GasMono: Geometry-Aided Self-Supervised Monocular Depth Estimation for Indoor Scenes

[1]Chaoqiang Zhao   [2]Matteo Poggi   [2]Fabio Tosi

[1]Lei Zhou   [1]Qiyu Sun   [1,*]Yang Tang   [2]Stefano Mattoccia

[1]East China University of Science and Technology   [2]University of Bologna

Abstract

This paper tackles the challenges of self-supervised monocular depth estimation in indoor scenes caused by large rotation between frames and low texture. We ease the learning process by obtaining coarse camera poses from monocular sequences through multi-view geometry to deal with the former. However, we found that limited by the scale ambiguity across different scenes in the training dataset, a naïve introduction of geometric coarse poses cannot play a positive role in performance improvement, which is counter-intuitive. To address this problem, we propose to refine those poses during training through rotation and translation/scale optimization. To soften the effect of the low texture, we combine the global reasoning of vision transformers with an overfitting-aware, iterative self-distillation mechanism, providing more accurate depth guidance coming from the network itself. Experiments on NYUv2, ScanNet, 7scenes, and KITTI datasets support the effectiveness of each component in our framework, which sets a new state-of-the-art for indoor self-supervised monocular depth estimation, as well as outstanding generalization ability. Code and models are available at `https://github.com/zxcqlf/GasMono`

1. Introduction

Depth estimation from images is one of the fundamental tasks in computer vision and plays a key role in several higher-level applications [28, 44]. It has a long history and has been intensively studied building upon multi-view geometry [17, 37], exploiting image matching across two or multiple images and their camera positions [31, 35, 41]. The advent of deep learning rejuvenated this field and introduced new, exciting perspectives. Among many, the possibility of learning to estimate depth out of a single image – for long considered the holy grail in computer vision – be-
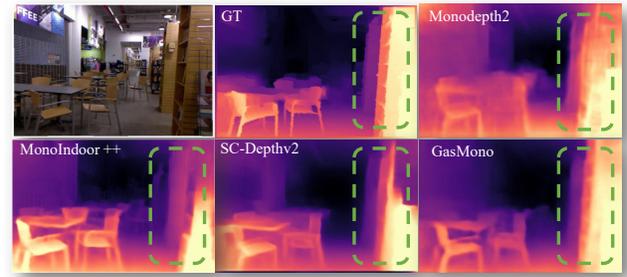
Figure 1. **Comparison between existing methods [2, 11, 27] and GasMono.** Our framework shows remarkable accuracy on thin objects and global structures.

came true [7, 9, 23, 38, 39]. However, this came at the cost of requiring a massive amount of images annotated with ground truth depth, often expensive to collect.

In light of this, multi-view geometry maintained a key role in softening this latter constraint, allowing for the development of self-supervised monocular depth estimation frameworks [57]. These replace the need for depth labels by exploiting a proxy signal based on image reconstruction of the frame given as input to the network, i.e. the *target*, starting from one or multiple *source* images. The only requirement consists of collecting raw video sequences [64] or rectified stereo images [12]. Among the two alternatives, the former [64] results as the cheapest and most flexible, since requiring a single camera only to move around and collect training data, with the relative camera poses between consecutive frames needing to be estimated by employing a dedicated pose network (PoseNet) alongside the depth estimation task. A large body of literature built ever more accurate self-supervised solutions, mainly focusing on outdoor environments – i.e. with driving context [10, 51] representing the preferred benchmark.

Nonetheless, the indoor setting is equally important for the development of, among others, navigation and assistive technologies, although featuring 1) much more complex ego-motion configurations and 2) large untextured regions,

---

* Corresponding author, yangtang@ecust.edu.cn.

making this setting itself challenging for self-supervised depth estimation frameworks [62]. Specifically, the different data collection equipment involved in outdoor and indoor scenes – car-mounted vs handheld cameras – leads to motion models largely different in the two cases. As an example, the average rotation between consecutive images in KITTI [10] is $0.25°$, while on NYUv2 [32] dataset it is $2.28°$ [2]. Larger rotations among the images hinder the training process, because of the discontinuous Euler angle representation [65] commonly used for this task. Additionally, the lack of texture tampers the image reconstruction process through which supervision is provided to the network, with several local minima in the training loss signal.

In this paper, we propose a novel, **G**eometry-**a**ided **s**elf-supervised framework for **Mono**cular depth estimation, dubbed *GasMono*, specifically designed to face these challenges. Specifically, we leverage classic structure-from-motion algorithms such as COLMAP [41] on the training sequences to initialize the pose estimation process, which is then refined to cope with the scale inconsistency occurring between the different monocular sequences part of the training set. To deal with the reduced texture characterizing indoor images, we combine recent architectures based on vision transformers [59] with an iterative, self-distillation scheme to obtain stronger supervision and, thus, train GasMono more effectively. Our main contributions are:

- We tackle the challenges of learning camera poses in indoor scenes by exploiting strong priors coming from classic structure-from-motion algorithms [41].

- This, however, is not sufficient: we explore the factors in such an approach making the training process unstable and a further, learning-based refinement strategy is proposed to optimize both rotation and translation/scale of the initial poses.

- We explore the effectiveness of transformer architecture in improving the depth estimation of low-texture regions for indoor scenes, coupled with an overfitting-aware iterative self-distillation method, iteratively distilling pseudo labels from the depth network itself.

- Our GasMono framework is evaluated on a variety of indoor datasets, establishing a new state-of-the-art for indoor, self-supervised monocular depth estimation.

Fig. 1 shows a comparison between existing frameworks and GasMono, which shows more coherent predictions.

2. Related Work

We review the basic framework and related works concerning self-supervised monocular depth estimation.

Outdoor Self-Supervised Methods. Pivotal works in self-supervised monocular depth estimation focused on the outdoor setting, in particular dealing with driving environments [10], by replacing ground truth labels with image reconstruction losses on stereo images [12] or monocular videos [64]. A rich literature of method following either the former [13, 14, 36, 46, 48, 56] or the latter [3, 11, 15, 54, 59] strategies have been developed, with monocular sequences being more practical, yet requiring 1) to estimate relative poses between frames at training time and 2) to deal with independently moving objects, violating the principles behind image reconstruction. This second challenge has been the object of study by several works, by masking the reprojection loss [11, 43], exploiting additional tasks – e.g. as optical flow [54], semantic segmentation [21] or both [47] – while additional constraints such as the consistency between predicted depth maps [3] or normals [52], adversarial losses [58] and uncertainty modeling [34] result in good cues to ease the learning process. Prior work [22] already shows how Structure-from-Motion can improve the learning process on videos, yet we will show how it cannot be exploited seamlessly in the indoor setting. On an orthogonal dimension, the development of new architectures also plays a crucial role in boosting the accuracy of self-supervised depth estimation. Among them, PackNet [15] and HRDepth [30] proved consistent improvements over UNet-like architectures using vanilla encoders such as VGG and ResNet. More recently, vision transformers [59] yielded further improvements thanks to joint local and global reasoning, allowing for predicting more detailed depth maps on outdoor scenes compared to established CNNs.

