Uncertainty in Self-supervised Depth Estimation Using Multi-scale Decoders

Mayank Mali

CMU-CS-22-124
August 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

Thesis Committee:
Dr. Jean Oh (advisor)
Dr. Ji Zhang

Submitted in partial fulfillment of the Masters degree in Computer Science

Keywords: Machine Learning, Depth Estimation, Self-Supervised, View Synthesis, Uncertainty, Predictive Uncertainty, Sparsification, Student-teacher framework, U-Net, Decoder, Multi-scale, ResNet

For my family, friends, and faculty that offered endless support and feedback.

Abstract

Depth estimation is an image translation problem that predicts depth-maps for a given camera image, and has fostered research in various applications including self-driving vehicles. Self-supervised depth estimation methods are of particular interest since ground truth LIDAR depth is expensive to acquire and instead use view synthesis as weaker supervision. Generally, the produced depth maps to date are only point estimates of an underlying depth distribution due to randomness in model training, resulting in noisy depth estimates that can propagate errors and lead to inaccurate or fatal decisions in real-world applications. Recent interest has been sparked in reducing such noise by modeling the uncertainty of depth estimates. Empirical uncertainty strategies seek to predict uncertainty via statistical methods on treating independent models as black-box predictors. Of particular interest are predictive strategies that seek to learn the inherent uncertainty of a depth model. For example, student-teacher frameworks train one network to learn the depth output distribution of another. Such methods are desirable due to the advantage of requiring fewer training and space resources compared to other empirical methods. In this work, we study self-supervised depth models with a U-Net architecture that output depths at multiple scales. In particular, we explore a novel predictive uncertainty model that only has access to these scales and the U-Net bottleneck feature. We evaluate and discuss the novel method alongside other uncertainty strategies on the KITTI dataset.

Acknowledgments

Very special thanks to Dr. Soonmin Hwang for investing generous time in mentoring, teaching, and encouragement, extending the limits of my self-confidence and accelerating my understanding of depth estimation. Warm thanks to my advisor Dr. Jean Oh for much-needed guidance, direction, and support. Also thanks to Dr. Ji Zhang for offering time to be part of the thesis committee.

I also extend my gratitude to all my peers in the Bot Intelligence Group at CMU, especially Ingrid Navarro, for their valuable time and effort in providing feedback for this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Self-supervised Depth Estimation Primer

1.1.1 Motivation

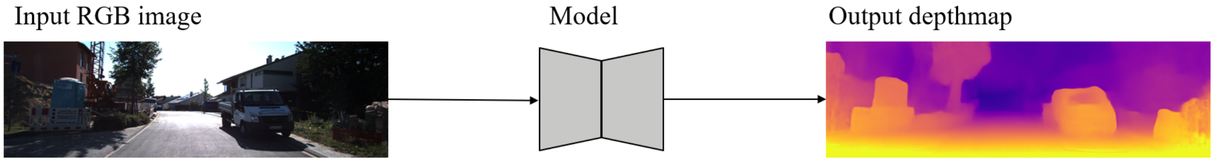

Figure 1.1: Monocular depth estimation is the problem of predicting pixel-wise depth-maps for a single camera image.

The problem of monocular depth estimation is to predict depthmaps from a given camera image as shown in Figure 1.2, where the underlying premise is that single images of an indoor/outdoor 3D scene contains various depth cues (i.e. object spatial arrangement, textures) that encode the distance from the camera i.e. depth. Furthermore, if camera images are collected via video or as stereo image pairs, then motion parallax between frames or stereo parallax respectively can give further depth cues.

However, ground truth depth points, usually from LIDAR (Light Detection and Ranging) hardware, are expensive to acquire. To address this problem, many works use view synthesis which is a geometry-based supervision. Sometimes called warping or reconstruction, this pipeline takes a camera image of a 3D scene from one viewpoint (i.e. camera pose) and predicts the image of the same scene for another viewpoint. For example, the KITTI dataset [4] offers stereo camera image pairs as the two views. Another option is to collect monocular video, and use different frames of time as the viewpoints [13]. The latter technique is called SfM (Structure from Motion) and involves using visual odometry to understand the transformation between viewpoints.

