


Figure 6: Quantitative evaluation of Snapshot and Siam performance on Monodepth2 [15] baseline model with seven depth metrics: Abs Rel, Sq Rel, RMSE, RMSE log, accuracy (ai) (Table 1). One radar chart illustrates one metric, varying by the training paradigms and CNN modules. An axis of the radar chart represents one training paradigm (M, S, MS), while the number represents ResNet18 or ResNet50.

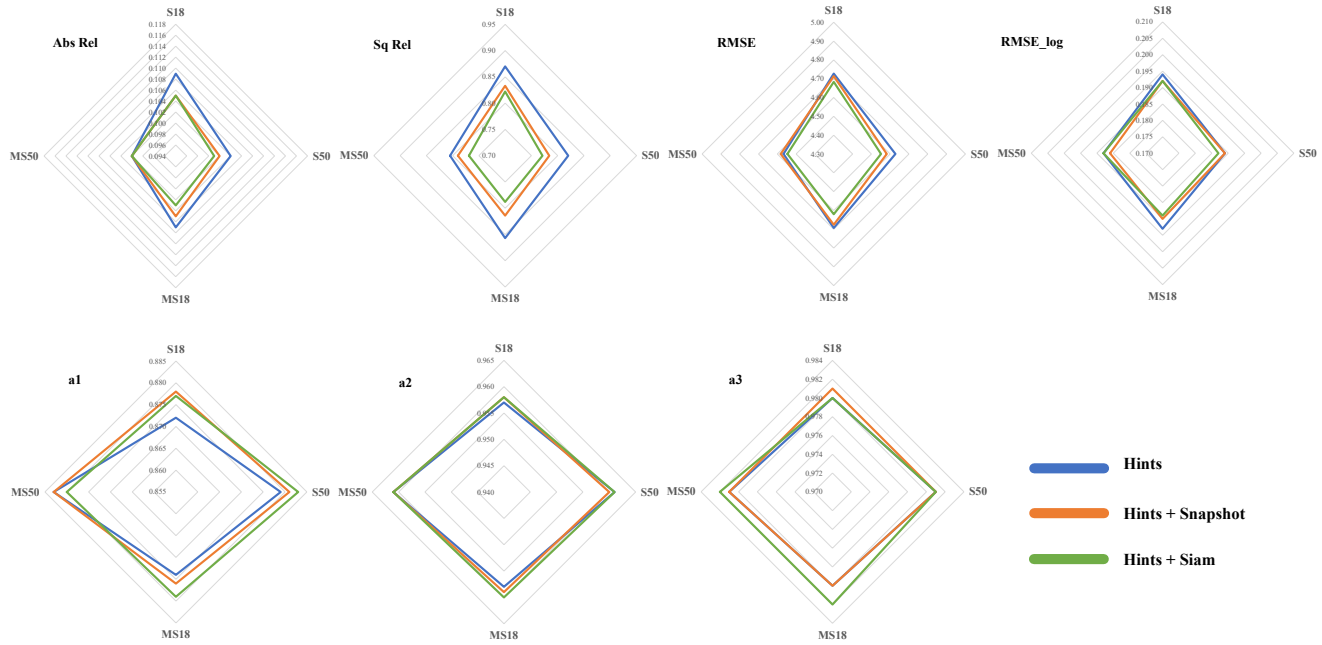


Figure 7: Quantitative evaluation of Snapshot and Siam performance on Hints [42] baseline model with seven depth metrics: Abs Rel, Sq Rel, RMSE, RMSE log, accuracy (ai) (Table 1). One radar chart illustrates one metric, varying by the training paradigms and CNN modules. An axis of the radar chart represents one training paradigm (M, S, MS), while the number represents ResNet18 or ResNet50.

Table 5: Ablation Study on baseline model Monodepth2 [15]. We conduct an ablation study by switching on/off UG and UP in three training paradigms.

Backbone	Train	UG	UP	Snapshot			Siam		
				Abs Rel↓	RMSE↓	a1↑	Abs Rel↓	RMSE↓	a1↑
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	✓	$\overline{D}$	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	✓	$\overline{D}$	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	✓	$\overline{D}$	**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	✓	$\overline{D}$	**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	✓	$\overline{D}$	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	✓	$\overline{D}$	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 6: Ablation Study on the baseline model Hints [42]. We conduct an ablation study by switching on/off UG and UP in three training paradigms.

