Semi-Supervised Learning with Mutual Distillation for Monocular Depth Estimation

Jongbeom Baek*, Gyeongnyeon Kim*, and Seungryong Kim

Abstract—We propose a semi-supervised learning framework for monocular depth estimation. Compared to existing semi-supervised learning methods, which inherit limitations of both sparse supervised and unsupervised loss functions, we achieve the complementary advantages of both loss functions, by building two separate network branches for each loss and distilling each other through the mutual distillation loss function. We also present to apply different data augmentation to each branch, which improves the robustness. We conduct experiments to demonstrate the effectiveness of our framework over the latest methods and provide extensive ablation studies.

I. INTRODUCTION

Monocular depth estimation, estimating a depth map from a single image, can facilitate numerous Computer Vision and Robotics applications, such as scene understanding, SLAM, and autonomous driving [1], [2], [3].

Early approaches [4], [5], [6], [7], [8] that solve the task with deep Convolutional Neural Networks (CNNs) were formulated in a supervised manner, which relies on large ground-truth depth data. But, constructing such data is very costly and labour-intensive [9], [10]. In addition, they have a limited generalization ability to the regions where the ground-truth depths do not exist. To alleviate the reliance on large ground-truth data, unsupervised, also called self-supervised, learning based methods [9], [10], [11], [12], [13] have been proposed, which cast the task as an image synthesis from stereo image pairs or monocular video sequences. Although it turns out that the unsupervised loss is an appealing alternative, but it often leads to blurry results around depth boundaries [11], [14].

To learn monocular depth estimation networks in a semi-supervised manner, some methods [15], [16] directly combine the sparse supervised and unsupervised loss functions, but such straightforward approach inherits limitations of both loss functions and accumulates the errors by each loss, which the networks trained with each loss solely may handle.

Meanwhile, some methods [17], [18], [19] attempted to train monocular depth estimation networks through pseudo depth labels from stereo image pairs generated by traditional or pre-trained stereo matching modules [18], [19]. To mitigate performance degeneration by inaccurate pseudo depth labels, the confidence of the pseudo depth is also predicted. However, demanding additional stereo matching and confidence estimation modules hinders their applicability.

To overcome the aforementioned limitations, we propose a novel semi-supervised learning framework for monocular depth estimation, as in Fig. 1. To achieve the complementary advantages of both sparse supervised and unsupervised loss functions, we present to build separate networks tailored for each loss, called supervised and unsupervised branches. We train the networks in a probabilistic inference framework to predict both the depth and its uncertainty, so as to distil confident depth knowledge between the branches. To consider the sparsity nature of supervised loss, we introduce unprojected point filtering loss at supervised branch to localize the regions, where ground-truth depths do not exist. By leveraging the proposed mutual distillation loss, defined with depth and confidence maps from each branch, two networks converge to better depth solutions in a mutual and boosting manner. In addition, our framework enables applying different data augmentation to each branch, which improves the robustness. Experiments on standard benchmarks for monocular depth estimation such as KITTI [23] and Cityscapes [24] prove the effectiveness of our approach over the latest methods. We also provide an ablation study to validate and analyze components in our approach.

II. RELATED WORK

Monocular Depth Estimation. Eigen et al. [4] pioneered the approach with deep CNNs for monocular depth estimation. Following [4], several methods have been proposed to improve the performance [5], [6], [7], [25], [8]. Traditional
methods mentioned above were often formulated in a supervised manner that require massive ground-truth depth maps.

To address this limitation, self-supervised approaches based on other forms of supervision, i.e., from stereo image pairs or video, have been introduced [10], [9], [11], [26], [13]. [9] used the stereo reconstruction. [11] used additional loss to enforce left-right consistency of the predicted disparities. [26] and [13] simultaneously learn depth and pose networks from video. Although they seem to be an appealing alternative, the reconstruction loss has problems, e.g., missing objects and over-smoothing boundaries.

To take advantages of both sparse supervised and unsupervised learning methods, semi-supervised learning methods have also been presented [15], [16]. [15] directly combined both loss functions. [16] improved performance by applying left-right consistency loss. They inherit limitations of both loss functions, and overcoming this is the topic of this paper.

