

Toward Practical Monocular Indoor Depth Estimation

Cho-Ying Wu², Jialiang Wang¹, Michael Hall¹, Ulrich Neumann² and Shuochen Su¹

¹Meta Reality Labs, ²University of Southern California

{jialiawg,michaelhall,shuochsu}@fb.com, {choyingw, uneumann}@usc.edu

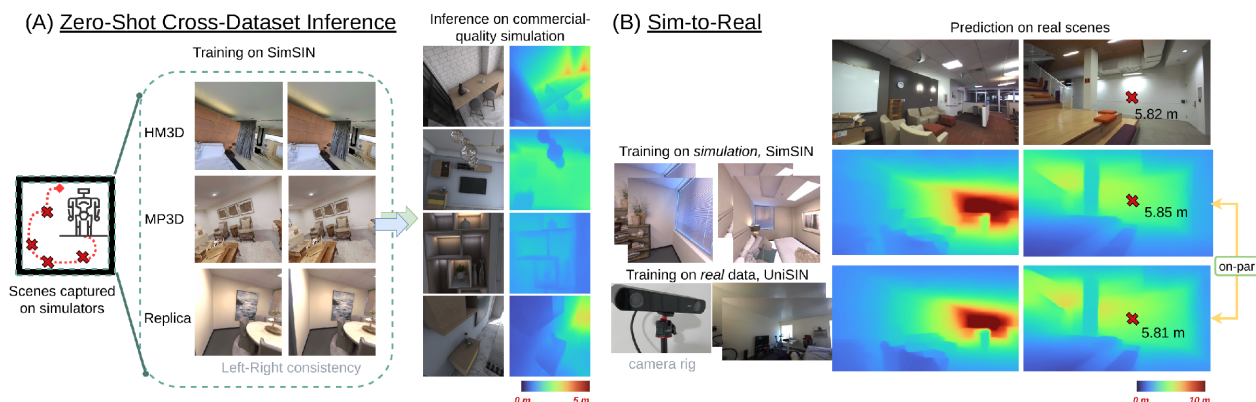


Figure 1. **Advantages of our framework.** (A) We attain zero-shot cross-dataset inference. (B) Our framework trained on simulation data produces on-par results with the one trained on real data.

Abstract

The majority of prior monocular depth estimation methods without groundtruth depth guidance focus on driving scenarios. We show that such methods generalize poorly to unseen complex indoor scenes, where objects are cluttered and arbitrarily arranged in the near field. To obtain more robustness, we propose a structure distillation approach to learn knacks from an off-the-shelf relative depth estimator that produces structured but metric-agnostic depth. By combining structure distillation with a branch that learns metrics from left-right consistency, we attain structured and metric depth for generic indoor scenes and make inferences in real-time. To facilitate learning and evaluation, we collect SimSIN, a dataset from simulation with thousands of environments, and UniSIN, a dataset that contains about 500 real scan sequences of generic indoor environments. We experiment in both sim-to-real and real-to-real settings, and show improvements, as well as in downstream applications using our depth maps. This work provides a full study, covering methods, data, and applications aspects.

1. Introduction

This work proposes a **practical** indoor depth estimation framework that has the following features: *learning from off-the-shelf estimators and left-right image pairs without their depth annotations*, *efficient training data collection*, *high generalizability to cross-dataset inference*, and *accurate and real-time depth sensing*. Our work applies to consumer-level AR/VR, such as 3D indoor scene reconstruction and virtual object insertion and interaction with environment [32]

Although self-supervised depth estimation, especially using left-right consistency, has attracted much research interest recently, popular works, such as MonoDepth [23], MonoDepth2 [24], DepthHints [74], and ManyDepth [75], mainly focus on driving scenes and are trained on large-scale driving datasets like KITTI [22] and Cityscapes [13], and it is unclear how these methods apply on indoor environments. Learning *indoor* depth via self-supervision is arguably more challenging for a number of reasons: (1) *structure priors*: depth estimation for driving scenes imposes a strong scene structure prior to the learning paradigm. The upper parts of images, commonly occupied by the sky or buildings, are typically farther away; on the other hand, the

lower parts are usually roads extending to the distance [15]. By contrast, the structure priors are much weaker for indoor environments since objects can be cluttered and arranged arbitrarily in the near field. (2) *distribution*: scene depth for driving scenarios tends to distribute more evenly across near to far ranges on roads, whereas indoor depth can be concentrated in either near or far ranges, such as zoom-in views of desks or ceilings. The uneven depth distribution makes it challenging to predict accurate metric depth for indoor scenes. (3) *camera pose*: depth-sensing devices can move in 6DoF for indoor captures, but they are typically anchored on cars for collecting driving data where translations are usually without elevation and rotations are dominated by yaw angle. Therefore, a desirable network needs to be more robust to arbitrary camera poses and complex scene structures for indoor cases. (4) *untextured surfaces*: large untextured regions, such as walls, make the commonly used photometric loss ambiguous.

In this work we propose *DistDepth*, a structure distillation approach to enhance depth accuracy trained by self-supervised learning. *DistDepth* uses an off-the-shelf relative depth estimator, DPT [59, 60] that produces *structured but only relative depth* (output values reflect depth-ordering relations but are metric-agnostic). Our structure distillation strategy encourages depth structural similarity both statistically and spatially. In this way, depth-ordering relations from DPT can be effectively blended into metric depth estimation branch trained by left-right consistency. Our learning paradigm only needs an off-the-shelf relative depth estimator and stereo image inputs without their curated depth annotations. Given a monocular image at test time, our depth estimator can predict structured and metric-accurate depth with high generalizability to unseen indoor scenes (Sec. 3.2). Distillation also helps downsize DPT’s large vision transformer to a smaller architecture, which enables real-time inference on portable devices.

