

Pseudo RGB-D for Self-Improving Monocular SLAM and Depth Prediction

Lokender Tiwari¹[0000-0002-9217-4248], Pan Ji²[0000-0001-6213-554X], Quoc-Huy Tran²[0000-0003-1396-6544], Bingbing Zhuang²[0000-0002-2317-3882], Saket Anand¹[0000-0002-6229-3940], and Manmohan Chandraker^{2,3}[0000-0003-4683-2454]

¹IIT-Delhi ²NEC Labs America ³UCSD

Abstract. Classical monocular Simultaneous Localization And Mapping (SLAM) and the recently emerging convolutional neural networks (CNNs) for monocular depth prediction represent two largely disjoint approaches towards building a 3D map of the surrounding environment. In this paper, we demonstrate that the coupling of these two by leveraging the strengths of each mitigates the others shortcomings. Specifically, we propose a joint narrow and wide baseline based self-improving framework, where on the one hand the CNN-predicted depth is leveraged to perform *pseudo RGB-D* feature-based SLAM, leading to better accuracy and robustness than the monocular RGB SLAM baseline. On the other hand, the bundle-adjusted 3D scene structures and camera poses from the more principled geometric SLAM are injected back into the depth network through novel wide baseline losses proposed for improving the depth prediction network, which then continues to contribute towards better pose and 3D structure estimation in the next iteration. We emphasize that our framework only requires *unlabeled monocular* videos in both training and inference stages, and yet is able to outperform state-of-the-art *self-supervised monocular* and *stereo* depth prediction networks (*e.g.*, Monodepth2) and feature-based monocular SLAM system (*i.e.*, ORB-SLAM). Extensive experiments on KITTI and TUM RGB-D datasets verify the superiority of our self-improving geometry-CNN framework.

Keywords: self-supervised learning, self-improving, monocular depth prediction, monocular SLAM

1 Introduction

One of the most reliable cues towards 3D perception from a monocular camera arises from camera motion that induces multiple-view geometric constraints [20] wherein the 3D scene structure is encoded. Over the years, Simultaneous Localization And Mapping (SLAM) [6, 21, 32] has been long studied to simultaneously recover the 3D scene structure of the surrounding and estimate the ego-motion of the agent. With the advent of Convolutional Neural Networks (CNNs), unsupervised learning of single-view depth estimation [13, 16, 60] has emerged as a promising alternative to the traditional geometric approaches. Such methods rely

on CNNs to extract meaningful depth cues (*e.g.*, shading, texture, and semantics) from a single image, yielding very promising results.

Despite the general maturity of monocular geometric SLAM [10, 30, 9] and the rapid advances in unsupervised monocular depth prediction approaches [29, 44, 57, 2, 15, 38], they both still have their own limitations.

Monocular SLAM. Traditional monocular SLAM has well-known limitations in robustness and accuracy as compared to those leveraging active depth sensors, *e.g.*, RGB-D SLAM [31]. This performance issue is due to the inherent scale ambiguity of depth recovery from monocular cameras, which causes the so-called scale drift in both the camera trajectory and 3D scene depth, and thus lowers robustness and accuracy of conventional monocular SLAM. In addition, the triangulation-based depth estimation employed by traditional SLAM methods is degenerate under pure rotational camera motion [20].

Unsupervised Monocular Depth Prediction. Most of the unsupervised and self-supervised methods [60, 16, 15, 2] formulate single image depth estimation as a novel-view synthesis problem, with appearance based photometric losses being central to the training strategy. Usually, these models train two networks, one each for *pose* and *depth*. As photometric losses largely rely on the brightness consistency assumption, nearly all existing self-supervised approaches operate in a narrow-baseline setting optimizing the loss over a snippet of 2-5 consecutive frames. Consequently, models like MondoDepth2 [15], work very well for close range points, but generate inaccurate depth estimates for points that are farther away (*e.g.*, see 0th iteration in Fig. 6). While it is well known that a wide-baseline yields better depth estimates for points at larger depth, a straightforward extension of existing CNN based approaches is inadequate for the following two reasons. A wide baseline in a video sequence implies a larger temporal window, which in most practical scenarios will violate the brightness consistency assumption, rendering the photometric loss ineffective. Secondly, larger temporal windows (wider baselines) would also imply more occluded regions that behave as outliers. Unless these aspects are effectively handled, training of CNN based depth and pose networks in the wide baseline setting will lead to inaccuracies and biases.

In view of the limitations in both monocular geometric SLAM and unsupervised monocular depth estimation approaches, a particularly interesting question to ask is whether these two approaches can complement each other (see Sec. 5) and mitigate the issues discussed above. Our work makes contributions towards answering this question. Specifically, we propose a *self-supervised, self-improving* framework of these two tasks, which is shown to improve the robustness and accuracy on each of them.

While the performance gap between geometric SLAM and self-supervised learning-based SLAM methods is still large, incorporating depth information drastically improves the robustness of geometric SLAM methods (*e.g.*, see RGB-D SLAM vs. RGB SLAM on the KITTI Odometry leaderboard [14]). Inspired by this success of RGB-D SLAM, we postulate the use of an unsupervised CNN-based depth estimation model as a *pseudo depth sensor*, which allows us to design our self-supervised approach, pseudo RGB-D SLAM (pRGBD-SLAM) that only

uses monocular cameras and yet achieves significant improvements in robustness and accuracy as compared to RGB SLAM.

Our fusion of geometric SLAM and CNN-based monocular depth estimation turns out to be symbiotic and this complementary nature sets the basis of our self-improving framework. To improve the depth predictions, we make use of two main modifications in the training strategy. First, we eschew the learning based pose estimates in favor of geometric SLAM based estimates (an illustrative motivation is shown in Fig. 1). Second, we make use of common tracked keypoints from neighboring *keyframes* and impose a symmetric depth transfer and a depth consistency loss on the CNN model. These adaptations are based on the observation that both pose estimates and sparse 3D feature point estimates from geometric SLAM are robust, as most techniques typically apply multiple bundle adjustment iterations over wide baseline depth estimates of common keypoints. This simple observation and the subsequent modification is key to our self-improving framework, which can leverage any unsupervised CNN-based depth estimation model and a modern monocular SLAM method. In this paper, we test our framework, with ORBSLAM [31] as the geometric SLAM method and MonoDepth2 [15] as the CNN-based model. We show that our self-improving framework outperforms previously proposed self-supervised approaches that utilizes monocular, stereo, and monocular-plus-stereo cues for self-supervision (see Tab. 1) and a strong feature based RGB-SLAM baseline (see Tab. 5). The framework runs in a simple alternating update fashion: first, we use depth maps from the CNN-based depth network and run pRGBD-SLAM; second, we inject the outputs of pRGBD-SLAM, *i.e.*, the relative camera poses and common tracked keypoints and keyframes to fine-tune the depth network parameters to improve the depth prediction; then, we repeat the process until we see no improvement. Our specific contributions are summarized here:

- We propose a self-improving strategy to inject into depth prediction networks the supervision from SLAM outputs, which stem from more generally applicable geometric principles.
- We introduce two wide baseline losses, *i.e.*, the symmetric depth transfer loss and the depth consistency loss on common tracked points, and propose a joint narrow and wide baseline based depth prediction learning setup, where appearance based losses are computed on narrow baselines and purely geometric losses on wide baselines (non-consecutive temporally distant keyframes).
- Through extensive experiments on KITTI [14] and TUM RGB-D [40], our framework is shown to outperform both monocular SLAM system (*i.e.*, ORBSLAM [30]) and the state-of-the-art unsupervised single-view depth prediction network (*i.e.*, Monodepth2 [15]).