Meanwhile, as a proxy loss signal, using stereo knowledge for monocular depth estimation is also proposed [17], [18], [19]. Tonioni et al. [18] showed the possibility to substitute the ground-truth with proxy labels obtained through traditional stereo matching and confidence measure [27], [17], [19] utilize pseudo labels from pre-trained stereo matching network to train student by distilling the knowledge. However, requiring additional stereo matching and confidence estimation modules limits their applicability. Unlike these methods [17], [18], [19], our approach only exploits the monocular depth estimation networks.

**Self-training.** Self-training is one of popular semi-supervised learning approaches that encourages a model to follow the pseudo label from the model’s prediction itself for entropy minimization [28], which has been successfully utilized in many tasks [29], [30], [31]. However, when the pseudo labels are inaccurate, vanilla methods often produce confirmation bias [32]. To overcome this, [33] learns the teacher as well to enhance the student’s performance on labeled dataset. Our approach is the first attempt to exploit self-training paradigm for monocular depth estimation.

**Uncertainty Estimation.** Estimating the uncertainty of predictions is critical for safety-critical applications [34], [35], [36]. There are two main types of uncertainty, i.e., epistemic uncertainty and aleatoric uncertainty [37]. Epistemic uncertainty is in the model, which captures ignorance about the models due to the lack of training data. Aleatoric uncertainty, on the other hand, captures noise inherent in the environment such as measurement noise. Kendall et al. [38] studied the benefits of modeling uncertainty in Bayesian deep learning models for vision tasks. For monocular depth prediction, Poggi et al. [20] studied how to measure the uncertainty for monocular depth estimator. In this paper, we propose to use uncertainty for mutual distillation.

### III. Preliminaries

Let us denote an image and depth as $I$ and $D$, respectively. Our objective is to learn monocular depth estimation network that takes the image $I$ as input and produces its corresponding depth $D$. To learn the network in a supervised manner, the ground-truth depth $D_{GT}(i)$, defined at all points $i$, is required, but establishing a large-scale dense depth data is very costly and labour-intensive [39]. Instead, sparse ground-truth depth labels can be relatively easily acquired, e.g., by the LiDAR sensor, since the LiDAR laser measurements are projected to a sparse subset in an image with a limited amount of scan lines [15]. Recent research trends thus focus on leveraging the unlabeled data, i.e., adjacent images or the points having no ground-truth depth labels, while mitigating the reliance on labeled data, i.e., sparse depth maps.

When sparse supervision is available to train the network, we can minimize sparse supervised loss function $L_S$ between predicted $D$ and sparse ground-truth $D_{GT}$ such that

$$L_S = \frac{1}{N_D} \sum_{i \in \Omega_D} \|D(i) - D_{GT}(i)\|_1, \quad (1)$$

where $\Omega_D$ indicates a set of pixels where ground-truth depths are available, and $N_D$ is the number of pixels in $\Omega_D$. Due to sparsity of the ground-truth (e.g., 3% density in KITTI depth maps [40]), minimizing the loss function $L_S$ solely cannot guarantee high-precision depth estimation, e.g., especially at sky regions or transparent object regions where the LiDAR sensor does not capture [41].

Instead of using ground-truth depths for training, unsupervised or self-supervised learning-based methods [11], [26],...
A. Motivation and Overview

To overcome the aforementioned limitations of conventional methods [15], [19], we present a novel semi-supervised learning framework for learning monocular depth estimation networks. To achieve the complementary advantages of both sparse supervised loss and unsupervised loss, as illustrated
in Fig. 3 we design two independent networks with same architecture tailored for each loss function, called supervised branch and unsupervised branch. They are learned independently and distilled each other through the proposed mutual distillation loss function. To extract confident pseudo depth labels from each network, we learn the networks in a probabilistic fashion to estimate the distribution of output depth. The uncertainties are used to adjusting the reliability of the depth maps when distilling the depth knowledge of one network to the other network. We also apply two different data augmentation to two branches.

B. Mutual Distillation for Semi-Supervised Learning

In this section, we describe the loss functions defined for each branch and mutual distillation loss.