We next describe our dataset-level contributions. Current publicly available stereo datasets are either targeting driving scenarios [9, 13, 20, 22, 76], small-scale and lacking scene variability [65, 66], rendered from unrealistic-scale 3D animations [6, 52], or collected in-the-wild [35, 70]. Popular indoor datasets are either small-scale (Middlebury [65]) or lacking stereo pairs (NYUv2 [54]). There is currently no large-scale indoor stereo dataset to facilitate left-right consistency for self-supervised studies. We utilize the popular Habitat simulator [63, 69] to collect stereo pairs in 3D indoor environments. Commonly-used environments are chosen, including Replica [68], Matterport3D (MP3D) [8], and Habitat-Matterport 3D (HM3D) [57], to create *SimSIN*, a novel dataset consisting of about 500K simulated stereo indoor images across about 1K indoor environments (Sec. 4). With *SimSIN*, we are able to investigate performances of prior self-supervised frameworks on indoor

scenes [24, 74, 75]. We show that we can fit on *SimSIN* by directly training those models, but such models generalize poorly to heterogeneous domain of unseen environments. Using our structure distillation strategy, however, can produce highly structured and metric-accurate depth on unseen data (Sec. 5).

Several commercial-quality simulations and real data are utilized for evaluation, including a challenging virtual apartment (VA) sequence [1, 2], pre-rendered scenes in *HyperSim* [61], and real monocular images in *NYUv2* [67]. To further investigate the gap between training on simulation v.s. training on real data, we further collect *UniSIN*, a dataset including 500 real stereo indoor sequences, amounting to 200K images, in a university across buildings and spaces using off-the-shelf high-performing stereo cameras. We show that our *DistDepth* trained on simulation data only has on-par performance with those trained on real data.

Our *DistDepth* is especially capable of 1. *attaining zero-shot cross-dataset inference*, and 2. *closing the gap between sim-to-real and real-to-real learning*, as shown in Fig. 1. Throughout the work we visualize depth maps in actual *metric* ranges unless marked as relative depth. We summarize our contributions as follows.

1. We propose *DistDepth*, a framework that distills depth-domain structure knowledge into a self-supervised depth estimator to obtain highly structured and metric-accurate depth maps.
2. We present *SimSIN*, a large-scale indoor simulation dataset that fuels the study of indoor depth estimation via left-right consistency, and a real dataset, *UniSIN*, that targets at studying the gap between training on simulation and real data.
3. We attain a practical indoor depth estimator: learning without curated depth groundtruth, efficient and effective data collection by simulation, high generalizability, and accurate and real-time inference for depth sensing.

2. Related Work

Monocular Scene Depth Estimation. Much research interest focuses on learning-based methods to learn a mapping: $\mathcal{I} \rightarrow \mathcal{D}$, from image to depth domain.

2.1. Supervised Scene Depth Estimation

Supervised learning requires pixel-level depth annotation. Early methods [17–19, 33, 45, 46, 49, 58, 79, 81] perform pixel-level regression of depth values using convolutional neural networks to minimize a loss between predictions and groundtruth. Recently, Bhat *et al.* [4] adopt adaptive bins for depth regression and use vision transformers [16]. MiDaS [60] and Wei *et al.* [82] smartly mix several datasets to attain large-scale depth training using scale-and-shift-invariant losses. BoostingDepth [53] fuses multi-scale depth cues in MiDaS based on observations from [34], but

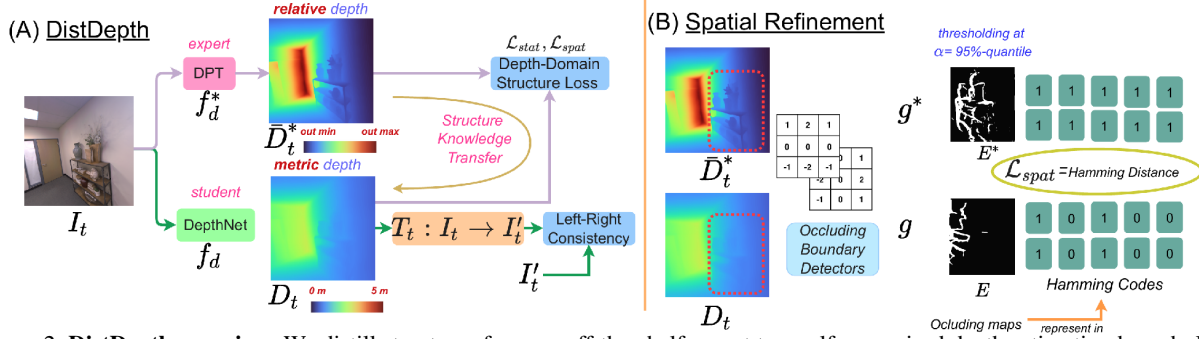


Figure 2. **DistDepth overview.** We distill structures from an off-the-shelf expert to a self-supervised depth estimation branch, DepthNet. Such an approach enables us to obtain metric depth maps with fine structures and still work without curated depth annotations. Note that we omit the temporal warping and PoseNet here for simplicity.

it takes minutes to post-process a depth map. DPT (MiDaS-V3) [59] designs a dense vision transformer and achieves better results than the original MiDaS.

Although state-of-the-art MiDaS [60] and DPT [59] can estimate fine-grained depth structures for *in-the-wild* images, they only provide relative depth, which is up to unknown scale and shift factors to align with actual size due to the mixed-dataset training strategy. Our DistDepth adopts such pretraining on in-the-wild scenes as an expert for depth structure distillation to attain both structured and metric depth from a branch trained by left-right consistency.

2.2. Left-Right and Temporal Consistency

Left-Right and temporal consistency help attain self-supervised learning to lift requirements on groundtruth depth for training, which steps closer to practical depth sensing [27, 41, 43, 44, 55]. MonoDepth [23] learns depth from stereo pairs and uses left-right depth reprojection with photometric loss minimization. MonoDepth2 [24] further includes temporal neighboring frames and also minimizes photometric consistency losses. DepthHints [74] adopts pre-computed SGM [30, 31] depth from stereo pairs as a proxy and still stay self-supervised. ManyDepth [75] uses test-time multi-frame inputs with cost-volume minimization to obtain more accurate predictions. However, these methods all focus on driving scenarios, and their applicability to indoor data is yet to be investigated. Our work is based on the left-right and temporal consistency with depth structure distillation to attain structured, metric, and generalizable depth estimation, while most works do not discuss generalizability [24–26, 75, 80]. Note that our structure distillation is different from regular distillation [29, 62, 84] since our expert is capable of only estimating depth ordering, and it needs to be combined with metrics inferred by the student to obtain the end output. Some other works attain self-supervision by only temporal consistency [5, 36], which makes scale less robust. Another work [47], based on Manhattan world assumption, minimizes co-planar and normal losses but only shows robustness to planar regions

with inherently ambiguous scale.

