

Article

Object-Centric Seamless Pose Estimation in Multi-Object Scenes by Scale Alignment of Ray Diffusion and Iterative Closest Point [†]

YeonChang Jeong , Dong-Uk Seo , Kwanwoo Park  and Soon-Yong Park * 

School of Electronic and Electrical Engineering, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, Republic of Korea; jdd9695@naver.com (Y.J.); sseodw@naver.com (D.-U.S.); kevin2760@naver.com (K.P.)

* Correspondence: sypark@knu.ac.kr

[†] This article is a revised and expanded version of a paper entitled “Camera Pose Estimation in Multi-Object Scenes Using Ray Diffusion and Point Cloud Alignment”, which was presented at 2025 40th International Conference on Image and Vision Computing New Zealand (IVCNZ), 19–21 November 2025, Wellington, New Zealand.

Abstract

Robust estimation of camera trajectories from unconstrained image sequences remains a fundamental problem in computer vision and robotics. Recently, a diffusion-based camera tracking network has shown strong performance in sparse-view and single-object-centric settings, where a consistent object is observed across frames. However, when multiple objects appear sequentially in a video, the initially observed object may disappear as the sequence progresses, which prevents maintaining the “single-object-centric” paradigm across all frames and degrades pose estimation when the conventional method is applied to the multi-object sequence. In this work, we propose an object-centric camera pose estimation framework that handles such sequences by partitioning a video into object-level sub-scenes. As a baseline network, Ray Diffusion is applied to single-object sub-scenes, while frame-to-frame camera motion in multi-object sub-scenes is estimated using monocular video depth, object masks, and point cloud alignment using Iterative Closest Point (ICP). Since the domain of pose estimation from different sub-scenes is inconsistent in terms of pose scale, it requires seamless concatenation of pose estimation results through all sub-scenes. In this regard, we introduce a scale alignment strategy based on reprojection error minimization. This enables the pose estimates from individual sub-scenes to be integrated into a single and seamless camera trajectory. We evaluate the proposed method on a newly collected indoor dataset consisting of 40 multi-object video sequences. Experimental results compare our camera trajectory estimation with both the diffusion-based method and the state-of-the-art visual SLAM methods.

Keywords: camera pose estimation; monocular depth estimation; point cloud alignment



Academic Editor: Christos Bouras

Received: 20 May 2026

Revised: 23 June 2026

Accepted: 24 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Camera pose estimation from unconstrained image sequences is a long-standing problem in computer vision and robotics. The goal of pose estimation is to find the six degrees of freedom (6-DoF) of the camera, comprising its three-dimensional position and orientation, for each frame in a sequence. Accurate camera pose estimation provides the geometric basis for modeling spatial relationships across views and enables reconstruction of the three-dimensional scene structure.

Accurately recovering these spatial relationships and reconstructing the 3D scene are critical for a wide range of applications. For example, neural rendering techniques such as NeRF [1] and 3D Gaussian Splatting [2] require precise camera poses to synthesize photorealistic images from novel viewpoints. In robotics, pose accuracy supports Simultaneous Localization and Mapping (SLAM) [3] and autonomous navigation, where even small estimation errors can accumulate and cause substantial drift. Similarly, in AR/VR environments, accurate pose information is essential to maintain spatial alignment between real and virtual content, and in large-scale 3D reconstruction, reliable poses are indispensable for preserving the geometric fidelity of reconstructed scenes.

When there are sufficient visual overlaps between images, Structure-from-Motion (SfM) pipelines such as COLMAP [4] are capable of achieving highly accurate pose estimation and dense 3D reconstruction. When enhanced with modern local feature extraction and matching methods such as SuperPoint [5] and SuperGlue [6], SfM remains one of the most reliable methods for recovering camera pose under favorable conditions. However, the success of SfM is critically dependent on robust feature matching. Its performance degrades significantly when viewpoint changes are large, textureless regions dominate the scene, or the number of available images is limited, underscoring its inherent sensitivity to visual overlap and feature distinctiveness.