Learning Depth and Uncertainty. Considering the uncertainty and ignoring the regions with high uncertainty enable transferring reliable depth knowledge to each other in our framework. To this end, we leverage a negative log-likelihood minimization to infer the uncertainty, as well as depth, as in [20]. Specifically, the predictive distribution of the output depth can be modelled as the Laplacian likelihood [38] as

\[ L_{S,a} = \frac{1}{Nd} \sum_{i \in \Omega_D} \left( \frac{\| D_S(i) - D_G(i) \|_1}{\sigma_S(i)} + \mu_S \log(\sigma_S(i)) \right), \tag{8} \]

where \( D_S \) and \( \sigma_S \) denote the predicted depth and its corresponding uncertainty, respectively. \( \mu_S \) is a hyperparameter. The additional logarithmic term prevents the uncertainty from approaching infinite predictions. However, due to the sparsity nature of ground-truth depth maps, minimizing the above loss function cannot recover all the pixels in an image. For instance, the estimated depth maps and uncertainty maps would be ambiguous at the sky or upper parts of an image, out of the field of view of LiDAR, as the networks have never seen the ground-truth depths at such regions. Even though the depth quality is poor at the regions, if the uncertainty estimation is reliably measured, such unreliable depth labels could be ignored when transferring depth knowledge. To overcome this, we introduce additional loss term to deal with such regions, by indicating such regions unreliable, and the loss function is then reformulated such that

\[ L_{S,a} = L_{S,a} + \frac{1}{Nd} \sum_{j \in \Omega_D} \left( \frac{1}{\sigma_S(j)} + \mu_S \log(\sigma_S(j)) \right), \tag{9} \]

where \( \Omega_D \) denotes the pixels outside \( \Omega_D \), and \( M \) is a hyperparameter. The latter term is called unprojected point filtering loss. Fig. 4 visualizes the effect of the term. Note that our loss function \( L_{S,a} \) is the first attempt to simultaneously predict the depth and its uncertainty under sparse depth supervisions. We similarly learn the unsupervised branch with the loss:

\[ L_{U,a} = \frac{1}{N} \sum_{i \in \Omega} \left( \frac{\text{pe}(I(i),I'(i))}{\sigma_U(i)} + \mu_U \log(\sigma_U(i)) \right), \tag{10} \]

Distillation of Depth with Uncertainty. By using the predicted depth map and its uncertainty map, we formulate a mutual distillation loss function to mutually boost each network. Specifically, to transfer the confident depth knowledge from unsupervised branch to supervised branch, we define the distillation loss function \( L_{S,d} \) such that

\[ L_{S,d} = \frac{1}{N} \sum_{i \in \Omega} \frac{\| D_S(i) - D_G(i) \|_1}{\sigma_S(i)}, \tag{11} \]

where \( N = \sum(1/\sigma_U(i)) \). The loss function \( L_{U,d} \) is similarly defined such that

\[ L_{U,d} = \frac{1}{N} \sum_{i \in \Omega} \frac{\| D_U(i) - D_G(i) \|_1}{\sigma_S(i)}. \tag{12} \]

Unlike conventional methods that leverage the stereo knowledge and its thresholded confidence [17], [18], [19], [53], we exploit the predicted uncertainty itself since it empirically yields better performance. In addition, unlike conventional semi-supervised methods [15], [16] that only leverage the fixed, sparse ground-truth depth maps as labeled data, our approach enlarges the pseudo labels as evolving the training, dramatically boosting the performance. Fig. 5 visualizes the depth and confidence at two branches as evolving iterations, which shows complementary information of them.

Total Loss. By considering the loss functions discussed so far, the total loss functions are defined such that: for supervised branch, \( L_{S,total} = L_{S,a} + \lambda_S L_{S,d} \) and for unsupervised branch, \( L_{U,total} = L_{U,a} + \lambda_U L_{U,d} + \lambda_{\text{smooth}} L_{\text{smooth}} \), where \( \lambda_S, \lambda_U, \) and \( \lambda_{\text{smooth}} \) are weighting parameters.

