

MonoIndoor: Towards Good Practice of Self-Supervised Monocular Depth Estimation for Indoor Environments

Pan Ji^{*1}, Runze Li^{*1,2}, Bir Bhanu², Yi Xu¹

¹OPPO US Research Center, InnoPeak Technology, Inc. ²University of California Riverside

Abstract

Self-supervised depth estimation for indoor environments is more challenging than its outdoor counterpart in at least the following two aspects: (i) the depth range of indoor sequences varies a lot across different frames, making it difficult for the depth network to induce consistent depth cues, whereas the maximum distance in outdoor scenes mostly stays the same as the camera usually sees the sky; (ii) the indoor sequences contain much more rotational motions, which cause difficulties for the pose network, while the motions of outdoor sequences are pre-dominantly translational, especially for driving datasets such as KITTI. In this paper, special considerations are given to those challenges and a set of good practices are consolidated for improving the performance of self-supervised monocular depth estimation in indoor environments. The proposed method mainly consists of two novel modules, i.e., a depth factorization module and a residual pose estimation module, each of which is designed to respectively tackle the aforementioned challenges. The effectiveness of each module is shown through a carefully conducted ablation study and the demonstration of the state-of-the-art performance on three indoor datasets, i.e., EuRoC, NYUv2 and 7-Scenes.

1. Introduction

Depth estimation plays an essential role in a variety of 3D perceptual tasks, such as autonomous driving, virtual reality (VR), and augmented reality (AR). In this paper, we tackle the problem of estimating the depth map from a single image in a self-supervised manner. Compared to the supervised methods [5, 8], self-supervision [9, 46, 12] frees us from having to capture the ground-truth depth using depth sensors (e.g., LiDAR) and therefore, it is more attractive in scenarios where obtaining the ground-truth is not possible.

Recently, self-supervised methods [12] have achieved significant success, producing depth prediction that is comparable to that produced by the supervised methods [14, 8]. For example, on the KITTI dataset [10], Monodepth2 [12] achieves an absolute relative depth error (AbsRel) of 10.6%, which is not far from the AbsRel of 7.2% by supervised DORN [8]. However, most of these self-supervised depth prediction methods [9, 46, 12] are only evaluated on outdoor datasets such as KITTI, leaving their performance opaque for indoor environments. A few methods [45, 44] have considered indoor self-supervised depth prediction, but their performance still trail far behind the one on the outdoor datasets by methods such as [9, 46, 12] or the supervised counterparts [8, 41] on indoor datasets. For instance, on the indoor NYUv2 dataset [33], the method by Zhao *et al.* [44] reaches an AbsRel of 18.9%, which is much higher than what Monodepth2 can achieve on KITTI.

In view of the performance discrepancies between the indoor and outdoor scenes, we examine what makes indoor depth prediction more challenging than the outdoor case. Our first conjecture is that this is partly due to the fact that the scene depth range of indoor sequences varies a lot more than in the outdoor. This results in more difficulties for the depth network in inducing consistent depth cues across images. Our second observation is that the pose network, which is commonly used in self-supervised methods [46, 12], tends to have large errors in rotation prediction. A similar finding in [47] shows that predicted poses have much higher rotational errors (e.g., 10 times larger) than geometric SLAM [26] even after using a recurrent pose network. This problem is not prominent on KITTI because the motions therein are mostly translational. However, since indoor datasets are often captured by hand-held cameras [33] or MAVs [31] which inevitably undergo frequent rotations, the inaccurate rotation prediction becomes detrimental to the self-supervised training of a depth model for indoor environments.

Given the above considerations, we propose **MonoIndoor**, a monocular self-supervised depth estimation method tailored for indoor environments. Our MonoIndoor consists of two novel modules: a *depth factorization* module and a

^{*}Joint first authorship. P. Ji is the corresponding author (perji530@gmail.com). R. Li's contribution was made during an internship with OPPO US Research Center.

residual pose estimation module. In the depth factorization module, we factorize the depth map into a global depth scale (for the current image) and a relative depth map. The depth scale factor is separately predicted by an extra branch in the depth network. In such a way, the depth network has more model plasticity to adapt to the depth scale changes during training. In the residual pose estimation module, we mitigate the issue of inaccurate rotation prediction by performing residual pose estimation in addition to an initial large pose prediction. Such a residual approach leads to more accurate computation of the photometric loss [12], which in turn leads to better model training for the depth network.

In summary, our contributions are:

- A novel depth factorization module that helps the depth network adapt to the rapid scale changes;
- A novel residual pose estimation module that mitigates the inaccurate rotation prediction issue in the pose network and in turn improves depth prediction;
- State-of-the-art performance of self-supervised depth prediction on three publicly available indoor datasets, *i.e.*, EuRoC [31], NYUv2 [33], and 7-Scenes [32].

2. Related Work

In this section, we review both supervised and self-supervised methods for monocular depth estimation.