To overcome these limitations, learning-based approaches have recently been explored. Among them, Ray Diffusion [7] has introduced a strong capability in estimating relative camera poses and recovering intrinsics from sparsely sampled, object-centric image collections. This framework performs even in scenarios where traditional SfM pipelines fail. However, Ray Diffusion is inherently designed for single-object scenes, and its performance declines significantly in multi-object scenarios.

In this study, we present a camera pose estimation framework that extends Ray Diffusion by integrating monocular depth estimation and Iterative Closest Point (ICP) [8] alignment, enabling robust pose recovery in multi-object image sequences. In our previous study [9], we have proposed to divide a multi-object video sequence into several sub-scenes to solve the inherent problem of Ray Diffusion. In this paper, the proposed method is evaluated against four state-of-the-art pose estimation frameworks across six evaluation metrics and a completeness criterion. Furthermore, to evaluate the performance of the proposed method under varying experimental conditions, two ablation studies are conducted. The first study is done without scale alignment between sub-scenes, whereas the second study investigates the impact of varying the image patch size utilized to select the reference 3D point for scale alignment.

An overview of the proposed pipeline is shown in Figure 1. The dataset is organized such that a single object initially appears, followed by frames where two or more objects coexist, resulting in alternating single- and multi-object sub-scenes. To facilitate pose integration, we define a one-frame overlap at each sub-scene boundary. Consequently, two distinct strategies are applied:

- Single-object sub-scenes: The original Ray Diffusion framework is applied for relative pose estimation, using the official pretrained model in an inference-only manner without additional training.
- Multi-object sub-scenes: When multiple objects are present, direct application of Ray Diffusion yields low accuracy. To address this, monocular depth is estimated using Video Depth Anything (VDA) [10], object masks are extracted via Segment Anything Model2 (SAM2) [11], depth maps are converted into object-specific 3D point clouds, and frame-to-frame transformations are recovered using Generalized ICP (GICP) [12].

Although the above strategies enable reliable pose estimation within each sub-scene, the resulting camera trajectories are defined with different translation scales and therefore

cannot be directly integrated into a single global trajectory. To resolve this inconsistency, we introduce an unknown translation scale for each single-object sub-scene. The scale is estimated by minimizing the reprojection error of a reference 3D point that is propagated through the chained frame-to-frame transformations across neighboring multi-object sub-scenes. By optimizing this reprojection consistency, the translation scale of each single-object sub-scene becomes aligned with that of the corresponding multi-object sub-scene. As a result, the independently estimated sub-scene trajectories can be sequentially composed into a unified and geometrically coherent camera trajectory.

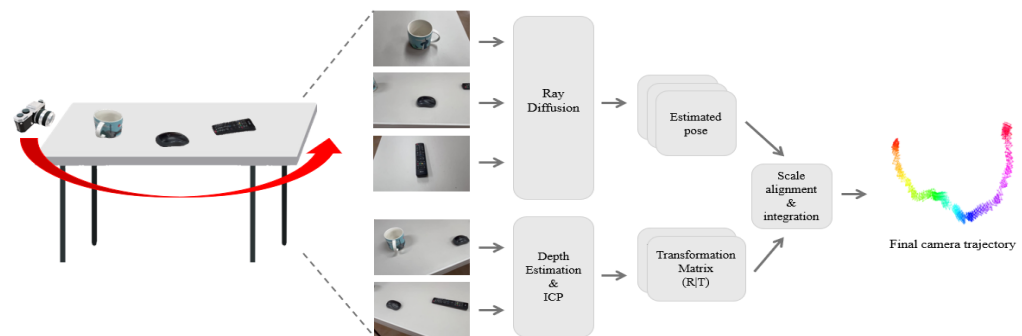


Figure 1. Overview of the proposed pipeline. The red-colored arrow shows the motion of the camera.

2. Related Work

In this work, we combine a Ray Diffusion model with point cloud registration, while leveraging monocular video depth estimation as well as object detection and segmentation to enable this integration. Based on this framework, we now review the research areas most relevant to our approach, namely camera pose estimation, monocular depth estimation, and point cloud alignment.