3. Method

3.1. Basic Problem Setup

We describe the commonly adopted left-right and temporal photometric consistency in self-supervised methods such as MonoDepth2, DepthHints, and ManyDepth in this section. During training, I_t and I'_t are stereo pairs at timestep t . DepthNet f_d is used to predict depth of I_t , $D_t = f_d(I_t)$. With known camera intrinsic K and transformation $T_t : I_t \rightarrow I'_t$ using stereo baseline, one can back-project I_t into 3D space and then re-project to the imaging plane of I'_t by utilizing K , D_t , and T_t . $\hat{I}'_t = I_t \langle \text{proj}(D_t, T_t, K) \rangle$ denotes the reprojection. The objective is to minimize photometric loss $\mathcal{L} = pe(I'_t, \hat{I}'_t)$, where pe is shown as follows.

$$pe(I'_t, \hat{I}'_t) = \kappa \frac{1 - \text{SSIM}(I'_t, \hat{I}'_t)}{2} + (1 - \kappa) L_1(I'_t, \hat{I}'_t), \quad (1)$$

where κ is commonly set to 0.85, SSIM [72] is used to measure the image-domain structure similarity, L_1 is used to compute the pixel-wise difference. $pe(I'_t, \hat{I}'_t)$ measures photometric reconstruction error of a stereo pair to attain left-right consistency.

Temporal neighboring frames are also utilized to compute photometric consistency. PoseNet calculates relative camera pose between timestep t and $t + k$: $T_{t+k \rightarrow t} = f_p(I_t, I_{t+k})$ with $k \in \{1, -1\}$. Then, temporal consistency is attained by warping an image from $t + k$ to t and calculating photometric consistency in Eq. 1. At inference time, depth is predicted from a monocular image via $D = f_d(I)$.

Applicability. We train MonoDepth2, DepthHints, and ManyDepth on the SimSIN dataset and exemplify the scene fitting later in Fig. 4. Prior arts fit the training set but do not generalize well for cross-dataset inference due to unseen complex object arrangements for indoor environments.

3.2. DistDepth: Structure Distillation from Expert

To overcome the generalizability issue when applying self-supervised frameworks to indoor environments, we

propose DistDepth (Fig. 2). DPT [59] using dense vision transformer can produce highly structured but only relative depth values by $D_t^* = f_d^*(I_t)$ ¹ as explained in Sec. 2.1. We extract the depth-domain structure of D_t^* and transfer to the self-supervised learning branches, including DepthNet f_d and PoseNet f_p . The self-supervised branch learns metric depth since it leverages stereo pairs with known camera intrinsic and baseline with depth warping operation $I_t \langle \text{proj}(D_t, T_t, K) \rangle$. Our distillation enables f_d to produce both highly structured and metric depth and still work in a fashion without groundtruth depth for training.

We first estimate rough alignment factors of scale a_s and shift a_t from DPT’s output D_t^* to predicted depth D_t by minimizing differences between $\bar{D}_t^* = a_s D_t^* + a_t$ and D_t with closed-form expressions from the least-square optimization (see the supplementary).

Statistical loss. Compared with image-domain structures, depth-domain structures exclude depth-irrelevant low-level cues such as textures and painted patterns on objects and show geometric structures. Image structure similarity can be obtained by SSIM [71–73] w.r.t. statistical constraints. Depth-domain structures also correlate to depth distribution represented by mean, variance, and co-variance for similarity measures. Thus, we compute the SSIM with depth map input \bar{D}_t^* and D_t and use the negative metric as the loss term

$$\mathcal{L}_{stat} = 1 - \text{SSIM}(\bar{D}_t^*, D_t), \quad (2)$$

Unlike the widely-used appearance loss that combines SSIM with L_1 loss, we find that pixel-wise difference measures lead to unstable training since inverting from disparity to depth magnifies prediction uncertainty and produces much larger outliers in arbitrary ranges. In contrast, the SSIM loss constrains on the mean and variance terms for two distributions instead of per-pixel differences and becomes a desirable choice.

Spatial refinement loss. SSIM loss only constrains statistical depth distribution but loses spatial information. We next propose a spatial control using depth occluding boundary maps (Fig. 2 (B)). The Sobel filter, which is a first-order gradient operator [37], is applied to compute depth-domain gradients: $g = (\frac{\partial X}{\partial u}, \frac{\partial X}{\partial v})$, where $X \in \{\bar{D}_t^*, D_t\}$ and u, v represent horizontal and vertical direction on 2D grids. Then we calculate a turn-on level $\alpha = \text{quantile}(\|g\|_2, 95\%)$ at the 95%-quantile level of gradient maps to determine the depth occluding boundaries, where gradients are larger than α . We compute the 0/1 binary-value maps, E^* and E , to represent occluding boundary locations by thresholding \bar{D}_t^* and D_t with their respective α terms. Last we calculate the

¹DPT (and MiDaS) outputs relative relation in disparity (inverse depth) space since it trains on diverse data sources (laser-based depth, depth from SfM, or stereo with unknown calibration). We inverse its outputs and compute losses in the depth space since our training data source is single

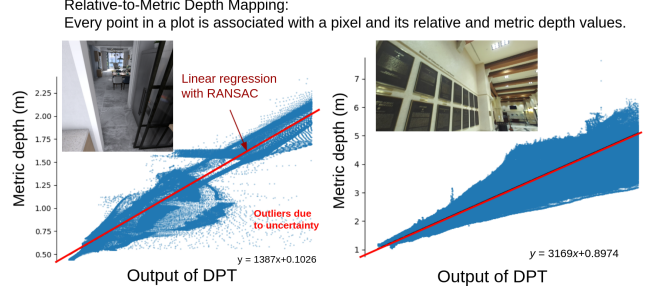


Figure 3. **Mapping from relative depth to metric depth is non-trivial.** x -axis captures (inverse) output values of DPT and y -axis represents metric depth from simulation systems or sensors. We run least-square linear regression with RANSAC to produce the optimal fitted lines between relative depth and metric depth for random scenes from our dataset.