2.1. Supervised Monocular Depth Estimation

Early depth estimation methods are mostly supervised. Saxena *et al.* [30] regress the depth from a single image with superpixel features and a Markov Random Field (MRF). Eigen *et al.* [6] propose the first deep-learning based method for monocular depth estimation using a multi-scale convolutional neural network (CNN). Later methods improve the performance of depth prediction either by better network architecture [19] or via more sophisticated training losses [21, 8, 41]. A few methods [36, 34] rely on two networks, one for depth prediction and the other for motion, to mimic geometric Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) in a supervised framework. Training these methods needs ground-truth depth data, which is often expensive to capture. Some other methods then resort to generating pseudo ground-truth depth labels with traditional 3D reconstruction methods [23, 22], such as SfM [31] and SLAM [26], or 3D movies [28]. Such methods have better capacity of generalization across different datasets, but can not necessarily achieve the best performance for the dataset at hand.

2.2. Self-Supervised Monocular Depth Estimation

Self-supervised depth estimation has attracted a lot of attention recently as it does not require training with the ground truth. Along this line, Garg *et al.* [9] propose the

first self-supervised method to use color consistency loss between stereo images to train a monocular depth model. Zhou *et al.* [46] employ two networks (*i.e.*, one depth network and one pose network) to construct the photometric loss across temporal frames. Many follow-up methods then try to improve the self-supervision by new loss terms. Godard *et al.* [11] incorporate a left-right depth consistency loss for the stereo training. Bian *et al.* [1] put forth a temporal depth consistency loss to encourage neighboring frames to have consistent depth predictions. Wang *et al.* [37] observe the diminishing issue of the depth model during training and come up with a simple normalization method to counter this effect. Yin *et al.* [42] and Zou *et al.* [48] use three networks (*i.e.*, one depth network, one pose network, and one extra flow network) to enforce cross-task consistency between optical flow and dense depth. Wang *et al.* [39] and Zou *et al.* [47] leverage recurrent neural networks, such as LSTMs, to model long-term dependency in the pose network and/or the depth network. Tiwari *et al.* [35] form a self-improving loop with monocular SLAM and a self-supervised depth model [12] to improve the performance of each one. Notably, Monodepth2 [12] significantly improves the performance over previous methods via a set of techniques: a per-pixel minimum photometric loss to handle occlusions, an auto-masking method to mask out static pixels, and a multi-scale depth estimation strategy to mitigate the texture-copying issue in depth. Due to its good performance, we implement our self-supervised depth estimation framework based on Monodepth2, but make important changes to both the depth and the pose networks.

Most of the aforementioned methods are only evaluated on outdoor datasets such as KITTI. A few other recent methods [45, 44, 2, 43] focus on indoor self-supervised depth estimation. Zhou *et al.* [45] propose an optical-flow based training paradigm and handle large rotational motions by a pre-processing step that removes all the image pairs with “pure rotation”. Yu *et al.* [43] utilize patch-match and plane-regularization to improve self-supervised structure-from-motion. Zhao *et al.* [44] adopt a geometry-augmented strategy that solves for the depth via two-view triangulation and then uses the triangulated depth as supervision. Bian *et al.* [2] argue that “the rotation behaves as noise during training” and thus propose a rectification step to remove the rotation between consecutive frames. We have an observation similar to [45] and [2] that large rotations cause difficulties for the network. However, we take a different strategy. Instead of removing rotations from training data, we progressively estimate them via a novel residual pose module. This in turn improves depth prediction.

3. Method

In this section, we give detailed descriptions of performing self-supervised depth estimation using **MonoIndoor**.

