A technique to jointly estimate depth and depth uncertainty for unmanned aerial vehicles

Michaël Fonder[1], and Marc Van Droogenbroeck[1]

[1] Montefiore Institute, University of Liège, Liège, Belgium

michael.fonder@uliege.be

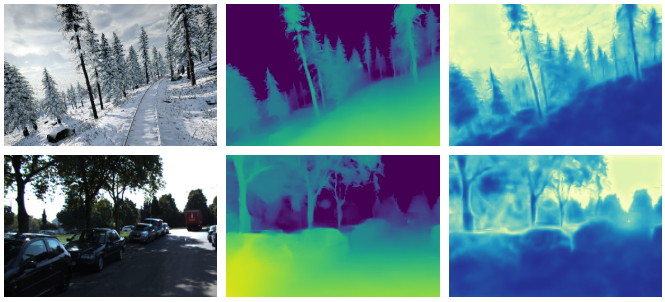
***Abstract*—When used by autonomous vehicles for trajectory planning or obstacle avoidance, depth estimation methods need to be reliable. Therefore, estimating the quality of the depth outputs is critical. In this paper, we show how M4Depth, a state-of-the-art depth estimation method designed for unmanned aerial vehicle (UAV) applications, can be enhanced to perform joint depth and uncertainty estimation. For that, we present a solution to convert the uncertainty estimates related to parallax generated by M4Depth into uncertainty estimates related to depth, and show that it outperforms the standard probabilistic approach. Our experiments on various public datasets demonstrate that our method performs consistently, even in zero-shot transfer. Besides, our method offers a compelling value when compared to existing multi-view depth estimation methods as it performs similarly on a multi-view depth estimation benchmark despite being 2.5 times faster and causal, as opposed to other methods. The code of our method is publicly available at the following URL: https://github.com/michael-fonder/M4DepthU.**

***Index Terms*—Depth estimation, uncertainty estimation, autonomous aerial vehicles, parallax**

I. Introduction

One of the many applications of depth estimation is to replace depth sensors in autonomous vehicles for path planning [1] or obstacle avoidance [2], [3]. Such practice is common for small unmanned aerial vehicles (UAVs) as their size, weight and power constraints prevent the use of dedicated depth sensors. For such applications, being able to predict the quality of the estimates is essential to anticipate potentially erroneous data and take action accordingly. However, to the best of our knowledge, the task of joint depth and uncertainty estimation for drone-specific constraints, such as being robust to a wide variety of conditions and environments while being computationally lightweight enough to run in real-time on limited hardware, has not been addressed yet.

In a previous work [4], we introduced M4Depth, a depth estimation method specifically designed for unstructured environments and UAV applications that shows state-of-the-art performance for depth estimation in such environments and in generalization. In this work, we detail how it is possible to adapt the architecture of M4Depth to jointly estimate depth and its uncertainty for a negligible additional computational cost. Section II discusses the related works about uncertainty estimation. In Section III, we explain how M4Depth can be adapted for joint depth and uncertainty estimation. Our experiments, presented in Section IV, test our proposal in various conditions including zero-shot transfer on public datasets and on an existing benchmark for multi-view depth (MVD) estimation methods. Section V concludes this work.



RGB input Estimated depth Estimated uncertainty

Figure 1. Illustration of depth and uncertainty estimates produced by the method presented in this work for two setups. Row 1: trained and tested on the Mid-Air [5] dataset. Row 2: tested in zero-shot transfer on the KITTI [6] dataset. Lighter colors correspond to higher uncertainty values.

Our main contributions are as follows. (i) Our method is the first to address joint monocular depth and uncertainty estimation for the specific constraints of autonomous drones. (ii) We show that our method for uncertainty estimation performs consistently in zero-shot transfer in different environments. (iii) On a benchmark for MVD, we show that our method performs on par with existing MVD methods for joint depth and uncertainty estimation despite being 2.5 times faster and causal, as opposed to MVD methods.

