Revisit Self-supervised Depth Estimation with Local Structure-from-Motion

Shengjie Zhu and Xiaoming Liu

Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
zhusheng@msu.edu, liuxm@cse.msu.edu

Abstract. Both self-supervised depth estimation and Structure-from-Motion (SfM) recover scene depth from RGB videos. Despite sharing a similar objective, the two approaches are disconnected. Prior works of self-supervision backpropagate losses defined within immediate neighboring frames. Instead of learning-through-loss, this work proposes an alternative scheme by performing local SfM. First, with calibrated RGB or RGB-D images, we employ a depth and correspondence estimator to infer depthmaps and pair-wise correspondence maps. Then, a novel bundle-RANSAC-adjustment algorithm jointly optimizes camera poses and one depth adjustment for each depthmap. Finally, we fix camera poses and employ a NeRF, however, without a neural network, for dense triangulation and geometric verification. Poses, depth adjustments, and triangulated sparse depths are our outputs. For the first time, we show self-supervision within 5 frames already benefits SoTA supervised depth and correspondence models. Despite self-supervision, our pose algorithm has certified global optimality, outperforming optimization-based, learning-based, and NeRF-based prior arts. Codes and models will be released.

Keywords: Self-supervision · Depth · Pose · Structure-from-Motion

1 Introduction

Monocular depth estimation [18,31] infers depthmap from a single image. It is an essential vision task with applications in AR/VR [44], autonomous driving [20], and 3D reconstruction [9]. Most methods [6, 46, 48, 72] supervise the model with groundtruth collected from stereo cameras [74] or LiDAR [20]. Recently, self-supervised depth [21, 22, 79] has drawn significant attention due to its potential to scale up depth learning from massive unlabeled RGB videos.

Classic SfM methods [1, 2, 13, 50, 51, 55, 68] also reconstruct scene depth from unlabled RGB videos. Despite its relevance, SfM is rarely applied to self-supervised depth learning. We outline two potential reasons. First, SfM is an off-the-shelf algorithm unrelated to the depth estimator. Poses and depths from SfM have different scales compared to depth models. Second, self-supervision has a well-defined training scheme to work with universal unlabeled videos. It backpropagates through photometric loss computed within immediate neighboring frames, *e.g.*, red trajectory in Fig. 1. In contrast, SfM is more selective to

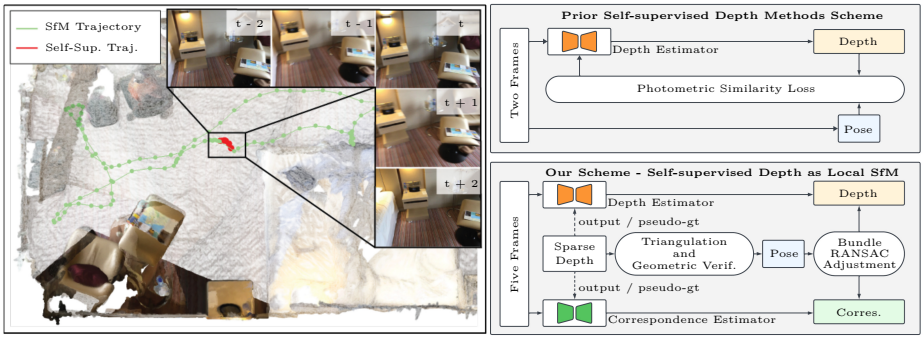


Fig. 1: Revisit Self-supervision with Local SfM. The work proposes alternating the learning-through-loss with a local SfM pipeline for self-supervised depth estimation. We summarize our differences. On self-supervision: (1) Instead of using naive two-view camera poses, we propose a Bundle-RANSAC-Adjustment pose optimization algorithm with multi-view constraints. (2) Instead of backpropagating through a loss, we produce a sparse point cloud with explicit triangulation and geometric verification. The point cloud serves as either output or pseudo-groundtruth for self-supervision. On SfM: (1) Our local SfM is adapted to use estimated monocular depthmaps and automatically resolve their scale inconsistency. (2) We maintain accuracy under significant sparse view variations, *e.g.*, red trajectories. We generalize SfM to as few as 5 frames, similar to the number of images used to define self-supervision loss.

input videos. It requires images of diverse view variations (green trajectory in Fig. 1), being inaccurate and unstable when applied to a small frame window.

This work connects self-supervision with SfM. We replace the self-supervision loss with a complete SfM pipeline that maintains robustness to a local window. Shown in Fig. 2, with N frames as input, our algorithm outputs $N-1$ camera poses, $N-1$ depth adjustments, and the sparse triangulated point cloud. In initialization, N monocular depthmaps and $N \times (N-1)$ pairwise correspondence maps are inferred. Next, we propose a Bundle-RANSAC-Adjustment pose estimation algorithm that retains accuracy for second-long videos. The algorithm utilizes the 3D priors from mono-depthmap to compensate for the deficient camera views. Correspondingly, we optimize $N-1$ depth adjustments to alleviate the depth scale ambiguity by temporally aligning to the root frame depth.

The Bundle-RANSAC-Adjustment extends two-view RANSAC with multi-view bundle-adjustment (BA). The algorithm has quadratic complexity, designed for parallel GPU computation. We RANdomly SAmple and hypothesize a set of normalized poses. In Consensus checking, we apply BA to evaluate a robust inlier-counting scoring function over multi-view images. Camera scales and depth adjustments are determined during BA to maximize the scoring function.

Next, we freeze the optimized poses and employ a Radiance Field (RF), *i.e.*, a NeRFF [42] without a neural network, for triangulation. We optimize RF to achieve multi-view depthmap and correspondence consistency within a shared 3D frustum volume. For outputs, we apply geometric verification to extract multi-view consistent point cloud, *i.e.*, a sparse root depthmap.

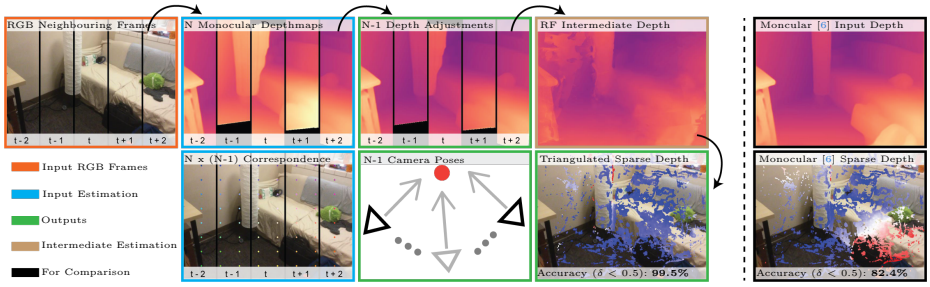


Fig. 2: Local Structure-from-Motion. With N neighboring frames, we extract monocular depthmaps and pairwise dense correspondence maps with methods, *e.g.*, ZoeDepth [6] and PDC-Net [60]. Next, skipping the root frame, we optimize the rest $N-1$ camera poses and depth adjustments. The depth adjustments render input depthmaps **temporally consistent**. Fixing poses and adjustments, we use the Radiance Field (RF) for triangulation. A geometrically verified sparse root depthmap is output. Our local SfM applies **self-supervision** with only 5 RGB frames. Yet, our sparse output already outperforms the input supervised depth with SoTA performance.

Fig. 1 contrasts our method with prior self-supervised depth and SfM methods. To our best knowledge, there has not been prior work showing geometry-based self-supervised depth benefits supervised models. However, self-supervision is supposed to augment supervised models with unlabeled data. In Fig. 2, our unique pipeline gives the **first** evident results, that self-supervision with **as few as** 5 frames already benefits supervised models.

Despite depths, our multi-view RANSAC pose has certified global optimality under a robust scoring function. It outperforms prior arts in optimization-based [51, 80], learning-based [58, 64], and NeRF-based [61] pose algorithms.

Beyond pose and depth, our method has diverse applications. The depth adjustments from our method provide empirically consistent depthmaps, being important for AR image compositing. When with RGB-D inputs, our method enables self-supervised correspondence estimation. Our accurate pose estimation gives improved projective correspondence than the SoTA supervised correspondence input. An example is in Fig. 9. We summarize our contributions as:

- We propose a novel local SfM algorithm with Bundle-RANSAC-Adjustment.
- We show the **first** evident result that self-supervised depth with **as few as** 5 frames already benefit SoTA supervised models.
- We achieve SoTA sparse-view pose estimation performance.
- We enable self-supervised temporally consistent depthmaps.
- We enable self-supervised correspondence estimation with 5 RGB-D frames.

2 Related Works

Structure-from-Motion. SfM is a comprehensive task [51, 68]. A typical pipeline is, correspondence extraction [8, 38, 62], two-view initialization [4, 32, 34], triangulation [33, 45], and local & global bundle-adjustment [51, 68]. Classic methods

require diverse view variations for accurate reconstruction. Our method compensates SfM on scarse camera views via introducing deep depth estimator. Further, we suggest SfM itself is a self-supervised learning pipeline, as in Fig. 1. Finally, our SfM is not up-to-scale and shares the metric space as the input depthmap.

Sparse Multi-view Pose Estimation. Estimating poses from sparse frames is crucial for self-supervision [12,21,49,73,77], video depth estimation [23,58,63,80], and sparse-view NeRF [15, 28, 36, 43, 61]. Camera poses are estimated either by learning [12, 21, 23, 58], optimization [76, 80] or together with NeRF [36, 61]. We propose an additional multi-view RANSAC pipeline with improved accuracy.

Self-supervised Depth and Correspondence Estimation. Multiple works improve self-supervised depth in different ways, including learning loss [21,47,66], architecture [24, 78], camera pose [7, 11, 41, 76], joint with semantics segmentation [79], and using large-scale data [56,69]. Recently, [56] shows self-supervision only performs on-par with supervised models under substantially more data. [69] shows the benefit of self-supervision via exploiting non-geometry monocular semantic consistency. Our method shows the first evident results where self-supervision benefits supervised models with only 5 consecutive frames.

Consistent Depth Estimation. AR applications necessitate temporally consistent depthmaps, *i.e.*, depthmaps from different temporal frames reside in the same 3D space. Recent works [39,75] align depthmap according to the poses and points from the off-the-shelf COLMAP algorithm. Our method seamlessly integrates SfM with monocular depthmaps, outputting consistent depth and poses.

Test Time Refinement (TTR). TTR aims to improve self-supervised / supervised depth estimators in testing time with RGB video [10, 11, 30, 53, 67]. Methods [26,59] rely on off-the-shelf algorithms for pseudo depth and pose labels. Recently, [26] first shows TTR improves supervised models. TTR is our downstream application, which details strategies for utilizing noisy pseudo-labels.

3 Methodology

Our method runs sequentially. From N calibrated images $\mathcal{I}$, we extract N monocular depthmaps $\mathcal{D}$ and $N \times (N-1)$ pair-wise dense correspondence $\mathcal{C}$. We split the N images into one root frame $\mathbf{I}_o$ in the center of the N-frame window where $o = \lfloor \frac{N+1}{2} \rfloor$, and $N-1$ support frames $\mathbf{I}_i$, where $i \in \mathbb{Z} = [1, N] \backslash \{o\}$. In Sec. 3.1, after setting the root frame as identity pose, we use Bundle-RANSAC-Adjustment to optimize $N-1$ poses $\mathcal{P}$ and $N-1$ depth adjustments $\mathcal{R}$. Next, in Sec. 3.2, we apply triangulation by optimizing a frustum Radiance Field (RF) $\mathbf{V}$, *i.e.*, a NeRF without network. Finally, in Sec. 3.3, we apply geometric verification by rendering multi-view consistent 3D points from RF. An overview is in Fig. 3.

3.1 Bundle-RANSAC-Adjustment Pose Estimation

We generalize two-view RANSAC with multi-view constraints through Bundle-Adjustment. Sec. 3.1.1 describes our pipeline. In Sec. 3.1.2, we propose Hough transform to accelerate computation. We discuss the time complexity in Sec. 3.1.3.