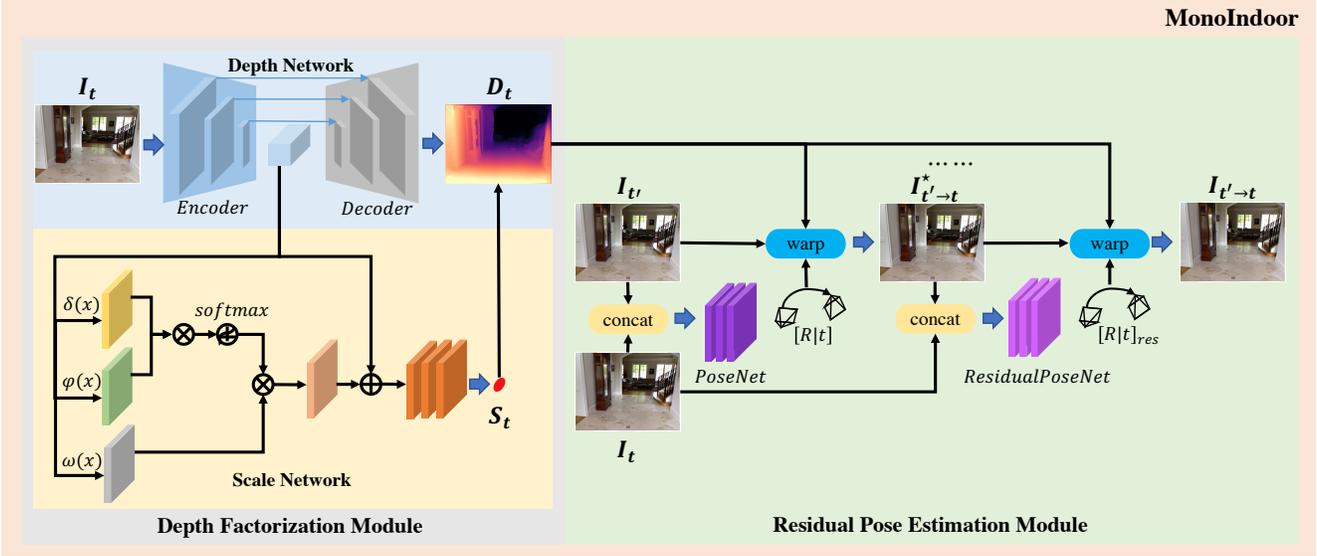


Figure 1. Overview of the proposed **MonoIndoor**. **Depth Factorization Module**: We use an encoder-decoder based depth network to predict a relative depth map and a non-local scale network to estimate a global scale factor. **Residual Pose Estimation Module**: We use a pose network to predict an initial camera pose of a pair of frames and residual pose network to iteratively predict residual camera poses based on the predicted initial pose.

Specifically, we first introduce the background of the self-supervised depth estimation. Then, we describe the good practices in predicting depth with our MonoIndoor.

3.1. Self-Supervised Depth Estimation

Similar to [46, 12, 47], we also consider the self-supervised depth estimation as a novel view-synthesis problem by training a model to predict the target image from different viewpoints of source images. The image synthesis process is trained and constrained by using the depth map as the bridging variable. Such a system requires both the predicted depth map of the target image and the estimated relative pose between a pair of target and source images. Specifically, given a target image I_t and a source image $I_{t'}$ from another view, the system is jointly trained to predict a dense depth map D_t of the target image and a relative camera pose $T_{t \rightarrow t'}$ from the target to the source. The photometric reprojection loss can then be constructed as follows:

$$\mathcal{L}_A = \sum_{t'} \rho(I_t, I_{t' \rightarrow t}), \quad (1)$$

and

$$I_{t' \rightarrow t} = I_{t'} \langle \text{proj}(D_t, T_{t \rightarrow t'}, K) \rangle, \quad (2)$$

where ρ denotes the photometric reconstruction error [46, 12]. It is a weighted combination of the L1 and Structured Similarity (SSIM) loss defined as

$$\rho(I_t, I_{t' \rightarrow t}) = \frac{\alpha}{2} (1 - \text{SSIM}(I_t, I_{t' \rightarrow t})) + (1 - \alpha) \|I_t, I_{t' \rightarrow t}\|_1. \quad (3)$$

$I_{t' \rightarrow t}$ is the source image warped to the target coordinate frame based on the depth of the target image. $\text{proj}(\cdot)$ is the transformation function to map image coordinated p_t from the target image to its $p_{t'}$ on the source image following

$$p_{t'} \sim K T_{t \rightarrow t'} D_t(p_t) K^{-1} p_t, \quad (4)$$

and $\langle \cdot \rangle$ is the bilinear sampling operator which is locally sub-differentiable. Following [12], the camera intrinsics K of all images are assumed to be the same, and an edge-aware smoothness term is employed as

$$\mathcal{L}_s = |\partial_x d_t^*| e^{-|\partial_x I_t|} + |\partial_y d_t^*| e^{-|\partial_y I_t|}, \quad (5)$$

where $d_t^* = d/\bar{d}_t$ is the mean-normalized inverse depth from [37]. During training, we adopt the auto-masking scheme [12] to handle static pixels.

Similar to [1], we use an additional depth consistency loss to enforce consistent depth prediction across neighboring frames. We first warp the depth image $D_{t'}$ of the source image by Equation (2) to generate $D_{t' \rightarrow t}$, which is a corresponding depth map in the coordinate system of the source image. We then transform $D_{t' \rightarrow t}$ to the coordinate system of the target image via Equation (4) to produce a synthesized target depth map $\tilde{D}_{t' \rightarrow t}$. The depth consistency loss can be written as

$$\mathcal{L}_c = \frac{|D_t - \tilde{D}_{t' \rightarrow t}|}{D_t + \tilde{D}_{t' \rightarrow t}}. \quad (6)$$

The overall objective to train the model is

$$\mathcal{L} = \mathcal{L}_A + \tau \mathcal{L}_s + \gamma \mathcal{L}_c, \quad (7)$$

where τ and γ are the weights for the edge-aware smoothness loss and the depth consistency loss respectively.

Even though existing monocular self-supervised methods are able to produce competitive depth maps in outdoor environments, these methods still suffer from worse performance in indoor environments, especially compared with fully-supervised methods. As discussed in Section 1, the main challenges in indoor environments come from the fact that the depth range changes a lot and indoor sequences contain regular rotational motions which are difficult to predict. To handle these issues, we propose **MonoIndoor**, a monocular self-supervised depth estimation framework, as shown in Figure 1, to enable improved predicted depth quality in indoor environments.

The system takes as input a single color image and outputs a depth map via our MonoInoor which consists of two core parts: a depth factorization module and a residual pose estimation module. We present our main contributions in the following sections.