1.1.2 View Synthesis Pipeline

During training, given a pair of images I_t, I_c of two camera poses, the supervision signal comes by using predicted depth-map d_t of pixels in I_t to warp I_t into the pose of the I_c, and then comparing the warped image $\widetilde{I}_c$ with I_c using photometric loss. This is a window-based loss that penalizes structural differences of the two images $\hat{I}_c, I_c$, meaning if the same neighborhood of pixels for the two images have similar spatial arrangements, then the loss contribution from that region is smaller.

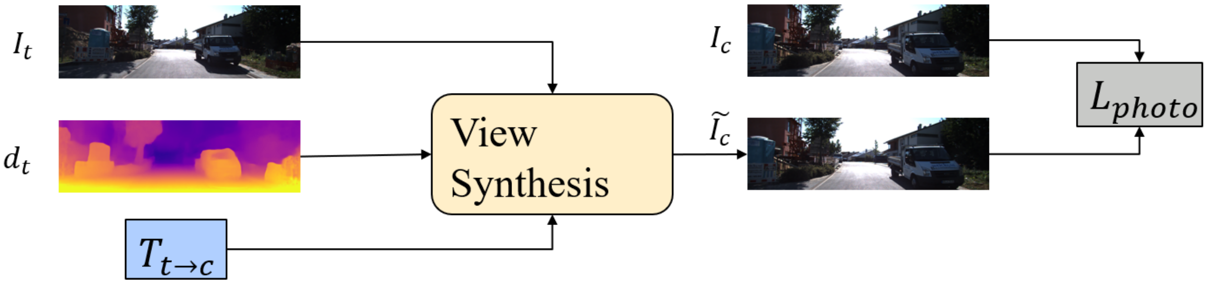

Figure 1.2: Overview of the view synthesis problem. Given two camera images, I_t, I_c, $\widetilde{I}_c$ is synthesized from I_t, predicted depth d_t of I_t, and transformation between the views $T_{t\to c}$. Photometric loss L_{photo} compares I_c and $\widetilde{I}_c$, jointly supervising d_t and $T_{t\to c}$ (if learned).

During training, the images I_t, I_c are offered either as stereo camera pairs [3, 5], or as successive frames in a video [6, 7, 11, 13].

In the geometry-based view synthesis pipeline, pixels from I_t must be unprojected into 3D space using the predicted depth d_t and the camera's inverse intrinsics K^{-1}, which is used when the camera is assumed to have a pinhole geometry, where the intrinsics K is determined by the focal lengths f_x, f_y and pixel center c_x, c_y

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}.$$

Once in 3D space in I_t's camera reference frame, it is transformed to I_c's reference frame by either additionally predicting the transformation pose $T_{t\to c}$ between the frames in 3D space via a pose model that maps $(I_t, I_c) \mapsto T_{t\to c}$ or calculating it based on odometry hardware. In the stereo case where I_t, I_c form the left and right images, camera extrinsics (the transformation from world to camera reference frame). Finally, the 3D point in I_c's reference frame is projected to I_c's camera via the intrinsics (assuming both cameras are have identical pinhole geometries). The pipeline is summarized below:

1. Predict depthmap d_t from image I_t.
2. Unproject points as homogenous coordinates $p = (u, v, 1)$ in I_t to I_c's 3D reference frame using d_t and camera inverse intrinsics K^{-1}:

$$\phi(p, d_t) = d_t K^{-1} I_t(p)$$

3. Transform 3D points $\phi(p, d_t)$ to I_c's reference frame.

$$P = T_{t \to c}\phi(p, d_t)$$

4. Project 3D points in I_c's reference frame to I_c's camera with camera intrinsics K:

$$\widetilde{I}_c(p) = \pi(P) = \frac{1}{P_z}KP = \frac{1}{P_z}KT_{t \to c}d_tK^{-1}I_t(p)$$

The prediction image $\widetilde{I}_c$ is compared with I_c using photometric loss. This pipeline transforms points from one camera to another based on the pinhole geometry (camera intrinsics) and the relative geometry of the camera poses (called extrinsics, the transformation between the 3D scene and camera coordinates).