2.1. Camera Pose Estimation

Camera pose estimation from images has traditionally relied on geometric Structure-from-Motion (SfM) pipelines, which remain widely used in current research and applications. Classical SfM jointly estimates camera trajectories and a sparse 3D structure from multi-view images by extracting features, establishing correspondences across views, and refining camera parameters and 3D points through bundle adjustment. Systems such as COLMAP follow this structure and can recover highly accurate camera poses when images exhibit sufficient texture and viewpoint overlap. In this work, COLMAP combined with the learned local feature descriptors and matchers, SuperPoint and SuperGlue, is used to obtain reference trajectories and sparse reconstructions for evaluation [13]. However, the performance of feature-based SfM is strongly dependent on the quality and quantity of feature correspondences across views. In scenarios with few frames, extensive textureless regions, or limited viewpoint overlap, stable feature tracks are difficult to obtain, and the result may degrade substantially or even fail.

To alleviate these limitations, learning-based camera pose estimation methods for sparsely sampled views have been actively explored in recent years. For example, RelPose [14] and RelPose++ [15] employ energy-based models to estimate camera poses from sparsely sampled views of arbitrary objects. RelPose estimates only camera rotation, whereas RelPose++ recovers full 6-DoF camera poses by additionally estimating translation. PoseDiffusion [16] applies diffusion models to camera pose estimation, starting from an initial noisy pose representation and progressively denoising it to recover both camera rotations and translations. PF-LRM [17] and DUS3R [18] directly predict point clouds from images, allowing camera pose recovery and 3D reconstruction within a unified framework.

Ray Diffusion, which is employed in this work, estimates camera poses (i.e., camera extrinsic parameters) from a set of images of a single object captured under sparse-view settings. Instead of directly regressing camera parameters using a diffusion model, Ray Diffusion introduces a novel pose representation by predicting image rays at the patch level. Based on this representation, the method can recover the intrinsic camera parameters and estimate relative rotations and translations between cameras.

2.2. Monocular Depth Estimation

Monocular depth estimation predicts per-pixel depth from images captured by a single camera and serves as a foundational component for various applications, including camera pose estimation, 3D reconstruction, and autonomous driving. Figure 2 shows an example of monocular depth estimation, illustrating an input RGB image and its corresponding predicted depth map. The field covers not only depth prediction from a single static image, but also video-based approaches that aim to produce temporally consistent depth estimates across frames. Models such as MiDaS [19] and Marigold [20] are trained on large-scale datasets and operate robustly in diverse scenes and conditions, allowing stable prediction of relative depth from input images. For video depth estimation, enforcing temporal consistency is essential. For instance, ChronoDepth [21] and DepthCrafter [22] leverage diffusion-based video generation frameworks to learn inter-frame context and generate depth maps that remain consistent over time. The VDA model used in this work is also trained on large-scale datasets and demonstrates strong generalization across diverse scenes while maintaining temporal consistency throughout a video. In particular, VDA provides two variants, relative depth and metric depth, which makes it applicable to a broad range of 3D vision tasks.



Figure 2. Example of monocular depth estimation. From a single RGB image (**top**), a monocular depth model predicts a dense per-pixel depth map (**bottom**), with warmer colors indicating closer regions and cooler colors representing farther regions.

2.3. Iterative Closest Point Algorithm

The Iterative Closest Point (ICP) algorithm is a classical method that registers a source point cloud to a target point cloud and estimates the 3D rigid transformation that aligns the source to the target. When point clouds are generated in their respective camera coordinate frames, the estimated rigid transformation can be interpreted as the relative camera pose between the two frames. Given an initial transformation, ICP iteratively constructs correspondence pairs by matching each point in the source point cloud to

its nearest neighbor in the target point cloud, and then updates the transformation by minimizing the distances between the matched pairs. Figure 3 provides a qualitative example of the ICP process, where a source point cloud (red) is aligned with a target point cloud (green) by estimating a 3D rigid transformation.