2 Related Work

Monocular SLAM. Visual SLAM has a long history of research in the computer vision community. Due to its well-understood underlying geometry, various geometric approaches have been proposed in the literature, ranging from the

classical MonoSLAM [6], PTAM [21], DTAM [32] to the more recent LSD-SLAM [10], ORB-SLAM [30] and DSO [9]. More recently, in view of the successful application of deep learning in a wide variety of areas, researchers have also started to exploit deep learning approaches for SLAM, in the hope that it can improve certain components of geometric approaches or even serve as a complete alternative. Our work makes further contributions along this line of research.

Monocular Depth Prediction. Inspired by the pioneering work by Eigen et al. [8] on learning single-view depth estimation, a vast amount of learning methods [3, 26, 12] emerge along this line of research. The earlier works often require ground truth depths for fully-supervised training. However, per-pixel depth ground truth is generally hard or prohibitively costly to obtain. Therefore, many self-supervised methods that make use of geometric constraints as supervision signals are proposed. Godard et al. [16], relies on the photo-consistency between the left-right cameras of a calibrated stereo. Zhou et al. [60] learn monocular depth prediction as well as ego-motion estimation, thereby permitting unsupervised learning with only a monocular camera. This pipeline has inspired a large amount of follow-up works that utilize various additional heuristics, including 3D geometric constraints on point clouds [29], direct visual odometry [44], joint learning with optical flow [57], scale consistency [2], and others [15, 38, 5, 63].

Using Depth to Improve Monocular SLAM. Approaches [41, 56, 53, 27] leveraging CNN-based depth estimates to tackle issues in monocular SLAM have been proposed. CNN-SLAM [41] uses learned depth maps to initialize keyframes' depth maps in LSD-SLAM [10] and refines them via a filtering framework. Yin et al. [56] use a combination of CNNs and conditional random fields to recover scale from the depth predictions and iteratively refine ego-motion and depth estimates. DVSO [53] trains a single CNN to predict both the left and right disparity maps, forming a virtual stereo pair. The CNN is trained with photo-consistency between stereo images and consistency with depths estimated by Stereo DSO [46]. More recently, CNN-SVO [27] uses depths learned from stereo images to initialize depths of keypoints and reduce their corresponding uncertainties in SVO [11]. In contrast to our self-supervised approach, [41, 56] use *ground truth* depths for training depth networks while [53, 27] need *stereo* images.

Using SLAM to Improve Monocular Depth Prediction. Depth estimates from geometric SLAM have been leveraged for training monocular depth estimation networks in recent works [22, 1]. In [1], sparse depth maps by Stereo ORB-SLAM [31] are first converted into dense ones via an auto-encoder, which are then integrated into geometric constraints for training the depth network. [22] employ depths and poses by ORB-SLAM [30] as supervision signals for training the depth and pose networks respectively. This approach only considers five consecutive frames, thus restricting its operation in the narrow-baseline setting.

3 Method: A Self-Improving Framework

Our self-improving framework leverages the strengths of each, the unsupervised single-image depth estimation and the geometric SLAM approaches, to mitigate

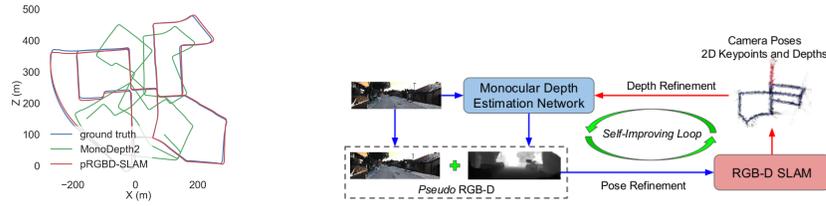


Fig. 1. MonoDepth2 [15] pose network camera poses vs. pseudo RGBD-SLAM. **Fig. 2. Overview of Our Self-Improving Framework.** It alternates between pose re-camera poses where depth(D) is from finement (blue arrows; Sec. 3.1) and depth CNN. The pose network from [15] leads refinement (red arrows; Sec. 3.2) to significant drift.

the other’s shortcomings. On one hand, the depth network typically generates reliable depth estimates for nearby points, which assist in improving the geometric SLAM estimates of poses and sparse 3D points (Sec. 3.1). On the other hand, geometric SLAM methods rely on a more holistic view of the scene to generate robust pose estimates as well as identify *persistent* 3D points that are visible across many frames, thus providing an opportunity to perform wide-baseline and reliable sparse depth estimation. Our framework leverages these sparse, but robust estimates to improve the noisier depth estimates of the farther scene points by minimizing a blend of the symmetric transfer and depth consistency losses (Sec. 3.2) and the commonly used appearance based loss. In the following iteration, this improved depth estimate further enhances the capability of geometric SLAM and the cycle continues until the improvements become negligible. Even in the absence of ground truth, our self-improving framework continues to produce better pose and depth estimates.

An overview of the proposed self-improving framework is shown in in Fig. 2, which iterates between improving poses and improving depths. Our pose refinement and depth refinement steps are then detailed in Sec. 3.1 and 3.2 respectively. An overview of narrow and wide baseline losses we use for improving the depth network is shown in Fig. 3 and details are provided in Sec. 3.2.