Hamming distance, i.e. bitwise difference, of E^* and E and normalize it by the map size and use it as the spatial loss term,

$$\mathcal{L}_{spat} = E^* \oplus E / |E|, \quad (3)$$

where \oplus is the XOR operation for two boolean sets, and $|E|$ computes the size of a set. In implementation, to make thresholding and binarization operations differentiable, we subtract respective α from \bar{D}_t^* and D_t and apply soft-sign function, which resembles the sign function but back-propagates smooth and non-zero gradients, to obtain maps with values in $\{-1, 1\}$. After the division by 2, we arrive at the element-wise Hamming distance between the maps. The loss function for structure distillation is $\mathcal{L}_{dist} = \mathcal{L}_{stat} + 10^{-1} \mathcal{L}_{spat}$. The final loss function \mathcal{L}_t for I_t is combined with left-right consistency $\mathcal{L}_{LR} = pe(I_t', \hat{I}_t')$, temporal consistency $\mathcal{L}_{temp} = pe(I_t, \hat{I}_{t+k \rightarrow t})$, where $\hat{I}_{t+k \rightarrow t}$ is forward warping and backward warping, $k \in \{1, -1\}$, and \mathcal{L}_{dist} :

$$\mathcal{L}_t = \mathcal{L}_{LR} + \mathcal{L}_{temp} + 10^{-1} \mathcal{L}_{dist}. \quad (4)$$

The designed structure distillation is key to gearing up the self-supervised depth estimator with high generalizability to unseen textures such that it better separates depth-relevant and depth-irrelevant low-level cues. From another perspective, the student trained by left-right consistency helps DPT learn ranges across different indoor scenes.

Another alternative way is to predict scale and shift factors to align relative depth to metric depth based on the alignment relation [60]. This seemingly simple method, however, suffers from the disadvantage that depth estimation from neural networks inevitably includes uncertainty, which is either caused by neural network model or caused by data [38, 39, 51]. Conversion between relative depth and metric depth shows overall linear but noisy trends, and the optimal line equations can vary a lot for different scenes, as shown in Fig. 3. Thus, this alternative approach cannot factorize noise and outliers with only scale and shift terms. We show experiments using this approach in the supplementary.

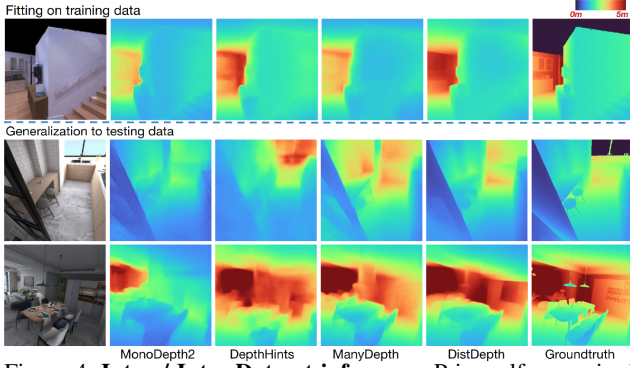


Figure 4. **Intra-/ Inter-Dataset inference.** Prior self-supervised works can fit training data (SimSIN) shown in the first row, but they generalize poorly to unseen testing dataset (VA), shown in the second and third rows. Our DistDepth can produce more structured and accurate ranges with reference to groundtruth.

We adopt ResNet [28] as DepthNet f_d . Although one can use dense vision transformers for higher prediction accuracy, they suffer from low inference speed and cannot meet real-time on-device depth sensing due to larger network size and complex operations. Therefore, we maximally exploit structure knowledge embedded in DPT and also downsize the large vision transformer to smaller-size ResNet, which enables us to run depth sensing at an interactive rate (35+ fps v.s. 8-11 fps for different version DPT, measured on a laptop with RTX 2080 GPU) to fulfill the practical depth estimator purpose. See the supplementary for demonstration.

4. Datasets

4.1. Training: SimSIN

To utilize the left-right and temporal neighboring frames to attain photometric consistency for self-supervised training, we adopt the popular Habitat simulator [63, 69] that initiates a virtual agent and renders camera-captured 3D indoor environments. We adopt Replica, MP3D, and HM3D as the backend 3D models following prior embodied AI works [10–12, 14, 21, 56].

We adopt a stereo baseline of 13cm following the camera setting in [7] and render at 512×512 resolution. The agent navigates multiple times and captures stereo sequences. We then manually filter out failure sequences, such as when the agent gets too close to walls or navigates to null spaces. Our dataset consists about 80K, 205K, and 215K images from Replica, MP3D, and HM3D respectively, amounting to 500K stereo images from ~ 1000 various environments in our proposed SimSIN dataset, which is by far the largest stereo dataset for generic indoor environments.

4.2. Training: UniSIN

To investigate the gap between simulation and reality and compare performances of models trained on simulation and trained on real data. We use ZED-2i [3], a high-

performing stereo camera system, to collect large-scale stereo sequences from various interior spaces around a university and create the UniSIN dataset. Its training split contains 500 sequences, and each sequence has 200 stereo pairs, amounting to 200K training images.

4.3. Evaluation Sets

Commercial-Quality simulation. We select a delicately designed virtual apartment (VA) and render about 3.5K photorealistic images along a trajectory as the evaluation set [1, 2]. The VA dataset contains challenging indoor scenes for depth sensing, such as cabinet cubes with different lighting, thin structures, and complex decorators. These scenes enable us to conduct a detailed study of depth sensing in private indoor spaces, the most common use cases for AR/VR. We further include samples from pre-rendered *Hypersim* [61] dataset, which contains monocular images of virtual environments, for qualitative demonstration.

Real Data. We adopt popular NYUv2 [67] whose test set contains 654 monocular images with depth maps from time-of-flight laser using Kinect v1. To compensate for Kinect’s older imaging system and low resolution to serve more practical AR/VR use, we collect 1K high-definition images with finely optimized depth delivered by ZED-2i for numerical evaluation.