Backbone	Train	UG	UP	Snapshot			Siam		
				Abs Rel↓	RMSE↓	a1↑	Abs Rel↓	RMSE↓	a1↑
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	✓	$\overline{D}$	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	✓	$\overline{D}$	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	✓	$\overline{D}$	**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	✓	$\overline{D}$	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

Depth Accuracy Evaluation. Fig. 6 and Fig. 7 report depth accuracy on the two baseline models Monodepth2 and Hints variants implementing, respectively. Snapshot and Siam have improved the model accuracy of Monodepth2 and Hints on all training paradigms and ResNet18/50. Concerning evaluations on four loss metrics, our Snapshot and Siam lie inside the baseline models. On the accuracy metrics, our Snapshot and Siam lie outside baseline models. Evaluation results have validated the effectiveness of our method.

Uncertainties Evaluation. We quantitatively evaluate the uncertainty measurement strategy using the two uncertainty metrics(Ause and Aurg). Monodepth2-Snap+Self-M50 is the best model in [32]. Monodepth2-Snapshot-M50 and Hint-Siam-MS50 are the optimal models under three training paradigms, respectively. Table 4 summarizes the effectiveness of modeled uncertainties. In the uncertainties evaluation, we can see that our results are better than [32].

Table 7: Depth evaluation on the KITTI. Comparisons with state-of-the-art methods. *: the result of the model we trained. Method marked by gray is baseline method, and light-gray is the uncertainty method from Poggi *et al.* [32]. The best results in each category are written in **boldface.**

Method	Year	Periodical	Train	Abs Rel↓	Sq Rel ↓	RMSE↓	RMSE log ↓	a1↑	a2↑	a3↑
Zhou *et al.* [48]	2017	CVPR	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang *et al.* [45]	2018	AAAI	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian *et al.* [27]	2018	CVPR	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet [46]	2018	CVPR	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DDVO [40]	2018	CVPR	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [49]	2018	ECCV	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [44]	2018	CVPR	M	0.162	1.352	6.276	0.252	-	-	-
Ranjan *et al.* [35]	2019	CVPR	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
Struct2depth [5]	2019	AAAI	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [15]*	2019	ICCV	M	0.118	0.912	4.887	0.196	0.874	0.958	0.980
Klingner *et al.* [21]	2020	ECCV	M	0.117	0.907	4.844	0.196	0.875	0.958	0.980
EPC++ [26]	2020	PAMI	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
PackNet [16]	2020	CVPR	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Johnston *et al.* [20]	2020	CVPR	M	0.106	0.861	4.699	0.185	**0.899**	0.962	0.982
CoMoDA [22]	2021	WACV	M	**0.103**	0.862	4.594	0.183	**0.899**	0.961	0.981
Poggi *et al.* [32]	2020	CVPR	M	0.112	0.838	4.691	0.186	0.881	0.961	0.983
SC-Depth [3]	2021	IJCV	M	0.126	0.920	5.245	0.208	0.840	0.949	0.979
SD-SSMDE [30]	2022	CVPR	M	0.108	**0.751**	**4.485**	**0.180**	0.885	**0.964**	**0.984**
Ours(Monodepth2-Snapshot-50)	-	-	M	0.109	0.792	4.551	0.184	0.885	0.963	0.983
Garg *et al.* [11]	2016	ECCV	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth R50 [14]	2017	CVPR	S	0.133	1.142	5.533	0.230	0.830	0.936	0.970
StrAT [28]	2018	3DV	S	0.128	1.019	5.403	0.227	0.827	0.935	0.971
3Net [33]	2018	3DV	S	0.129	0.996	5.281	0.223	0.831	0.939	0.974
SuperDepth (1024x382) [31]	2019	ICRA	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Monodepth2 [15]*	2019	ICCV	S	0.110	0.903	5.001	0.209	0.867	0.949	0.975
MonoResMatch [39]	2019	CVPR	S	0.115	0.920	4.913	0.208	0.850	0.945	0.970
Hints [42]*	2019	ICCV	S	0.109	0.870	4.812	0.194	0.872	0.957	0.980
Poggi *et al.* [32]	2020	CVPR	S	0.108	0.835	4.856	0.202	0.865	0.951	0.977
Wavelet Decomposition [34]	2021	CVPR	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**
UnDeepVO [23]	2018	ICRA	MS	0.183	1.730	6.571	0.268	-	-	-
Hints [42]*	2019	ICCV	MS	0.107	0.857	4.780	0.193	0.874	0.958	0.980
Monodepth2 [15]*	2019	ICCV	MS	0.107	0.829	4.788	0.197	0.873	0.957	0.979
EPC++ [26]	2020	TPAMI	MS	0.128	0.936	5.011	0.209	0.831	0.945	0.979
Poggi *et al.* [32]	2020	CVPR	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**

Ablation Study. We conducted an ablation study to validate the effectiveness of each component (uncertainty guidance and post-processing) in our proposed uncertainty quantification method. We switch on and switch off uncertainty guidance and uncertainty post-processing in all three training paradigms on baseline models Monodepth2 and Hints. We present the complete results in Table 5 and Table 6. We can seen that the uncertainty guidance and uncertainty post-processing can improve the accuracy of the depth estimation model, respectively. The best result is use the uncertainty guidance and uncertainty post-processing.