3.2. Depth Factorization

We use the Monodepth2 [12] as the backbone model for depth prediction. The depth model in Monodepth2 employs an auto-encoder structure with skip connections between the encoder and the decoder. The depth encoder takes as input a color image I , and the decoder outputs its depth map. Note that the final depth prediction is not directly from the convolutional layers, but after a sigmoid activation function and a linear scaling function as follows,

$$d = 1/(a\sigma + b), \quad (8)$$

where σ is the value after the sigmoid function, a and b are specified to constrain the depth map D within a certain depth range. Practically, a and b are respectively pre-defined as a minimum depth value and a maximum depth value which can be obtained in a known environment. For instance, on the KITTI dataset [10], a is chosen as 0.1 and b as 100. The reason for setting a and b as fixed values is that the depth range is consistent across the video sequences when the camera always sees the sky at the far point. However, this setting is not valid for most indoor environments. As scene varies, the depth range varies a lot. For example, the depth range in a bathroom (e.g. 0.1m~3m) can be very different from the one in a lobby (e.g. 0.1m~10m). Pre-setting depth range will act as an inaccurate guidance that is harmful for the model to capture accurate depth scales. This is especially true when there are rapid scale changes, which are commonly observed in indoor scenes. To overcome this problem, we propose a depth factorization module (see Figure 1) to learn a disentangled representation in the form of a relative depth map and a global scale factor. We employ the depth network of Monodepth2 [12] to predict relative depth and propose a self-attention-guided scale

regression network to predict the global scale factor for the current view.

Scale Network. We design the scale network as a new branch which takes as input a color image and outputs its global scale factor. Since the global scale factor is closely informed by certain areas (e.g., the far point) in the images, we explore to use a self-attention block [40] so that the network can be guided to pay more attention to a certain area which is informative to induce the depth scale factor of the current view in a scene. Given the feature representations \mathcal{F} learnt from the input image, we utilize a self-attention block to take \mathcal{F} as input, forming the query, the key and the value output by

$$\begin{aligned} \psi(\mathcal{F}) &= \mathbf{W}_\psi \mathcal{F}, \\ \phi(\mathcal{F}) &= \mathbf{W}_\phi \mathcal{F}, \\ h(\mathcal{F}) &= \mathbf{W}_h \mathcal{F}, \end{aligned} \quad (9)$$

where \mathbf{W}_ψ , \mathbf{W}_ϕ and \mathbf{W}_h are parameters to be learnt. The query and key values are then combined in $\mathcal{G}_\mathcal{F} = \text{softmax}(\mathcal{F}^T \mathbf{W}_\psi^T \mathbf{W}_\phi \mathcal{F}) h(\mathcal{F})$ as the learnt self-attentions. Finally, the self-attention $\mathcal{G}_\mathcal{F}$ and \mathcal{F} jointly contribute to the output $\mathcal{S}_\mathcal{F}$ by using

$$\mathcal{S}_\mathcal{F} = \mathbf{W}_{\mathcal{S}_\mathcal{F}} \mathcal{G}_\mathcal{F} + \mathcal{F}. \quad (10)$$

Once we obtain the attentive representations as $\mathcal{S}_\mathcal{F}$, we apply two residual blocks including two convolutional layers in each, followed by three fully-connected layers with dropout layers in-between, to output the global scale factor S for the current image.

Probabilistic Scale Regression Head. To predict a global scale, a high-dimensional feature map has to be mapped into a single positive number. One straightforward way is to let the network directly regress the scale number. However, we observe unstable training using this approach. To mitigate this issue, inspired by [4], we propose to use a probabilistic scale regression head to estimate this continuous value. Given a maximum bound that the global scale factor is within, the probability of each scale s is calculated from the output of the scale network $\tilde{\mathcal{S}}$ via the softmax operation $\text{softmax}(\cdot)$. The predicted global scale S is calculated as the sum of each scale s weighted by its probability as

$$S = \sum_{s=0}^{D_{max}} s \times \text{softmax}(\tilde{\mathcal{S}}). \quad (11)$$

By doing so, the regression problem is smoothly resolved by a probabilistic classification-based strategy (see Section 4.1.1 for more ablation results).

3.3. Residual Pose Estimation

As mentioned in Section 3.1, self-supervised depth estimation is built upon the novel view synthesis, which requires both accurate depth maps and camera poses. Es-

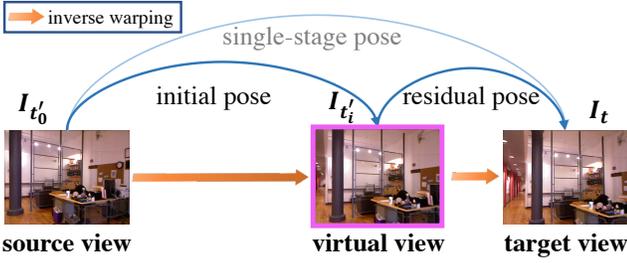


Figure 2. Residual Pose Estimation. Here we give an illustrative example of how a single-stage pose can be decomposed into an *initial pose* and a *residual pose* by virtual view synthesis.

timating accurate relative poses is key for the photometric reprojection loss because inaccurate poses might lead to wrong correspondences between the target and source pixels, causing problems in predicting the depth. Existing methods mostly employ a standalone PoseNet to estimate the 6 Degrees-of-Freedom (DoF) pose between two images. In outdoor environments (*e.g.*, driving scenes like KITTI), the relative camera poses are fairly simple because the cars are mostly moving forward with large translations but minor rotations. This means that pose estimation is normally less challenging. In contrast, in indoor environments, the sequences are typically recorded with hand-held devices (*e.g.*, Kinect), so there are more complicated ego-motions involved as well as much larger rotational motions. It is thus more difficult for the pose network to learn accurate camera poses.