II. Related works

Our M4Depth paper [4] already covers related works in depth estimation, and uncertainty in neural networks is well covered in the survey of Gawlikowski *et al.* [7]. Therefore, we focus on uncertainty estimation for pixel-wise computer vision regression tasks in this section.

Kendall and Gal [8] showed that a part of the uncertainty in a deep neural network, called the *aleatoric* uncertainty, is due to the noise in the input data. They also showed that this part of the uncertainty can be estimated by training a network to learn the parameters of a probabilistic distribution that represents the noise in the output. The output noise for a method trained with a L1 loss is assumed to follow a Laplace distribution [9]–[12]

as it allows learning its parameters, the location and the scale, with a Maximum Log-Likelihood loss function [8], [10], [13].

Estimating the aleatoric uncertainty can be done either by creating a new architecture designed around uncertainty, such as done by Ke *et al.* [9] and Su *et al.* [14], or by modifying an existing architecture for the desired task. In the latter case, the simplest way to proceed consists of adding a channel for the uncertainty at the output of the network [12], [13], [15]–[17]. However, some methods create a distinct head for the uncertainty by duplicating the last layers of their network [18]–[20], which provides more trainable parameters to avoid potentially sub-optimal performances due to the shared weights.

III. Uncertainty estimation using M4Depth

In this section, we briefly remind the working principles of M4Depth [4] and explain how the network architecture can be modified to jointly estimate the parallax and its aleatoric uncertainty. We then detail how to get the uncertainty on depth from the uncertainty on the parallax.

A. M4Depth working principles

M4Depth is a multi-level pyramidal architecture where each level has the same structure and outputs a parallax estimate. The parallax $\rho > 0$ is linked to the depth z of a point P by the motion of the camera between two poses:

$$\mathrm{z} = \frac{\sqrt{\left(f_x \mathrm{t}_x - \mathrm{t}_z i_V\right)^2 + \left(f_y \mathrm{t}_y - \mathrm{t}_z j_V\right)^2}}{\rho \ \mathrm{z}_V} - \frac{\mathrm{t}_z}{\mathrm{z}_V}, \qquad (1)$$

where f_x and f_y are the respective focal lengths along the x and y camera axes, $\begin{bmatrix} \mathrm{t}_x & \mathrm{t}_y & \mathrm{t}_z \end{bmatrix}$ expresses the known translation of the camera between the two poses, and where i_V, j_V and z_V are solely functions of the projection coordinates (i, j) of P and the rotation of the camera between the two poses [4].

The network starts with a first rough low-resolution parallax estimate and then refines it progressively at higher resolutions to get the final estimate. Each intermediate parallax map can be converted into a depth map using Eq. (1) for each pixel. The only architectural modification required for joint uncertainty inference is to add an output for uncertainty at each level of the architecture. As for the parallax, the additional outputs are refined progressively to get the final estimate. Details on the architecture modifications can be found in our code.

As mentioned in [4], M4Depth is trained for depth estimation on a weighted sum of the L_1 distance of the logarithm of the depth for each architecture level l:

$$\mathcal{L}_t = \frac{1}{HW} \sum_{l=1}^{M} \sum_{\mathrm{z}_{ij} \in \mathbf{d}_t^l} 2^{-l} \ \left|\log(\mathrm{z}_{ij}) - \log(\hat{\mathrm{z}}_{ij})\right| . \qquad (2)$$

B. Correspondence between depth and parallax uncertainties

Since M4Depth works with parallax instead of depth values, we need to make some adaptations to get the uncertainty on depth. We first detail the baseline approach, which relies on the standard probabilistic framework, to get depth uncertainties from M4Depth. We then present a new and more elaborate method to get depth uncertainty estimates from the parallax ones. As confirmed by experiments, our new method better evaluates the depth uncertainty.