It is important to note that due to the geometric grounding of the view synthesis pipeline, we can generalize this process to datasets with stereo video, and even videos from multi-camera rigs. Once unprojected via ϕ, we can chain transformations between cameras *and* between time frames simultaneously (e.g. via an additional matrix multiplication step). In this case, a viewpoint is generalized to any camera image in any time, and additional supervision comes from multiple photometric losses from reconstructions between these viewpoints.

Even more surprising is that some works have shown that we can predict the camera intrinsics K if unknown [1], and even the pinhole assumption can be relaxed. For cameras without pinhole geometries (e.g. fish-eye camera), a method called NRS (Neural Ray Surfaces) are used to learn the unprojection and projection operations ϕ, π themselves [11].

Regardless, during evaluation the depth model is considered separately and can make depth-map predictions on single camera images.

There has been a lot of attention on using CNNs (Convolutional Neural Networks).

Due to the high cost of LIDAR ground truth depths and ease of capturing many images via video, much attention has been paid to self-supervised methods, including those that use photometric warping loss between consecutive video frames.

TODO Why self-supervision using photometric warping loss is weaker.

Need to explain scale aware (stereo, fsm, anything where camera extrinsics are known -¿ provides metric scale) vs scale ambiguous (monodepth, transformation between pose based on visual odometry or hardware odometry).

Photometric assumptions about scene.

How some works address some of the broken photometric assumptions.

1.2 Uncertainty

Uncertainty is supposed to encode the errors.

No ground truth in uncertainty for evaluation. If uncertainty encodes the errors via sparsification-based metrics (AUSE, AURG).

Chapter 2

Related Works

2.1 Self-supervised monodepth

2.2 Uncertainty in flow estimation

2.3 Self-teaching

2.4 Empirical methods

2.4.1 Bootstrap ensembles

Figure 2.1.

2.4.2 Snapshot ensembles

Figure 2.2

2.4.3 Dropout sampling

Figure 2.3

Uncertainty studied for flow estimation in SfM.

Empirical (Greybox/BlackBox uncertainty inference) and predictive methods so far.

Some new things people have tried in research (self-teaching for self-supervised, Bayesian NNs [intractable, approx.],).

SAFENet and multi-task learning where features inform each other (only applicable to joint teaching).

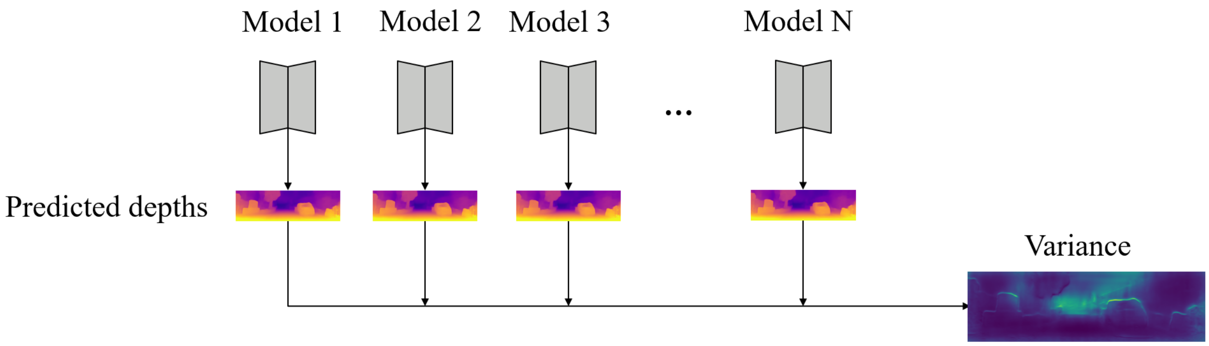

Figure 2.1: Using variance across bootstrap ensembles as uncertainty.