Unlike existing methods [45, 2] that concentrate on “removing” or “reducing” rotational components during data preprocessing, we instead propose a residual pose estimation module to learn the relative camera pose between the target and source images in an iterative manner (see Figure 2). In the first stage, the pose network takes a target image I_t and a source image $I_{t'_0}$ as input and predicts an initial camera pose $T_{t'_0 \rightarrow t}$, where the subscript 0 in t'_0 indicates that no transformation is applied yet. We then follow Equation (2) to bilinearly sample from the source image, reconstructing a virtual view $I_{t'_i \rightarrow t}$ which is expected to be the same as the target image I_t if the correspondences match accurately. However, it will not be the case due to inaccurate pose prediction. Note here the transformation is defined as

$$I_{t'_0 \rightarrow t} = I_{t'} \langle \text{proj}(D_t, T_{t'_0 \rightarrow t}^{-1}, K) \rangle. \quad (12)$$

Next, we utilize a residual pose network (see *Residual-PoseNet* in Figure 1) which takes the target image and the synthesized view $I_{t'_0 \rightarrow t}$ as input and outputs a residual camera pose $T_{(t'_0 \rightarrow t) \rightarrow t}^{\text{res}}$, representing the camera pose of the synthesized image $I_{t'_0 \rightarrow t}$ with respect to the target image.

Now, we bilinearly sample from the synthesized image as

$$I_{(t'_0 \rightarrow t) \rightarrow t} = I_{t'_0 \rightarrow t} \langle \text{proj}(D_t, T_{(t'_0 \rightarrow t) \rightarrow t}^{\text{res}-1}, K) \rangle. \quad (13)$$

Once we obtain a new synthesized view, we can continue to estimate the next residual poses for next view synthesis. For simplicity of notation in Equation (13), we replace the subscript $t'_0 \rightarrow t$ with t'_1 to indicate that one warping transformation is applied, and similarly for the i^{th} transformation. Thus, a general form of Equation (13) is defined by

$$I_{t'_i \rightarrow t} = I_{t'_i} \langle \text{proj}(D_t, T_{t'_i \rightarrow t}^{\text{res}-1}, K) \rangle, i = 0, 1, \dots \quad (14)$$

After we estimate multiple residual poses, the camera pose of source image $I_{t'}$ with respect to the target image I_t can be written as $T_{t' \rightarrow t} = T_{t' \rightarrow t}^{-1}$ where

$$T_{t' \rightarrow t} = \prod_i T_{t'_i \rightarrow t}, i = \dots, k, \dots, 1, 0. \quad (15)$$

By iteratively estimating residual poses, we expect to obtain more accurate camera poses compared with the pose predicted from a single-stage pose network, so that a more accurate photometric reprojection loss can be built up for better depth prediction.

4. Experiments

Datasets. We evaluate the proposed framework *MonoIndoor* on two challenging indoor datasets: the EuRoC MAV [31] dataset, the NYUv2 depth dataset [33] and RGB-D 7-Scenes dataset [32](see supplementary materials for more quantitative results).

Evaluation Metrics. For evaluation, we follow [6] to use the mean absolute relative error (AbsRel), root mean squared error (RMS), and the accuracy under threshold ($\delta_i < 1.25^i, i = 1, 2, 3$) on both datasets.

Implementation Details. We implement our model using PyTorch [27]. In the depth factorization module, we use the same depth network as in [12]; for the scale network, we use two basic residual blocks followed by three fully-connected layers with a dropout layer in-between. The dropout rate is set to 0.5. In the residual pose module, we let the residual pose networks use a common architecture [12] which consists of a shared pose encoder and an independent pose regressor. Each experiment is trained for 40 epochs using the Adam [17] optimizer and the learning rate is set to 10^{-4} for the first 20 epochs and it drops to 10^{-5} for remaining epochs. The smoothness term τ and consistency term γ are set as 0.001 and 0.05, respectively.

4.1. EuRoC MAV Dataset

The EuRoC MAV Dataset [31] contains 11 video sequences captured in two main scenes, a machine hall and a vicon room. Sequences are categorized as *easy*, *medium*

Table 1. Ablation results of design choices and the effectiveness of components in the depth factorization module of our model (**MonoIndoor**) on EuRoC [31]. Prob. Reg.: the probabilistic scale regression block. Note: here we also use the residual pose estimation module when experimenting with different network designs for the depth factorization module.