To simplify the notations in the following, we rewrite Eq. (1) for a given pixel and a given camera motion as:

$$\mathrm{z} = Z(\rho) = \frac{a}{\rho} + c, \qquad (3)$$

where

$$a = \frac{\sqrt{\left(f_x \mathrm{t}_x - \mathrm{t}_z i_V\right)^2 + \left(f_y \mathrm{t}_y - \mathrm{t}_z j_V\right)^2}}{\mathrm{z}_V} \geq 0, \qquad (4)$$

and $c = -\frac{\mathrm{t}_z}{\mathrm{z}_V}$ are independent from the depth of the considered point.

1) Baseline: the probabilistic framework

As the aleatoric uncertainty is assumed to be proportional to the variance of the estimated output distribution, the natural solution to get the uncertainty on depth is to find the relation between the variance of the parallax output distribution and the one of depth. From the literature, we know that training a network as the log-likelihood of the L_1 distance on the depth z makes the assumption that its outputs follow a Laplace distribution whose mean and standard deviation are respectively equal to $\hat{\mu}(\mathrm{z})$ and $\hat{\sigma}(\mathrm{z})$. Unfortunately, the inverse relation linking depth and parallax (see Eq. (3)) means that a direct conversion between the variances, and therefore the standard deviations, is impossible as they may not be finite in both domains at the same time. However, training the network to infer the inverse parallax solves this issue since injecting the variable change $\zeta = 1/\rho$ in Eq. (3) gives:

$$\mathrm{z} = a\zeta + c \Rightarrow \sigma(\mathrm{z}) = \sigma(a\zeta + c) = |a|\,\sigma(\zeta) = a\sigma(\zeta) . \qquad (5)$$

In practice, we can train M4Depth to infer the uncertainty $\hat{\sigma}(\mathrm{z})$ jointly to depth, from the inverse parallax by adding this term to its loss function:

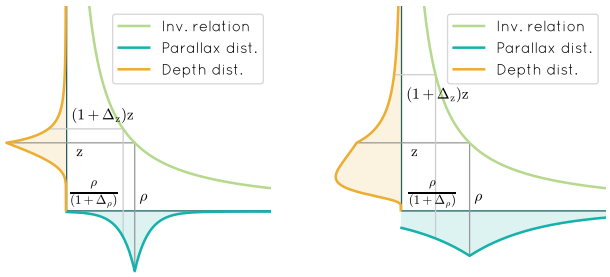
$$\mathcal{L}_{\mathrm{z},t} = \frac{1}{HW} \sum_{l=1}^{M} \sum_{\mathrm{z}_{ij} \in \mathbf{d}_t^l} 2^{-l} \ \left[\frac{\oslash\left(\left|\mathrm{z}_{ij} - \hat{\mathrm{z}}_{ij}\right|\right)}{a\hat{\sigma}(\zeta_{ij})} + \beta \log\left(a\hat{\sigma}(\zeta_{ij})\right)\right], \qquad (6)$$

where gradients are not propagated to the variables enclosed in the $\oslash()$ expression to avoid interference with the gradients generated by the $\mathcal{L}_t$ term of the loss, and where β is an arbitrary weighting factor for the uncertainty (we use $\beta = 0.02$ in our experiments). Note that this loss is only computed for pixels whose depth is lower than $400\,m$ to avoid any convergence issues.

In the following, we will refer to our modified version of M4Depth trained with this loss term as M4Depth+$\mathrm{U_z}$.

2) Elaborate conversion of the uncertainty

One of the strengths of M4Depth is the direct link that exists between the parallax and the disparity sweeping cost volumes, which are the main sources of information available to infer the parallax. We assume that, as the cost volumes provide valuable information on the parallax, they should also provide valuable information on the related uncertainty. This information is best used if there is a trivial relation between the distribution to learn and the cost volumes. However, such a



(a) Low standard deviation (b) Large standard deviation

Figure 2. Illustration of the correspondence between a Laplace distribution (blue curve) and its inverse (orange curve) when applied to the relation linking parallax to depth for different standard deviations of the Laplace distribution. We propose to use Δ_ρ as an uncertainty measure on the parallax, whose correspondence for depth is Δ_z.

trivial relation does not exist when learning the distribution of the inverse parallax because of the inverse relation. As a result, the probabilistic approach is not well suited for M4Depth, and we propose another approach to get depth uncertainty estimates from the parallax domain.