Indoor Self-Supervised Methods. Unlike the outdoor setting considered by early approaches, monocular sequences collected in indoor environments expose much more complex motions between frames, with much larger rotational components. Because of Euler angle representation of 3D rotation hampering the learning process of the PoseNet [65], the self-supervised framework being effective in outdoor environments cannot achieve satisfactory results on indoor scenes. To soften this problem, Zhao *et al*. [60] recover relative pose from dense optical flow to fully replace the PoseNet. Bian *et al*. [2, 3] demonstrate the role of rotation in unsupervised training, and they propose an Auto-Rectify Network to estimate and eliminate the rotation in advance. MonoIndoor series [20, 27] consider the depth range changes between different indoor scenes and design a transformer for depth factorization. Moreover, a residual pose estimation module is proposed in [20, 27] to iteratively optimize the relative pose predicted by networks. A second challenge when dealing with indoor scenes consist of the much lower texture present in images, making the image reconstruction loss often ineffective. To tackle this, Li *et al*. [25] use Manhattan normal constraint, while, Zhou *et al*. [62] propose to use optical flow to both obtain stronger supervision in low-textured regions, as well as to

assist the training of the PoseNet.

Our framework faces both challenges, respectively 1) by exploiting geometry by means of structure-from-motion algorithms [41] to ease the pose estimation process and 2) by deploying vision transformers. However, these two strategies alone are ineffective: although being a good initialization, coarse poses [41] are noisy and scale-inconsistent across different sequences. Thus, they need to be properly refined and further optimized.

Self-Distillation. Different from the knowledge distillation methods, self-distillation approaches treat the model as its own teacher and distill labels by itself. This strategy is widely used in the classification task [19] and, more recently, has been introduced for self-supervised depth estimation as well [33] by exploiting stereo images during training. Inspired by this latter work, we introduce self-distillation into the monocular sequence-based self-supervised framework and prove that this approach alone yields sub-optimal improvements. Indeed, we cast self-distillation as an iterative approach by generating and selecting more and more accurate pseudo-labels.

3. GasMono Framework

We now introduce the key component of our framework, sketched in Fig. 2.

3.1. Geometry-Aided Pose Estimation

The standard supervision protocol for self-supervised monocular depth estimation frameworks trained on video sequences consists of reprojecting pixels from a source image $\mathcal{I}_s$ into the target $\mathcal{I}_t$ according to estimated depth D_t and relative camera pose $E_{t\to s} = R_{t\to s}|t_{t\to s}$. This means that, for pixel p_t in the target view, its coordinates p_s in the source view can be obtained as follows:

$$p_s \sim K[R_{t\to s}D_t(p_t)K^{-1}p_t + t_{t\to s}] \qquad (1)$$

Given the challenges of learning an accurate relative pose between images in presence of large rotations, we propose to get rid of the commonly used PoseNet and replace it with a traditional pose estimation algorithm. For this purpose, we leverage COLMAP [41] to obtain camera poses $E_i^{CP} = R_i^{CP}|t_i^{CP}$ for images $\mathcal{I}_i$ in every single indoor sequence in the training set. Then, for a given image pair $\mathcal{I}_t$, $\mathcal{I}_s$, respectively the target and source frames, we can obtain the relative pose between the two as $E_{t\to s}^{CP} = R_{t\to s}^{CP}|t_{t\to s}^{CP} = E_s^{CP}E_t^{CP-1}$. Differently from two-frame pose estimation, a structure-from-motion pipeline such as COLMAP reasons globally over the entire sequence. We argue that, since pose estimation represents a side task for learning monocular depth, it is worth exploiting sequences as a whole.

Nonetheless, poses estimated by COLMAP, which we will refer to as *coarse* from now on, suffers from some prob-

lems, specifically 1) scale inconsistency across different sequences and scale drift within the very same scene because of monocular ambiguity, 2) noise in estimated rotation and translation due to the lack of texture. This makes COLMAP alone ineffective at seamlessly replacing the PoseNet to train a monocular depth network.

3.1.1 Translation Rescaling and Refinement

To solve the former issue, we deploy a shallow network, AlignNet, to refine the coarse translation and to rescale it during training, allowing to overcome scale inconsistency across the many sequences in the training set.

Accordingly, AlignNet processes the target and source images and predicts a scale factor s^{NN} and a residual shift Δt^{NN} applied to the translation component $t_{t\to s}^{CP}$ estimated by COLMAP. Then, estimated translation $t_{t\to s}$ from target to source views is obtained as

$$t_{t\to s}^* = s_{t\to s}^{NN}t_{t\to s}^{CP} + \Delta t_{t\to s}^{NN} \qquad (2)$$

This refined translation vector is then used in Eq. 1, leading to scale adjustment when learning to estimate monocular depth maps by exploiting $R_{t\to s}^{CP}|t_{t\to s}^*$, both within the same sequence as well as across all the scenes in the training dataset, shown as "Optim_t" in Fig. 2.

We can consider AlignNet as a training optimization tool, refining coarse poses to make them scale-consistent as a whole. As such, it loses its utility once the training procedure is completed.

3.1.2 Rotation Optimization

Bian *et al.* [2] has proved that the rotation in self-supervised training is independent of depth learning. Nonetheless, in case of inaccurate rotations being estimated between target and source frames, noisy gradients will arise and affect the training of the depth network as well. However, we argue that such an error is a small residual rotation, largely smaller compared to the original rotation between the two frames, and thus a deep network could learn such a correction.

Inspired by the rotation and pose rectify methods in recent works [2, 20], we re-introduce the PoseNet to solely estimate the residual pose $R_{t\to s}^r|t_{t\to s}^r$ between the image being reconstructed from $R_{t\to s}^{CP}|t_{t\to s}^*$ and the target one, by computing an additional reprojection loss using the same, estimated depth and the newly computed, residual pose – "Optim_R" in Fig. 2. Although the PoseNet cannot deal properly with large rotations, it can effectively estimate the small, residual rotations result of the COLMAP errors.