3.1 Pose Refinement

Pseudo RGB-D for Improving Monocular SLAM. We employ a well explored and widely used geometry-based SLAM system, i.e., the RGB-D version of ORB-SLAM [31], to process the pseudo RGB-D data, yielding camera poses as well as 3D map points and the associated 2D keypoints. Any other geometric SLAM system that provides these output estimates can also be used in place of ORB-SLAM. A trivial direct use of pseudo RGB-D data to run RGB-D ORB-SLAM is not possible, because CNN might predict depth at a very different scale compared to depth measurements from real active sensors, *e.g.*, LiDAR. Keeping the above difference in mind, we discuss an important adaptation in order for RGB-D ORB-SLAM to work well in our setting. We first note that RGB-D

ORB-SLAM transforms the depth data into disparity on a virtual stereo to reuse the framework of stereo ORB-SLAM. Specifically, considering a keypoint with 2D coordinates (u_l, v_l) (*i.e.*, u_l and v_l denote the horizontal and vertical coordinates respectively) and a CNN-predicted depth d_l , the corresponding 2D keypoint coordinates (u_r, v_r) on the virtual rectified right view are $u_r = u_l - \frac{f_x b}{d_l}$, $v_r = v_l$, where f_x is the horizontal focal length and b is the virtual stereo baseline.

Adaptation. In order to have a reasonable range of disparity, we mimic the setup of the KITTI dataset [14] by making the baseline adaptive, $b = \frac{b^{\text{KITTI}}}{d_{max}^{\text{KITTI}}} * d_{max}$, where d_{max} represents the maximum CNN-predicted depth of the input sequence, and $b^{\text{KITTI}} = 0.54$ and $d_{max}^{\text{KITTI}} = 80$ (both in meters) are respectively the actual stereo baseline and empirical maximum depth value of the KITTI dataset.

We also summarize the overall pipeline of RGB-D ORB-SLAM here. The 3D map is initialized at the very first frame of the sequence due to the availability of depth. After that, the following main tasks are performed: i) track the camera by matching 2D keypoints against the local map, ii) enhance the local map via local bundle adjustment, and iii) detect and close loops for pose-graph optimization and full bundle adjustment to improve camera poses and scene depths. As we will show in Sec. 4.4, using pseudo RGB-D data leads to better robustness and accuracy as compared to using only RGB data.

3.2 Depth Refinement

We start from the pre-trained depth network of Monodepth2 [15], a state-of-the-art monocular depth estimation network, and fine-tune its network parameters with the camera poses, 3D map points and the associated 2D keypoints produced by the above pseudo RGB-D ORB-SLAM (pRGBD-SLAM). In contrast to Monodepth2, which relies only on the narrow baseline photometric reconstruction loss between adjacent frames for short-term consistencies, we propose wide baseline symmetric depth transfer and sparse depth consistency losses to introduce long-term consistencies. Our final loss (Eq. (4)) consists of both narrow and wide baseline losses. The narrow baseline losses, *i.e.*, photometric and smoothness losses, involve the current keyframe \mathcal{I}_c and its temporally adjacent frames \mathcal{I}_{c-1} and \mathcal{I}_{c+1} , while wide baseline losses are computed on the current keyframe \mathcal{I}_c and the two neighboring keyframes \mathcal{I}_{k1} and \mathcal{I}_{k2} that are temporally farther than \mathcal{I}_{c-1} and \mathcal{I}_{c+1} (see Fig. 3). Next, we introduce the notation and describe the losses.

Notation. Let \mathcal{X} represent the set of common tracked keypoints visible in all the three keyframes \mathcal{I}_{k1} , \mathcal{I}_c and \mathcal{I}_{k2} obtained from pRGBD-SLAM. Note that $k1$ and $k2$ are two neighboring keyframes of the current frame c (*i.e.*, $k1 < c < k2$) in which keypoints are visible. Let $\mathbf{p}_{k1}^i = [p_{k1}^{i1}, p_{k1}^{i2}]$, $\mathbf{p}_c^i = [p_c^{i1}, p_c^{i2}]$ and $\mathbf{p}_{k2}^i = [p_{k2}^{i1}, p_{k2}^{i2}]$ be the 2D coordinates of the i^{th} common tracked keypoint in the keyframes \mathcal{I}_{k1} , \mathcal{I}_c and \mathcal{I}_{k2} respectively, and the associated depth values obtained from pRGBD-SLAM are represented by $d_{k1}^i(\text{SLAM})$, $d_c^i(\text{SLAM})$, and $d_{k2}^i(\text{SLAM})$ respectively. The depth values corresponding to the keypoints \mathbf{p}_{k1}^i , \mathbf{p}_c^i and \mathbf{p}_{k2}^i can also be obtained from the depth network and are represented by $d_{k1}^i(\mathbf{w})$, $d_c^i(\mathbf{w})$, and $d_{k2}^i(\mathbf{w})$ respectively, where \mathbf{w} stands for the depth network parameters.

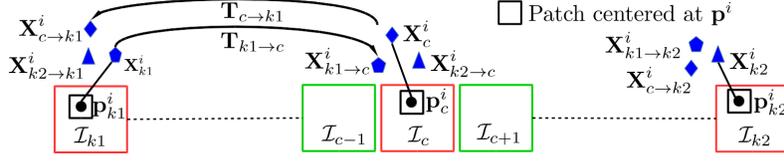


Fig. 3. Narrow and Wide Baseline Losses. Narrow baseline photometric and smoothness losses involve keyframe \mathcal{I}_c and temporally *adjacent* frames \mathcal{I}_{c-1} and \mathcal{I}_{c+1} , and wide baseline symmetric depth transfer and depth consistency losses involve keyframe \mathcal{I}_c and temporally *farther* keyframes \mathcal{I}_{k1} and \mathcal{I}_{k2} . Refer to the text below.

Symmetric Depth Transfer Loss. Given the camera intrinsic matrix \mathbf{K} , and the depth value $d_c^i(\mathbf{w})$ of the i^{th} keypoint \mathbf{p}_c^i , the 2D coordinates of the keypoint \mathbf{p}_c^i can be back-projected to its corresponding 3D coordinates as: $\mathbf{X}_c^i(\mathbf{w}) = \mathbf{K}^{-1}[\mathbf{p}_c^i, 1]^T d_c^i(\mathbf{w})$. Let $\mathbf{T}_{c \rightarrow k1}^{\text{SLAM}}$ represent the relative camera pose of frame $k1$ w.r.t. frame c obtained from pRGBD-SLAM. Using $\mathbf{T}_{c \rightarrow k1}^{\text{SLAM}}$, we can transfer the 3D point $\mathbf{X}_c^i(\mathbf{w})$ from frame c to $k1$ as: $\mathbf{X}_{c \rightarrow k1}^i(\mathbf{w}) = \mathbf{T}_{c \rightarrow k1}^{\text{SLAM}} \mathbf{X}_c^i(\mathbf{w}) = [x_{c \rightarrow k1}^i(\mathbf{w}), y_{c \rightarrow k1}^i(\mathbf{w}), d_{c \rightarrow k1}^i(\mathbf{w})]^T$. Here, $d_{c \rightarrow k1}^i(\mathbf{w})$ is the transferred depth of the i^{th} keypoint from frame c to frame $k1$. Following the above procedure, we can obtain the transferred depth $d_{k1 \rightarrow c}^i(\mathbf{w})$ of the same i^{th} keypoint from frame $k1$ to frame c . The symmetric depth transfer loss of the keypoint \mathbf{p}_c^i between frame pair c and $k1$, is the sum of absolute errors (ℓ_1 distance) between the transferred network-predicted depth $d_{c \rightarrow k1}^i(\mathbf{w})$ and the existing network-predicted depth $d_{k1}^i(\mathbf{w})$ in the target keyframe $k1$, and vice-versa. It can be written as:

$$\mathcal{T}_{c \leftrightarrow k1}^i(\mathbf{w}) = |d_{c \rightarrow k1}^i(\mathbf{w}) - d_{k1}^i(\mathbf{w})| + |d_{k1 \rightarrow c}^i(\mathbf{w}) - d_c^i(\mathbf{w})|. \quad (1)$$

Similarly, we can compute the symmetric depth transfer loss of the same i^{th} keypoint between frame pair c and $k2$, *i.e.*, $\mathcal{T}_{c \leftrightarrow k2}^i(\mathbf{w})$, and between $k1$ and $k2$, *i.e.*, $\mathcal{T}_{k1 \leftrightarrow k2}^i(\mathbf{w})$. We accumulate the total symmetric transfer loss between frame c and $k1$ in $\mathcal{T}_{c \leftrightarrow k1}$, which is the loss of all the common tracked keypoints and the points within the patch of size 5×5 centered at the common tracked keypoints. Similarly, we compute the total symmetric depth transfer loss $\mathcal{T}_{c \leftrightarrow k2}$ and $\mathcal{T}_{k1 \leftrightarrow k2}$ between frame pair $(c, k2)$, and $(k1, k2)$ respectively.

Depth Consistency Loss. The role of the depth consistency loss is to make depth network’s prediction consistent with the refined depth values obtained from the pRGBD-SLAM. Note that depth values from pRGBD-SLAM undergo multiple optimization over wide baselines, hence are more accurate and capture long-term consistencies. We inject these long-term consistent depths from pRGBD-SLAM to depth network through the depth consistency loss. The loss for the frame c can be written as follows:

$$\mathcal{D}_c = \frac{\sum_{i \in \mathcal{X}} |d_c^i(\mathbf{w}) - d_c^i(\text{SLAM})|}{|\mathcal{X}|}. \quad (2)$$

Photometric Reconstruction Loss. Denote the relative camera pose of frame \mathcal{I}_{c-1} and \mathcal{I}_{c+1} w.r.t. current keyframe \mathcal{I}_c obtained from pRGBD-SLAM by

$\mathbf{T}_{c-1 \rightarrow c}^{\text{SLAM}}$ and $\mathbf{T}_{c+1 \rightarrow c}^{\text{SLAM}}$ respectively. Using frame \mathcal{I}_{c+1} , $\mathbf{T}_{c+1 \rightarrow c}^{\text{SLAM}}$, network-predicted depth map $d_c(\mathbf{w})$ of the keyframe \mathcal{I}_c , and the camera intrinsic \mathbf{K} , we can synthesize the current frame \mathcal{I}_c [15, 16]. Let the synthesized frame be represented in the functional form as: $\mathcal{I}_{c+1 \rightarrow c}(d_c(\mathbf{w}), \mathbf{T}_{c+1 \rightarrow c}^{\text{SLAM}}, \mathbf{K})$. Similarly we can synthesize $\mathcal{I}_{c-1 \rightarrow c}(d_c(\mathbf{w}), \mathbf{T}_{c-1 \rightarrow c}^{\text{SLAM}}, \mathbf{K})$ using frame \mathcal{I}_{c-1} . The photometric reconstruction error between the synthesized and the original current frame [13, 16, 60] is then computed as:

$$\mathcal{P}_c = pe(\mathcal{I}_{c+1 \rightarrow c}(d_c(\mathbf{w}), \mathbf{T}_{c+1 \rightarrow c}^{\text{SLAM}}, \mathbf{K}), \mathcal{I}_c) + pe(\mathcal{I}_{c-1 \rightarrow c}(d_c(\mathbf{w}), \mathbf{T}_{c-1 \rightarrow c}^{\text{SLAM}}, \mathbf{K}), \mathcal{I}_c), \quad (3)$$

where we follow [16, 15] to construct the photometric reconstruction error function $pe(\cdot, \cdot)$. Additionally, we adopt the more robust per-pixel minimum error, multi-scale strategy, auto-masking, and depth smoothness loss \mathcal{S}_c from [15].

Our final loss for fine-tuning the depth network at the depth refinement step is the weighted sum of narrow baseline losses (*i.e.*, photometric (\mathcal{P}_c) and smoothness loss (\mathcal{S}_c)), and wide baseline losses (*i.e.*, symmetric depth transfer ($\mathcal{T}_{c \leftrightarrow k1}$, $\mathcal{T}_{c \leftrightarrow k2}$, $\mathcal{T}_{k1 \leftrightarrow k2}$) and depth consistency loss (\mathcal{D}_c)):

$$\mathcal{L} = \alpha \mathcal{P}_c + \beta \mathcal{S}_c + \gamma \mathcal{D}_c + \mu(\mathcal{T}_{c \leftrightarrow k1} + \mathcal{T}_{c \leftrightarrow k2} + \mathcal{T}_{k1 \leftrightarrow k2}). \quad (4)$$

4 Experiments

We conduct experiments to evaluate depth refinement and pose refinement steps of our self-improving framework with the state-of-the-arts in self-supervised depth estimation and RGB-SLAM based pose estimation respectively.

4.1 Datasets and Evaluation Metrics

KITTI Dataset. Our experiments are mostly performed on the KITTI dataset [14], which contains outdoor driving sequences for road scene understanding [39, 7]. We further split KITTI experiments into two parts: one focused on depth refinement evaluation and the other on pose refinement. For depth refinement evaluation we train/fine-tune the depth network using the Eigen train split [8] which contains 28 training sequences and evaluate depth prediction on the Eigen test split [8] following the baselines [60, 55, 29, 57, 44, 64, 54, 35, 28, 4]. For pose refinement evaluation, we train/fine-tune the depth network using KITTI odometry sequences 00-08 and test on sequences 09-10 and 11-21. Note, for evaluation on sequences 09-10 we use the ground-truth trajectories provided by [14], while for evaluation on sequences 11-21, since the ground-truth is not available we use the pseudo ground-truth trajectories obtained by running stereo version of ORB-SLAM on these sequences.