Substituting the loss term $\mathcal{L}_{z,t}$ defined in Eq. (6) by the following $\mathcal{L}_{\rho,t}$ term in the training loss of M4Depth allows to train the network to produce uncertainty estimates $\hat{\sigma}(\rho)$ directly related to parallax estimates $\hat{\rho}$:

$$\mathcal{L}_{\rho,t} = \frac{1}{HW} \sum_{l=1}^{M} \sum_{\rho_{ij} \in \boldsymbol{\rho}_t^l} 2^{-l} \left[\frac{\oslash \left(|\rho_{ij} - \hat{\rho}_{ij}| \right)}{\hat{\sigma}(\rho_{ij})} + \beta \log \left(\hat{\sigma}(\rho_{ij}) \right) \right] . \quad (7)$$

In the following, we will refer to our modified version of M4Depth trained with this loss term as M4Depth+U_ρ.

We want to find a value $\Delta_z > 0$ in the depth domain that represents the uncertainty carried by $\hat{\sigma}(\rho)$. Stated otherwise, for any corresponding pair $(\sigma(\rho), \Delta_z)$ and with everything else being equal, we want:

$$\sigma_1(\rho) < \sigma_2(\rho) \;\Leftrightarrow\; \Delta_{z1} < \Delta_{z2} . \quad (8)$$

Assuming that $\hat{\sigma}(\rho)$ is a valid indicator for the uncertainty, we derive a notion of relative uncertainty Δ_ρ on the parallax defined as follows:

$$\Delta_\rho = \frac{\hat{\sigma}(\rho)}{\hat{\rho}} > 0 . \quad (9)$$

This allows us to derive a range of values $\left[\frac{\hat{\rho}}{1+\Delta_\rho}, \hat{\rho} \right]$ that is representative of the uncertainty as it monotonously increases with uncertainty. As shown in Fig. 2, the equivalent of this range in the depth domain can be defined as $[\hat{z}, (1+\Delta_z)\hat{z}]$ where $\hat{z} = Z(\hat{\rho})$. With this definition, Δ_z has properties similar to that of Δ_ρ, and it is also representative of the uncertainty on the parallax since Eq. (8) is verified.

To find the relation between Δ_z and Δ_ρ, we use Eq. (3) as follows:

$$(1+\Delta_z)\hat{z} = Z\left(\frac{\hat{\rho}}{1+\Delta_\rho} \right) \Leftrightarrow \Delta_z = \frac{c}{\hat{z}} + (1+\Delta_\rho)\left(1 - \frac{c}{\hat{z}}\right) - 1 > 0 . \quad (10)$$

Since $\Delta_\rho > 0$ and $\hat{z} > 0$, the inequality is verified if $z_V \hat{z} + t_z > 0$ which is the same condition of existence than for the parallax itself [4]. Therefore, the Δ_z quantity is defined for any possible value of the parallax.

In a nutshell, getting joint depth and uncertainty estimates with M4Depth+U_ρ amounts to training the network to infer the parallax and its related uncertainty, then to convert them into depth z and its related uncertainty Δ_z by using Eq. (3) and (10) respectively.

IV. Experiments

In the experiments, we compare our elaborate approach for uncertainty estimation to (1) the probabilistic baseline in various conditions, and (2) existing methods on a benchmark for MVD methods. Before presenting the results, let us first describe the experimental setup.

A. Experimental setup

Datasets. We base our experiments on three datasets, namely Mid-Air [5], KITTI [6], and TartanAir [21]: we use Mid-Air to train and test the method in unstructured environments, KITTI for zero-shot transfer tests on real data in urban environments, and TartanAir for further tests either in urban or unstructured environments. We use the same splits and image resolution as for the original experiments for M4Depth [4].