C. Noising the Networks: Data Augmentation

In many literature for semi-supervised learning [54], [29], [30], some methods attempted to learn robust features by giving different perturbations to separate networks and aligning the features. Inspired by these, since our network consists of two separate branches, different kinds of data augmentation can be applied to improve the robustness. To improve the ability to recover an object instance without relying on the bias from the background context or geometric structure of background such as the vanishing point, we present a novel augmentation technique for monocular depth estimation that
TABLE I: Quantitative results on KITTI dataset [40]. The best results are in bold, and the second best results are underlined. Self, Sup and Semi respectively indicate self-supervised, supervised and semi-supervised learning, with (M), (S) and (MS) respectively indicating monocular, stereo, and both. Our method trained with self-supervised learning on monocular stereo videos is exactly same as Monodepth2 [13] (denoted Ours (Monodepth2 [13]).

Table:
| Methods          | Supervision | # param. | time | Abs Rel | Sq Rel | RMSE | RMSE log | δ < 1.25 | δ² < 1.25 | δ³ < 1.25 |
|------------------|-------------|----------|------|---------|--------|------|----------|----------|----------|----------|
| Eigen et al. [4] | Sup         | 54M      | -    | 0.203   | 0.154  | 6.307| 0.622    | 0.702    | 0.890    | 0.890    |
| Liu et al. [47]  | Sup         | 14M      | 2.9ms| 0.201   | 0.158  | 6.471| 0.273    | 0.680    | 0.898    | 0.967    |
| Ours             | Sup         | 14M      | 2.9ms| 0.206   | 0.158  | 6.471| 0.273    | 0.680    | 0.898    | 0.967    |
| Monodepth [11]   | Self (S)    | 56M      | 9.4ms| 0.138   | 1.186  | 5.650| 0.234    | 0.813    | 0.930    | 0.996    |
| MonoResMatch [22]| Self (S)    | 41M      | 8.3ms| 0.111   | 0.867  | 4.714| 0.199    | 0.864    | 0.954    | 0.997    |
| Uncertainty [20]  | Self (S)    | 14M      | 3.6ms| 0.107   | 0.811  | 4.796| 0.200    | 0.866    | 0.952    | 0.978    |
| PackNet-SfM [48] | Self (M)    | 122M     | 9.5ms| 0.111   | 0.785  | 4.601| 0.189    | 0.878    | 0.960    | 0.982    |
| DepthHint [49]   | Self (S)    | 33M      | 6.6ms| 0.102   | 0.762  | 4.602| 0.189    | 0.880    | 0.960    | 0.981    |
| Insta-DM [50]    | Self (S)    | 14M      | 3.0ms| 0.112   | 0.777  | 4.772| 0.191    | 0.872    | 0.959    | 0.982    |
| Ours (Monodepth2 [13]) | Semi (S) | 14M      | 2.9ms| 0.106   | 0.818  | 4.750| 0.196    | 0.874    | 0.957    | 0.978    |
| SVSM FT [51]     | Semi (S)    | -        | -    | 0.102   | 0.700  | 4.681| 0.200    | 0.872    | 0.954    | 0.978    |
| Kuznetsov et al. [15] | Semi (S) | 81M      | -    | 0.113   | 0.741  | 4.621| 0.189    | 0.862    | 0.960    | 0.986    |
| OnboardDepth. [52]| Semi (S)    | -        | -    | 0.115   | 0.766  | 4.665| 0.189    | 0.861    | 0.957    | 0.983    |
| Cho et al. [19]  | Semi (S)    | -        | -    | 0.099   | 0.748  | 4.599| 0.183    | 0.850    | 0.959    | 0.983    |
| Ours             | Semi (S)    | 18M      | 2.9ms| 0.117   | 0.753  | 4.423| 0.273    | 0.870    | 0.946    | 0.965    |
| Ours             | Semi (M)    | 18M      | 2.9ms| 0.101   | 0.673  | 4.292| 0.176    | 0.892    | 0.965    | 0.984    |
| Ours             | Semi (MS)   | 18M      | 2.9ms| 0.101   | 0.657  | 4.262| 0.176    | 0.892    | 0.966    | 0.984    |

V. EXPERIMENTAL RESULTS

A. Implementation Details

We implement our networks with the Pytorch library [55]. Our monocular depth estimation network is based on U-net architecture [56], similar to Monodepth2 [13], in which one shared encoder extracts low resolution, high-dimensional features from the input image I and two separate decoders estimate the depth D and its uncertainty σ, respectively. We design the decoder as a mirrored version of the encoder.