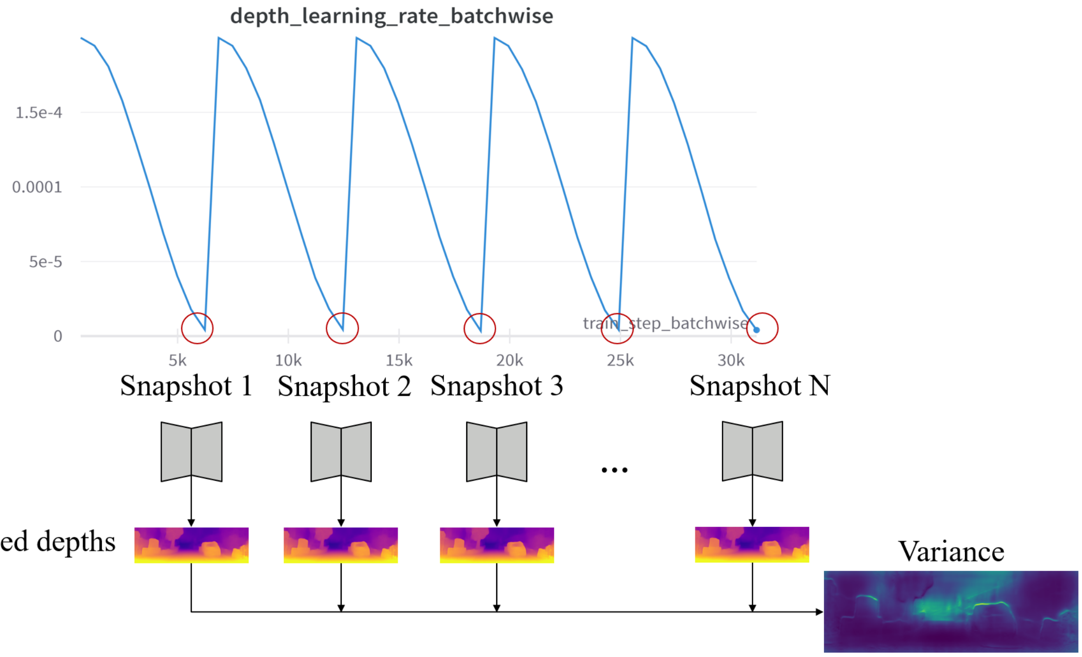

Figure 2.2: Using variance across snapshot ensembles as uncertainty. Snapshots of the model are taken at low points of a cyclic learning rate throughout a single training session.

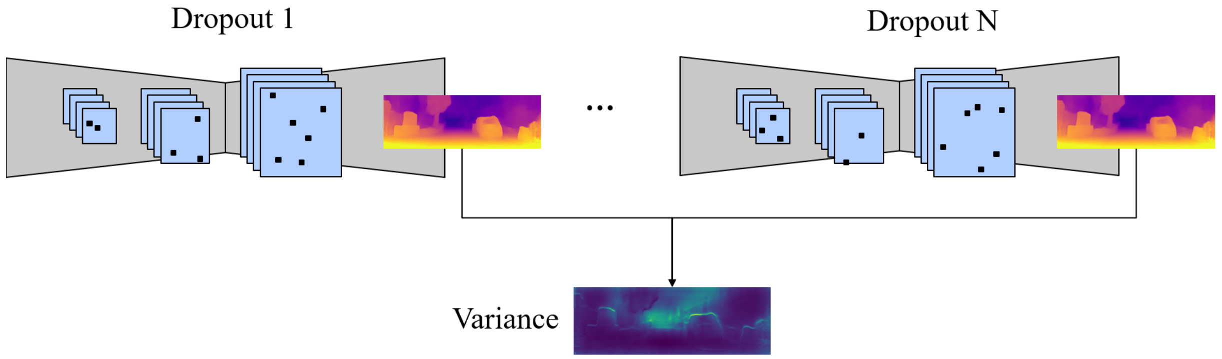

Figure 2.3: Using random dropout to zero-out certain neurons. Uncertainty is taken as variance across N independent dropout forward passes with the *same* model after training.