Network Design	Attention	Prob. Reg.	Error Metric		Accuracy Metric		
			AbsRel	RMSE	δ_1	δ_2	δ_3
I. ScaleCNN	✓	✓	0.140	0.518	0.821	0.956	0.985
II. ScaleNet	✓	✓	0.141	0.519	0.817	0.959	0.988
III. ScaleRegressor	✗	✗	0.139	0.508	0.817	0.960	0.987
III. ScaleRegressor	✓	✗	0.135	0.501	0.825	0.964	0.989
III. ScaleRegressor	✓	✓	0.125	0.466	0.840	0.965	0.993

and *difficult* according to the varying illumination and camera motions. For the training, we use three sequences of “Machine hall” (MH_01, MH_02, MH_04) and two sequences of “Vicon room” (V1_01 and V1_02). Images are rectified with provided camera intrinsics to remove image distortion. During training, images are resized to 512×256. Following [13], we use the Vicon room sequence V2_01 for testing where the ground-truth depths are generated by projecting Vicon 3D scans onto the image planes.

4.1.1 Ablation Study

We perform ablation studies for our design choices of the depth factorization module on the EuRoC MAV dataset. Firstly, we consider the following designs as the backbone of our scale network: I) a pre-trained ResNet-18 [15] followed by a group of Conv-BN-ReLU layers; II) a pre-trained ResNet-18 [15] followed by two residual blocks; III) a lightweight network with two residual blocks which shares the feature maps from the depth encoder as input. These three choices are referred to as the ScaleCNN, ScaleNet and ScaleRegressor, respectively in Table 1. Next, we validate the effectiveness of adding new components into our backbone design. As described in Section 3.2, we mainly integrate two sub-modules: i) a self-attention block and ii) a probabilistic scale regression block.

As shown in Table 1, the best performance is achieved by ScaleRegressor that uses self-attention and probabilistic scale regression. It proves that sharing features with the depth encoder is beneficial to scale estimation. Comparing the results of three ScaleRegressor variants, the performance gradually improves as we add more components (*i.e.*, attention and Prob. Reg.). Specifically, adding the self-attention block improves the overall performance over the baseline backbone; adding the probabilistic regression block leads to a further improvement, which validates the effectiveness of our proposed sub-modules.

4.1.2 Quantitative Results

Since there are not many public results reported on the EuRoC MAV [31] dataset, we mainly compare our model with

Table 2. Ablation results of our MonoIndoor and quantitative comparison with the baseline on the test sequence V2_01 of EuRoC. **Best results are in bold.**

Method	Depth Factorization	Residual Pose	Error Metric		Accuracy Metric		
			AbsRel	RMSE	δ_1	δ_2	δ_3
Monodepth2 [12]	✗	✗	0.157	0.567	0.786	0.941	0.986
MonoIndoor	✓	✗	0.149	0.535	0.805	0.955	0.987
MonoIndoor	✗	✓	0.141	0.518	0.815	0.961	0.991
MonoIndoor	✓	✓	0.125	0.466	0.840	0.965	0.993

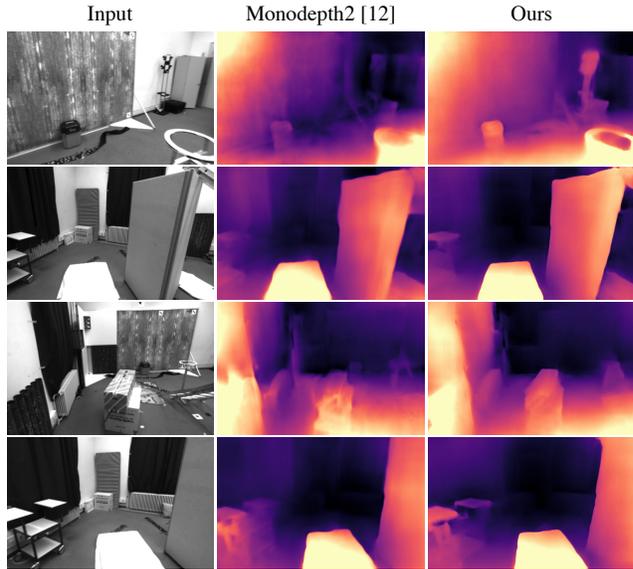


Figure 3. Qualitative comparison of depth prediction on EuRoC. Our model produces more accurate and cleaner depth maps.

the baseline model Monodepth2 [12] and validate the effectiveness of each module of our MonoIndoor. As shown in Table 2, adding our depth factorization module reduces the AbsRel from 15.7% to 14.9%, and our residual pose module decreases the AbsRel to 14.1%, which verifies the usefulness of each module. Our full model achieves the best performance across all evaluation metrics. Specifically, compared to Monodepth2, the AbsRel by our **MonoIndoor** is significantly decreased from 15.7% to 12.5% and the δ_1 is improved by around 6%, from 78.6% to 84.0%.