3.2. Low-Texture Regions

In self-supervised training, back-propagation behavior replies to the photometric gradient changes on RGB images.
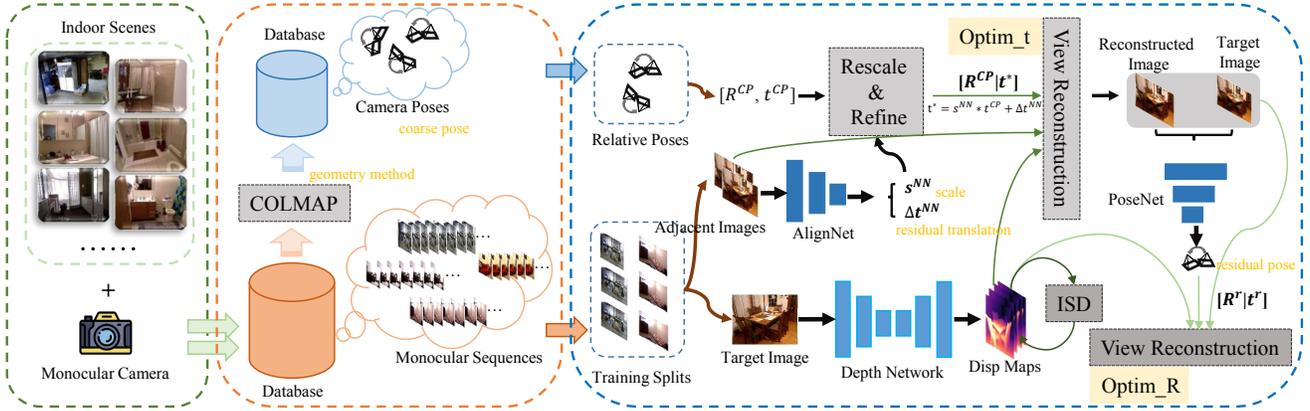
Figure 2. **GasMono: a geometry-aided self-supervised monocular depth estimation framework for indoor scenes.** Note that no ground truth labels are used in the training process. With the monocular image sequences selected by the camera from several indoor scenes, the structure-from-motion package COLMAP is used to estimate the coarse poses for the camera on each sequence. Then, the image sequences and coarse poses are used to train the depth model. To improve the coarse translation, an AlignNet is designed to estimate the scale s^{NN} and residual translation Δt. Moreover, a PoseNet is also designed to further improve the pose, especially the coarse rotation, based on the reconstructed and target images. AlignNet and PoseNet are only used during training.

Those regions with effective photometric changes provide strong gradients for depth learning, while regions with low-texture regions, like walls and floors, cannot provide effective supervisory signals, because multiple depth hypotheses lead to photometric errors being close to zero [62], thus driving the network into local minima. Therefore, the depth estimation process mainly relies on the inferring ability of the network itself for low-texture regions. The use of some additional constraints, coming from cues such as optical flow or plane normals [25,62], might help. Nonetheless, this requires additional supervision and can suffer the same issues due to low texture, as in the case of optical flow. Thus, we choose to tackle it on the architectural side, in particular by overcoming the limited receptive fields of CNNs with a vision transformer. Moreover, previous works proved the effectiveness of label distillation [33, 34] to improve the accuracy of the depth network. Therefore, we propose a more effective solution for obtaining more accurate distilled labels in order to provide stronger supervision to GasMono.

3.2.1 Network Architecture

Our framework consists of three networks, a Depth Network for monocular depth estimation, an Alignment Network (AlignNet) for scale correction and residual translation prediction, and a PoseNet for residual pose estimation. The overall training architecture is shown in Fig. 2.

Considering the outstanding performance of the transformer in long-range relationship modeling between features, in order to enhance the global feature extraction for the depth inferring of low-texture regions, we introduce a transformer encoder, MPViT [24], as the depth encoder by

following [59]. The self-attention mechanism in the encoder is implemented in an efficient factorized way [24]:

$$\text{FactorAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \frac{\mathbf{Q}}{\sqrt{C}}(\text{softmax}(\mathbf{K})^T \mathbf{V}) \quad (3)$$

where C refers to the embedding dimension. Query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V} \in \mathbb{R}^{N \times C}$) vectors are projected from vision tokens. Moreover, for the depth decoder, we replace the standard upsampling implemented by Monodepth2 [11] and others with convex upsampling [45], bringing the 4 scale disparity maps to full resolution, at which they are used for the iterative self-distillation operations described in the remainder. To build both the AlignNet and PoseNet, we use the same architecture used to implement the PoseNet in previous works [11,59], which is based on ResNet-18 [18].

3.2.2 Iterative Self-Distillation Method

Inspired by [33], we propose an overfitting-driven Iterative Self-Distillation procedure (ISD) to obtain the depth map with the lowest per-pixel minimum reprojection error, yielding more accurate labels for any specific training sample.

The key steps of ISD are listed in Algorithm 1. For each single training image, we iterate this process multiple times (row 4). At the first iteration, we select the per-pixel minimum reprojection error across all scales and its corresponding predicted depth (rows 6-14). Then, we update the network by minimizing a depth loss between the current best depth map and predictions at each scale (rows 15-16). This procedure is repeated for a number of iterations.

We will show in our experiments how this allows for improving the supervision provided by naïve distillation [33].

Algorithm 1: Iterative Self-Distillation **ISD**

Input:
Target image: $\mathcal{I} \in R^{b \times 3 \times H \times W}$;
Reconstructed target image: $\tilde{\mathcal{I}}^s \in R^{b \times 3 \times H \times W}$;
Disparity map: $D^s \in R^{b \times 1 \times H \times W}$;
Output: Self-distillation loss (L_{ISD})

1   *total_iterations* $= n > 0$;
2   *curr_iterations* $= 0$;
3   *Dict* $= \{\}$;
4   **for** *curr_iteration < total_iterations* **do**
5   *curr_iterations* += 1;
6   **for** *s in range(4)* **do**
7   Rec_error $= \mathcal{F}(\tilde{\mathcal{I}}^s, \mathcal{I}) \in R^{b \times 1 \times H \times W}$;
8   **if** *Dict is empty* **then**
9   Dict['disp_best'] $= D^0$;
10   Dict['error_min'] = Rec_error;
11   **end**
12   Dict['disp_best'] = torch.where(Rec_error<
           Dict['error_min'], D^s, Dict['disp_best']);
13   Dict['error_min'] = torch.min(torch.cat(
           [Rec_error, Dict['error_min']]));
14   **end**
15      $L_{ISD} = \sum_{s=1}^{4}(log(|Dict['disp\_best'] - D^s| + 1))$;
16   L_{tot}.backward()
17 **end**

3.3. Training Losses

Different from competitors [2, 27], we follow outdoor methods [11], the key term of our training loss consists of the minimum view reconstruction loss, accompanied by a smoothness term.