TUM RGB-D Dataset. For completeness and to demonstrate the capability of our self-improving framework on indoor scenes, we evaluate on the TUM RGB-D dataset [40], which consists of indoor sequences captured by a hand-held camera. We choose *freiburg3* sequences because only they have *undistorted* RGB images

and ground truth available to train/fine-tune and evaluate respectively. We use 6 of 8 *freiburg3* sequences for training/fine-tune and the remaining 2 for evaluation. **Metrics for Pose Evaluation.** For quantitative pose evaluation, we compute the Root Mean Square Error (*RMSE*), Relative Translation (*Rel Tr*) error, and Relative Rotation (*Rel Rot*) error of the predicted camera trajectory. Since monocular SLAM systems can only recover camera poses up to a global scale, we align the camera trajectory estimated by each method with the ground truth one using the EVO toolbox [18]. We then use the official evaluation code from the KITTI Odometry benchmark to compute the *Rel Tr* and *Rel Rot* errors for all sub-trajectories with length in $\{100, \dots, 800\}$ meters.

Metrics for Depth Evaluation. For quantitative depth evaluation, we use the standard metrics, including the Absolute Relative (*Abs Rel*) error, Squared Relative (*Sq Rel*) error, *RMSE*, *RMSE log*, $\delta < 1.25$ (namely *a1*), $\delta < 1.25^2$ (namely *a2*), and $\delta < 1.25^3$ (namely *a3*) as defined in [8]. Again, since the depths from monocular images can only be estimated up to scale, we align the predicted depth map with the ground truth one using their median depth values. Following [8] and other baselines, we also clip the depths to 80 meters.

Note. In all tables, best performances are in **bold** and second bests are underlined.

4.2 Implementation Details

We implement our framework based on Monodepth2 [15] and ORB-SLAM [31], *i.e.*, we use the depth network of Monodepth2 and the RGB-D version of ORB-SLAM for depth refinement and pose refinement respectively. We would like to emphasize, that our self-improving strategy is not specific to MonoDepth2 or ORB-SLAM. Any other depth network that allows to incorporate SLAM outputs and any SLAM system that can provide the desired SLAM outputs can be put into the self-improving framework. We set the weight of the smoothness loss term of the final loss (Eq. (4)) $\beta = 0.001$ similar as in [15] and α, γ , and μ to 1. The ablation study results on disabling different loss terms can be found in Tab. 3. A single self-improving loop takes 0.6 hour on a NVIDIA TITAN Xp 8GB GPU.

KITTI Eigen Split/Odometry Experiments. We pre-train MonoDepth2 using monocular videos of the KITTI Eigen split training set with the hyperparameters as suggested in MonoDepth2 [15]. We use an input/output resolution of 640×192 for training/fine-tuning and scale it up to the original resolution while running pRGBD-SLAM. We use same hyperparameters as for KITTI Eigen split to train/fine-tune the depth model on KITTI Odometry train sequences mentioned in Sec. 4.1. During a self-improving loop, we *discard* pose network of MonoDepth2 and instead use camera poses from pRGBD-SLAM.

Outlier Removal. Before running a depth refinement step, we run an outlier removal step on the SLAM outputs. Specifically, we filter out outlier 3D map points and the associated 2D keypoints that satisfy at least one of the following conditions: i) it is observed in less than 3 keyframes, ii) its reprojection error in the current keyframe \mathcal{I}_c is larger than 3 pixels.

Camera Intrinsic. Monodepth2 computes the average camera intrinsics for the KITTI dataset and uses it for the training. However, for our fine-tuning of the

depth network, using the average camera intrinsics leads to inferior performance, because we use the camera poses from pRGBD-SLAM, which runs with different camera intrinsics. Therefore, we use different camera intrinsics for different sequences when fine-tuning the depth network.

For fine-tuning the depth network pre-trained on KITTI Eigen split training sequences, we run pRGBD-SLAM on all the training sequences, and extract camera poses, 2D keypoints and the associated depths from keyframes. For pRGBD-SLAM(RGB-D ORB-SLAM), we use the default setting of ORB-SLAM, except for the adjusted b described in Sec. 3.1. The same above procedure is followed for depth model pre-trained on KITTI Odometry training sequences. The average number of keyframes used in a self-improving loop is $\sim 9K$ and $\sim 10K$ for KITTI Eigen split and KITTI Odometry experiments respectively. At each depth refinement step, we fine-tune the depth network parameters with 1 epoch only, using learning rate 1e-6, keeping all the other hyperparameters the same as pre-training. For both KITTI Eigen split and KITTI Odometry experiments we report results after 5 *self-improving loops*.

TUM RGB-D Experiments. For TUM RGB-D, we pre-train/fine-tune the depth network on 6 *freiburg3* sequences, and test on 2 *freiburg3* sequences. The average number of keyframes in a self-improving loop is $\sim 3.5K$. We use an input/output resolution of 480×320 for pre-training/fine-tuning and scale it up to the original resolution while running pRGBD-SLAM. We report results after 3 *self-improving loops*. Other details can be found in the supplementary material.

4.3 Monocular Depth/Depth Refinement Evaluation

In the following, we evaluate the performance of our depth estimation on the KITTI Raw Eigen split test set and TUM RGB-D *frieburg3* sequences.

Results on KITTI Eigen Split Test Set. We show the depth evaluation results on the Eigen split test set in Tab. 1. From the table, it is evident that our refined depth model (pRGBD-Refined) outperforms all the competing monocular (M) unsupervised methods by non-trivial margins, including MonoDepth2-M re-trained depth model, and even surpasses the unsupervised methods with stereo (S) training, *i.e.*, Monodepth2-S, and combined monocular-stereo (MS) training, *i.e.*, MonoDepth2-MS, in most metrics. Our method also outperforms several ground-truth depth supervised methods [8, 26]. The reason is probably that the aggregated cues from multiple views with wide baseline losses (*e.g.*, our symmetric depth transfer, depth consistency losses) lead to more well-posed depth recovery, and hence even higher accuracy than learning with the pre-calibrated stereo rig with smaller baselines. Further analysis is provided in Sec. 5. Fig. 4 shows some qualitative results, where our method (pRGBD-Refined) shows visible improvements. Refer supplementary material for more qualitative results.

Results on TUM RGB-D Sequences. The depth evaluation results on the two TUM *frieburg3* RGB-D sequences is shown in Tab. 2. Our refined depth model (pRGBD-Refined) outperforms pRGBD-Initial/Monodepth2-M in both sequences and all metrics. Refer supplementary material for qualitative results.