4.1.3 Qualitative Results

Figure 3 gives a qualitative comparison of depth maps predicted by Monodepth2 [12] and our **MonoIndoor**. From Figure 3, it is clear that the depth maps generated by our model are much better than the ones by Monodepth2. For instance, in the first row, our model can predict precise depths for the *hole* region at the right-bottom corner whereas such a hole structure in the depth map by Monodepth2 is missing. Besides, in the second row, our model can predict much sharper depth map of the *ladder* at the right-top area while Monodepth2 cannot. These observations are also consistent with the better quantitative results

Table 3. Ablation results of the effectiveness of each module of our MonoIndoor on NYUv2. “No. Residual Pose Block” means the number of residual poses we estimate in the residual pose estimation module.

Model	Depth Factorization	No. Residual Pose Block	Error Metric		Accuracy Metric		
			AbsRel	RMS	δ_1	δ_2	δ_3
Monodepth2 [12]	✗	0	0.16	0.601	0.767	0.949	0.988
MonoIndoor	✓	0	0.152	0.576	0.792	0.951	0.987
MonoIndoor	✗	1	0.142	0.553	0.813	0.958	0.988
MonoIndoor	✓	1	0.134	0.526	0.823	0.958	0.989
MonoIndoor	✗	2	0.141	0.548	0.814	0.958	0.988
MonoIndoor	✓	2	0.141	0.546	0.818	0.958	0.989

in Table 2, proving the superiority of our model.

4.2. NYUv2 Depth Dataset

In this section, we evaluate our **MonoIndoor** on the NYUv2 depth dataset [33] which contains 464 indoor video sequences captured by a hand-held Microsoft Kinect RGB-D camera with a resolution of 640×480 . We use the official training and validation splits which include 302 and 33 sequences respectively. We rectify the images with provided camera parameters to remove distortions. Following [44, 2], the raw dataset is firstly downsampled 10 times along the temporal dimension to remove redundant frames, resulting in $\sim 20K$ images for training. During training, images are resized to 320×256 . We use officially provided 654 images with dense labelled depth maps for testing.

4.2.1 Ablation Study

We perform another ablation study for the depth factorization module on NYUv2 [33]. In Table 3, comparing with Monodepth2 which predicts depth without any guidance of global scales, using the depth factorization module with a separate scale network can improve the performance, decreasing the AbsRel from 16% to 15.2% and increasing δ_1 to 79.2%. Next, we experiment to validate the effectiveness of the residual pose estimation module. Comparing the rows in Table 3, by adding the residual pose estimation module with one residual pose block, we observe an improved performance from 16.0% down to 14.2% for the AbsRel and from 76.7% up to 81.3% for δ_1 . Furthermore, by applying both the depth factorization module and the residual pose estimation module (*i.e.*, our full **MonoIndoor**), significant improvements can be achieved across all evaluation metrics. For instance, the AbsRel is reduced to 13.4% and the δ_1 is increased to 82.3%. However, referring to the last two rows, when adding more residual pose blocks and training with/without the depth factorization module, the performance does not significantly improve or even becomes worse. We will leave the investigation of this phenomenon for future work.

Table 4. Comparison of our method to existing supervised and self-supervised methods on NYUv2 [33]. Best results among supervised and self-supervised methods are in **bold**.

Methods	Supervision	Error Metric		Accuracy Metric		
		AbsRel	RMS	δ_1	δ_2	δ_3
Make3D [30]	✓	0.349	1.214	0.447	0.745	0.897
Depth Transfer [16]	✓	0.349	1.210	-	-	-
Liu <i>et al.</i> [25]	✓	0.335	1.060	-	-	-
Ladicky <i>et al.</i> [18]	✓	-	-	0.542	0.829	0.941
Li <i>et al.</i> [20]	✓	0.232	0.821	0.621	0.886	0.968
Roy <i>et al.</i> [29]	✓	0.187	0.744	-	-	-
Liu <i>et al.</i> [24]	✓	0.213	0.759	0.650	0.906	0.976
Wang <i>et al.</i> [38]	✓	0.220	0.745	0.605	0.890	0.970
Eigen <i>et al.</i> [5]	✓	0.158	0.641	0.769	0.950	0.988
Chakrabarti <i>et al.</i> [3]	✓	0.149	0.620	0.806	0.958	0.987
Laina <i>et al.</i> [19]	✓	0.127	0.573	0.811	0.953	0.988
Li <i>et al.</i> [21]	✓	0.143	0.635	0.788	0.958	0.991
DORN [8]	✓	0.115	0.509	0.828	0.965	0.992
VNL [41]	✓	0.108	0.416	0.875	0.976	0.994
Fang <i>et al.</i> [7]	✓	0.101	0.412	0.868	0.958	0.986
Zhou <i>et al.</i> [45]	✗	0.208	0.712	0.674	0.900	0.968
Zhao <i>et al.</i> [44]	✗	0.189	0.686	0.701	0.912	0.978
Monodepth2 [12]	✗	0.160	0.601	0.767	0.949	0.988
Bian <i>et al.</i> [2]	✗	0.147	0.536	0.804	0.950	0.986
MonoIndoor(Ours)	✗	0.134	0.526	0.823	0.958	0.989

4.2.2 Quantitative Results

We present the quantitative results of our model **MonoIndoor** and both state-of-the-art (SOTA) supervised and self-supervised methods on NYUv2 in Table 4. It shows that our model outperforms previous self-supervised SOTA methods, reaching the best results across all metrics. Specifically, compared to a recent self-supervised method by Bian *et al.* [2] which removes rotations via “weak rectification”, our method reduces AbsRel by 1.3% and increases δ_1 by 1.9%, reaching an AbsRel of 13.4% and δ_1 of 82.3%. In addition to that, our model outperforms a group of supervised methods and close the performance gap between the self-supervised methods and fully-supervised methods.