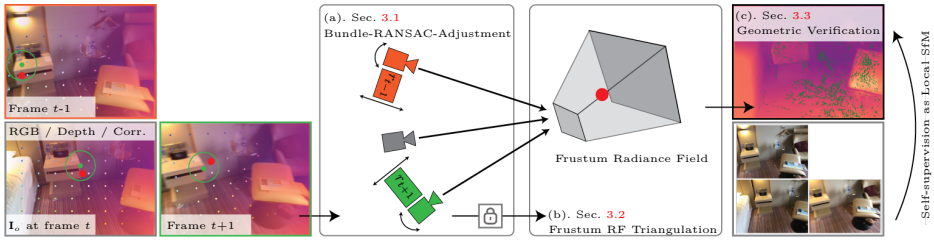


Fig. 3: Algorithm Overview. After extracting monodepths and correspondence maps from inputs: (a) We apply Bundle-RANSAC-Adjustment to optimize $N-1$ camera poses $\mathcal{P}$ and $N-1$ depth adjustments $\mathcal{R}$. (b) We fix poses and depth adjustments and optimize a frustum Radiance Field (RF) for triangulation. (c) We apply geometric verification to extract multi-view consistent 3D points via rendering with RF. We further detail step (a) in Fig. 4, 5, and 6, and steps (b) and (c) in Fig. 7.

3.1.1 Optimization Pipeline

RANdom SAmple. We use five-point algorithm [34] as the minimal solver. We execute it between root and each support frame on GPU, extracting a pool of $(N-1)\times K$ normalized poses $\overline{\mathcal{Q}} = \{\overline{\mathbf{P}}_i^k \mid i \in \mathbb{Z}, k \in [1,K]\}$, where $\overline{\mathbf{P}}_i^k \in \mathbb{R}^{3\times 4}$. The K is the number of normalized poses extracted per frame. We term a set of $N-1$ normalized poses as a group $\overline{\mathcal{P}} \in \mathbb{R}^{(N-1)\times 3\times 4}$. Two-view RANSAC enumerates over single normalized pose $\overline{\mathbf{P}}$. Our multi-view algorithm hence enumerates over normalized pose group $\overline{\mathcal{P}}$. We initialize the optimal group $\overline{\mathcal{P}}^*$ as the top candidate from K poses of $\overline{\mathcal{Q}}$ for each frame. See examples in Fig. 4.

Bundle-Adjustment Consensus. While computing consensus counts, the camera scales $\mathcal{S}$ and depth adjustments $\mathcal{R}$ are automatically determined with bundle-adjustment to maximize a robust scoring function:

$$\rho_i = \phi(\overline{\mathcal{P}}) = \max_{\mathcal{S},\mathcal{R}} f(\mathcal{S},\mathcal{R} \mid \overline{\mathcal{P}},\mathcal{D},\mathcal{C}). \tag{1}$$

Search for Optimal Group. Our multi-view RANSAC has a significantly larger solution space than two-view RANSAC. With N view inputs, we determine the optimal group out of K^{N-1} combinations. Hence, we iteratively search for the optimal group with a greedy strategy. For each epoch, we ablate $(N-1)(K-1)$ additional pose groups:

$$\overline{\mathcal{P}}_i^k = \overline{\mathcal{P}}_i^* \setminus \{\overline{\mathbf{P}}_i^*\} \cup \{\overline{\mathbf{P}}_i^k\}, \tag{2}$$

where $i \in \mathbb{Z}$ and $k \in [1,K]$. Combine Eq. (2) and Fig. 4, taking frame i as an example, we replace the optimal pose $\overline{\mathbf{P}}_i^*$ by its $K-1$ other candidates $\overline{\mathbf{P}}_i^k$, generating $K-1$ groups. For N frames, we have $(N-1)(K-1)+1$ groups. We apply bundle-adjustment to each group to evaluate Eq. (1). As shown in Fig. 3 and Fig. 4, we select the normalized pose together with its optimized scales and depth adjustments that maximize the scores as the output,

$$\mathcal{P}_i^* = b(\overline{\mathcal{P}}_i^*, \mathcal{S}_i^*),\ \mathcal{R}_i^* = \mathcal{R}_i^k,\ \text{where}\, k = \arg\max\{\rho_i^k\},\ \overline{\mathcal{P}}_i^* = \overline{\mathcal{P}}_i^k,\ \mathcal{S}_i^* = \mathcal{S}_i^k, \tag{3}$$

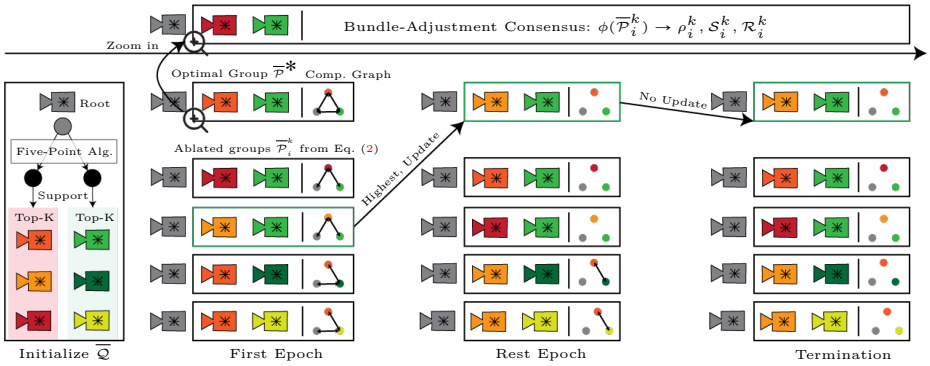


Fig. 4: Pose Optimization Pipeline. We show a sample execution when $N=3$ and $K=3$. We initialize normalized pose candidates pool $\overline{\mathcal{Q}}$. Optimal group $\overline{\mathcal{P}}^*$ is set to top candidates within $\overline{\mathcal{Q}}$. In each epoch, Eq. (2) ablates pose group $\overline{\mathcal{P}}_i^k$. Each group is scored with Eq. (1) via BA with Hough Transform, detailed in Sec. 3.1.2. The optimal group with the highest score is updated with Eq. (3). Termination occurs when the maximum score stabilizes. We maintain quadratic complexity by avoiding repetitive computation after the first epoch, shown with the Comp. Graph, detailed in Sec. 3.1.3.

where $b(\cdot)$ combines normalized poses with scales. Fig. 2 third column plots an adjusted temporal consistent depthmap after applying $\mathcal{R}^*$. In Fig. 4, the algorithm terminates when the maximum score stops increasing.

Scoring Function. Similar to other RANSAC methods, we adopt robust inlier-counting based scoring functions. Expand Eq. (1) for a specific group $\overline{\mathcal{P}}$:

$$\phi(\overline{\mathcal{P}}) = \sum_{i,i\neq j} \sum_{j} f_{i,j}(s_i, s_j, r_i, r_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, \mathbf{D}_i, \mathbf{D}_j, \mathbf{C}_{i,j}), \tag{4}$$

where i, j are frame index. We set the per-frame camera scale, depth, and correspondence as $s \in \mathcal{S}$, $\mathbf{D} \in \mathcal{D}$, and $\mathbf{C} \in \mathcal{C}$. The discrete scoring function $f_{i,j}(\cdot)$ has various forms. We provide three. First, we describe a 2D scoring function:

$$f_{i,j}^{\text{2D}}(\cdot) = \sum_{m} \|\pi(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_i^m) - \mathbf{c}_{i,j}^m\|_2 < \lambda^{\text{2D}}, \tag{5}$$

where $m \in [1, M]$ indexes sampled pixels per frame pair. $f_{i,j}^{\text{2D}}(\cdot)$ measures the inlier count between depth projected correspondence and input correspondence. $\pi(\cdot)$ is projection process. Intrinsic is skipped. d and $\mathbf{c}$ are depth and correspondence sampled from $\mathbf{D}$ and $\mathbf{C}$. An example is in Fig. 5. The projected pixel is an inlier if it resides within the circle of radius λ^{2D} and center at the correspondence $\mathbf{c}_{i,j}^m$ (denoted as $\mathbf{p}_j$ in Fig. 5). Second, we introduce a 3D scoring function:

$$f_{i,j}^{\text{3D}}(\cdot) = \sum_{m} \|\pi^{-1}(s_i \mid \overline{\mathbf{P}}_i, r_i, d_i^m) - \pi^{-1}(s_j \mid \overline{\mathbf{P}}_j, r_j, d_j^m)\|_2 < \lambda^{\text{3D}}. \tag{6}$$

The depth pair d_i and d_j is determined by input correspondence. Unlike the 2D one, the 3D function fixes depth adjustment r. We elaborate its reason in Supp. We also include a Sampson error [40] based scoring function in Supp.

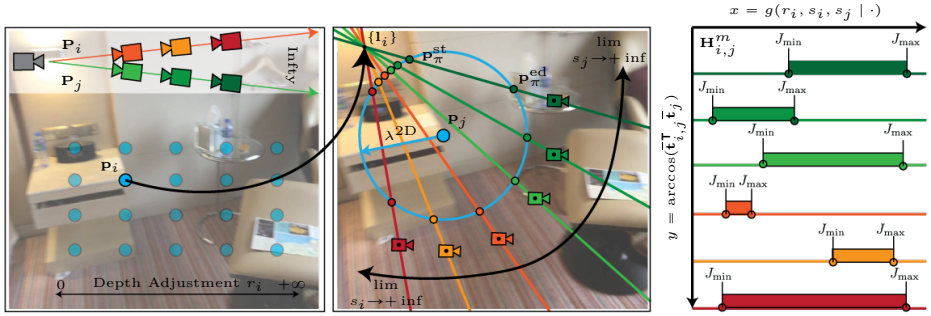


Fig. 5: Hough Transform between Two Normalized Poses. With fixed normalized poses, there exists three variables, *i.e.*, scales of $\mathbf{P}_i$ & $\mathbf{P}_j$ and adjustment r_i. Pixel $\mathbf{p}_i$ and $\mathbf{p}_j$ are corresponded. Ablating pose scales maps pixel $\mathbf{p}_i$ to a set of epipolar lines $\{\mathbf{l}_i\}$, however, bounded by Red and Green at infinite scales. We have three observations. First, with fixed normalized poses, epipolar lines $\mathbf{l}_i$ have limited possibilities. Second, scale s and depth adjustment d are equivalent, both adjusting projection on the epipolar line. Third, per epipolar line, to be an inlier, the projection has to reside within the line-circle intersection, between $\mathbf{p}_\pi^{\text{st}}$ and $\mathbf{p}_\pi^{\text{ed}}$. The observations motivate us to discretizes the solution space to a 2D matrix, *i.e.*, Hough Transform. Right figure plots an example transformation $\mathbf{H}_{i,j}^m$ from frame i to j on the mth pixel $\mathbf{p}_i$.

3.1.2 Hough Transform Acceleration

Maximizing Eq. (1) for each pose group is computationally prohibitive, as shown in Fig. 4. We propose Hough Transform for acceleration. We use Eq. (5), the 2D function $f^{\text{2D}}(\cdot)$ as an example for illustration. See our motivation in Fig. 5.