We show a sample collection for all the datasets in the supplementary.

5. Experiments and Analysis

We set input size to 256×256 , batch size to 24, and epoch number to 10. Adam [40] is used as the optimizer with an initial learning rate of 2×10^{-4} that drops by a factor of 10 at epoch 8 and 10. We adopt common data augmentation of color jittering and random flipping. We use ResNet50 for our PoseNet f_p and ResNet152 for DepthNet f_d , and the same for MonoDepth2, DepthHints, and ManyDepth for comparison in this section. DPT adopts large-size dense transformer network to apply on in-the-wild scenes. Thus we choose larger architecture for DepthNet to demonstrate generalizability but can still run in interactive frame rate (illustrated in the supplementary).

5.1. Experiments on Simulation Data

We use SimSIN as the training dataset in Sec. 5.1 and evaluate on various commercial-quality simulation data.

Prior self-supervised methods trained on SimSIN. We first directly train MonoDepth2, DepthHints, and ManyDepth on SimSIN following settings in their papers and show fitting on the training data and inference on VA in Fig. 4 to investigate the generalizability. ManyDepth and DepthHints attain better results than MonoDepth2. Our DistDepth produces highly regularized structures with robustness to unseen examples, w.r.t groundtruth. The range

Table 1. **Quantitative comparison on the VA dataset.** Our DistDepth attains much lower errors than prior works of left-right consistency. DistDepth-M further uses the test-time multi-frame strategy in ManyDepth. See the main text.

Method	Test-Time Single-Frame				Test-Time Multi-Frame		
	MonoDepth2 [24]	DepthHints [74]	DistDepth	Improvement	ManyDepth [75]	DistDepth-M	Improvement
MAE	0.295	0.291	0.253	-14.2%	0.275	0.239	-13.1%
AbsRel	0.203	0.197	0.175	-13.8%	0.189	0.166	-12.2%
RMSE	0.432	0.427	0.374	-13.4%	0.408	0.357	-12.5%
RMSElog	0.251	0.248	0.213	-15.1%	0.241	0.210	-12.9%

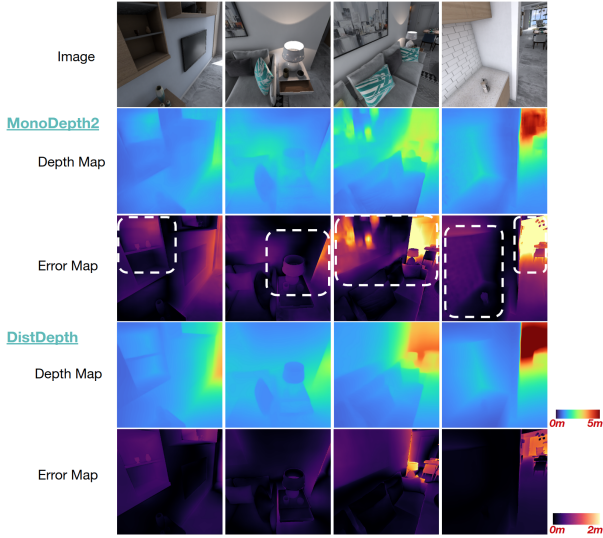


Figure 5. **Qualitative results on VA sequence.** Depth and error maps are shown for DistDepth and MonoDepth2 for comparison. These examples demonstrate that our DistDepth predicts geometrically structured depth for common indoor objects.

prediction also improves, which we believe is due to better structure occluding boundary reasoning.

Error analysis on VA. The VA dataset includes various challenging scenes in indoor spaces. We show qualitative error analysis in Fig. 5. Highlighted in error maps, our DistDepth has better generalizability on estimating the underlying geometric structures such as paintings, shelves, and walls under various lighting conditions. See supplementary for more examples.

We further show numerical comparison on the entire VA sequence in Table 1. All the methods in comparison are trained on SimSIN. We further equip DistDepth with the test-time multi-frame strategy with cost-volume minimization introduced in ManyDepth and denote this variant by DistDepth-M. Methods are categorized into test-time single-frame and test-time multi-frame. In both cases, DistDepth attains lower errors than prior arts. This validates our network design: with an expert used for depth-domain structure distillation, a student network f_d can produce both structured and metric depth that is closer to the groundtruth.

Ablation study on VA. We first study the expert network and adopt different versions of DPT (hybrid and legacy), whose network sizes are different. Table 2 shows that the student network taught by the larger-size expert, DPT-

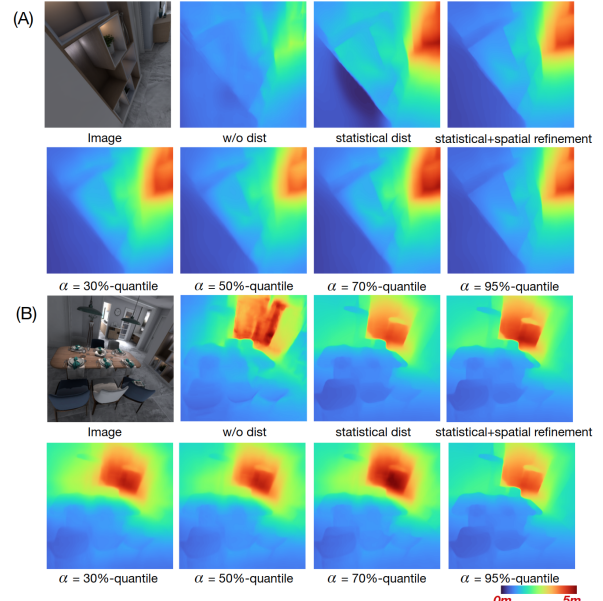


Figure 6. **Qualitative study for depth-domain structure improvement.** Two examples (A) and (B) are shown to study the effects of distillation (dist) losses and turn-on level α in spatial refinement to validate our design in Sec. 3.2.

Table 2. **Study on the choice of the expert network for distillation.** Different versions of DPT [59] that vary in network sizes (# of params) are adopted as the expert to teach the student. DPT-legacy localizes occluding contours better and leads to a better-performing student network. The results of supervised learning are provided as a reference.