4.2.3 Qualitative Results

Figure 4 visualizes the predicted depth maps on NYUv2. Compared with the results from the Monodepth2 [12], depth maps predicted from our model (**MonoIndoor**) are more precise and closer to the ground-truth. For instance, looking at the third column in the first row, the depth in the region of *chairs* predicted from our model is much sharper and cleaner, being close to the ground truth (the last column). On the rightmost area of the same image where there is a *shelf*, our model can produce better depth predictions that reflect its shape. These observations are consistent with our quantitative results in Table 4.

5. Conclusions

In this work, we have presented a novel monocular self-supervised depth estimation model, namely **MonoIndoor**, to study good practices towards predicting accurate depth



Figure 4. Qualitative comparison on NYUv2 [33]. Compared with Monodepth2 [12], our model produces accurate depth maps (in the third column) that are closer to the ground-truth.

maps in indoor environments. We first introduce the *depth factorization module* to jointly learn a global scale factor and a relative depth map from an input image. To estimate accurate camera poses for novel view synthesis, we propose a *residual pose estimation module* that decomposes a global pose into an initial pose and one or a few residual poses, which in turn improves the depth model. We have shown that our model achieves the state-of-the-art performance among the self-supervised methods on three chal-

lenging indoor datasets, *i.e.*, EuRoC, NYUv2 and 7-Scenes.

It is to be noted that our depth factorization module is in itself agnostic to the types of supervision, so it may also be helpful for supervised depth prediction. In the future, we plan to investigate its effectiveness in a supervised setup. Another interesting future direction would be to train our method on multiple datasets with various depth ranges and then test it for zero-shot cross-dataset transfer as in [28].

References

- [1] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *arXiv preprint arXiv:1908.10553*, 2019. [2](#), [3](#)
- [2] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian Reid. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. *arXiv preprint arXiv:2006.02708*, 2020. [2](#), [5](#), [7](#)
- [3] Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NeurIPS*, 2016. [7](#)
- [4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. [4](#)
- [5] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, December 2015. [1](#), [7](#)
- [6] David Eigen, Christian Puhres, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. [2](#), [5](#)
- [7] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *WACV*, March 2020. [7](#)
- [8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, June 2018. [1](#), [2](#), [7](#)
- [9] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756, 2016. [1](#), [2](#)
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. [1](#), [4](#)
- [11] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017. [2](#)
- [12] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3828–3838, 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [13] 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. [6](#)
- [14] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, pages 484–500, 2018. [1](#)
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016. [6](#)
- [16] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depthtransfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 2014. [7](#)
- [17] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR*, pages 1–15, 2015. [5](#)
- [18] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *CVPR*, June 2014. [7](#)
- [19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248. IEEE, 2016. [2](#), [7](#)
- [20] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, June 2015. [7](#)
- [21] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *ICCV*, Oct 2017. [2](#), [7](#)
- [22] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, pages 4521–4530, 2019. [2](#)
- [23] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. [2](#)
- [24] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, June 2015. [7](#)
- [25] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *CVPR*, June 2014. [7](#)
- [26] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. [1](#), [2](#)
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035. 2019. [5](#)
- [28] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. [2](#), [8](#)
- [29] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, June 2016. [7](#)
- [30] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 31(5):824–840, 2008. [2](#), [7](#)
- [31] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. [1](#), [2](#), [5](#), [6](#)
- [32] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. [2](#), [5](#)

- [33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. [1](#), [2](#), [5](#), [7](#), [8](#)
- [34] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. [2](#)
- [35] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. In *ECCV*, pages 437–455, 2020. [2](#)
- [36] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, pages 5038–5047, 2017. [2](#)
- [37] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, pages 2022–2030, 2018. [2](#), [3](#)
- [38] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, June 2015. [7](#)
- [39] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *CVPR*, pages 5555–5564, 2019. [2](#)
- [40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. [4](#)
- [41] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, October 2019. [1](#), [2](#), [7](#)
- [42] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018. [2](#)
- [43] Zehao Yu, Lei Jin, and Shenghua Gao. P² net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *ECCV*, pages 206–222, 2020. [2](#)
- [44] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, pages 9151–9161, 2020. [1](#), [2](#), [7](#)
- [45] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *ICCV*, pages 8618–8627, 2019. [1](#), [2](#), [5](#), [7](#)
- [46] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1858, 2017. [1](#), [2](#), [3](#)
- [47] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. *arXiv preprint arXiv:2007.10983*, 2020. [1](#), [2](#), [3](#)
- [48] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018. [2](#)