Hough Transform. The relative pose between $\overline{\mathbf{P}}_i$ and $\overline{\mathbf{P}}_j$ is defined as:

$$\mathbf{P}_{i,j} = \mathbf{P}_j \mathbf{P}_i^{-1} = \left[\mathbf{R}_{i,j}\ s_{i,j}\bar{\mathbf{t}}_{i,j}\right] = \left[\mathbf{R}_j\mathbf{R}_i^{-1}\ -s_i\mathbf{R}_j\mathbf{R}_i^{-1}\bar{\mathbf{t}}_i + s_j\bar{\mathbf{t}}_j\right], \tag{7}$$

where $\mathbf{R}, \bar{\mathbf{t}}$, and s are rotation, normalized translation and pose scale. From Eq. (7) and Fig. 5, $\bar{\mathbf{t}}_{i,j}$ is controlled by the scale s_i and s_j, and thus we have:

$$\lim_{s_i \to +\inf} \bar{\mathbf{t}}_{i,j} = -\mathbf{R}_j\mathbf{R}_i^{-1}\bar{\mathbf{t}}_i, \qquad \lim_{s_j \to +\inf} \bar{\mathbf{t}}_{i,j} = \bar{\mathbf{t}}_j. \tag{8}$$

For a pixel $\mathbf{p}_i$ on frame i, its corresponding epipolar line $\mathbf{l}_i$ on frame j is:

$$\mathbf{l}_i = \mathbf{K}^{-\intercal}[\bar{\mathbf{t}}_{i,j}]_\times \mathbf{R}_{i,j}\mathbf{K}^{-1}\mathbf{p}_i. \tag{9}$$

Eq. (8) and Eq. (9) suggest the epipolar line has limited possibilities. Operation $[\cdot]_\times$ is the cross product in matrix form. Further, as the projected pixel $\mathbf{p}_\pi$ of $\mathbf{p}_i$ always locate on the epipolar line $\mathbf{l}_i$ [25], we have:

$$\mathbf{l}_i^{\intercal}\mathbf{p}_\pi = 0,\ \mathbf{p}_\pi = \pi(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_i). \tag{10}$$

To be an inlier of the scoring function $f^{\text{2D}}(\cdot)$, we have:

$$\|\mathbf{p}_\pi - \mathbf{p}_j\|_2 \leqslant \lambda^{\text{2D}}. \tag{11}$$

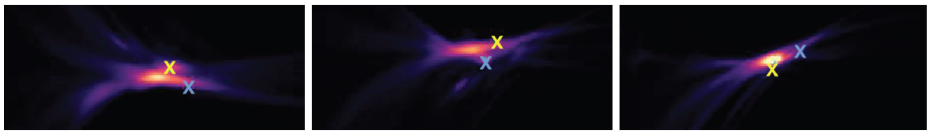


Fig. 6: Visualize Hough Transform Matrix $\mathbf{H}_i^j$ from Eq. (18). Area with higher intensity suggests more inlier counts. Given normalized pose group, for N views, there exists $N \times (N-1)$ matrices $\mathbf{H}_i^j$, constraining $N-1$ scale and $N-1$ adjustments. We plot the start and end points after optimizing Eq. (19) in the figure.

Combining Eq. (10), Eq. (11) and Fig. 5, to be an inlier, the projected pixel $\mathbf{p}_\pi$ has to reside within the line segment, with two end-points computed by the line-circle intersection. The circle centers at corresponded pixel $\mathbf{p}_j$ on frame j with a radius λ^{2D}. We denote the two end-points $\mathbf{p}_\pi^{\text{st}}$ and $\mathbf{p}_\pi^{\text{ed}}$. We add their calculation in Supp. Following [80], there exists a function $J(\cdot)$ that maps a projected pixel $\mathbf{p}_\pi$ and adjusted depth $r_i d_i$ to camera scale $s_{i,j}$ as: $s_{i,j} = J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi)$.

Corollary 1. *A pixel is an inlier iff:*

$$J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi^{st}) \leqslant s_{i,j} \leqslant J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi^{ed}). \tag{12}$$

Corollary 2. *Scale and depth are equivalent as;*

$$s_{i,j} = J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi) = r_i \cdot J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi). \tag{13}$$

See Fig. 5 and proof in Supp. Combine Eqs. (12) and (13),

$$J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{st}}) \leqslant \frac{s_{i,j}}{r_i} \leqslant J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{ed}}). \tag{14}$$

Set $g(\cdot)$ maps the variables under optimization to intermediate term $\frac{s_{i,j}}{r_i}$:

$$J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{st}}) \leqslant g(r_i, s_i, s_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j) \leqslant J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{ed}}). \tag{15}$$

Pixel $\mathbf{p}_i$ is an inlier if and only if its projection satisfies Eq. (15). Note, the value space of function $g(\cdot)$ can be mapped to a 2D space $\mathbf{H}$ after Hough Transform:

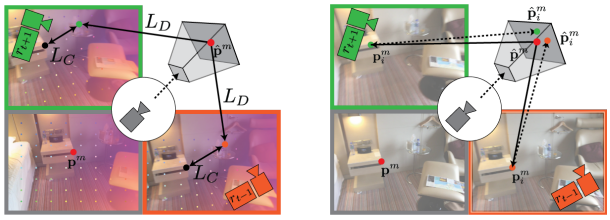
$$x = g(r_i, s_i, s_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j),\ y = \arccos(\bar{\mathbf{t}}_{i,j}^{\intercal} \bar{\mathbf{t}}_j), \tag{16}$$

where x and y are transformed coordinates. From Eq. (16), x is a synthesized translation magnitude and y is angular variable. We then set $x \in [0, x_{\max}]$, and $y \in [0, \theta_{\max}]$, where $\theta_{\max} = \arccos(-\bar{\mathbf{t}}_j^{\intercal} \mathbf{R}_{i,j} \bar{\mathbf{t}}_i)$. Finally, the value of $\mathbf{H}$ is:

$$\forall y \in [0, \theta_{\max}],\ \mathbf{H}(x \mid y) = 1, \text{if}\ \ x \in [J_{\min}, J_{\max}], \tag{17}$$

where $J_{\min}$ and $J_{\max}$ are the two bounds from Eq. (15). The transformation over the scoring function $f_{i,j}^{\text{2D}}$ with all M sampled pixels between frame $\mathbf{I}_i$ and $\mathbf{I}_j$:

$$\mathbf{H}_{i,j} = \sum_m \mathbf{H}_{i,j}^m,\ f_{i,j}^{\text{2D}}(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j) = \mathbf{H}_{i,j}(x, y), \tag{18}$$



(a) Triangulation **(b)** Geometric Verification

Fig. 7: Triangulation optimizes frustum RF for multiview consistency *w.r.t.* depth and correspondence. **Geometric Verification** inferences RF for sparse multiview consistent 3D points. For simplicity, in (a), we only plot L_c defined from the root frame.

where x and y are functions of s_i, s_j, r_i. Eq. (1) becomes:

$$\phi(\overline{\mathcal{P}}) = \max_{\mathcal{S},\mathcal{R}} \sum_i \sum_{j,j\neq i} \mathbf{H}_{i,j}(x(\mathcal{S},\mathcal{R}), y(\mathcal{S},\mathcal{R})). \tag{19}$$

In our implementation, we discretize $\mathbf{H}_{i,j}$ to a 2D matrix.

Accelerate Bundle-Adjustment Consensus. The BA determines $N-1$ camera scales and $N-1$ depth adjustments to maximize the scoring function $\phi(\cdot)$ in Eq. (19). With Hough transform, BA maximizes the summarized intensity via **indexing** $N \times (N-1)$ Hough transform matrices $\mathbf{H}$. It avoids BA repetitively enumerating all sampled pixels. Fig. 6 shows an example optimization process.

Certified Global Optimality of robust inlier-counts scoring function Eq. (5) and Eq. (6) are achieved after optimization. See Fig. 8 for more analysis.

Optimization with RGB-D. With GT depthmap, the algorithm switches to the 3D scoring function $f_{i,j}^{\text{3D}}(\cdot)$. The depth adjustment is fixed to 1 and the 2D line-circle intersection becomes 3D line-sphere intersection. See Supp. for details.

3.1.3 Computational Complexity

Naive Time Complexity. From Eq. (2) and Fig. 4, in each epoch, we evaluate $(N-1)(K-1)$ pose groups with Hough Transform Acceleration. Suppose each group takes T iterations to optimize Eq. (19), the time complexity is:

$$\mathcal{O}((N-1)(K-1) \cdot N(N-1) \cdot (M+T)), \tag{20}$$

where each group computes $N(N-1)$ Hough matrices $\mathbf{H}$. Each matrix enumerates M sampled pixels, see Eq. (18). Maximizing Eq. (19) becomes indexing $\mathbf{H}$, hence has constant time complexity T, where $T << M$.

Counting Unique Hough Matrices. Most computation is spent on Hough matrices. In Fig. 4, each connection in the computation graph suggests two unique Hough matrices. We minimize time complexity by only computing **unique** Hough matrices. In Fig. 4 first epoch, the initial optimal group $\overline{\mathcal{P}}^*$ has $N(N-1)$ matrices. Each ablated group only differs by one pose, hence introducing $2(N-1)(N-1)(K-1)$ matrices. The first-epoch complexity is then:

$$\mathcal{O}^H(N(N-1)M + 2(N-1)^2(K-1)M) + \mathcal{O}^{\text{BA}}(N(N-1)(K-1)T). \tag{21}$$

Only the Hough transform is accelerated. As $T << M$, the complexity of BA is neglectable. After the first epoch, $\overline{\mathcal{P}}^*$ only updates one pose per epoch, hence introducing $2(N-2)(K-1)$ matrices. The complexity for the rest epochs is,

$$\mathcal{O}^H(2(N-2)(K-1)M) + \mathcal{O}^{\text{BA}}(N(N-1)(K-1)T). \tag{22}$$

While Eq. (22) has linear complexity, our method only updates one pose per epoch. Updating poses in all frames like other SfM methods is still quadratic.

3.2 Frustum Radiance Field Triangulation

Frustum Radiance Field. Now, we fix the optimized pose $\mathcal{P}^*$. Then we employ a frustum radiance field $\mathbf{V}$ of size $H \times W \times D$ for dense triangulation. Field $\mathbf{V}$ is defined over the root frame $\mathbf{I}_o$ and shares similarity with the categorical depthmap [5,18]. We follow [61,65] in rendering the depth d. The RGB estimation is skipped as unrelated. A 3D ray originated from pixel $\mathbf{p}_i$ at frame i is discretized into a set of 3D points and depth labels. With slight abuse of notation, we denote $\{\hat{\mathbf{p}}_{i,t} = \mathbf{o} + d_t\mathbf{r} \mid t \in [1,T]\}$, where $\hat{\mathbf{p}}$ is a 3D point, d_t is depth label and $\mathbf{r}$ is ray direction. Set integration interval $\delta_t = d_{t+1} - d_t$, depth d is:

$$d(\mathbf{p}_i) = \sum_t \alpha_t d_t,\ \alpha_t = T_t(1 - \exp(-\sigma_t\delta_t)),\ T_t = \exp(-\sum_{t' \in [1,t]} \sigma_{t'}\delta_{t'}). \tag{23}$$

We set the camera origin of frame i as $\mathbf{o}$. Instead of regressing occupancy δ with MLP [61,65], we directly interpolate the radiance field $\mathbf{V}$:

$$\delta_t = \mathbf{V}(u,v,w), \text{where } \begin{bmatrix} u & v & w \end{bmatrix}^\intercal = \pi(\mathbf{K}, \hat{\mathbf{p}}_{i,t}). \tag{24}$$

Compared to MLP, frustum radiance field $\mathbf{V}$ is computationally efficient [17].

Triangulation. Classic triangulation method [51] operates on a single 3D point. The RF provides additional constraints where all optimized points share a canonical 3D volume. In Fig. 7, we supervise $\mathbf{V}$ for multi-view consistency between dense depthmap $\mathcal{D}$ and correspondence map $\mathcal{C}$. On depth:

$$L_D = \frac{1}{NM} \sum_i \sum_m \|\pi(\mathbf{P}_i, \hat{\mathbf{p}}^m) - d_i^m\|_1. \tag{25}$$

Here, $\hat{\mathbf{p}}^m$ is rendered from the root frame, following depth computed with Eq. (23). To apply correspondence consistency, we have:

$$L_C = \frac{1}{N(N-1)M} \sum_i \sum_{j,j\neq i} \sum_m \|\pi(\mathbf{P}_j, \hat{\mathbf{p}}_i^m) - \mathbf{q}_{i,j}^m\|_1, \tag{26}$$

where $\hat{\mathbf{p}}_i^m = \pi^{-1}(\mathbf{P}_i, \mathbf{p}_i^m, d(\mathbf{p}_i^m)), \mathbf{p}_i^m = \pi(\mathbf{P}_i, \hat{\mathbf{p}}^m)$. With slight abuse of notation, function $\pi(\cdot)$ returns depth for L_D, and location for L_C. We **always** first render from the root frame and subsequently project to N frames. From there, we project to other supported frames again, forming $N(N-1)$ pairs.