View Reconstruction Loss. Having obtained a reconstructed image $\tilde{\mathcal{I}}$, its error with respect to the target image $\mathcal{I}$ is measured by means of Structured Similarity Index Measure (SSIM) and L1 difference, combined as in [11]:

$$\mathcal{F}(\tilde{\mathcal{I}}, \mathcal{I}) = \alpha \cdot \frac{1 - \text{SSIM}(\tilde{\mathcal{I}}, \mathcal{I})}{2} + (1 - \alpha) \cdot |\tilde{\mathcal{I}} - \mathcal{I}| \quad (4)$$

with α commonly set to 0.85 [11]. Besides, to soften the effect of occlusions between two views, the minimum among losses obtained with respect to forward and backward adjacent frames being warped is computed:

$$\mathcal{L}_{rec}(p) = \min_{i \in [1, -1]} \mathcal{F}(\tilde{\mathcal{I}}_i(p), \mathcal{I}(p)) \quad (5)$$

with '1' and '-1' referring to the forward and backward adjacent frames, respectively.

Smoothness Loss. The edge-aware smoothness loss is used to further improve the inverse depth map d:

$$\mathcal{L}_{smooth} = |\partial_x d^*| e^{\partial_x I} + |\partial_y d^*| e^{\partial_y I}, \quad (6)$$

where $d^* = d/\hat{d}$ represents the mean-normalized inverse depth. Besides, following [11], an auto-mask μ is calculated to filter static frames and some repeated texture regions.

Iterative Self-Distillation Loss. As described before, GasMono self-distills proxy labels to be used as additional supervision. Given proxy labels d^{best} obtained according to Algo. 1, we minimize the log-error of predicted depth d with respect to it:

$$\mathcal{L}_{ISD} = log(|d^{best} - d| + 1) \quad (7)$$

Total Loss. Finally, the view reconstruction loss $\mathcal{L}_{rec}$, the smoothness loss $\mathcal{L}_{smooth}$ and the distillation term $\mathcal{L}_{ISD}$ are computed on outputs at any given scale – brought to full resolution – to obtain the total loss term $\mathcal{L}_{tot}$. More specifically, two reconstruction losses are computed, respectively $\mathcal{L}_{rec}^{Optim\_t}$ and $\mathcal{L}_{rec}^{Optim\_R}$:

$$\mathcal{L}_{tot} = \mathcal{L}_{rec}^{Optim\_t} + \beta \cdot \mathcal{L}_{rec}^{Optim\_R}$$
$$+ \lambda \cdot \mathcal{L}_{smooth} + \mu \cdot \mathcal{L}_{ISD} \quad (8)$$

with $\mathcal{L}_{rec}^{Optim\_t}$ and $\mathcal{L}_{rec}^{Optim\_R}$ being computed with reconstructed images based on the poses obtained after translation and rotation optimizations respectively, and with β, λ and μ being set to 0.2, 0.001 and 0.1. Finally, total losses are averaged across scales.

4. Experimental Results

We now evaluate the performance of GasMono with respect to existing methods from the literature. We first describe the datasets and protocol used for our experiments, then conduct an exhaustive study on GasMono behavior when trained with COLMAP poses and our pose optimization strategies, followed by an ablation study on our model and a final, comparison with state-of-the-art approaches.

4.1. Implementation Details

We start by describing the datasets involved in our evaluation and the implementation details of our method.

4.1.1 Datasets

We conduct our experiments on three popular indoor datasets: NYUv2 [32], 7scenes [42], and ScanNet [4]. To validate our iterative-self distillation strategy, we also evaluate on the outdoor KITTI [10] dataset.

Indoor Datasets: NYUv2 contains 464 indoor sequences with dense depth ground truth captured by a handheld RGB-D camera at 640×480 resolution. Following previous works [2], we use the officially provided 654 densely labeled images for testing and the remaining 335 sequences for training and validation. Since COLMAP failed in some short training sequences, these have been excluded. Differently from the MonoIndoor series [20, 27], which rectifies

	Details	scale_std	lower is better				higher is better		
			Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
(a)	$\mathbf{T}(R^{NN}, t^{NN})$ – Monodepth2 [11]	0.259	0.167	0.137	0.603	0.208	0.754	0.944	0.985
(b)	$\mathbf{T}^{\ddagger}(R^{CP}, t^{CP})$	0.597	0.298	0.521	1.079	0.346	0.534	0.808	0.925
(c)	$\mathbf{T}(R^{CP}, t^{NN})$	0.215	0.160	0.131	0.584	0.203	0.771	0.945	0.985
(d)	$\mathbf{T}^{\ddagger}(R^{NN}, t^{CP})$	-	0.311	0.460	0.999	0.369	0.511	0.779	0.904
(e)	$\mathbf{T}^{\ddagger}(R^{CP}, t^{CP} * s^{NN})$	0.237	0.183	0.169	0.661	0.231	0.718	0.920	0.978
(f)	$\mathbf{T}^{\ddagger}(R^{CP}, t^{CP} + \Delta t^{NN})$	0.501	0.255	0.326	0.888	0.309	0.583	0.845	0.946
(g)	$\mathbf{T}(R^{CP}, t^{CP} * s^{NN} + \Delta t^{NN})$	0.240	0.159	0.132	0.590	0.203	0.775	0.945	0.986
(h)	Monodepth2 [11]	*	0.167	0.137	0.603	0.208	0.754	0.944	0.985
(i)	+ MPViT	*	0.145	0.109	0.546	0.186	0.804	0.959	0.991
(j)	+ MPViT + CP	*	0.311	0.460	0.999	0.369	0.511	0.779	0.904
(k)	+ MPViT + CP + Optim_t	*	0.124	0.090	0.490	0.165	0.850	0.968	0.991
(l)	+ MPViT + CP + Optim_t + Optim_R	*	0.114	0.085	0.469	0.155	0.867	0.972	0.992
(m)	+ MPViT + ISD	*	0.137	0.100	0.523	0.177	0.821	0.964	0.991
(n)	+ MPViT + CP + Optim_t + ISD	*	0.122	0.091	0.486	0.162	0.857	0.969	0.991
(o)	+ MPViT + CP + Optim_t + Optim_R + ISD	*	0.113	0.083	0.459	0.153	0.871	0.973	0.992