We conduct all our experiments with 24GB RTX-3090 GPU. We set the learning rate as 10⁻⁴ and batches of images downsampled to 640 × 192 as 16. We use Adam optimizer with β₁ = 0.9 and β₂ = 0.999. We set the SSIM weight as α = 0.85, following [11]. We set the weight parameters such that λ_S = 1, λ_U = 0.05, Lsmooth = 0.001, µ_S = 3, and µ_U = 0.03, determined by cross-validation. We use the proposed data augmentation, as well as flipping and jittering, widely used in many literature [57], [58]. For uncertainty estimation, we train the network to predict the log variance log(σ) because it is more numerically stable than regressing the variance as the loss avoids any division by zero. Similar to Monodepth2 [13], we use multi-scale loss functions. We will make our code publicly available in case of acceptance.

B. Experimental Settings

In this section, we conduct extensive evaluations to verify the robustness of our framework in comparison with existing methods such as MonoDepth [11], Monodepth2 [13], Uncertainty [20], MonoResMatch [22], DepthHint [49], PackNet-SfM [48], Kuznetsov et al. [15], SVSM FT [51], and Amiri et al. [16]. In experiments, we use KITTI benchmark [40] and Cityscape benchmark [24]. We train the proposed networks on KITTI benchmark [40] with pre-processing by the Zhou et al. [26] to remove static frames, which results in 39,810 monocular triplets for training and 4,424 for validation. We use annotated depth maps, refined by [41], to train supervised

TABLE II: Quantitative results on Cityscape dataset [24] without fine-tune.

| Methods          | Abs | Sq | RMSE | RMSE log | δ < 1.25 |
|------------------|-----|----|------|----------|----------|
| Monodepth [11]   | 0.631| 10.257| 13.424| 0.525    | 0.281    |
| MonoResMatch [22]| 0.241| 2.149| 9.064 | 0.296    | 0.570    |
| PackNet-SfM [48]| 0.245| 2.240| 8.920 | 0.298    | 0.557    |
| Monodepth2 [13]| 0.242| 2.308| 8.563 | 0.290    | 0.591    |
| DepthHint [49]| 0.220| 2.008| 8.363 | 0.273    | 0.613    |
| Ours             | 0.220| 1.955| 8.234 | 0.270    | 0.612    |


### C. Experimental Results

**Results on KITTI.** We evaluated the monocular depth estimation performance on the KITTI Eigen Split [4]. Table I shows the comparison of our method with the state-of-the-art methods. Our approach achieves competitive performance in comparison to other methods. Regarding the model capacity, e.g., parameters or time, our method is highly comparable to methods that have bigger networks, e.g., PackNet [48]. In addition, although our current model was based on unsupervised loss functions presented in Monodepth2 [13], better unsupervised loss functions, as in DepthHint [49], could boost the performance. In qualitative evaluations of Fig. 6, our model shows robust prediction results while preserving the shape of the object well compared to other methods.

**Results on Cityscape.** In Table II and Fig. 7, we also evaluated our method on the Cityscapes dataset [24], where we evaluate the generalization ability of the networks learned from the KITTI dataset [40]. The evaluation is performed with the depth by the Semi-Global Matching (SGM) [60] as a ground-truth. The outstanding performance of our method demonstrates the proposed mutual distillation shows a satisfactory generalization capability for different datasets.

### D. Ablation Study

**Comparison of Mutual Distillation Modules.** We analyze four different kinds of techniques to define our mutual distillation loss. We consider the uncertainty learned in our method, and the confidence learned from CCNN [45], as done in [61]. We first evaluate uncertainty weighting (UW) in our method, and uncertainty thresholding (UT) (with parameter $\tau = 0.1$), and both (UWT). In addition, we evaluate the confidence thresholding (CT), used in stereo knowledge distillation framework [11], [17], [19], [18]. In this study, our uncertainty weighting (UW) for mutual distillation shows the best performance. We hypothesize that since our uncertainty was simultaneously trained with depth prediction, its magnitude itself can be a reliable cue to adjust the loss when transferring depth knowledge, while confidence learned from additional networks, e.g., CCNN [45] needs the proper thresholding, which is hard to tune.