Expert	Self-Supervised			Supervised
	w/o distillation	DPT - hybrid	DPT - legacy	with groundtruth
# of params	-	123M	344M	-
MAE	0.295	0.276	0.253	0.221
AbsRel	0.203	0.188	0.175	0.158
RMSE	0.432	0.394	0.374	0.325
RMSElog	0.251	0.227	0.213	0.188

legacy, achieves lower depth estimation errors. Without distillation, results are worse because its estimation relies only on the photometric loss, which fails on untextured areas like walls. As a sanity check, we also provide results of supervised training using SimSIN’s groundtruth depth with pixel-wise MSE loss and test on the VA dataset, which shows the gap between training on curated depth and depth from expert network’s predictions.

We next study the training strategy with different distillation losses and effects of turn-on level α in Sec. 3.2. We

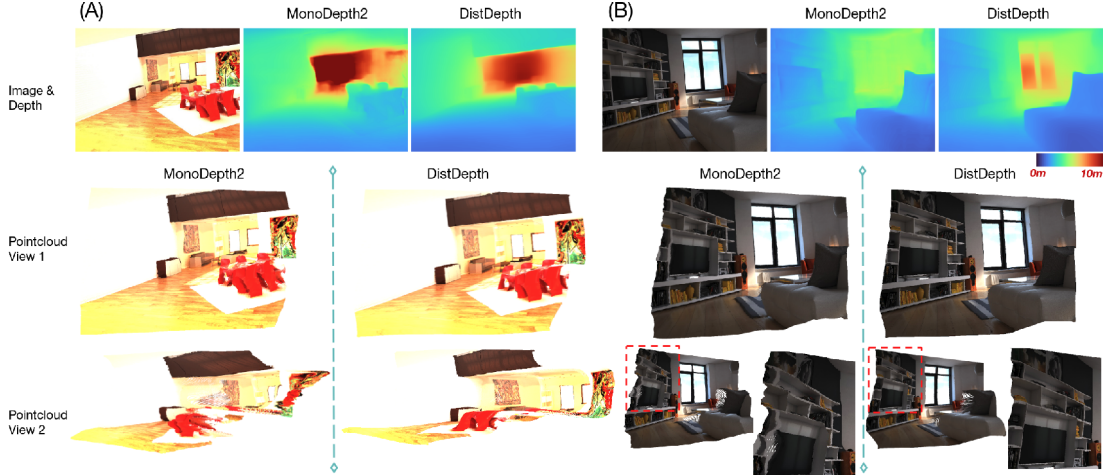


Figure 7. **Results on Hypersim.** Depth map and textured pointcloud comparison of MonoDepth2 and our DistDepth. With structure distillation, DistDepth attains better object structure predictions, such as tables and paintings on the wall shown in (A) and much less distortion for the large bookcase in (B).

compare (1) w/o distillation, (2) distillation with statistical loss only, and (3) distillation with statistical and spatial refinement loss. We demonstrate qualitative results in Fig. 6 to show the depth-domain structure improvements. Without distillation, spatial structures cannot be reasoned crisply. With statistical refinement, depth structures are more distinct. Adding spatial refinement, the depth-domain structures show fine-grained details. We further analyze the effects of different turn-on levels of α . Low α makes structures blurry since the refinement does not focus on the high-gradient occluding boundaries as high α does, which identifies only high-gradient areas as occluding boundaries and benefits structure knowledge transfer.

Comparison on Hypersim. We next exhibit depth and textured pointcloud in Fig. 7 for some scenes in Hypersim. Two different views are adopted for pointcloud visualization. One can find that our DistDepth predicts better geometric shapes in both depth map and pointcloud. See the supplementary for more examples.

5.2. Experiments on Real Data

Closing sim-to-real gap. We compare results of training on simulation (SimSIN)² and real data (UniSIN) to investigate the performance gap. We examine (1) training MonoDepth2 on simulation and evaluate on real data, (2) training MonoDepth2 on real data and evaluate on real data, (3) training DistDepth on simulation and evaluate on real data, and (4) training DistDepth on real and evaluate on real data. Fig. 8 illustrates the results of the four settings. Comparing (1) and (2), one can find that MonoDepth2 trained on real data produces more reliable results than on simulation. By contrast, this gap becomes unobvious when comparing (3)

and (4) using DistDepth. Results of (3) are on-par with (4) and sometimes even produce better geometric shapes like highlighted areas. We further include numerical analysis in the supplementary.

The results validate our proposals on both method and dataset levels. First, DistDepth utilizes an expert network to distill knacks to the student. The distillation substantially adds robustness to models trained on simulation data and makes the results comparable to models trained on real data. This shows the ability of DistDepth for *closing the gap between simulation and real data*. Second, stereo simulation data provide a platform for left-right consistency to learn metric depth from stereo triangulation. We show a collection of results in Fig. 9 using DistDepth that is trained purely on simulation.

Evaluation on NYUv2. Table 3 shows evaluations on NYUv2. We first train our DistDepth on SimSIN and fine-tune on NYUv2 with only temporal consistency. Note that one finetuned model (Sup:△) is categorized as semi-supervised since it utilizes an expert that has been trained with NYUv2’s curated depth. The finetuned models produce the best results among methods without NYUv2’s depth supervision and even attain comparable results to many supervised methods. We next train DistDepth only on simulation (SimSIN) or real data (UniSIN) and evaluate on NYUv2. Performances of the model trained on SimSIN only drop a little compared with that trained on UniSIN, which justifies our sim-to-real advantage again. Without involving any training data in NYUv2, DistDepth still achieves comparable performances to many supervised and self-supervised methods, which further validates our *zero-shot cross-dataset advantage*. We exhibit real-time depth sensing, 3D photos, and depth-aware AR applications in supplementary.

²To balance the training dataset size, we randomly subsample about 200K images here in SimSIN to match UniSIN’s dataset size.