Table 1. **Ablation Studies**. The upper part is based on the monodepth2 [11], which explores the utilization and optimization methods for the coarse pose from COLMAP. "$(R/t)^{CP/NN}$" refers to the (R/t) generated by $COLMAP/NeuralNetwork$. "scale_std" stands for the std. error of the scale alignment factor on the testset. "$\ddagger$" denotes that the training process has a high probability of not converging, and we only record the results of the converged case here. The other part is ablation experiments to prove the effectiveness of each module in this paper. CP refers to the model trained by using coarse pose $[R^{CP}, t^{CP}]$.

the image distortion with provided camera intrinsics, we directly use raw images for training. **7-Scenes** consists of 7 indoor scenes, each containing several image sequences at 640×480 resolution. We follow the official train/test split for each scene. Following [2], we extract the first image from every 10 frames for each scene for testing. **ScanNet** provides 1513 indoor RGB-D videos, captured by handheld devices. To evaluate the generalization performance of our depth model trained on NYUv2, we use the officially released test set following [2, 27]. On these datasets, images are resized to 320×256 during both training and testing.

Outdoor Dataset: KITTI contains 61 scenes, with a typical image size of 1242×375, captured using a stereo rig mounted on a moving car equipped with a LiDAR sensor. Following the literature [12, 64], we use the image split of Eigen *et al.* [8] for training and validation. To compare with the existing solutions, we evaluate the model on the test split of [8] with the same evaluation metrics. Images are resized to 640×192 during both training and testing.

4.1.2 Settings

We implement GasMono in PyTorch, training it for 40 epochs on the NYUv2 dataset and 20 epochs on the KITTI dataset by using AdamW [29] optimizer. The batch size is set to 12 and the number of iterations for ISD to 2. The translation, pose encoder and depth encoders are pre-trained on ImageNet [5]. For training, we use a single NVIDIA Tesla P40 GPU. Source code and generated coarse poses will be open-sourced in case of acceptance.

4.2. Ablation Studies

We start by inquiring about the impact of any component in GasMono, to assess their effectiveness at dealing with the challenges of indoor monocular depth estimation.

4.2.1 Coarse Poses and Network Convergence

To analyze the difficulties as well as improvements yielded by employing geometry-based pose estimation algorithms such as COLMAP, we build a set of experiments based over the Monodepth2 baseline [11] on the NYUv2 dataset [32], collected in Tab. 1. For training, only photometric and smoothness losses from [11] are used in any setting.

Row (a) contains the results achieved by the baseline – i.e., Monodepth2 using a PoseNet to estimate poses. Then, rotation R^{CP} and translation t^{CP} estimated by COLMAP are directly used to replace the pose network in (b), showing that this approach does not yield satisfactory accuracy.

Then, we select only one among rotation R^{NN} and translation t^{NN} components estimated by the PoseNet to respectively replace the COLMAP counterpart. From the results, we can notice that using only translation t^{NN} estimated by the network, while keeping the rotation component being estimated by COLMAP (c), allows for outperforming the baseline. Besides, using the R^{NN} and t^{CP} (d) yields to the worst results observed so far. These two experiments confirm that 1) the translation component t^{CP} is the main cause of failure when using COLMAP poses – indeed, replacing it with t^{NN} is sufficient to boost performance – and 2) COLMAP estimated rotations results largely better than those learned through a PoseNet. Concerning t^{CP}, we as-
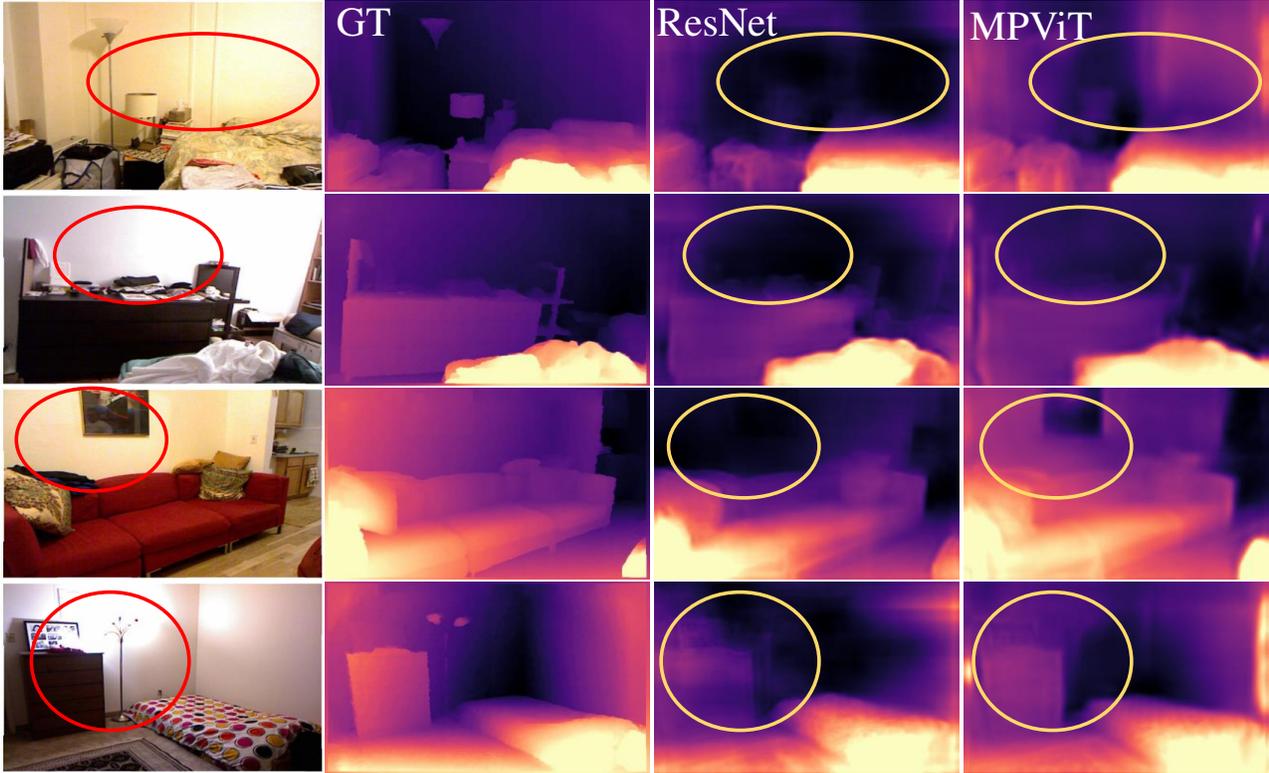
Figure 3. **Effects of different encoders on low-texture depth estimation.** The transformer-based encoder, MPViT, can effectively improve the depth inferring on low-texture regions.

cribe two main factors preventing it from making the learning process work properly: scale and shift components.