Performance evaluation. The performance analysis is based on a subset of metrics proposed by Eigen *et al.* [22] for depth estimation, that is "Abs rel", "RMSE log", and $\delta < 1.25$. We also report the quality of uncertainty estimates with the *Area under the Sparsification Error* (AuSE) proposed Ilg *et al.* [10]. This value, derived from so-called *sparsification plots* [23]–[26], has to be minimized for each performance metric for depth estimation. Similar to related works, distant points (ground-truth depth $> 80\,m$) are excluded from the performance metric computations.

Network training. All the performance reported and analyzed in this section are based on networks with six levels trained on the training set of the Mid-Air dataset. We use the same hyper-parameters and the same data augmentation steps as the ones used for M4Depth [4]. However, we let the network train on more iterations (250 k steps). We compute the performance of the network in validation after each epoch and use the set of weights that performed the best in validation for our performance analysis.

B. Results

M4Depth+U_ρ vs M4Depth+U_z. In Section III, we explain how the probabilistic framework can be used as a baseline, referred as M4Depth+U_z, to get the uncertainty on depth estimates for M4Depth. We also propose a more elaborate method to get this uncertainty with M4Depth+U_ρ. We trained our network with both methods on the Mid-Air dataset, and tested the performances in zero-shot transfer on various datasets. Some output results for M4Depth+U_ρ are shown in Fig. 1.

The results given in Table I and the sparsification error curves displayed in Fig. 3 show that estimating depth and uncertainty jointly with M4Depth works well. As hypothesized,

Table I
PERFORMANCE OF M4DEPTH+U_ρ COMPARED TO M4DEPTH+U_z.

Set	Method	Abs Rel		RMSE log		$\delta < 1.25$	
		Perf. ↓	AuSE ↓	Perf. ↓	AuSE ↓	Perf. ↑	AuSE ↓
Mid-Air	M4Depth	0.127	–	0.185	–	0.907	–
	M4Depth+U_z	0.145	0.028	0.190	0.084	0.906	0.009
	M4Depth+U_ρ	0.134	**0.007**	0.188	**0.020**	0.906	**0.006**
KITTI	M4Depth	0.193	–	0.224	–	0.849	–
	M4Depth+U_z	0.140	0.025	0.195	0.046	0.858	0.021
	M4Depth+U_ρ	0.147	**0.021**	0.195	**0.041**	0.858	**0.019**
TtA-W	M4Depth	0.614	–	0.593	–	0.652	–
	M4Depth+U_z	0.618	0.176	0.597	0.217	0.636	0.031
	M4Depth+U_ρ	0.478	**0.058**	0.592	**0.157**	0.646	**0.028**
TtA-O	M4Depth	0.446	–	0.355	–	0.793	–
	M4Depth+U_z	0.468	0.077	0.410	0.155	0.776	**0.020**
	M4Depth+U_ρ	0.268	**0.032**	0.382	**0.122**	0.789	**0.020**

The network was trained and tested with the two loss functions on the Mid-Air dataset, and tested in zero shot transfer on the other datasets. We used the seasons forest winter (TtA-W) and neighborhood (TtA-N) environments of the TartanAir dataset. The best AuSEs for each set are highlighted in bold.

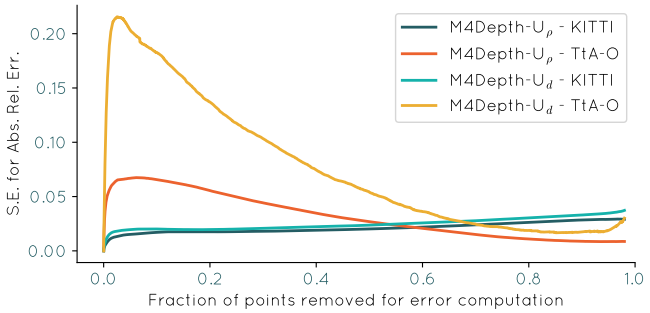


Figure 3. Sparsification error (S.E.) curves on the absolute relative error for M4Depth+U_z and M4Depth+U_ρ on the KITTI dataset, and on the "Old Town" set of TartanAir (TtA-O).