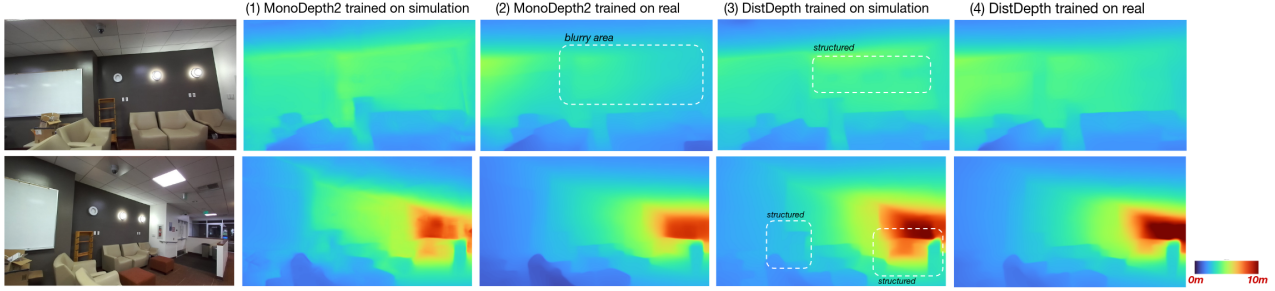


Figure 8. **Comparison on UniSIN.** Geometric shapes produced from DistDepth are better than MonoDepth2. DistDepth concretely reduces the gap for sim-to-real: (3) and (4) attain on-par results and sometimes training on simulation shows better structure than training on real.

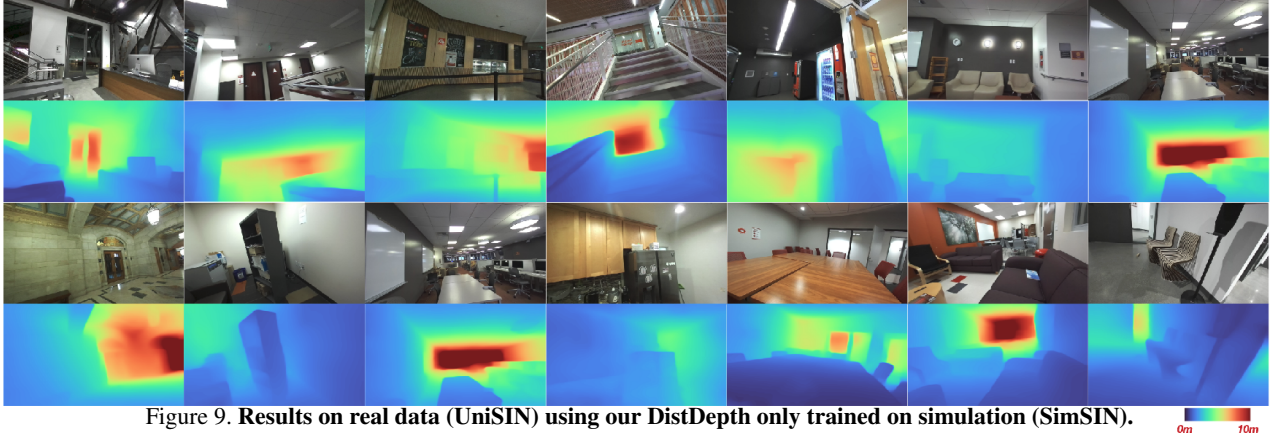


Figure 9. **Results on real data (UniSIN) using our DistDepth only trained on simulation (SimSIN).**

Table 3. **Evaluation on NYUv2.** Sup: ✓ - supervised learning using groundtruth depth, ✗ - not using groundtruth depth, and △ - semi-supervised learning (we use the expert finetuned on NYUv2, where we have indirect access to the groundtruth). We achieve the best results among all self-supervised methods, and our semi-supervised and self-supervised finetuned on NYUv2 even outperform many supervised methods. The last two rows show results without groundtruth supervision and without training on NYUv2. In this challenging *zero-shot cross-dataset evaluation*, we still achieve comparable performances to many methods trained on NYUv2. Error and accuracy (yellow/green) metrics are reported.

Methods	Sup	Train on NYUv2	AbsRel	RMSE	δ_1	δ_2	δ_3
Make3D [64]	✓	✓	0.349	1.214	44.7	74.5	89.7
Li <i>et al.</i> [48]	✓	✓	0.143	0.635	78.8	95.8	99.1
Eigen <i>et al.</i> [17]	✓	✓	0.158	0.641	76.9	95.0	98.8
Laina <i>et al.</i> [45]	✓	✓	0.127	0.573	81.1	95.3	98.8
DORN [19]	✓	✓	0.115	0.509	82.8	86.5	99.2
AdaBins [4]	✓	✓	0.103	0.364	90.3	98.4	99.7
DPT [59]	✓	✓	0.110	0.357	90.4	98.8	99.8
Zhou <i>et al.</i> [87]	✗	✓	0.208	0.712	67.4	90.0	96.8
Zhao <i>et al.</i> [85]	✗	✓	0.189	0.686	70.1	91.2	97.8
Bian <i>et al.</i> [5]	✗	✓	0.157	0.593	78.0	94.0	98.4
P ² Net+PP [83]	✗	✓	0.147	0.553	80.4	95.2	98.7
StructDepth [47]	✗	✓	0.142	0.540	81.3	95.4	98.8
MonoIndoor [36]	✗	✓	0.134	0.526	82.3	95.8	98.9
DistDepth (finetuned)	✗	✓	0.130	0.517	83.2	96.3	99.0
DistDepth (finetuned)	△	✓	0.113	0.444	87.3	97.4	99.3
DistDepth (SimSIN)	✗	✗	0.164	0.566	77.9	93.5	98.0
DistDepth (UniSIN)	✗	✗	0.158	0.548	79.1	94.2	98.5

6. Conclusion and Discussion

This work targets at a practical indoor depth estimation framework with following features: training without depth

groundtruth, effective training on simulation, high generalizability, and accurate and real-time inference. We first identify the challenges of indoor depth estimation and study the applicability of existing self-supervised methods with left-right consistency on SimSIN. Geared up with the depth-domain structure knowledge distilled from an expert, we see substantial improvement in both inferring finer structures and more accurate metric depth. We show zero-shot cross-dataset inference that proves its generalizability to work on heterogeneous data domains and attain a broadly applicable depth estimator for indoor scenes. Even more, depth learned from simulation data transfers well to real scenes, which shows the success of our distillation strategy. At inference time, it only takes a single feed-forward pass to DepthNet to produce structured metric depth and reach 35+ fps on a portable device which serves real-time needs.