Scale Factor: since the training set contains hundreds of monocular sequences, the scale ambiguity and scale drift caused by monocular SfM result in scale inconsistency of poses between different training samples, which affects the depth network during training as well. In the table, we report the standard deviation of the scale alignment factor on the test set (scale std), to reflect how depth scale is affected. We can notice how using COLMAP coarse poses (b) greatly increases the standard deviation, pointing out scale inconsistency across the scenes. By means of estimating a rescaling factor s^{NN} (e), we can significantly reduce it, making it even lower with respect to the baseline and, consequently, improving the accuracy of the estimated depth map.

Shift Factor: although poses are generated by COLMAP, they can be noisy and inaccurate. Therefore, we use a neural network to regress the residual translation Δt^{NN} to be applied to t^{CP}. Estimating this alone (f) results ineffective because of the scale ambiguity issue discussed so far. However, when both scale and shift are estimated (g), we achieve the best results in terms of depth accuracy.

Methods	lower is better			higher is better		
	Abs Rel↓	RMSE↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
MonoUncertainty [34]	0.111	4.756	0.188	0.881	0.961	0.982
HR-Depth [30]	0.109	4.632	0.185	0.884	0.962	0.983
CADepth [50]	0.105	4.535	0.181	0.892	0.964	0.983
DIFFNet [61]	0.102	4.445	0.179	0.897	0.965	0.983
DevNet [63]	0.100	4.412	0.174	0.893	0.966	**0.985**
BRNet [16]	0.105	4.462	0.179	0.890	0.965	0.984
SC-Depthv2 [2]	0.118	4.803	0.193	0.866	0.958	0.981
Monodepth2 [11]	0.115	4.863	0.193	0.877	0.959	0.981
+ SD	0.112	4.814	0.191	0.879	0.860	0.981
+ ISD	0.111	4.732	0.189	0.880	0.961	0.982
MonoViT [59]	0.099	4.372	0.175	0.900	0.967	0.984
+ SD	0.099	4.359	0.174	0.902	0.967	0.984
+ ISD	**0.098**	**4.303**	**0.173**	**0.903**	**0.968**	0.984

Table 2. **Testing our ISD with different baselines on the outdoor KITTI Dataset [10]**; The depth network is trained by outdoor monocular image sequences in a self-supervised manner.

4.2.2 From Monodepth2 to GasMono

Having assessed the impact of coarse poses and how to use them effectively, we now measure the improvements yielded by any single component differentiating GasMono from Monodepth2, by sequentially adding one component at a time. Results are collected in the bottom part of Tab. 1.

Transformer Encoder. One of the main challenges in

Methods	Supervision	Error Metric		Accuracy Metric		
		Abs Rel↓	RMSE↓	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$
Make3D [40]	Sup.	0.349	1.214	44.7	74.5	89.7
Li *et al.* [26]	Sup.	0.143	0.635	78.8	95.8	99.1
Eigen *et al.* [6]	Sup.	0.158	0.641	76.9	95.0	98.8
Laina *et al.* [23]	Sup.	0.127	0.573	81.1	95.3	98.8
DORN [9]	Sup.	0.115	0.509	82.8	86.5	99.2
AdaBins [1]	Sup.	**0.103**	0.364	90.3	98.4	99.7
DPT [38]	Sup.	0.110	**0.357**	90.4	98.8	99.8
Zhou *et al.* [62]	Self-Sup.	0.208	0.712	67.4	90.0	96.8
Zhao *et al.* [60]	Self-Sup.	0.189	0.686	70.1	91.2	97.8
SC-Depthv1 [3]	Self-Sup.	0.157	0.593	78.0	94.0	98.4
P²Net+PP [55]	Self-Sup.	0.147	0.553	80.4	95.2	98.7
StructDepth [25]	Self-Sup.	0.142	0.540	81.3	95.4	98.8
MonoIndoor [20]	Self-Sup.	0.134	0.526	82.3	95.8	98.9
SC-Depthv2 [2]	Self-Sup.	0.138	0.532	82.0	95.6	98.9
MonoIndoor++ [27]	Self-Sup.	0.132	0.517	83.4	96.1	99.0
DistDepth [49]	Self-Sup.	0.130	0.517	83.2	96.3	99.0
GasMono (Ours)	Self-Sup. + Pose	**0.113**	**0.459**	**87.1**	**97.3**	**99.2**

Table 3. **Evaluation on NYUv2 [32].** Sup.: trained with ground truth; Self-Sup.: trained on image sequences; Pose: uses COLMAP poses at training time.

Methods	Supervision	Error Metric		Accuracy Metric		
		Abs Rel↓	RMSE↓	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$
Laina *et al.* [23]	Sup.	0.141	0.339	0.811	.958	0.990
VNL [53]	Sup.	0.123	0.306	0.848	0.964	0.991
DPT [38]	Sup.	**0.089**	**0.220**	**0.917**	**0.985**	**0.997**
TrainFlow [60]	Self-Sup.	0.179	0.415	0.726	0.927	0.980
SC-Depthv1 [3]	Self-Sup.	0.169	0.392	0.749	0.938	0.983
SC-Depthv2 [2]	Self-Sup.	0.156	0.361	0.781	0.947	0.987
Monodepth2 [11]	Self-Sup.	0.170	0.401	0.730	0.948	0.991
MonoIndoor [20]	Self-Sup.	0.154	0.373	0.779	0.951	0.988
MonoIndoor++ [27]	Self-Sup.	0.138	0.347	0.810	0.967	**0.993**
GasMono (Ours)	Self-Sup. + Pose	**0.120**	**0.301**	**0.856**	**0.972**	**0.993**

Table 4. **Zero-shot generalization results on ScanNet [4].** Sup: trained with ground truth, Self-Sup.: trained on image sequences; Pose: uses COLMAP poses at training time.