Table 1. Depth evaluation result on KITTI Eigen split test set. M: self-supervised monocular supervision, and S: self-supervised stereo supervision, D: depth supervision. ‘-’ means the result is not available from the paper. pRGBD-Refined outperforms all the self-supervised monocular methods and several stereo only and combined monocular and stereo methods. Our results are after 5 *self-improving loops*.

	Method	Train	Lower is better				Higher is better		
			Abs Rel	Sq Rel	RMSE	RMSE log	a1	a2	a3
self-supervised	Yang[55]	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
	Mahjourian[29]	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	Klodt[22]	M	0.166	1.490	5.998	-	0.778	0.919	0.966
	DDVO[44]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
	GeoNet[57]	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
	DF-Net[64]	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
	Ranjan[35]	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
	EPC++[28]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
	Struct2depth(M)[4]	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
	WBAF [59]	M	0.135	0.992	5.288	0.211	0.831	0.942	0.976
	MonoDepth2-M (re-train) [15]	M	0.117	0.941	4.889	0.194	0.873	0.957	0.980
	MonoDepth2-M (original) [15]	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	pRGBD-Refined	M	0.113	0.793	4.655	0.188	0.874	0.960	0.983
	Garg[13]	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
	3Net (R50)[34]	S	0.129	0.996	5.281	0.223	0.831	0.939	0.974
	Monodepth2-S[15]	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
	SuperDepth [33]	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
	monoResMatch [43]	S	0.111	0.867	4.714	0.199	0.864	0.954	0.979
	DepthHints [49]	S	0.106	0.780	4.695	0.193	0.875	0.958	0.980
	DVSO[53]	S	0.097	0.734	4.442	0.187	0.888	0.958	0.980
	UnDeepVO [24]	MS	0.183	1.730	6.570	0.268	-	-	-
	EPC++ [28]	MS	0.128	0.935	5.011	0.209	0.831	0.945	0.979
	Monodepth2-MS[15]	MS	0.106	0.818	4.750	0.196	0.874	0.957	0.979
	Eigen[8]	D	0.203	1.548	6.307	0.282	0.702	0.890	0.890
	Liu[26]	D	0.201	1.584	6.471	0.273	0.680	0.898	0.967
	Kuznetsov[23]	DS	0.113	0.741	4.621	0.189	0.862	0.960	0.986
	SVSM FT[28]	DS	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Guo[19]	DS	0.096	0.641	4.095	0.168	0.892	0.967	0.986	
DORN[12]	D	0.072	0.307	2.727	0.120	0.932	0.984	0.994	

Method	TUM RGBD Sequences						
	Lower is better				Higher is better		
	AbRel	SqRel	RMSE	RMSElog	a1	a2	a3
pRGBD-I	0.397	0.848	1.090	0.719	0.483	0.722	0.862
pRGBD-R	0.307	0.341	0.743	0.655	0.522	0.766	0.873

Table 2. Depth evaluation results on two *friberg3* RGB-D sequences. pRGBD-R results are after 3 *self-improving loops*.

4.4 Monocular SLAM/Pose Refinement Evaluation

In this section, we evaluate pose estimation/refinement on the KITTI Odometry sequences 09 and 10, KITTI Odometry test set sequences 11-21, and two TUM *friberg3* RGB-D sequences.

Results on KITTI Odometry Sequences 09 and 10. We show the quantitative results on seqs 09 and 10 in Tab. 4. It can be seen that our pRGBD-Initial outperforms RGB ORB-SLAM [30] both in terms of RSME and Rel Tr. Our pRGBD-Refined further improves pRGBD-Initial in all metrics, which verifies the effectiveness of our self-improving mechanism in terms of pose estimation. The higher Rel Rot errors of our method compared to RGB ORB-SLAM could be due to the high uncertainty of CNN-predicted depths for far-away points, which affects our rotation estimation [20]. In addition, our methods outperform

Table 3. Ablation study on 1st self-improving loop using KITTI Eigen split. The best performance is in **bold**

Loss	Lower is better				Higher is better		
	AbsRel	SqRel	RMSE	RMSElog	a1	a2	a3
w/o \mathcal{D}_c	0.117	0.958	4.956	0.194	0.862	0.955	0.980
w/o \mathcal{T}_c	0.118	0.955	4.867	0.194	0.872	0.957	0.980
w/o \mathcal{P}_c	0.117	0.942	4.855	0.194	0.873	0.958	0.980
all losses	0.117	0.931	4.809	0.192	0.873	0.958	0.981

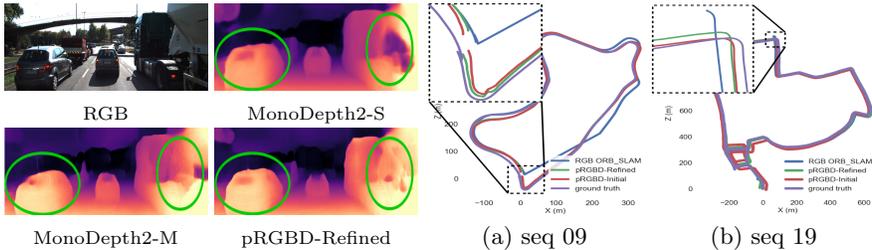


Fig. 4. Qualitative depth evaluation on KITTI Raw Eigen’s split test set. **Fig. 5.** Qualitative pose evaluation results on KITTI sequences.

all the competing supervised and self-supervised methods by a large margin, except for the supervised method of [51] with lower Rel Tr than ours on sequence 10. Note that we evaluate the camera poses produced by the pose network of Monodepth2-M [15] in Tab. 4, yielding much higher errors than ours. Fig. 5(a) shows the camera trajectories estimated for sequence 09 by RGB ORB-SLAM, our pRGBD-Initial, and pRGBD-Refined. It is evident that, although all the methods perform loop closure successfully, our methods generate camera trajectories that align better with the ground truth.

Table 4. Quantitative pose evaluation results on KITTI Odometry Sequences 09 and 10. - means the result is not available from the paper.