4.4. Comparisons

We make comprehensive comparisons with the current representative methods. Table 7 shows the depth of quantitative results. For the baseline model Monodepth2, Monodepth2-Snapshot-M50 achieves the best result. For baseline model Hints, Hints-Siam-MS50 achieves the best result. Although recently proposed methods surpass Monodepth2 and Hints, our proposed uncertainty quantification improved their accuracy. In Fig. 8, we demonstrate a group of depth maps from current methods. The objects in our depth maps have complete structures and sharp boundaries.

Compared to the uncertainty work [32], our results are quantitatively and qualitatively superior to theirs. In Table 7, we mark the results [32] in light gray color and backbone in gray. We can find that our uncertainty quantification advantage over them is very significant. Such a significant advantage also exists in visual comparison Fig. 8. The reason that our uncertainty quantification better understands and applies uncertainty in depth estimation brings solid advantages.

5. Conclusion

In this paper, we proposed a novel uncertainty quantification strategy to train self-supervised monocular depth es-

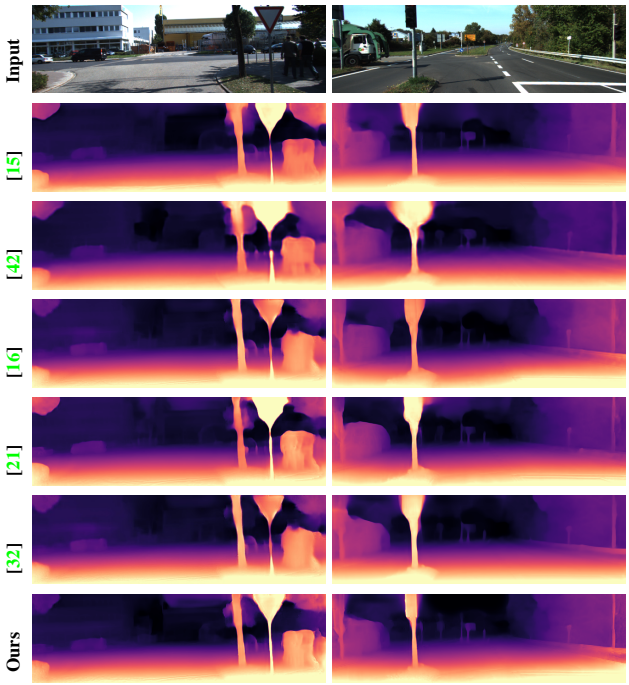


Figure 8: Examples of qualitative comparison. Comparisons with the state-of-the-art self-supervised monocular depth estimation methods: Monodepth2 [15], Hints [42], PackNet [16], Klingner *et al.* [21], Poggi *et al.* [32]. Our results of left: Monodepth2-Snapshot-M50 and right: Hints-Siam-MS50.

timation. Our uncertainty quantification strategy contains uncertainty measurement, guidance, and post-processing. First, we use consecutive training epochs or the Siam network to measure the uncertainty of the depth map. Then, the estimation uncertainty is used to guide the depth network model learning. Finally, the uncertainty post-processing adaptively produces final depth results with a balance of accuracy and robustness. Our method has achieved the SOTA results compared with existing uncertainty quantification methods. We want to investigate how to construct a more efficient self-supervised uncertainty quantification method for future work. Another problem we intend to explore is finding effective depth cues to fix the uncertainty.

Acknowledgement

This work is partially supported by NSFC (No. 61972298). Chunxia Xiao and Fei Luo are the Corresponding authors.