**Components Analysis.** To better understand how the components of our model contribute to the overall performance, in Table IV we conduct ablation study on key components, mutual distillation (D), unprojected point filtering (M), and noise (N). The results demonstrate that the baseline model, without any our contributions, performs the worst. However, when combined together, all of our components lead to a significant improvement.

**Final Depth.** Table V demonstrates the final performance, after convergence of training, of our two branches, supervised branch and unsupervised branch. Because each branch has its own loss functions, the performance may be different during training, but after convergence, the performance gap was marginal thanks to the proposed mutual distillation loss functions. Empirically, we select the depths from unsupervised branch as our final depth prediction results as they are more plausible.

### VI. Conclusion

In this paper, we have proposed a novel semi-supervised learning (SSL) framework for monocular depth estimation. To achieve the complementary advantages of both sparse supervised and unsupervised loss functions for monocular depth estimation, we defined two separate branches for each loss, and distilled the depth knowledge each other with the learned uncertainties. We also proposed to apply different augmentation to each network, which boosts the robustness of the networks. We have shown that our method surpasses the current state-of-the-art in several benchmarks.

---

**Acknowledgements.** This research was supported by the MSIT, Korea (IITP-2022-2020-0-01819, ICT Creative Convergence program), and National Research Foundation of Korea (NRF-2021R1C1C1006897).

---

**TABLE III: Evaluation of mutual distillation strategy:** using uncertainty weighting (UW) (Ours), uncertainty thresholding (UT), both (UWT) and confidence thresholding (CT).

| Methods      | Abs  | Sq   | RMSE | RMSElog | $\delta < 1.25$ |
|--------------|------|------|------|---------|-----------------|
| UW (Ours)    | 0.102| 0.664| 4.277| 0.119   | 0.892           |
| UT           | 0.104| 0.669| 4.401| 0.180   | 0.886           |
| UWT          | 0.103| 0.665| 4.422| 0.180   | 0.885           |
| CT           | 0.107| 0.730| 4.301| 0.179   | 0.887           |

**TABLE IV: Ablation study on main components:** distillation (D), unprojected points filtering (M), and noise (N).

| Methods        | Abs  | Sq   | RMSE | RMSElog | $\delta < 1.25$ |
|----------------|------|------|------|---------|-----------------|
| Baseline       |      |      |      |         |                 |
| Ours (D)       | ✓    | 0.103| 4.327| 0.889   |                 |
| Ours (D+M)     | ✓    | 0.102| 4.272| 0.892   |                 |
| Ours (N)       | ✓    | 0.107| 4.686| 0.884   |                 |
| Ours (D+M+N)   | ✓    | 0.101| 4.262| 0.892   |                 |

**TABLE V: Evaluation of final depth quality.**

| Methods        | Abs  | Sq   | RMSE | RMSElog | $\delta < 1.25$ |
|----------------|------|------|------|---------|-----------------|
| Baseline       |      |      |      |         |                 |
| Ours (sup.)    | 0.101| 0.652| 4.264| 0.176   | 0.891           |
| Ours (unsup.)  | 0.101| 0.657| 4.262| 0.176   | 0.892           |
[53] J. Watson, O. Mac Aodha, D. Turmukhambetov, G. J. Brostow, and M. Firman, “Learning stereo from single images,” in ECCV, 2020, pp. 722–740.

[54] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” Neural computation, vol. 29, no. 9, pp. 2352–2449, 2017.

[55] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[56] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI, 2015, pp. 234–241.

[57] D. Dwibedi, I. Misra, and M. Hebert, “Cut, paste and learn: Surprisingly easy synthesis for instance detection,” in ICCV, 2017, pp. 1301–1310.

[58] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, “Simple copy-paste is a strong data augmentation method for instance segmentation,” in CVPR, 2021, pp. 2918–2928.

[59] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, “Unsupervised adversarial depth estimation using cycled generative networks,” in 3DV, 2018, pp. 587–595.

[60] H. Hirschmuller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in CVPR, vol. 2, 2005, pp. 807–814.

[61] H. Choi, H. Lee, S. Kim, S. Kim, S. Kim, K. Sohn, and D. Min, “Adaptive confidence thresholding for monocular depth estimation,” arXiv preprint arXiv:2009.12840, 2020.