the probabilistic framework underpinning the M4Depth+U_z baseline is sub-optimal while our elaborate uncertainty conversion method, M4Depth+U_ρ, consistently performs better. Also, the AuSE score for M4Depth+U_ρ varies less between datasets when compared to M4Depth+U_z, therefore hinting at more consistent generalization performances. Finally, it is worth noting that both approaches for estimating depth and its uncertainty preserve the raw performance for depth estimation of M4Depth.

The sparsification error curves show that the uncertainty produced by our network is better at discriminating low errors than higher ones, since the sparsification error is higher for lower sparsification values. The upward trend at the very end of the sparsification error curve for M4Depth+U_z on the TartanAir set hints that the network is very confident in some areas with higher errors, which is not desired. This behavior is not observed with M4Depth+U_ρ which further motivates its interest over the baseline.

Robust MVD benchmark. As our method targets autonomous UAV applications, it has to produce estimates for the latest available frame. Therefore, it cannot use future information as opposed to generic multi-view depth estimation methods which can use all the past and upcoming frames of the sequence. Since we are the first to target this specific use case, there is no existing baseline to compare to directly. Nonetheless, we assess the value proposition of M4Depth+U_ρ over some other existing methods on the benchmark proposed by Schröppel *et al.* [11] for joint multi-view depth and uncertainty estimation. Results on the KITTI set of the benchmark are reported in Table II. Despite working with fewer data than other methods, M4Depth+U_ρ outperforms most of the baseline and comes close to the state of the art on this benchmark. This, combined with the fact that M4Depth+U_ρ is at least 2.5 times faster than other methods, leads us to conclude that our method performs on par with existing methods tested on this benchmark, and that M4Depth+U_ρ has a real benefit for practical use.

Table II
PERFORMANCE OF M4DEPTH+U_ρ ON THE UNCERTAINTY BENCHMARK PROPOSED BY SCHRÖPPEL *et al.* [11] FOR MVS METHODS.

Method	Causal	Abs. Rel. (↓)	AuSE	Time [ms]
MVSNet [27]	✗	0.140	0.025	150
Fast-MVSNet [28]	✗	0.121	0.034	350
Vis-MVSNet [12]	✗	0.103	0.028	820
Robust MVD [11]	✗	0.071	0.017	60
M4Depth+U_ρ	✓	0.086	0.020	26

Performances are reported in zero-shot transfer on the 93-images test set for the KITTI dataset used for this benchmark. Inference timings are reported for full-size KITTI images. Note that M4Depth+U_ρ is causal and only uses a sequence of frames that precedes the frame considered for depth inference, while MVD methods are anti-causal as they also use upcoming frames.

Inference statistics. In the configuration used in our experiments, our method has 5.7 M parameters, and requires up to 840 Mo of VRAM to run. On a NVidia V100 GPU and for input samples with a size of 384×384 pixels, M4Depth+U_ρ jointly estimates depth and uncertainty in 18 ms. This is 1 ms more than M4Depth, which means that estimating the uncertainty requires a negligible additional computational time when compared to depth estimation alone.

V. CONCLUSION

In this paper, we showed that it is possible to adapt M4Depth, an efficient depth estimation network designed for autonomous vehicles applications, for joint depth and uncertainty estimation at minimal cost. We also demonstrated that converting the uncertainty values produced by the network into uncertainty values related to depth is better done with an elaborate conversion method, referred as M4Depth+U_ρ, than with the standard probabilistic approach. The performance on the Mid-Air dataset and our tests in zero-shot transfer on the KITTI and TartanAir datasets show that our method emerges as an excellent joint depth and uncertainty estimator. In addition, testing M4Depth+U_ρ on the Robust MVD benchmark in zero-shot transfer confirm that our method performs similarly to other multi-view stereo methods, while being 2.5 times faster and causal, as opposed to these methods.