Scenes	SC-Depthv2 [2]		**GasMono (Ours)**		Monoindoor++* [27]	
	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$
Chess	0.179	0.689	**0.148**	**0.791**	0.157	0.750
Fire	0.163	0.751	**0.131**	**0.844**	0.150	0.768
Heads	0.171	0.746	**0.151**	**0.802**	0.171	0.727
Office	0.146	0.799	**0.112**	**0.878**	0.130	0.837
Pumpkin	0.120	0.841	0.136	0.852	**0.102**	**0.895**
RedKitchen	0.136	0.822	**0.130**	**0.827**	0.144	0.795
Stairs	**0.143**	**0.794**	0.151	0.782	0.155	0.753
Average	0.151	0.778	**0.137**	**0.825**	0.144	0.789

Table 5. **Zero-shot generalization results on on RGB-D 7-Scenes [42].** Note that Monoindoor++* [27] extracts one image every 30 frames in each video sequence for testset, while we follow SC-Depthv2 [2] that extracts the first image every 10 frames.

Scenes	**Fine-tuned**			
	SC-Depthv2 [2]		**GasMono (Ours)**	
	AbsRel↓	$\delta_1\uparrow$	AbsRel↓	$\delta_1\uparrow$
Chess	0.150	0.780	**0.124**	**0.867**
Fire	0.105	0.918	**0.090**	**0.928**
Heads	0.143	0.833	**0.111**	**0.887**
Office	0.128	0.855	**0.102**	**0.914**
Pumpkin	**0.097**	**0.922**	0.110	0.908
RedKitchen	0.124	0.853	**0.111**	**0.894**
Stairs	0.134	0.823	**0.114**	**0.846**
Average	0.126	0.854	**0.108**	**0.892**

Table 6. **Fine-tuned results on on RGB-D 7-Scenes [42].**

indoor environments concerns the lack of texture and the global receptive field of a transformer can help deal with it. By replacing the ResNet18 encoder in Monodepth2 (h) with a state-of-the-art vision transformer (i) – MPViT [24] – allows for much more accurate depth predictions. Fig. 3 shows some qualitative examples: the ResNet-based depth network cannot infer a reasonable disparity for large regions having low texture, whereas the transformer-based depth network benefits from the long-range feature and global feature modeling and yields much better predictions. More ablation studies on the transformer encoder – and the depth decoder – are reported in the supplementary material.

Geometry-Aided Pose Estimation. We now introduce coarse poses estimated by COLMAP. As previously discussed, using such poses seamlessly (j) leads to instability during training, while optimizing the translation component (k) greatly improves the results over (i). However, the rotation component estimated by COLMAP is still sub-optimal, preventing GasMono from unleashing its full potential. Indeed, by both optimizing translation and rotation compo-

nents (l) allows to further reduce the errors.

Iterative Self-Distillation. To conclude, we verify the impact of ISD. By simply enabling it (m), the accuracy of our architecture starts improving. Moreover, it results effective also when combined with translation optimization (n). Eventually, enabling ISD to supervise GasMono while optimizing both translation and rotation components (o) yields the overall best results. To further validate the effectiveness of ISD, we run additional experiments on the KITTI dataset. Tab. 2 shows results obtained by plugging distillation strategies – SD [33] and our ISD – into Monodepth2 and MonoViT. We can notice how both frameworks are improved, with our method resulting more effective than SD.

4.3. Comparison with State-of-the-art

We conclude our experiments by evaluating GasMono with respect to state-of-the-art approaches.

In-Domain Evaluation. Tab. 3 collects results concerning training and testing on the NYUv2 splits. Gas-Mono outperforms existing self-supervised indoor methods [2, 27, 49] by a large margin, achieving the best results across all metrics. Especially, compared with our baseline Monodepth2 [11], our method improves the AbsRel error from 0.167 to 0.113 and δ_1 from 75.4% to 87.1%. Moreover, GasMono outperforms the previous top-performing method DistDepth [49], by 3.9% on δ_1, despite this latter is first trained on SimSIN [49] and then fine-tuned on NYUv2,
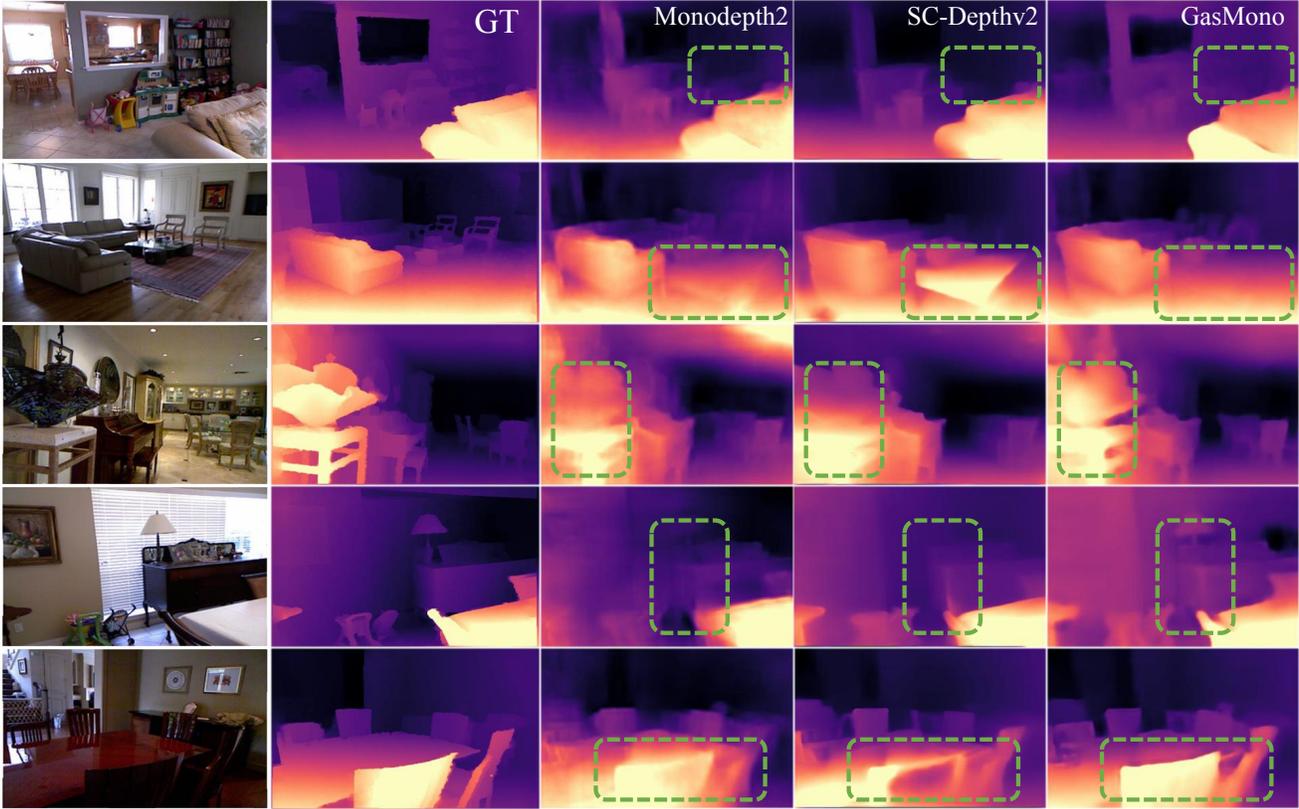
Figure 4. **Qualitative comparison on NYUv2 [32].** GasMono outperforms the baseline [11] and the recent SC-Depthv2 [2].