Table 1: Self-Supervised Depth Estimation. We apply self-supervision with 5 frames via executing the local SfM. We output improved sparse depthmaps over SoTA supervised inputs. The evaluation is conducted over the root frame.

Dataset	Method	Density	$\delta_{0.5}$	δ_1	$\mathbf{SI}_{\log}$	A.Rel	S.Rel	RMS	$\mathbf{RMS}_{\log}$
ScanNet [14]	ZoeDepth [6]	9.1%	0.877	0.963	6.655	0.056	0.016	0.154	0.075
	↳ Ours		**0.902**	**0.976**	**5.901**	**0.050**	**0.014**	**0.149**	**0.070**
	ZeroDepth [37]	5.6%	0.641	0.834	12.860	0.124	0.086	0.337	0.152
	↳ Ours		**0.686**	**0.877**	**9.463**	**0.106**	**0.067**	**0.295**	**0.133**
	Metric3D [71]	2.6%	0.804	0.946	6.708	0.067	0.020	0.150	0.084
	↳ Ours		**0.854**	**0.968**	**4.170**	**0.055**	**0.014**	**0.125**	**0.068**
KITTI360 [35]	ZoeDepth [6]	4.0%	0.677	0.899	14.154	0.103	0.490	3.521	0.153
	↳ Ours		**0.719**	**0.910**	**13.220**	**0.094**	**0.474**	**3.499**	**0.145**
	ZeroDepth [37]	4.5%	0.584	0.844	16.468	0.132	0.819	3.486	0.183
	↳ Ours		**0.654**	**0.877**	**13.881**	**0.115**	**0.772**	**3.395**	**0.164**
	Metric3D [71]	3.2%	0.846	0.958	9.226	0.072	0.508	2.194	0.104
	↳ Ours		**0.860**	**0.963**	**8.896**	**0.068**	**0.487**	**2.139**	**0.101**

Table 2: Consistent Depth Estimation. We measure the numerical improvement by aligning the support frame depthmaps to the root frame with our depth adjustment scalars. The evaluation is conducted on support frames on ScanNet [14].

Method	$\delta_{0.5}$	δ_1	$\mathbf{SI}_{\log}$	A.Rel	S.Rel	RMS	$\mathbf{RMS}_{\log}$
ZoeDepth [6]	0.658	0.894	**9.242**	0.104	0.039	0.255	0.128
↳ Ours	**0.793**	**0.942**	**9.242**	**0.079**	**0.024**	**0.203**	**0.105**
ZeroDepth [37]	0.351	0.589	**20.145**	0.254	0.223	0.565	0.287
↳ Ours	**0.490**	**0.725**	**20.145**	**0.199**	**0.156**	**0.457**	**0.237**
Metric3D [71]	0.533	0.753	**12.425**	0.216	0.339	0.495	0.228
↳ Ours	**0.664**	**0.838**	**12.425**	**0.137**	**0.126**	**0.345**	**0.175**

3.3 Geometric Verification

With the RF optimized, we apply geometric verification to acquire sparse multi-view consistent 3D points, as in Fig. 7:

$$\mathcal{C} = \{ \sum_{i,i\neq o} c_i^m \geqslant n^c \},\ c_i^m = 1 \text{ if } \sum_{i,i\neq o} \|\hat{\mathbf{p}}_i^m - \hat{\mathbf{p}}^m\|_2 \leqslant \lambda^c. \tag{27}$$

We follow the same rendering process as training, where $\hat{\mathbf{p}}_i^m$ is computed with Eq. (26). First, we render 3D points from the root frame, project them to other views, and render 3D points from there again. A point is valid if a minimum of n^c views are consistent with the root.

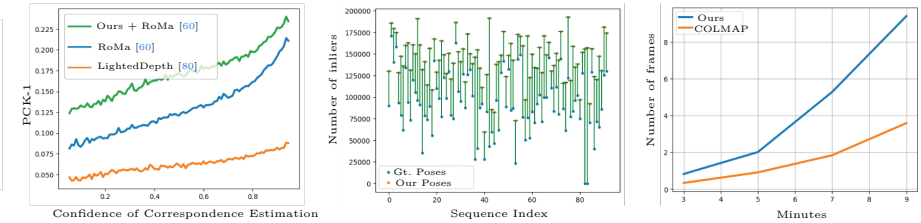
4 Experiments

4.1 Self-supervised Depth Estimation

We benchmark whether self-supervision benefits supervised depth in unseen test data. For the correspondence estimator, we use PDC-Net [60]. For depth estimators, we adopt recently published in-the-wild depth estimator, including

Table 3: Self-Supervised Correspondence Estimation. We improve correspondence with RGB-D inputs, using metrics from [60]. The entry train and test are training and testing datasets of correspondence estimators. [Key: M=MegaDepth, S=ScanNet]

Method	Train	Test	PCK-1	PCK-3	PCK-5	AEPE
PDC-Net [60]			0.119	0.511	0.743	4.612
↳ LightedDepth [80]	M	S	0.061	0.341	0.563	6.590
↳ Ours			**0.178**	**0.658**	**0.866**	**2.898**
RoMa [16]			0.144	0.583	0.815	3.333
↳ LightedDepth [80]	S	S	0.066	0.359	0.588	5.974
↳ Ours			**0.183**	**0.638**	**0.844**	**3.067**



(a) Self-supervised Correspondence (b) Certified Global Optimality (c) Run-time Comparison

Fig. 8: Ablation Studies on the ScanNet.

ZoeDepth [6], ZeroDepth [37], and Metric3D [71]. We evaluate with ScanNet [14] and KITTI360 [35] where all models perform zero-shot prediction.

Test Data. In dense correspondence estimation, methods [60,61,81] output confidence score per correspondence. We follow [60,61] to set a minimum threshold of 0.95. We run on ScanNet test split and it returns 92 sequences with sufficient correspondence. We form our test split by sampling 5 neighboring frames per valid sequence. Similarly, we run on KITTI360 data and randomly select 100×5 test split, *i.e.*, 100 sequences with 5 frames each. We consider it a comprehensive experiment. Similar to SPARF [61], our triangulation trains a NeRF-like structure. For reference, SPARF experiment on DTU dataset [27] includes only 15 sequences each with 3 images. In comparison, we include around 100 sequences.

Evaluation Protocols. We evaluate on **root** frame. We remove the scale ambiguity in the local SfM system to correctly reflect depth improvement. Specifically, we adjust all 5 depthmaps by an identical scalar computed between estimated root and GT depthmap, by median scaling [22]. This eliminates scale ambiguity in the root frame while preserving it in support frames.

Results. In Tab. 1, our point cloud has a density of $2.6\% - 9.1\%$, which amounts to $\mathbf{10 - 30}$k points on a 480×640 image. On accuracy, we have **unanimous** improvement over all supervised models of both datasets. Especially, we outperform strong baselines of ZoeDepth on ScanNet and Metric3D on KITTI360.

4.2 Consistent Depth Estimation

We evaluate on ScanNet. We follow Sec. 4.1 data split but evaluate the **support** frames. Temporal consistent depth is essential for AR applications [39]. Tab. 2

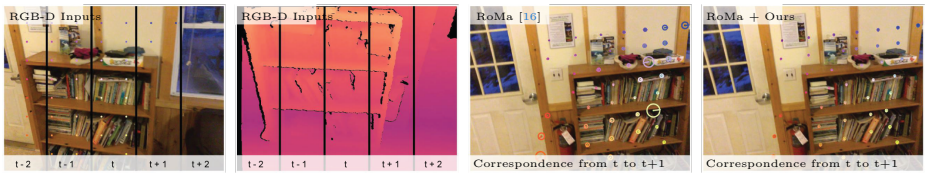


Fig. 9: Self-supervised Correspondence Estimation enabled by our method with RGB-D inputs. The correspondence error is marked by the radius of the circle.

reflects the performance gain by aligning support frames to root with adjustments, which are jointly estimated with camera poses, see Fig. 2 and Fig. 3.

4.3 Self-supervised Correspondence Estimation

Real-world image correspondence label is expensive, *e.g.* KITTI provides only 200 optical flow labels. Existing datasets, such as MegaDepth and ScanNet, require large-scale 3D reconstruction with manual verification. Hence, correspondence estimators can not fine-tune on general RGB-D datasets like NYUv2 [54] or KITTI [19]. But our method enables self-supervised correspondence estimation on RGB-D data when using 3D scoring function Eq. (6). The camera poses are optimized with the point cloud specified by depthmap and correspondence. The accurate pose in turn improves projective correspondence. In Tab. 3, with 5 RGB-D frames, our method improves projective correspondence over inputs. We use the same test split as Sec. 4.1. The evaluation accumulates correspondence of each frame pair. Fig. 8a shows our improvement is **unanimous** over both confident and unconfident estimation. A visual example is in Fig. 9.

4.4 Sparse-view Pose Estimation

Comparison with Optimization-based and Learning-based Poses. Following Sec. 4.1 ScanNet split, we keep root frame and gradually add neighboring frames. In Tab. 4, LightedDepth [80] and ours both use PDC-Net [60] correspondence and ZoeDepth [6] mono-depth. COLMAP [51] uses PDC-Net correspondence. In evaluation, we follow [61] in aligning to GT poses. In Tab. 4, our **zero-shot** pose accuracy significantly outperforms all prior arts, including [23, 58, 64] with ScanNet [14] or ScanNet++ [70] in their training set. See Supp. Tab. 2 for complete comparison from 3 to 9 frames. In Fig. 8, we attribute our superiority to certified global optimality over robust measurements. *E.g.*, Tab. 2 PCK-3 is Eq. (5) with GT depthmap and threshold $\lambda^{\text{2D}} = 3$. More in Supp.

Comparison with NeRF-based Poses. Sparse view NeRF methods optimize NeRF jointly with camera poses, mandating a sophisticated and time-consuming optimization scheme. *E.g.*, Sparf [61], takes one day to optimize the pose and NeRF. Typically, their poses are initialized with COLMAP. Our method provides an alternative initialization with superior performance. In Tab. 5, our initialization achieves better or on-par pose performance than SoTA [61] while only taking ∼3 minutes (Fig. 7). Our lower performance on Replica dataset might be due to

Table 4: Sparse-view Pose Comparison with optimization-based and learning-based methods. We only compare against COLMAP on its success sequences. Please see the complete comparison from 3 to 9 frames in Supp. Tab. 1. Our method performs **zero-shot** testing on ScanNet while outperforming DeepV2D [58], DRO [23], and DUSt3R [64] with ScaNet [14] or ScanNet++ [70] in their training set.

Frames	Method	Zero-shot	Suc. (%)	PCK-3	C3D-3	Rot.	Trans.
5	COLMAP [51]	✓	36.7	0.584	0.863	0.577	1.296
	Ours	✓	100.0	**0.727**	**0.904**	**0.422**	**1.062**
	DeepV2D [58] - ScanNet	✗		0.526	0.805	0.945	1.496
	DeepV2D [58] - NYUv2	✓		0.530	0.771	1.041	1.568
	DeepV2D [58] - KITTI	✓		0.125	0.387	4.908	4.231
	LightedDepth [80]	✓		0.651	0.832	0.469	1.550
	DRO [23] - ScanNet	✗	100.0	0.656	0.853	0.385	1.200
	DRO [23] - KITTI	✓		0.003	0.211	3.610	5.469
	DUSt3R [64] *w.o.* Intrinsic	✗		0.364	0.705	0.487	2.074
	DUSt3R [64] *w.t.* Intrinsic	✗		0.594	0.824	0.570	1.759
	Ours	✓		**0.799**	**0.900**	**0.368**	**1.120**

Table 5: Sparse-view Pose Comparison with NeRF-based methods following [61].