ACKNOWLEDGEMENT

This work was partly supported by the Walloon Region (Service Public de Wallonie Recherche, Belgium) under grant n°2010235 (ARIAC by DigitalWallonia.ai).

References

[1] F. Mumuni, A. Mumuni, and C. K. Amuzuvi, "Deep learning of monocular depth, optical flow and ego-motion with geometric guidance for UAV navigation in dynamic environments," *Mach. Learn. Appl.*, vol. 10, pp. 2–15, Dec. 2022. [Online]. Available: https://doi.org/10.1016/j.mlwa.2022.100416 1

[2] X. Yang, H. Luo, Y. Wu, Y. Gao, C. Liao, and K.-T. Cheng, "Reactive obstacle avoidance of monocular quadrotors with online adapted depth prediction network," *Neurocomputing*, vol. 325, pp. 142–158, Jan. 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.10.019 1

[3] D. Wang, W. Li, X. Liu, N. Li, and C. Zhang, "UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution," *Comput. Electron. Agric.*, vol. 175, pp. 1–11, Aug. 2020. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105523 1

[4] M. Fonder, D. Ernst, and M. Van Droogenbroeck, "Parallax inference for robust temporal monocular depth estimation in unstructured environments," *Sensors*, vol. 22, no. 23, pp. 1–22, Dec. 2022. [Online]. Available: https://doi.org/10.3390/s22239374 1, 2, 3

[5] M. Fonder and M. Van Droogenbroeck, "Mid-air: A multi-modal dataset for extremely low altitude drone flights," in *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW), UAVision*. Long Beach, CA, USA: Inst. Electr. Electron. Eng. (IEEE), Jun. 2019, pp. 553–562. [Online]. Available: https://doi.org/10.1109/CVPRW.2019.00081 1, 3

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 3354–3361. [Online]. Available: https://doi.org/10.1109/CVPR.2012.6248074 1, 3

[7] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *CoRR*, vol. abs/2107.03342, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2107.03342 1

[8] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 5574–5584. 1, 2

[9] T. Ke, T. Do, K. Vuong, K. Sartipi, and S. I. Roumeliotis, "Deep multi-view depth estimation with predicted uncertainty," in *IEEE Int. Conf. Robot. Autom. (ICRA)*. Xian, China: Inst. Electr. Electron. Eng. (IEEE), May 2021, pp. 9235–9241. [Online]. Available: https://doi.org/10.1109/ICRA48506.2021.9560873 1, 2

[10] E. Ilg, Ö. Çiçek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Eur. Conf. Comput. Vis. (ECCV)*, ser. Lect. Notes Comput. Sci., vol. 11211. Springer Int. Publ., 2018, pp. 677–693. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_40 1, 2, 3

[11] P. Schröppel, J. Bechtold, A. Amiranashvili, and T. Brox, "A benchmark and a baseline for robust multi-view depth estimation," *CoRR*, vol. abs/2209.06681, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2209.06681 1, 4

[12] J. Zhang, S. Li, Z. Luo, T. Fang, and Y. Yao, "Vis-MVSNet: Visibility-aware multi-view stereo network," *Int. J. Comput. Vis.*, vol. 131, no. 1, pp. 199–214, Oct. 2022. [Online]. Available: https://doi.org/10.1007/s11263-022-01697-3 1, 2, 4

[13] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Seattle, WA, USA: Inst. Electr. Electron. Eng. (IEEE), Jun. 2020, pp. 3224–3234. [Online]. Available: https://doi.org/10.1109/cvpr42600.2020.00329 2

[14] W. Su, Q. Xu, and W. Tao, "Uncertainty guided multi-view stereo network for depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7796–7808, Nov. 2022. [Online]. Available: https://doi.org/10.1109/TCSVT.2022.3183836 2