Method	Seq. 09			Seq. 10		
	RMSE	RelTr	RelRot	RMSE	RelTr	RelRot
DeepVO[47]	-	-	-	-	8.11	0.088
ESP-VO[48]	-	-	-	-	9.77	0.102
GFS-VO[50]	-	-	-	-	6.32	0.023
GFS-VO-RNN[50]	-	-	-	-	7.44	0.032
BeyondTracking[51]	-	-	-	-	3.94	0.017
DeepV2D[42]	79.06	8.71	0.037	48.49	12.81	0.083
SfMLearner [60]	24.31	8.28	0.031	20.87	12.20	0.030
GeoNet[57]	158.45	28.72	0.098	43.04	23.90	0.090
Depth-VO[58]	-	11.93	0.039	-	12.45	0.035
vid2depth[29]	-	-	-	-	21.54	0.125
UnDeepVO[24]	-	7.01	0.036	-	10.63	0.046
Wang <i>et al.</i> [45]	-	9.88	0.034	-	12.24	0.052
CC[35]	29.00	6.92	0.018	13.77	7.97	<u>0.031</u>
DeepMatchVO[37]	<u>27.08</u>	9.91	0.038	24.44	12.18	0.059
Li <i>et al.</i> [25]	-	8.10	0.028	-	12.90	0.032
Monodepth2-M[15]	55.47	11.47	0.032	<u>20.46</u>	7.73	0.034
SC-SfMLearner[2]	-	11.2	0.034	-	10.1	0.050
RGB ORB-SLAM	18.34	7.42	0.004	8.90	5.85	0.004
pRGBD-Initial	<u>12.21</u>	4.26	0.011	8.30	5.55	0.017
pRGBD-Refined	11.97	4.20	0.010	6.35	4.40	0.016

Results on KITTI Odometry Test Set. The KITTI Odometry leaderboard requires complete camera trajectories of all frames of all the sequences. Since

Seq	RGB ORB-SLAM			pRGBD-Initial			pRGBD-Refined		
	RMSE	RelTr	RelRot	RMSE	RelTr	RelRot	RMSE	RelTr	RelRot
11	14.83	7.69	0.003	6.68	3.28	0.016	3.64	2.96	0.015
13	6.58	2.39	0.006	6.83	2.52	0.008	6.43	2.31	0.007
14	4.81	5.19	0.004	4.30	4.14	0.014	2.15	3.06	0.014
15	3.67	1.78	0.004	2.58	1.61	0.005	2.07	1.33	0.004
16	6.21	2.66	0.002	5.78	2.14	0.006	4.65	1.90	0.004
18	6.63	2.38	0.002	5.50	2.30	0.008	4.37	2.21	0.006
19	18.68	4.91	0.002	23.96	2.82	0.007	13.85	2.52	0.006
20	9.19	6.74	0.016	8.94	5.43	0.027	7.03	4.50	0.022
12	X	X	X	X	X	X	94.2	32.94	0.026
17	X	X	X	14.71	8.98	0.011	12.23	7.23	0.011
21	X	X	X	X	X	X	X	X	X

Table 5. Pose evaluation results on KITTI Odometry test set. Since the ground truth for the KITTI Odometry test set is not available we run Stereo ORB-SLAM[31] to get the complete camera trajectories and use them as the pseudo ground truth to evaluate. ‘X’ denotes tracking failure.

	RGB SLAM			pRGBD-Initial			pRGBD-Refined		
	RMSE	RelTr	RelRot	RMSE	RelTr	RelRot	RMSE	RelTr	RelRot
W	X	X	X	0.23	0.02	0.52	0.09	0.01	0.30
L	1.72	0.02	0.32	1.40	0.01	0.34	0.39	0.01	0.33

Table 6. Pose evaluation results on two *frieburg3* RGB-D sequences, walking_xyz (W) and large_cabinet_validation (L).

we keep the default setting from ORB-SLAM, causing tracking failures in a few sequences, to facilitate quantitative evaluation on this test set (*i.e.*, sequences 11-21), we use pseudo-ground-truth computed as mentioned in Sec. 4.1 to evaluate all the competing methods in Tab. 5. From the results, RGB ORB-SLAM fails on three challenging sequences due to tracking failures, whereas our pRGBD-Initial fails on two sequences and our pRGBD-Refined fails only on one sequence. Among the sequences where all the competing methods succeed, our pRGBD-Initial reduces the RMSEs of RGB ORB-SLAM by a considerable margin for all sequences except for sequence 19. After our self-improving mechanism, our pRGBD-Refined further boosts the performance, reaching the best results both in terms of RMSE and Rel Tr. Fig. 5(b) shows qualitative comparisons on sequence 19.

Results on TUM RGB-D Sequences. Performance of pose refinement step on the two TUM RGB-D sequences is shown in Tab. 6. The result shows increased robustness and accuracy by pRGBD-Refined. In particular, RGB ORB-SLAM fails on walking_xyz, while pRGBD-Refined succeeds and achieves the best performance on both sequences. Refer supplementary material for qualitative results.

5 Analysis of Self-Improving Loops

In this section, we analyze the behaviour of three different evaluation metrics for depth estimation: Squared Relative (Sq Rel) error, RMSE error and accuracy metric a2, as defined in Sec. 4. The pose estimation is evaluated using the absolute trajectory pose error. In Fig. 6, we use the KITTI Eigen split dataset and report these metrics for each iteration of the self-improving loop. The evaluation metrics corresponding to the 0^{th} self-improving loop are of the pre-trained MonoDepth2-M. We summarize the findings from the plots in Fig. 6 as below:

- A comparison of evaluation metrics of farther scene points (*e.g.* max depth 80) with nearby points (*e.g.* max depth 30) at the 0^{th} self-improving loop

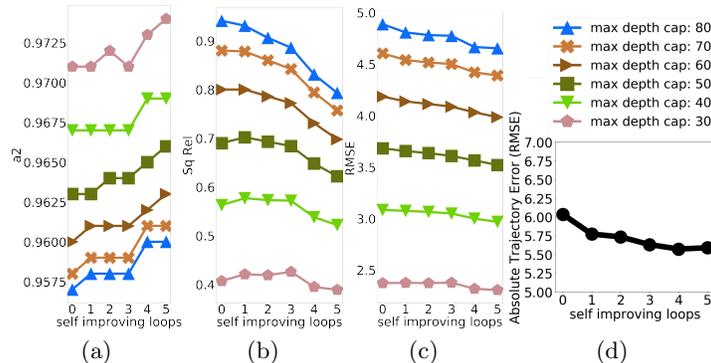


Fig. 6. Depth/Pose evaluation metric w.r.t. self-improving loops. Depth evaluation metrics in (a-c) are computed at different max depth caps ranging from 30-80 meters.

shows that the pre-trained MonoDepth2 performs poorly for farther scene points compared to nearby points.

- In the subsequent self-improving loops, we can see the rate of reduction in the Sq Rel and RMSE error is significant for farther away points compared to nearby points, *e.g.*, slope of error curves in Fig. 6(b-c) corresponding to max depth 80 is steeper than that of max depth 30. This validates our hypothesis of including wider baseline losses that help the depth network predict more accurate depth values for farther points. Overall, our joint narrow and wide baseline based learning setup helps improve the depth prediction of both the nearby and farther away points, and outperforms MonoDepth2 [15].
- The error plot in Fig. 6(d) shows a decrease in pose error with self-improving loops and complements the improvement in depth evaluation metrics as shown in Fig.6(a)-(c). We terminate the self-improvement loop once there is no further improvement, *i.e.*, at the 5th iteration.