Method	Frames	LLFF [52]		Replica [57]	
		Rot.	Trans.	Rot.	Trans.
BARF [36]	3	2.04	11.6	3.35	16.96
RegBARF [36, 43]		1.52	5.0	3.66	20.87
DistBARF [3, 36]		5.59	26.5	2.36	7.73
SCNeRF [28]		1.93	11.4	0.65	4.12
SPARF [61]		<u>0.53</u>	<u>2.8</u>	**0.15**	**0.76**
Ours		**0.46**	**1.9**	<u>0.52</u>	<u>4.09</u>

ZoeDepth not being trained on synthetic data. Our work suggests the straightforward "first-pose-then-NeRF" scheme can be applicable to short videos.

Certified Global Optimality. In Fig. 8b, our Bundle-RANSAC-Adjustment **always** finds more inliers than groundtruth poses. To our best knowledge, we are the **first** work that extends RANSAC to a multi-view system.

Run-time. In Fig. 8c, we run approximately 3× slower than COLMAP. But both have quadratic complexity. With 3/5/7/9 frames, we take 0.8/2.0/5.3/9.4 minutes on RTX 2080 Ti GPU, while COLMAP uses 0.3/0.9/1.8/3.6 minutes on Intel Xeon 4216 CPU. COLMAP runs sequentially. But our method is highly parallelized. Our core operation Hougn Transform scales up with more GPUs.

5 Conclusion

By revisiting self-supervision with local SfM, we first show self-supervised depth benefits SoTA supervised model with only 5 frames. We have SoTA sparse-view pose accuracy, applicable to NeRF rendering. We have diverse applications including self-supervised correspondence and consistent depth estimation.

Limitation. The NeRF-like triangulation constrains our method from applying to large-scale self-supervised learning. Its efficiency requires improvement.

======Supplementary======
Revisit Self-supervised Depth Estimation with Local Structure-from-Motion

No Author Given

No Institute Given

1 Extended Methodology

1.1 Sampson Scoring Function

In Sec. 3.1, we suggest a Sampson error [40] based scoring function. This specific scoring function is widely adopted in the two-view normalized pose estimation algorithm with RANSAC. Suppose we sample the m_{th} corresponded pixels $\mathbf{p}_i$ and $\mathbf{p}_j$ from the i_{th} and j_{th} frames. Following Eq. (9) and Eq. (7), denote essential matrix and normalized pixels as:

$$\mathbf{E}_{i,j} = [\,\overline{\mathbf{t}}_{i,j}]_{\times}\mathbf{R}_{i,j},\ \overline{\mathbf{p}}_i = \mathbf{K}^{-1}\mathbf{p}_i,\ \overline{\mathbf{p}}_j = \mathbf{K}^{-1}\mathbf{p}_j. \tag{1}$$

The Sampson error is defined as:

$$e(\cdot) = \overline{\mathbf{p}}_j^{\intercal}\mathbf{E}_{i,j}\overline{\mathbf{p}}_i / \sqrt{\|\overline{\mathbf{p}}_j^{\intercal}\mathbf{E}_{i,j}\|_2^2 + \|\mathbf{E}_{i,j}\overline{\mathbf{p}}_i\|_2^2}. \tag{2}$$

We define the Sampson scoring function as:

$$f_{i,j}^{\text{Epp}}(\cdot) = \sum_m e(s_i, s_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j) < \lambda^{\text{Epp}}. \tag{3}$$

The camera scales and normalized poses are used to compute the essential matrix. While Eq. (3) is robust in two-view normalized pose initialization, we find it is a weak constraint for multi-view pose estimation with bundle-adjustment.

1.2 Hough Transform with 3D Scoring Function

In Sec. 4.3 and Fig. 9, with RGB-D inputs, we utilize the 3D scoring function Eq. (6). For a pixel $\mathbf{p}_i$ on frame i, denote its backprojected 3D ray on the image coordinate system of frame j as $\hat{\mathbf{l}}_i$. The ray $\hat{\mathbf{l}}_i$ is determined by two 3D points:

$$\hat{\mathbf{p}}_1 = \mathbf{R}_{i,j}\mathbf{K}^{-1}\mathbf{p}_i + \mathbf{t}_{i,j}, \quad \hat{\mathbf{p}}_2 = \mathbf{t}_{i,j}. \tag{4}$$

The depth of 3D point $\hat{\mathbf{p}}_1$ can be set to arbitrary values. We set it to 1 for convenience. 3D point $\hat{\mathbf{p}}_2$ is the frame i camera origin on frame j. Correspondingly, the backprojected 3D point $\hat{\mathbf{p}}_\pi$ is:

$$\hat{\mathbf{l}}_i^{\intercal}\hat{\mathbf{p}}_\pi = 0,\ \hat{\mathbf{p}}_\pi = \pi^{-1}(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_i). \tag{5}$$

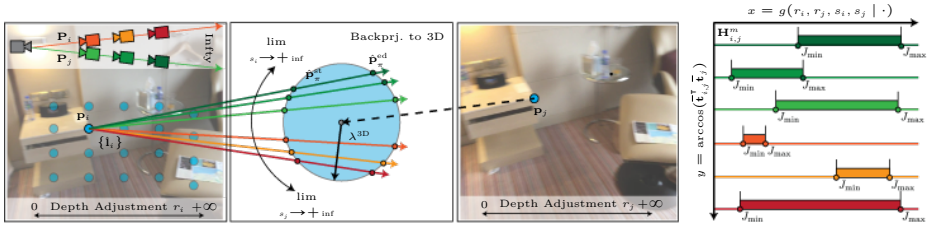


Fig. 1: Two-view Hough Transform on 3D Scoring Function. Pixels $\mathbf{p}_i$ and $\mathbf{p}_j$ are corresponded. Similar to Fig. 5, ablating pose scales map pixel $\mathbf{p}_i$ to a set of 3D rays originated from camera origin, denoted as $\{\hat{\mathbf{l}}_i\}$. To be an inlier, a backprojected 3D point $\hat{\mathbf{p}}_\pi$ has to reside within a sphere centered with $\hat{\mathbf{p}}_j$ with a radius λ^{3D}, *i.e.*, between 3D segments $\hat{\mathbf{p}}_\pi^{\text{st}}$ and $\hat{\mathbf{p}}_\pi^{\text{ed}}$. Different from Fig. 5, with fixed normalized poses, there exists **four** variables to optimize, including the additional frame j depth adjustment r_j.

To be an inlier of the scoring function $f^{\text{3D}}(\cdot)$, we have:

$$\|\hat{\mathbf{p}}_\pi - \hat{\mathbf{p}}_j\|_2 \leqslant \lambda^{\text{3D}},\ \hat{\mathbf{p}}_j = \pi^{-1}(s_i, s_j, r_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_j). \tag{6}$$

Eq. (6) suggests a 3D line and sphere intersection, with the two end-points denoted as $\hat{\mathbf{p}}_\pi^{\text{st}}$ and $\hat{\mathbf{p}}_\pi^{\text{ed}}$. Compare 2D intersection condition Eq. (11) with 3D intersection condition Eq. (6), we find Eq. (6) further relates to the depth adjustment r_j on frame j. Similarly, compare Fig. 5 with Fig. 1, the circle center of 2D scoring function is independent of depth adjustments r_j. The sphere center of the 3D scoring function, however, changes *w.r.t.* depth adjustments. This introduces one additional variable in Hough Transform, *i.e.*, the depth adjustment r_j at frame j, also shown in Fig. 1. The additional dimensionality creates an excessively high time and space complexity excessively for Hough Transform. Hence, we are unable to optimize depth adjustment in 3D scoring function. One viable solution is to first execute with 2D scoring function, fix the adjustments, and then employ the 3D scoring function. We did not experiment with this strategy since we already had competitive performance. When with RGB-D inputs, we use 3D scoring function as groundtruth depth does not require any scale adjustment. The rest is similar to Sec. 3.1.2 and skipped.

1.3 Proof of Eq. (8).

From Eq. (7), the relative pose translation magnitude $s_{i,j}$ is:

$$s_{i,j}^2 = s_i^2 + s_j^2 + k_\theta \cdot s_i s_j,\ k_\theta = 2\bar{\mathbf{t}}_j^\intercal \mathbf{R}_j \mathbf{R}_i^{-1} \bar{\mathbf{t}}_i. \tag{7}$$

The relative pose normalized translation vector $\bar{\mathbf{t}}_{i,j}$ is:

$$\bar{\mathbf{t}}_{i,j} = -s_i/s_{i,j} \mathbf{R}_j \mathbf{R}_i^{-1} \bar{\mathbf{t}}_i + s_j/s_{i,j} \bar{\mathbf{t}}_j. \tag{8}$$

Illustrate with s_i, when $s_i \to +\infty$, we have:

$$\lim_{s_i \to +\inf} s_i/s_{i,j} = 1,\ \lim_{s_i \to +\inf} s_j/s_{i,j} = 0. \tag{9}$$

Combined, we have:

$$\lim_{s_i \to +\inf} \bar{\mathbf{t}}_{i,j} = -\mathbf{R}_j \mathbf{R}_i^{-1} \bar{\mathbf{t}}_i. \tag{10}$$

The other case of Eq. (8) follows a similar logic.

1.4 Proof of Eq. (12) and Eq. (13).

1.4.0.1 Notations. For simplicity, the relative pose $\mathbf{P}_{i,j}$ between frame i and j is defined as $\mathbf{P}_{i,j} = \left[\mathbf{R}\ s \cdot \bar{\mathbf{t}}\right]$. Denote the 2D pixel locations on frame i as $\mathbf{p}$. Its projection is set to $\mathbf{q}$, *i.e.*, the variable $\mathbf{p}_\pi$ in Sec. 3.1.2. The projection is:

$$d'\mathbf{q} = d' \left[q_x\ q_y\ 1\right]^\intercal = r \cdot d\mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{p} + s\mathbf{K}\bar{\mathbf{t}}. \tag{11}$$

Variable r is the depth adjustment on frame i. Variable d is the depth of pixel $\mathbf{p}$. The pixel location q_x and q_y are:

$$q_x = \frac{r \cdot d\mathbf{m}_1^\intercal \mathbf{p} + s \cdot x}{r \cdot d\mathbf{m}_3^\intercal \mathbf{p} + s \cdot z},\ q_y = \frac{r \cdot d\mathbf{m}_2^\intercal \mathbf{p} + s \cdot y}{r \cdot d\mathbf{m}_3^\intercal \mathbf{p} + s \cdot z}. \tag{12}$$

The remaining variables are defined as follows:

$$\left[\mathbf{m}_1\ \mathbf{m}_2\ \mathbf{m}_3\right]^\intercal = \mathbf{K}\mathbf{R}\mathbf{K}^{-1},\ s \cdot \left[x\ y\ z\right]^\intercal = s \cdot \mathbf{K}\bar{\mathbf{t}}. \tag{13}$$

Here, we slightly abuse the notation of x and y without implying the Hough Transform locations. The mapping function $J(\cdot)$ from projection to camera scale is hence:

$$s = J(\overline{\mathbf{P}}, r \cdot d, \mathbf{q}) = r \cdot \frac{d(\mathbf{m}_1^\intercal \mathbf{p} - q_x \mathbf{m}_3^\intercal \mathbf{p})}{z \cdot q_x - x}. \tag{14}$$

There exists another analytical mapping from the axis-y direction. We skip this for simplicity.