[15] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural RGB-D sensing: Depth and uncertainty from a video camera," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Long Beach, CA, USA: Inst. Electr. Electron. Eng. (IEEE), Jun. 2019, pp. 10 978–10 987. [Online]. Available: https://doi.org/10.1109/CVPR.2019.01124 2

[16] W. Zhao, S. Liu, Y. Wei, H. Guo, and Y.-J. Liu, "A confidence-based iterative solver of depths and surface normals for deep multi-view stereo," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. Montreal, QC, Canada: Inst. Electr. Electron. Eng. (IEEE), Oct. 2021, pp. 6148–6157. [Online]. Available: https://doi.org/10.1109/iccv48922.2021.00611 2

[17] M. Mehltretter and C. Heipke, "Aleatoric uncertainty estimation for dense stereo matching via CNN-based cost volume analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 171, pp. 63–75, Jan. 2021. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2020.11.003 2

[18] C. Homeyer, O. Lange, and C. Schnörr, "Multi-view monocular depth and uncertainty prediction with deep SfM in dynamic environments," in *Int. J. Pattern Recognit. Artif. Intell.*, ser. Lect. Notes Comput. Sci., vol. 13363. Springer Int. Publ., 2022, pp. 373–385. [Online]. Available: https://doi.org/10.1007/978-3-031-09037-0_31 2

[19] M. Klodt and A. Vedaldi, "Supervising the new with the old: Learning SFM from SFM," in *Eur. Conf. Comput. Vis. (ECCV)*, ser. Lect. Notes Comput. Sci., vol. 11214. Springer Int. Publ., 2018, pp. 713–728. [Online]. Available: https://doi.org/10.1007/978-3-030-01249-6_43 2

[20] X. Yang, J. Chen, Y. Dang, H. Luo, Y. Tang, C. Liao, P. Chen, and K.-T. Cheng, "Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 156–167, Jan. 2021. [Online]. Available: https://doi.org/10.1109/TITS.2019.2955598 2

[21] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Las Vegas, NV, USA: Inst. Electr. Electron. Eng. (IEEE), Oct. 2020, pp. 4909–4916. [Online]. Available: https://doi.org/10.1109/IROS45743.2020.9341801 3

[22] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 2366–2374. 3

[23] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow, "Learning a confidence measure for optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1107–1120, May 2013. [Online]. Available: https://doi.org/10.1109/TPAMI.2012.171 3

[24] C. Kondermann, D. Kondermann, B. Jähne, and C. Garbe, "An adaptive confidence measure for optical flows based on linear subspace projections," in *Pattern Recognit.*, ser. Lect. Notes Comput. Sci., vol. 4713. Springer, 2007, pp. 132–141. [Online]. Available: https://doi.org/10.1007/978-3-540-74936-3_14 3

[25] A. S. Wannenwetsch, M. Keuper, and S. Roth, "ProbFlow: Joint optical flow and uncertainty estimation," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. Venice, Italy: Inst. Electr. Electron. Eng. (IEEE), Oct. 2017, pp. 1182–1191. [Online]. Available: https://doi.org/10.1109/ICCV.2017.133 3

[26] J. Kybic and C. Nieuwenhuis, "Bootstrap optical flow confidence and uncertainty measure," *Comput. Vis. Image Underst.*, vol. 115, no. 10, pp. 1449–1462, Oct. 2011. [Online]. Available: https://doi.org/10.1016/j.cviu.2011.06.008 3

[27] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth inference for unstructured multi-view stereo," in *Eur. Conf. Comput. Vis. (ECCV)*, ser. Lect. Notes Comput. Sci., vol. 11212. Springer, 2018, pp. 785–801. [Online]. Available: https://doi.org/10.1007/978-3-030-01237-3_47 4

[28] Z. Yu and S. Gao, "Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Seattle, WA, USA: Inst. Electr. Electron. Eng. (IEEE), Jun. 2020, pp. 1946–1955. [Online]. Available: https://doi.org/10.1109/cvpr42600.2020.00202 4