Chapter 3

Experimental Results

3.1 Metrics

3.1.1 Depth Metrics

For evaluation, LIDAR ground truth depth d^* is compared against the predicted depth d_t.

Metric	Equation

3.1.2 Uncertainty Metrics

No ground truth uncertainty.

Sparsification as a measure of how well uncertainty encodes error.

AUSE, AURG

3.2 Metrics

Show ResNet18 pose network graphs together, then make general observations in discussion.

Remember to compare groups of runs against each other (posenet type, freeze vs detach teaching, etc.)

Model	Abs Rel↓	Sq Rel↓	RMSE↓	$\mathrm{RMSE}_{log}\downarrow$	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
scale	**0.132**	**0.917**	**4.920**	**0.208**	0.845	0.951	**0.979**
boot	0.134	1.055	4.991	0.209	**0.849**	0.951	0.978
snap	0.139	1.139	5.172	0.212	0.842	0.951	0.977
scalenet teach	0.144	1.126	5.283	0.220	0.825	0.947	0.977

Model	AUSE↓	AURG↑
scale	1.658	2.415
boot	**0.951**	**3.209**
snap	1.631	2.664
scalenet teach	2.540	1.811

Table 3.1: depth (top) and uncertainty (bottom) metrics after gt-scaling and flip post-processing. All architectures have DepthResNet18 depth and PoseNet pose networks.

Model	Abs Rel↓	Sq Rel↓	RMSE↓	$\mathrm{RMSE}_{log}\downarrow$	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
scale	**0.132**	**0.917**	**4.920**	**0.208**	0.845	0.951	**0.979**
boot	0.134	1.055	4.991	0.209	**0.849**	0.951	0.978
snap	0.139	1.139	5.172	0.212	0.842	0.951	0.977
scalenet teach	0.144	1.126	5.283	0.220	0.825	0.947	0.977

Model	AUSE↓	AURG↑
scale	1.658	2.415
boot	**0.951**	**3.209**
snap	1.631	2.664
scalenet teach	2.540	1.811

Table 3.2: All metrics are computed after applying ground-truth median scaling and image flipping post-processing. All architectures have DepthResNet18 depth and PoseResNet18 pose networks.

Chapter 4

Discussion

What if depth estimates predicted as certain but not accurate? → most dangerous kind of estimate (false certainty).

Chapter 5

Conclusion

What makes architecture of depth decoder better/worse than baseline methods.

How to explain performance of scalenet using the abalation and difference between other methods.

TODO: [1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

Bibliography

[1] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019. 1.1.2, 5

[2] Jaehoon Choi, Dongki Jung, Donghwan Lee, and Changick Kim. Safenet: Self-supervised monocular depth estimation with semantic-aware feature extraction. *arXiv preprint arXiv:2010.02893*, 2020. 5

[3] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 1.1.2, 5

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1.1.1, 5

[5] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 1.1.2, 5

[6] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1.1.2, 5

[7] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. 1.1.2, 5

[8] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 5

[9] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters*, 7(2):5397–5404, 2022. 5

[10] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 5

[11] Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Wolfram Burgard, Greg Shakhnarovich, and Adrien Gaidon. Neural ray surfaces for self-supervised learning of depth and ego-motion. In *2020 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2020. 1.1.2, 1.1.2, 5

[12] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 340–349, 2018. 5

[13] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 1.1.1, 1.1.2, 5