Proof of Corollary 2. From Eq. (14), it is obvious that:

$$s = J(\overline{\mathbf{P}}, r \cdot d, \mathbf{q}) = r \cdot J(\overline{\mathbf{P}}, d, \mathbf{q}). \tag{15}$$

Proof of Corollary 1. We decompose the relative camera pose $\mathbf{P}$ from frame i and j into a rotation movement $\mathbf{P}^r$ followed by a translation movement $\mathbf{P}^t$:

$$\mathbf{P}^r = \left[\mathbf{R}\ \mathbf{0}\right],\ \mathbf{P}^t = \left[\mathbf{E}\ s \cdot \mathbf{R}\bar{\mathbf{t}}\right],\ \mathbf{P} = \mathbf{P}^t\mathbf{P}^r. \tag{16}$$

Correspondingly, we introduce an intermediate pixel $\mathbf{q}^r$ as a consequence of the pure rotation movement. That is to say, between pixels $\mathbf{p}$ and $\mathbf{q}^r$ are a pure rotation movement. And between pixels $\mathbf{q}^r$ and $\mathbf{q}$ are a pure translation movement. The pixels $\mathbf{q}^r$ and $\mathbf{q}$ have the following relationship:

$$\mathbf{q}^r = \lim_{s \to 0} \mathbf{q} = \lim_{r \cdot d \to +\infty} \mathbf{q}. \tag{17}$$

Eq. (17) is obvious. We skip its proof. We introduce it to illustrate two facts. First, from Fig. 5, the intersection point of all epipolar lines $\{\mathbf{l}_i\}$ is $\mathbf{q}^r$, *i.e.*, the projected pixel from pure rotation. Second, the pixel $\mathbf{q}^r$ and $\mathbf{q}$ formulates a pure

translation movement. It simplifies our proof if we replace the pixel $\mathbf{p}$ to pixel $\mathbf{q}^r$ in Eq. (12). Due to a pure translation movement, the $\mathbf{KRK}^{-1}$ is an identity matrix. We have:

$$q_x = \frac{r \cdot d^r \cdot q_x^r + s \cdot x}{r \cdot d^r + s \cdot z},\ q_y = \frac{r \cdot d^r \cdot q_y^r + s \cdot y}{r \cdot d^r + s \cdot z}. \tag{18}$$

We compute the squared magnitude of the vector $\|\overrightarrow{\mathbf{q}^r\mathbf{q}}\|_2^2$:

$$\begin{aligned} \|\overrightarrow{\mathbf{q}^r\mathbf{q}}\|_2^2 &= (q_x - q_x^r)^2 + (q_y - q_y^r)^2 \\ &= \frac{(s \cdot x)^2 + (s \cdot y)^2}{(r \cdot d^r + s \cdot z)^2} = \frac{x^2 + y^2}{(r \cdot d^r/s + z)^2}. \end{aligned} \tag{19}$$

Meanwhile, we underlyingly require the pixel $\mathbf{q}$ to be visible in both frames. This requires a positive depth:

$$r \cdot d^r/s + z > 0. \tag{20}$$

Combining Eq. (19) and Eq. (20), we conclude that the vector $\|\overrightarrow{\mathbf{q}^r\mathbf{q}}\|_2^2$ monotonously increases as translation magnitude s increases. It fits the intuition that the projected pixel $\mathbf{q}$ moves further as the camera translation magnitude increases. To prove the corollary 1, we have:

$$\|\overrightarrow{\mathbf{q}^r\mathbf{q}^{\text{st}}}\|_2^2 \leqslant \|\overrightarrow{\mathbf{q}^r\mathbf{q}}\|_2^2 \leqslant \|\overrightarrow{\mathbf{q}^r\mathbf{q}^{\text{ed}}}\|_2^2. \tag{21}$$

The s monotonously increases with vector magnitude, then:

$$J(\overline{\mathbf{P}}, r \cdot d, \mathbf{q}^{\text{st}}) \leqslant J(\overline{\mathbf{P}}, r \cdot d, \mathbf{q}) \leqslant J(\overline{\mathbf{P}}, r \cdot d, \mathbf{q}^{\text{ed}}). \tag{22}$$

We apologize for the abuse of the notation. The variable $\mathbf{q}^{\text{ed}}$ refers to $\mathbf{p}_\pi$ in 3.1.2. We next address some corner cases unattended in main paper due to space issues.

- The Epipolar line and the circle do not intersect:
  We exclude Hough Transform under this condition since there are no inlier projected pixels.
- Eq. (20) does not hold:
  To be visible in both views, Eq. (20) has to hold. Hence it provides an additional bound to $J_{\min}$ and $J_{\max}$, if needed.

Compute Start and End Intersected Points. To compute $\mathbf{p}_\pi^{\text{st}}$ and $\mathbf{p}_\pi^{\text{ed}}$ in Sec. 3.1.2, we follow the analytical line-circle intersection solution.

2 Extended Experiments

2.1 Implementation Details

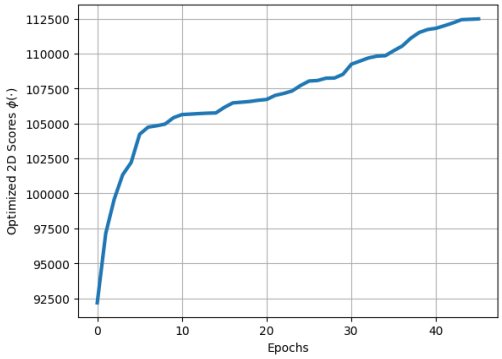
Sec. 3.1 Camera Pose Estimation. The proposed pose estimation algorithm runs on GPU devices. Per frame pair, we extract the top $K = 128$ normalized poses to formulate the candidate pool $\overline{\mathcal{Q}}$. We sample $M = 10,000$ points per

Table 1: Extended Sparse-view Pose Comparison with optimization-based and learning-based methods on ScanNet. We extend main paper Tab. 4 across 3 to 9 frames.

Frames	Method	Zero-shot	Suc. (%)	PCK-3	C3D-3	Rot.	Trans.
3	COLMAP [51]	✓	10.0	0.379	0.686	1.055	1.765
	Ours	✓	100.0	**0.750**	**0.890**	**0.210**	**0.624**
	DeepV2D [58] - ScanNet	✗	100.0	0.591	0.863	0.319	0.907
	DeepV2D [58] - NYUv2	✓		0.619	0.839	0.381	0.946
	LightedDepth [80]	✓		0.742	0.874	0.380	1.127
	DRO [23] - ScanNet	✗		0.757	0.883	0.368	0.881
	DRO [23] - KITTI	✓		0.006	0.258	2.043	3.435
	DUSt3R [64] *w.o.* Intrinsic	✗		0.409	0.770	0.356	1.355
	DUSt3R [64] *w.t.* Intrinsic	✗		0.574	0.827	0.467	1.245
	Ours	✓		**0.858**	**0.918**	**0.275**	**0.820**
5	COLMAP [51]	✓	36.7	0.584	0.863	0.577	1.296
	Ours	✓	100.0	0.727	0.904	0.422	1.062
	DeepV2D [58] - ScanNet	✗	100.0	0.526	0.805	0.945	1.496
	DeepV2D [58] - NYUv2	✓		0.530	0.771	1.041	1.568
	DeepV2D [58] - KITTI	✓		0.125	0.387	4.908	4.231
	LightedDepth [80]	✗		0.651	0.832	0.469	1.550
	DRO [23] - ScanNet	✗		0.656	0.853	0.385	1.200
	DRO [23] - KITTI	✓		0.003	0.211	3.610	5.469
	DUSt3R [64] *w.o.* Intrinsic	✗		0.364	0.705	0.487	2.074
	DUSt3R [64] *w.t.* Intrinsic	✗		0.594	0.824	0.570	1.759
	Ours	✓		**0.799**	**0.900**	**0.368**	**1.120**
7	COLMAP [51]	✓	70.6	0.571	0.852	0.596	1.521
	Ours	✓	100.0	**0.723**	**0.881**	**0.470**	**1.383**
	DeepV2D [58] - ScanNet	✗	100.0	0.395	0.667	1.719	2.673
	DeepV2D [58] - NYUv2	✓		0.434	0.674	1.830	2.740
	LightedDepth [80]	✓		0.560	0.801	0.551	1.909
	DRO [23] - ScanNet	✗		0.567	0.790	0.619	1.744
	DRO [23] - KITTI	✓		0.002	0.191	5.040	7.375
	DUSt3R [64] *w.o.* Intrinsic	✗		0.343	0.659	0.558	2.428
	DUSt3R [64] *w.t.* Intrinsic	✗		0.558	0.804	0.561	2.017
	Ours	✓		**0.739**	**0.873**	**0.458**	**1.428**
9	COLMAP [51]	✓	86.7	0.558	0.812	0.756	2.281
	Ours	✓	100.0	**0.687**	**0.852**	**0.538**	**1.675**
	DeepV2D [58] - ScanNet	✗	100.0	0.303	0.569	4.279	4.573
	DeepV2D [58] - NYUv2	✓		0.349	0.590	5.509	4.654
	LightedDepth [80]	✓		0.487	0.766	0.631	2.322
	DRO [23] - ScanNet	✗		0.483	0.710	1.313	3.093
	DRO [23] - KITTI	✓		0.002	0.174	6.397	9.142
	DUSt3R [64] *w.o.* Intrinsic	✗		0.333	0.631	0.615	2.740
	DUSt3R [64] *w.t.* Intrinsic	✗		0.568	0.801	0.584	2.278
	Ours	✓		**0.695**	**0.850**	**0.535**	**1.689**

frame pair. We exclude correspondence with a confidence lower than 0.2. We set λ^{2D} to 2 pixels and λ^{3D} to 0.025 meter. The Hough matrix $\mathbf{H}$ resolution is set to 100×200. The maximum Hough transform on axis-x, *i.e.*, the $x_{\max}$ is set to 1 meter. The BA optimization iterations are set to $T = 200$ at a learning rate of $5e^{-4}$ with an Adam optimizer [29].

Sec. 3.2 Triangulation. For the frustum radiance field $\mathbf{V}$, we configure its resolution as $240 \times 320 \times 128$ given a ScanNet image resolution of 480×640. We scale the depth consistent loss L_D by a factor of 0.01. This is due to the small magnitude of correspondence consistent loss L_C which is defined over normalized pixel coordinates, i.e., pixels divided by focal length. In triangulation,



(a) Inlier score $f^{2D}(\cdot)$ / ZoeDepth [6] / PDC-Net [60]

Fig. 2: Scores *w.r.t* Optimization Epochs. The inlier scores always rise throughout the optimization. With 5 frames, our algorithm terminates on average at 23.5 epochs. [Key: Inlier Score / Monodepth Estimator / Correspondence Estimator]

we optimize the Radiance Field $\mathbf{V}$ using the Adam optimizer with a learning rate of $1e^{-4}$ over $80,000$ iterations. In testing, we set the geometric verification threshold λ^c to 0.01 meter.

Sec. 3.3 Geometric Verification. The consistent threshold λ^c is set to 0.01 meters, and the minimum consistent view number n^c is set to 2 frames.

2.2 Additional Comparisons

Prior public benchmarks are inapplicable to measuring sparse-view pose estimation performance. DeepV2D [58], DRO [23], and LightedDepth [80] only report two-view relative pose performance. DUSt3R [64] reports on classic SfM and SLAM benchmarks with significantly more view variations. Only NeRF-based [61] methods benchmark sparse-view pose performance and have been included in the main paper Tab. 5. For a comprehensive comparison, we additionally experiment on the ScanNet with other SoTA methods.