Limitations. Although DistDepth is capable of producing structured and metric depth using a single forward pass of depth estimation, it operates on a per-frame basis, which can be refined to produce more temporally consistent depth for video inputs [42, 50]. Another issue commonly for depth estimation is the proper handling of reflective objects. With distillation, DistDepth can produce estimation for objects with clear contours, as illustrated in Fig. 1 of bulbs. However, our approach is still not yet robust to large mirrors. A possible solution is to locate mirrors and perform depth completion [77, 78, 86, 88] on raw estimates.

References

- [1] Unreal engine 4. <https://www.unrealengine.com/en-US/unreal>. 2, 5
- [2] Warm harbor environment. <https://www.unrealengine.com/marketplace/en-US/product/warmharbor>. 2, 5
- [3] Zed2i-stereolabs. <https://www.stereolabs.com/zed-2i/>. 5
- [4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021. 2, 8
- [5] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *IJCV*, 2021. 3, 8
- [6] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2
- [7] Rohan Chabra, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs. Stereodnet: Dilated residual stereonet. In *CVPR*, 2019. 5
- [8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 2
- [9] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 2
- [10] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *CVPR*, 2021. 5
- [11] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicens Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3D environments. In *ECCV*, 2020. 5
- [12] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *ICLR*, 2020. 5
- [13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2
- [14] Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. In *NeurIPS*, 2020. 5
- [15] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019. 2
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2
- [17] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2, 8
- [18] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014. 2
- [19] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 2, 8
- [20] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 2
- [21] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. Visualechoes: Spatial image representation learning through echolocation. In *ECCV*, 2020. 5
- [22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 2
- [23] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 3
- [24] Clement Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 1, 2, 3, 6
- [25] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *CVPR*, 2019. 3
- [26] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Chengzhou Tang, Siyu Zhu, and Ping Tan. Dro: Deep recurrent optimizer for structure-from-motion. *arXiv preprint arXiv:2103.13201*, 2021. 3
- [27] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 3
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5
- [29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3
- [30] Heiko Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *CVPR*, 2006. 3
- [31] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *TPAMI*, 2007. 3
- [32] Aleksander Holynski and Johannes Kopf. Fast depth densification for occlusion-aware augmented reality. In *SIGGRAPH Asia*, 2018. 1
- [33] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019. 2
- [34] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *CVPR*, 2019. 2

- [35] Yiwen Hua, Puneet Kohli, Pritish Uplavikar, Anand Ravi, Saravana Gunaseelan, Jason Orozco, and Edward Li. Holopix50k: A large-scale in-the-wild stereo image dataset. In *CVPRW*, 2020. 2
- [36] Pan Ji, Runze Li, Bir Bhanu, and Yi Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *ICCV*, 2021. 3, 8
- [37] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 1988. 4
- [38] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 4
- [39] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 4
- [40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [41] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, 2020. 3
- [42] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, 2021. 8
- [43] Varun Ravi Kumar, Marvin Klingner, Senthil Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mader. Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *WACV*, 2021. 3
- [44] Varun Ravi Kumar, Senthil Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick Mäder. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. In *IROS*, 2020. 3
- [45] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 2, 8
- [46] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2
- [47] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *ICCV*, 2021. 3, 8
- [48] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *CVPR*, 2017. 8
- [49] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 2
- [50] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (TOG)*, 2020. 8
- [51] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *NeurIPS*, 2019. 4
- [52] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 2
- [53] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *CVPR*, 2021. 2
- [54] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2
- [55] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020. 3
- [56] Senthil Purushwalkam, Sebastian Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. In *ICCV*, 2021. 5
- [57] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. *NeurIPS Datasets and Benchmarks Track*, 2021. 2
- [58] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *ICCVW*, 2019. 2
- [59] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 2, 3, 4, 6, 8
- [60] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 2, 3, 4
- [61] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *ICCV*, 2021. 2, 5
- [62] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3
- [63] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *CVPR*, 2019. 2, 5
- [64] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3D: Learning 3D scene structure from a single still image. *TPAMI*. 8
- [65] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*. Springer, 2014. 2

- [66] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad SLAM: Bundle adjusted direct RGB-D SLAM. In *CVPR*, 2019. 2
- [67] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 2, 5
- [68] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2
- [69] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *NeurIPS*, 2021. 2, 5
- [70] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *3DV*, 2019. 2
- [71] Zhou Wang and Alan C Bovik. A universal image quality index. *Signal Processing Letters*, 2002. 4
- [72] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 3, 4
- [73] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, 2003. 4
- [74] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, 2019. 1, 2, 3, 6
- [75] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *CVPR*, 2021. 1, 2, 3, 6
- [76] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2.0: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks Track*, 2021. 2
- [77] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *CVPR*, 2021. 8
- [78] Cho-Ying Wu and Ulrich Neumann. Scene completeness-aware lidar depth completion for driving scenario. In *ICASSP*, 2021. 8
- [79] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018. 2
- [80] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 3
- [81] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normals for depth prediction. In *ICCV*, 2019. 2
- [82] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *CVPR*, 2021. 2
- [83] Zehao Yu, Lei Jin, and Shenghua Gao. P²net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *ECCV*, 2020. 8
- [84] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. 2017. 3
- [85] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020. 8
- [86] Yiqi Zhong, Cho-Ying Wu, Suyu You, and Ulrich Neumann. Deep RGB-D canonical correlation analysis for sparse depth completion. *NeurIPS*, 2019. 8
- [87] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *CVPR*, 2019. 8
- [88] Luyang Zhu, Arsalan Mousavian, Yu Xiang, Hammad Mazhar, Jozef van Eenbergen, Shoubhik Debnath, and Dieter Fox. RGB-D local implicit function for depth completion of transparent objects. In *CVPR*, 2021. 8