6 Conclusion

In this work, we propose a self-improving framework to couple geometrical and learning based methods for 3D perception. A win-win situation is achieved — both the monocular SLAM and depth prediction are improved by a significant margin without any additional active depth sensor or ground truth label. Currently, our self-improving framework only works in an off-line mode, so developing an on-line real-time self-improving system remains one of our future works. Another avenue for our future works is to move towards more challenging settings [52], *e.g.*, rolling shutter cameras [36, 61] or uncalibrated cameras [17, 62].

Acknowledgements. This work was part of L. Tiwari’s internship at NEC Labs America, in San Jose. L. Tiwari was supported by Visvesvaraya Ph.D. Fellowship. S. Anand was supported by Infosys Center for Artificial Intelligence, IIIT-Delhi.

References

1. Andraghetti, L., Myriokefalitakis, P., Dovesi, P.L., Luque, B., Poggi, M., Pieropan, A., Mattoccia, S.: Enhancing self-supervised monocular depth estimation with traditional visual odometry. In: 3DV (2019)
2. Bian, J.W., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: NeurIPS (2019)
3. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: CodeSLAM-learning a compact, optimisable representation for dense visual slam. In: CVPR (2018)
4. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019)
5. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: ICCV (2019)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6), 1052–1067 (2007)
7. Dhiman, V., Tran, Q.H., Corso, J.J., Chandraker, M.: A continuous occlusion model for road scene understanding. In: CVPR (2016)
8. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NeurIPS (2014)
9. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(3), 611–625 (2017)
10. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular slam. In: ECCV (2014)
11. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: ICRA (2014)
12. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: CVPR (2018)
13. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: ECCV (2016)
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
15. Godard, C., Aodha, O.M., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019)
16. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017)
17. Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: CVPR (2019)
18. Grupp, M.: evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo> (2017)
19. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: ECCV (2018)
20. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
21. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: ISMAR (2007)

22. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning sfm from sfm. In: ECCV (2018)
23. Kuznetsov, Y., Stuckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: CVPR (2017)
24. Li, R., Wang, S., Long, Z., Gu, D.: Undeepvo: Monocular visual odometry through unsupervised deep learning. In: ICRA (2018)
25. Li, Y., Ushiku, Y., Harada, T.: Pose graph optimization for unsupervised monocular visual odometry. In: ICRA (2019)
26. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(10), 2024–2039 (2015)
27. Loo, S.Y., Amiri, A.J., Mashohor, S., Tang, S.H., Zhang, H.: CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction. In: ICRA (2019)
28. Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., Yuille, A.: Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. arXiv preprint arXiv:1810.06125 (2018)
29. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: CVPR (2018)
30. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics* **31**(5), 1147–1163 (2015)
31. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **33**(5), 1255–1262 (2017)
32. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: ICCV (2011)
33. Pillai, S., Ambrus, R., Gaidon, A.: Superdepth: Self-supervised, super-resolved monocular depth estimation. In: ICRA (2019)
34. Poggi, M., Tosi, F., Mattoccia, S.: Learning monocular depth estimation with unsupervised trinocular assumptions. In: 3DV (2018)
35. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: CVPR (2019)
36. Schubert, D., Demmel, N., Usenko, V., Stuckler, J., Cremers, D.: Direct sparse odometry with rolling shutter. In: ECCV (2018)
37. Shen, T., Luo, Z., Zhou, L., Deng, H., Zhang, R., Fang, T., Quan, L.: Beyond photometric loss for self-supervised ego-motion estimation. In: ICRA (2019)
38. Sheng, L., Xu, D., Ouyang, W., Wang, X.: Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep slam. In: ICCV (2019)
39. Song, S., Chandraker, M.: Robust scale estimation in real-time monocular sfm for autonomous driving. In: CVPR (2014)
40. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS (2012)
41. Tateno, K., Tombari, F., Laina, I., Navab, N.: CNN-SLAM: Real-time dense monocular slam with learned depth prediction. In: CVPR (2017)
42. Teed, Z., Deng, J.: DeepV2D: Video to depth with differentiable structure from motion. arXiv preprint arXiv:1812.04605 (2018)
43. Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S.: Learning monocular depth estimation infusing traditional stereo knowledge. In: CVPR (2019)

44. Wang, C., Miguel Buenaposada, J., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: CVPR (2018)
45. Wang, R., Pizer, S.M., Frahm, J.M.: Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In: CVPR (2019)
46. Wang, R., Schworer, M., Cremers, D.: Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In: ICCV (2017)
47. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: ICRA (2017)
48. Wang, S., Clark, R., Wen, H., Trigoni, N.: End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *International Journal of Robotics Research* **37**(4-5), 513–542 (2018)
49. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: ICCV (2019)
50. Xue, F., Wang, Q., Wang, X., Dong, W., Wang, J., Zha, H.: Guided feature selection for deep visual odometry. In: ACCV (2018)
51. Xue, F., Wang, X., Li, S., Wang, Q., Wang, J., Zha, H.: Beyond tracking: Selecting memory and refining poses for deep visual odometry. In: CVPR (2019)
52. Yang, N., Wang, R., Gao, X., Cremers, D.: Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters* **3**(4), 2878–2885 (2018)
53. Yang, N., Wang, R., Stuckler, J., Cremers, D.: Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: ECCV (2018)
54. Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R.: Lego: Learning edge with geometry all at once by watching videos. In: CVPR (2018)
55. Yang, Z., Wang, P., Xu, W., Zhao, L., Nevatia, R.: Unsupervised learning of geometry with edge-aware depth-normal consistency. In: AAAI (2018)
56. Yin, X., Wang, X., Du, X., Chen, Q.: Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In: ICCV (2017)
57. Yin, Z., Shi, J.: GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In: CVPR (2018)
58. Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: CVPR (2018)
59. Zhou, L., Kaess, M.: Windowed bundle adjustment framework for unsupervised learning of monocular depth estimation with u-net extension and clip loss. *IEEE Robotics and Automation Letters* **5**(2), 3283–3290 (2020)
60. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017)
61. Zhuang, B., Tran, Q.H.: Image stitching and rectification for hand-held cameras. In: ECCV (2020)
62. Zhuang, B., Tran, Q.H., Lee, G.H., Cheong, L.F., Chandraker, M.: Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam. In: IROS (2019)
63. Zou, Y., Ji, P., Tran, Q.H., Huang, J.B., Chandraker, M.: Learning monocular visual odometry via self-supervised long-term modeling. In: ECCV (2020)
64. Zou, Y., Luo, Z., Huang, J.B.: DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In: ECCV (2018)