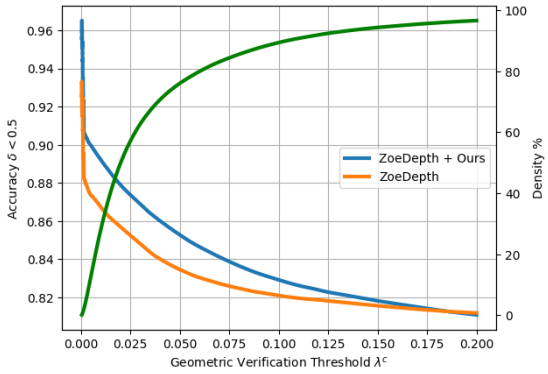
In Tab. 1, we divide the comparisons into two groups. The group above the dashed lines only evaluates the COLMAP successfully executed sequences. The group below the dashed lines includes all ScanNet test sequences. For two-view methods LightedDepth [80] and DRO [23], we repetitively apply two-view pose estimation between each support and root frame. For DUSt3R [64] with intrinsic, we set the intrinsic and disable its update in optimization. In Tab. 1, we achieve **unanimous** improvement on all frame numbers with **substantial** margin.

2.3 Additional Ablation Studies

Scores *w.r.t* Optimization Epochs. As shown in Fig. 2, inlier scores consistently rise during optimization. Despite optimizing over discrete local optimal sets, *i.e.*, predefined candidate pool $\overline{\mathcal{Q}}$, the convergence curve exhibits similar

Table 2: Ablation on Geometric Verification on ScanNet dataset. We additionally report the main paper Tab. 1 without applying the Sec. 3.3 Geometric Verification, *i.e.*, the radiance field intermediate depthmap with full density. See visual examples in the main paper Fig. 2 and supplementary Fig. 4.

Method	Geo. Ver.	Density	$\delta_{0.5}$	δ_1	SI_{log}	A.Rel	S.Rel	RMS	RMS_{log}
ZoeDepth [6]	✗	Dense	**0.808**	**0.937**	**9.592**	**0.074**	**0.028**	**0.211**	**0.102**
↳ Ours			0.800	0.925	11.006	0.082	0.032	0.227	0.118
ZoeDepth [6]	✓	9.1%	0.877	0.963	6.655	0.056	0.016	0.154	0.075
↳ Ours			**0.902**	**0.976**	**5.901**	**0.050**	**0.014**	**0.149**	**0.070**
ZeroDepth [37]	✗	Dense	0.557	0.771	20.124	0.166	0.118	0.427	0.211
↳ Ours			**0.595**	**0.809**	**17.744**	**0.147**	**0.094**	**0.388**	**0.190**
ZeroDepth [37]	✓	5.6%	0.641	0.834	12.860	0.124	0.086	0.337	0.152
↳ Ours			**0.686**	**0.877**	**9.463**	**0.106**	**0.067**	**0.295**	**0.133**
Metric3D [71]	✗	Dense	**0.747**	**0.905**	12.325	**0.089**	**0.043**	**0.259**	**0.129**
↳ Ours			0.746	0.883	**12.273**	0.109	0.089	0.306	0.147
Metric3D [71]	✓	2.6%	0.804	0.946	6.708	0.067	0.020	0.150	0.084
↳ Ours			**0.854**	**0.968**	**4.170**	**0.055**	**0.014**	**0.125**	**0.068**



(a) Accuracy $\delta < 0.5$ / ZoeDepth [6] / PDC-Net [60]

Fig. 3: Depth Density & Accuracy *w.r.t* Geometric Verification Threshold λ^c. Tab. 1 reports triangulated accuracy and density at $\lambda^c = 0.01$. From the plot, if set the threshold λ^c to a larger value, we maintain significant improvement while preserving a significantly larger density, *e.g.*, $\lambda^c = 0.05$. The unit is in meters. [Key: Accuracy $\delta < 0.5$ / Monodepth Estimator / Correspondence Estimator]

trends with smooth manifold optimization. We attribute this to a sufficiently large candidate pool size, set at $K = 128$ per frame in our experiment.

Geometric Verification Threshold. In Tab. 2, we benchmark the triangulated depth without geometric verification, *i.e.*, depth performance without filtering. On ZeroDepth [37], we outperform the supervised inputs even without geometric verification. On ZoeDepth [6] and Metric3D [71], we outperform the supervised inputs after the geometric verification, suggesting its necessity. Across all cases, geometric verification improves performance. In Fig. 3, we ablate different triangulation threshold λ^c values. From Fig. 3, our method maintains improvement at a much higher density than reported in the main paper Tab. 1.

2.4 Evaluation Metrics

Depth Related Metrics. In Tab. 1, we follow [5] for most evaluation metrics. As the accuracy metric δ_1 saturates, [46] introduces a stricter counterpart, $\delta_{0.5}$. The term $\text{SI}_{\log}$ is a scale-invariant metric [46]. As a result, the scale adjustment on support frames does not impact metric $\text{SI}_{\log}$, as in main paper Tab. 1 row 5 and 6.

Correspondence Related Metrics. In Tab. 3, we follow [60] in evaluation metrics. The metric AEPE refers to average end-point-error, *i.e.*, the L2 norm between estimated and groundtruth correspondence. PCK-K refers to the percentage of correct correspondence if its end-point-error is smaller than a given pixel threshold K.

Pose Related Metrics. In Tab. 4 and Tab. 5, we follow [61] in using Rot. and Trans.. The metric Rot. measures the average difference in degree after alignment. Similarly, metric Trans. measures the translation magnitude after multiplied by $\times 100$. We additionally report PCK-K and C3D-K. The metric PCK-K compares the projected correspondence using the groundtruth depthmaps. Its accuracy is only related to the estimated camera poses. The threshold is set to 3 pixels Similarly, the metric C3D-K compares the backprojected 3D points using the groundtruth depthmap with estimated poses. The threshold is set to 5 centimeters.

2.5 Visualization

We additionally visualize the self-supervised depth estimation, consistent depth estimation, and self-supervised correspondence estimation in Fig. 4, Fig. 5, and Fig. 6, respectively. Please see the captions for detailed analysis.

Self-supervised Depth Estimation. In Fig. 4, the Radiance Field (RF) rendered dense depthmap has poor visual quality, especially around object boundaries. This is expected, as the RF essentially applies triangulation, relying on consistent multi-view depth and correspondence estimation. Yet, both monocular depthmap and correspondence estimation exhibit lower quality around borders. When the inconsistency is significant, the triangulated results become underdefined, leading to noisy artifacts.

Self-supervised Consistent Depth. When generating Fig. 5, we produce groundtruth correspondence via groundtruth pose and depthmap. Using this correspondence, we composite multi-frame depthmaps with groundtruth correspondence depthmaps from various frames and subsequently backproject them to the root frame. Each column corresponds to sampling from a different frame's depthmap, and the final image is composed over all input frames.

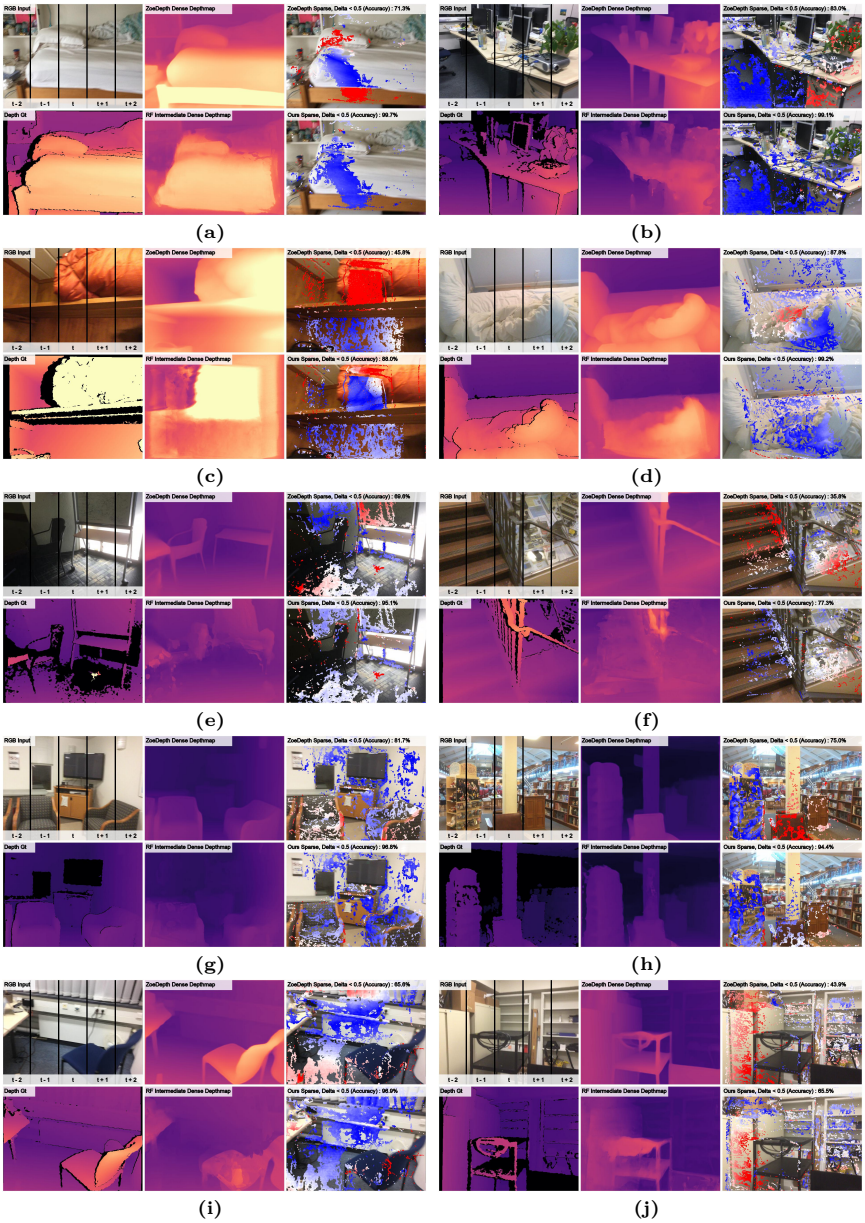


Fig. 4: Self-supervised Depth Estimation. Qualitative comparison with supervised model ZoeDepth [6]. We represent the depth quality of each pixel with the metric **A.Rel**. Blue to red scatter indicates small to large errors, ranging between 0.0 and 0.2.

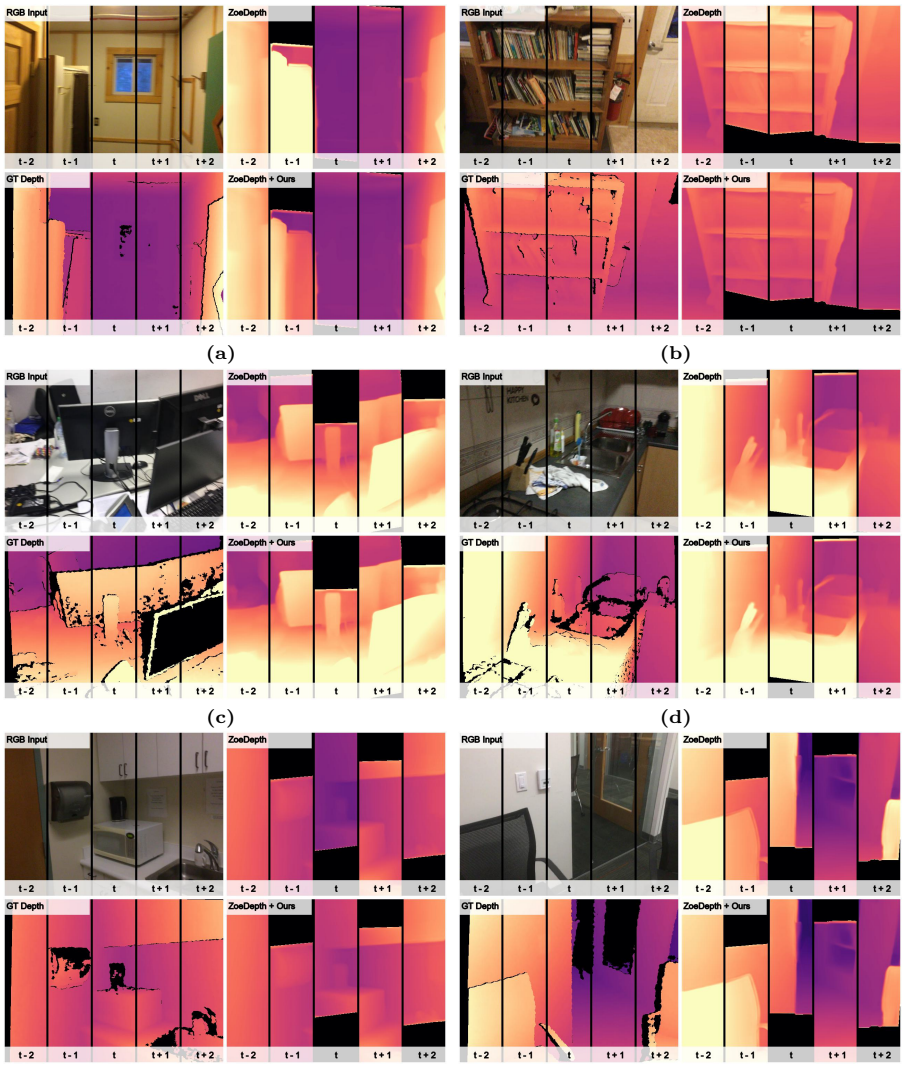


Fig. 5: Consistent Depth Estimation. Qualitative comparison with input SoTA monocular depth estimator ZoeDepth [6].