thus exploiting much more training data. This allows Gas-Mono for halving the gap between the best self-supervised method and the best supervised one – DistDepth vs DPT – on δ_1, reducing it from 6% to 3%.

Generalization. We further test GasMono on ScanNet and 7scenes datasets without retraining. Tabs. 4 and 5 collect the outcome of these experiments. On ScanNet, Gas-Mono consistently outperforms any existing self-supervised and supervised framework. On 7scenes, it achieves the best results on 5 out of 7 sets, being the absolute best on average.

For the sake of completeness, we also report the results achieved by fine-tuning GasMono on the 7-scenes dataset. Specifically, we follow [2] and train for 3 epochs an instance of GasMono for each scene alone, starting from weights pre-trained on NYUv2 dataset [32]. Table 6 shows that our method outperforms SC-Depthv2 [2], highlighting that a short fine-tuning process can already improve the performance of our model significantly, with important implications in terms of deployment in real-world applications.

Qualitative Results. Fig. 4 shows a qualitative comparison between existing approaches and GasMono. Depth maps predicted by our framework (last column) are more precise, especially in the flat regions and fine-grained de-

tails. More examples in the supplementary material.

5. Conclusions

This paper proposed GasMono, a geometry-aided self-supervised monocular depth estimation framework for challenging indoor scenes. With the rescale and refine methods proposed in this paper, our method tackles the problem of using coarse poses in self-supervised training, like scale inconsistency between poses, and makes full use of coarse poses from geometry methods. Therefore, our proposal vastly and consistently outperforms any existing approach on the NYUv2 and KITTI datasets. Moreover, experiments on ScanNet and 7Scenes datasets show that GasMono also achieves superior generalization performance.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021.

[2] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jin Chin, Chunhua Shen, and Ian Reid. Auto-rectify network for unsupervised indoor depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[3] 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.

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL, 2009. IEEE.

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *International Conference on Computer Vision (ICCV)*, 2019.

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[13] Juan Luis Gonzalez and Munchurl Kim. Plade-net: towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6851–6860, 2021.

[14] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33:12626–12637, 2020.

[15] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *CVPR*, 2020.

[16] Wencheng Han, Junbo Yin, Xiaogang Jin, Xiangdong Dai, and Jianbing Shen. Brnet: Exploring comprehensive features for monocular depth estimation. In *European Conference on Computer Vision*, pages 586–602. Springer, 2022.

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, 2016. IEEE.

[19] Mingi Ji, Seungjae Shin, Seunghyun Hwang, Gibeom Park, and Il-Chul Moon. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10664–10673, 2021.

[20] 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.

[21] 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.

[22] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[23] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[24] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2022.

[25] 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.

[26] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *CVPR*, 2017.

[27] Runze Li, Pan Ji, Yi Xu, and Bir Bhanu. Monoindoor++: Towards better practice of self-supervised monocular depth estimation for indoor environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[28] Chenxin Liu, Jiahu Qin, Shuai Wang, Lei Yu, and Yaonan Wang. Accurate rgb-d slam in dynamic environments based on dynamic visual feature removal. *Science China Information Sciences*, 65(10):202206, 2022.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[30] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. *In AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[31] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[32] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[33] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15560–15569, 2021.

[34] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020.

[35] Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2021.

[36] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *6th International Conference on 3D Vision (3DV)*, 2018.

[37] Hong Qiao, Shanlin Zhong, Ziyu Chen, and Hongze Wang. Improving performance of robots using human-inspired approaches: a survey. *Science China Information Sciences*, 65(12):221201, 2022.

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021.

[39] 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.

[40] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[41] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] 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*, pages 2930–2937, 2013.

[43] Qiyu Sun, Yang Tang, Chongzhen Zhang, Chaoqiang Zhao, Feng Qian, and Jürgen Kurths. Unsupervised estimation of monocular depth and vo in dynamic environments via hybrid masks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[44] Yang Tang, Chaoqiang Zhao, Jianrui Wang, Chongzhen Zhang, Qiyu Sun, Wei Xing Zheng, Wenli Du, Feng Qian, and Jürgen Kurths. Perception and navigation in autonomous systems in the era of learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[45] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[46] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[48] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, 2019.

[49] Cho-Ying Wu, Jialiang Wang, Michael Hall, Ulrich Neumann, and Shuochen Su. Toward practical monocular indoor depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3814–3824, 2022.

[50] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 464–473. IEEE, 2021.

[51] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[52] Zhenheng Yang, Peng Wang, Wang Yang, Wei Xu, and Nevatia Ram. Lego: Learning edge with geometry all at once by watching videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[53] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[54] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018.

[55] Zehao Yu, Lei Jin, and Shenghua Gao. P^{2}net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *ECCV*, 2020.

[56] Hongwei Zhang, Wenbo Xiong, Shuai Lu, Mengdan Chen, and Le Yao. Qa-ustnet: Yarn-dyed fabric defect detection via u-shaped swin transformer network based on quadtree attention. *Textile Research Journal*, page 00405175231158134, 2023.

[57] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9):1612–1627, 2020.

[58] Chaoqiang Zhao, Gary G Yen, Qiyu Sun, Chongzhen Zhang, and Yang Tang. Masked gan for unsupervised depth and pose prediction with scale consistency. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5392–5403, 2020.

[59] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022.

[60] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020.

[61] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021.

[62] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *CVPR*, 2019.

[63] Kaichen Zhou, Lanqing Hong, Changhao Chen, Hang Xu, Chaoqiang Ye, Qingyong Hu, and Zhenguo Li. Devnet: Self-supervised monocular depth learning via density volume construction. *European Conference on Computer Vision*, 2022.

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

[65] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.