References

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021. 3

[2] A. Asai, D. Ikami, and K. Aizawa. Multi-task learning based on separable formulation of depth estimation and its uncertainty. In *CVPR Workshops*, pages 21–24, 2019. 2, 3

[3] J.-W. Bian, H. Zhan, N. Wang, Z. Li, L. Zhang, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision (IJCV)*, 2021. 1, 2, 10

[4] T. Cao, F. Luo, Y. Fu, W. Zhang, S. Zheng, and C. Xiao. Dgecn: A depth-guided edge convolutional network for end-to-end 6d pose estimation. In *CVPR*, pages 3783–3792, 2022. 1

[5] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019. 10

[6] H. Choi, H. Lee, S. Kim, S. Kim, S. Kim, K. Sohn, and D. Min. Adaptive confidence thresholding for monocular depth estimation. In *ICCV*, pages 12808–12818, 2021. 2, 3

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015. 1, 6

[8] Y. Fu, Q. Yan, J. Liao, and C. Xiao. Joint texture and geometry optimization for rgb-d reconstruction. In *CVPR*, 2020. 1

[9] Y. Fu, Q. Yan, J. Liao, H. Zhou, J. Tang, and C. Xiao. Seamless texture optimization for rgb-d reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 1

[10] Y. Fu, Q. Yan, L. Yang, J. Liao, and C. Xiao. Texture mapping for 3d reconstruction with rgb-d sensor. In *CVPR*, pages 4645–4653, 2018. 1

[11] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756, 2016. 10

[12] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 1, 2

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 6

[14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017. 2, 6, 10

[15] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3827–3837, 2019. 1, 2, 6, 8, 9, 10, 11

[16] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, pages 2485–2494, 2020. 1, 10, 11

[17] H. Hirschmueller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 2, 6

[18] 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. 3

[19] S. Jamonnak, Y. Zhao, X. Huang, and M. Amiruzzaman. Geo-context aware study of vision-based autonomous driving models and spatial video data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1019–1029, 2022. 1

[20] A. Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *CVPR*, pages 4756–4765, 2020. 10

[21] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, pages 2619–2627, 2020. 1, 2, 10, 11

[22] Y. Kuznietsov, M. Proesmans, and L. Van Gool. Comoda: Continuous monocular depth adaptation using past experiences supplementary materials. In *ICCV*, 2019. 2, 10

[23] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *ICRA*, pages 7286–7291, 2018. 10

[24] Y. Li, F. Luo, and C. Xiao. Self-supervised coarse-to-fine monocular depth estimation using a lightweight attention module. *Computational Visual Media*, 8(4):631–647, 2022. 1

[25] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *CVPR*, pages 10986–10995, 2019. 3

[26] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2624–2641, 2020. 10

[27] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, pages 5667–5675, 2018. 10

[28] I. Mehta, P. Sakurikar, and P. Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *3DV*, pages 314–323, 2018. 10

[29] A. Mertan, Y. H. Sahin, D. J. Duff, and G. Unal. A new distributional ranking loss with uncertainty: Illustrated in relative depth estimation. In *3DV*, pages 1079–1088, 2020. 2, 3

[30] A. Petrovai and S. Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *CVPR*, pages 1578–1588, 2022. 2, 10

[31] S. Pillai, R. Ambruş, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, pages 9250–9256, 2019. 10

[32] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, pages 3227–3237, 2020. 2, 3, 7, 9, 10, 11

[33] M. Poggi, F. Tosi, and S. Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, pages 324–333, 2018. 10

[34] M. Ramamonjisoa, M. Firman, J. Watson, V. Lepetit, and D. Turmukhambetov. Single image depth prediction with wavelet decomposition. In *CVPR*, pages 11089–11098, 2021. 1, 10

[35] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, pages 12232–12241, 2019. 10

[36] Y. Shen, Z. Zhang, M. R. Sabuncu, and L. Sun. Real-time uncertainty estimation in computer vision via uncertainty-aware distribution distillation. In *WCACV*, pages 707–716, 2021. 3

[37] C. Song, C. Qi, S. Song, and F. Xiao. Unsupervised Monocular Depth Estimation Method Based on Uncertainty Analysis and Retinex Algorithm. *Sensors*, 20(18), 2020. 3

[38] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli. Aerial single-view depth completion with image-guided uncertainty estimation. *IEEE Robotics and Automation Letters*, 5(2):1055–1062, 2020. 2, 3

[39] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *CVPR*, pages 9799–9809, 2019. 10

[40] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, pages 2022–2030, 2018. 10

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. 6

[42] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, pages 2162–2171, 2019. 1, 2, 6, 8, 9, 10, 11

[43] X. Yang, Y. Gao, H. Luo, C. Liao, and K.-T. Cheng. Bayesian denet: Monocular depth prediction and frame-wise fusion with synchronized uncertainty. *IEEE Transactions on Multimedia*, 21(11):2701–2713, 2019. 2

[44] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *CVPR*, pages 225–234, 2018. 10

[45] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *AAAI*, volume 32, 2018. 10

[46] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018. 2, 10

[47] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss Functions for Image Restoration With Neural Networks. *IEEE transactions on computational imaging*, 3(1):47–57, 2017. 6

[48] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1858, 2017. 2, 10

[49] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, pages 36–53, 2018. 2, 10