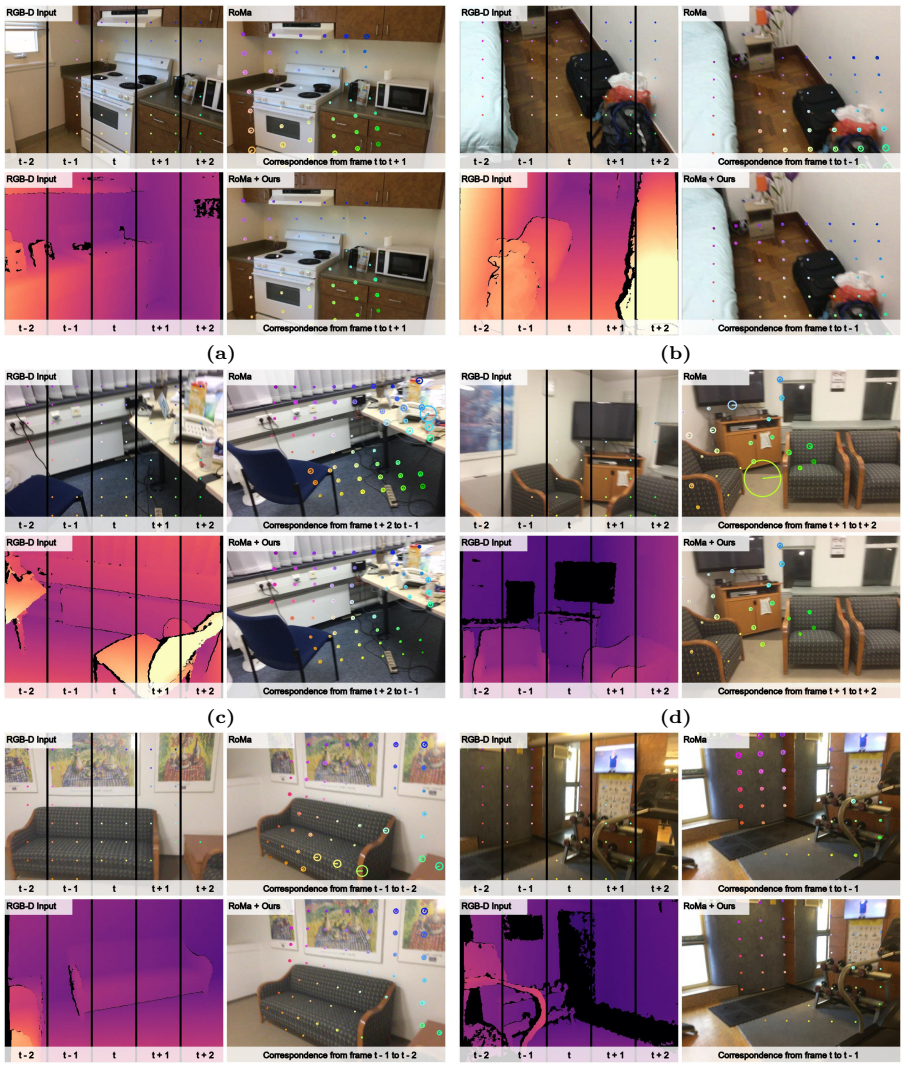


Fig. 6: Self-supervised Correspondence Estimation. Qualitative comparison with input SoTA correspondence estimator RoMa [16].

References

1. de Agapito, L., Hayman, E., Reid, I.: Self-calibration of a rotating camera with varying intrinsic parameters. In: BMVC (1998) 1
2. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM (2011) 1
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022) 14
4. Beder, C., Steffen, R.: Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In: Joint Pattern Recognition Symposium 3
5. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: CVPR (2021) 10, 8
6. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) 1, 3, 11, 12, 13, 6, 7, 9, 10
7. Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. NeurIPS (2019) 4
8. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. PAMI (2010) 3
9. Butime, J., Gutierrez, I., Corzo, L.G., Espronceda, C.F.: 3d reconstruction methods, a survey. In: VISAPP (2006) 1
10. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019) 4
11. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: ICCV (2019) 4
12. Clark, R., Bloesch, M., Czarnowski, J., Leutenegger, S., Davison, A.J.: Ls-net: Learning to solve nonlinear least squares for monocular stereo. ECCV (2018) 4
13. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: CVPR (2011) 1
14. Dai, A., Chang, A., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: PAMI (2017) 11, 12, 13, 14
15. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR (2022) 4
16. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: Roma: Revisiting robust losses for dense feature matching. arXiv preprint arXiv:2305.15404 (2023) 12, 13, 11
17. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 10
18. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: CVPR (2018) 1, 10
19. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. IJRR (2013) 13
20. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 1

21. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019) 1, 4
22. Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: ICCV (2019) 1, 12
23. Gu, X., Yuan, W., Dai, Z., Zhu, S., Tang, C., Dong, Z., Tan, P.: Dro: Deep recurrent optimizer for video to depth. IEEE Robotics and Automation Letters (2023) 4, 13, 14, 5, 6
24. Guizilini, V., Ambruș, R., Chen, D., Zakharov, S., Gaidon, A.: Multi-frame self-supervised depth with transformers. In: CVPR (2022) 4
25. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 7
26. Izquierdo, S., Civera, J.: Sfm-ttr: Using structure from motion for test-time refinement of single-view depth networks. In: CVPR (2023) 4
27. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR (2014) 12
28. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: CVPR (2021) 4, 14
29. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 5
30. Kuznietsov, Y., Proesmans, M., Van Gool, L.: Comoda: Continuous monocular depth adaptation using past experiences. In: WACV (2021) 4
31. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019) 1
32. Lepetit, V., Moreno-Noguer, F., Fua, P.: Ep n p: An accurate o (n) solution to the p n p problem. IJCV (2009) 3
33. Li, H.: A practical algorithm for l triangulation with outliers. In: CVPR (2007) 3
34. Li, H., Hartley, R.: Five-point motion estimation made easy. In: ICPR (2006) 3, 5
35. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 11, 12
36. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: ICCV (2021) 4, 14
37. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. arXiv preprint arXiv:2303.11328 (2023) 11, 12, 7
38. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004) 3
39. Luo, X., Huang, J.B., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation. ToG (2020) 4, 12
40. Luong, Q.T., Faugeras, O.D.: The fundamental matrix: Theory, algorithms, and stability analysis. IJCV (1996) 6, 1
41. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: CVPR (2018) 4
42. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM (2021) 2
43. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022) 4, 14

44. Niu, L., Cong, W., Liu, L., Hong, Y., Zhang, B., Liang, J., Zhang, L.: Making images real again: A comprehensive survey on deep image composition. arXiv preprint arXiv:2106.14490 (2021) 1
45. Olsson, C., Eriksson, A., Hartley, R.: Outlier removal using duality. In: CVPR (2010) 3
46. Piccinelli, L., Sakaridis, C., Yu, F.: idisc: Internal discretization for monocular depth estimation. In: CVPR (2023) 1, 8
47. Pillai, S., Ambruş, R., Gaidon, A.: Superdepth: Self-supervised, super-resolved monocular depth estimation. In: ICRA (2019) 4
48. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. PAMI 1
49. 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) 4
50. Sarlin, P.E., Lindenberger, P., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. PAMI (2023) 1
51. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 1, 3, 10, 13, 14, 5
52. Shafiei, M., Bi, S., Li, Z., Liaudanskas, A., Ortiz-Cayon, R., Ramamoorthi, R.: Learning neural transmittance for efficient rendering of reflectance fields (2021) 14
53. Shu, C., Yu, K., Duan, Z., Yang, K.: Feature-metric loss for self-supervised learning of depth and egomotion. In: ECCV (2020) 4
54. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. ECCV (2012) 13
55. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: Siggraph (2006) 1
56. Spencer, J., Russell, C., Hadfield, S., Bowden, R.: Kick back & relax: Learning to reconstruct the world by watching slowtv. In: ICCV (2023) 4
57. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019) 14
58. Teed, Z., Deng, J.: Deepv2d: Video to depth with differentiable structure from motion. In: ICLR (2020) 3, 4, 13, 14, 5, 6
59. Tiwari, L., Ji, P., Tran, Q.H., Zhuang, B., Anand, S., Chandraker, M.: Pseudo rgb-d for self-improving monocular slam and depth prediction. In: ECCV (2020) 4
60. Truong, P., Danelljan, M., Timofte, R., Van Gool, L.: Pdc-net+: Enhanced probabilistic dense correspondence network. PAMI (2023) 3, 11, 12, 13, 6, 7, 8
61. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: CVPR (2023) 3, 4, 10, 12, 13, 14, 6, 8
62. Tuytelaars, T., Mikolajczyk, K., et al.: Local invariant feature detectors: a survey. Foundations and trends® in computer graphics and vision (2008) 3
63. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: CVPR (2017) 4
64. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. arXiv preprint arXiv:2312.14132 (2023) 3, 13, 14, 5, 6

65. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021) 10
66. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: ICCV (2019) 4
67. Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth. In: CVPR (2021) 4
68. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV (2013) 1, 3
69. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. arXiv preprint arXiv:2401.10891 (2024) 4
70. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV (2023) 13, 14
71. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: ICCV (2023) 11, 12, 7
72. Yuan, W., Gu, X., Dai, Z., Zhu, S., Tan, P.: New crfs: Neural window fully-connected crfs for monocular depth estimation. In: CVPR (2022) 1
73. Zhan, H., Garg, R., Weerasekera, C.S., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: CVPR (2018) 4
74. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE multimedia **19**(2), 4–10 (2012) 1
75. Zhang, Z., Cole, F., Tucker, R., Freeman, W.T., Dekel, T.: Consistent depth of moving objects in video. TOG (2021) 4
76. Zhao, W., Liu, S., Shu, Y., Liu, Y.J.: Towards better generalization: Joint depth-pose learning without posenet. In: CVPR (2020) 4
77. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017) 4
78. Zhou, Z., Dong, Q.: Two-in-one depth: Bridging the gap between monocular and binocular self-supervised depth estimation. In: ICCV (2023) 4
79. Zhu, S., Brazil, G., Liu, X.: The edge of depth: Explicit constraints between segmentation and depth. In: CVPR (2020) 1, 4
80. Zhu, S., Liu, X.: Lighteddepth: Video depth estimation in light of limited inference view angles. In: CVPR (2023) 3, 4, 8, 12, 13, 14, 5, 6
81. Zhu, S., Liu, X.: Pmatch: Paired masked image modeling for dense geometric matching. In: CVPR (2023) 12