Although ICP is relatively simple and easy to implement, it is sensitive to initialization and vulnerable to outliers. To address these limitations, various extensions have been proposed. For example, point-to-plane ICP minimizes distances between points and local surface normals, while Generalized ICP (GICP) incorporates probabilistic modeling of local point distributions to improve robustness.

More recently, learning-based approaches have been investigated to replace or complement geometric registration. Methods such as Deep Closest Point [23], GCNet [24], and GeoTransformer [25] employ neural networks to predict point correspondences directly or to learn global feature representations for registration. Nevertheless, classical geometry-based methods, such as GICP remain competitive and are still widely used in practice due to their strong performance and reliability under appropriate conditions.

In this work, we apply GICP to monocular depth-based point clouds to estimate frame-to-frame 3D transformations within multi-object sub-scenes.

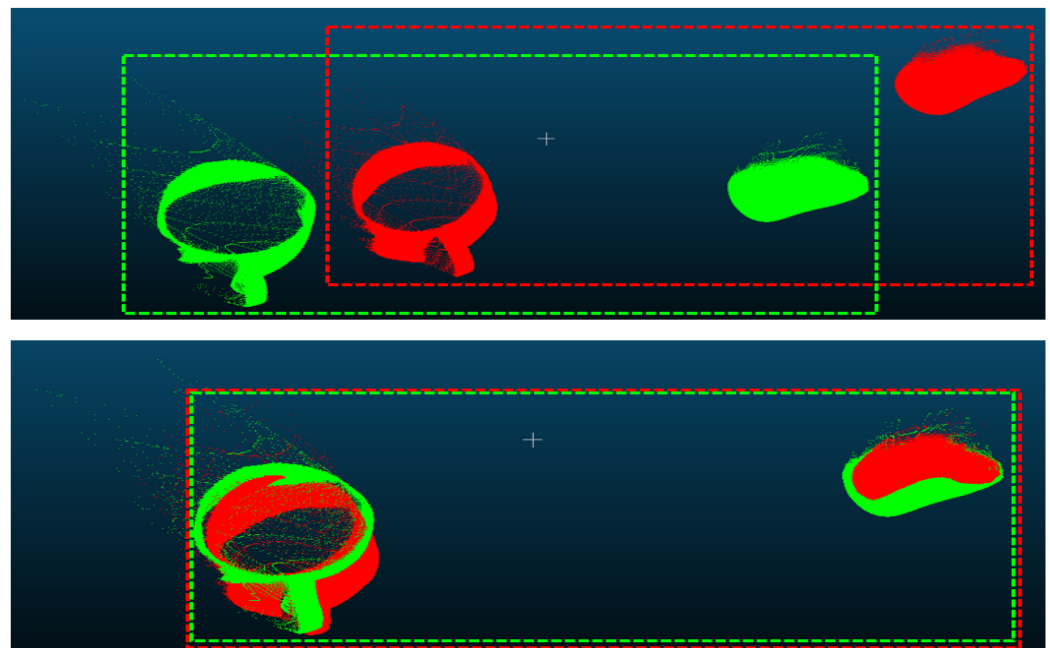


Figure 3. Illustration of point cloud registration using ICP. **Top:** before alignment, where the source point cloud (red) is misaligned with the target point cloud (green). **Bottom:** after ICP, the source point cloud is aligned to the target point cloud.

3. Proposed Method

3.1. Dataset and Problem Setting

To test and evaluate our method, we recorded a total of 40 indoor video sequences for our dataset. Figure 4 shows a sample image sequence. In each sequence, three different objects, denoted as (A, B, C) , appear sequentially, and the video transitions between single-object and multi-object sub-scenes. This yields five contiguous sub-scenes: $A \rightarrow A+B \rightarrow B \rightarrow B+C \rightarrow C$.

In our dataset, the total sequence length ranges from 110 to 150 frames. Single-object sub-scenes consist of 30–40 frames each, while multi-object sub-scenes consist of 10–15 frames each. We define boundary frames as the shared frames between adjacent sub-scenes, which serve as common references for the integration of sub-scene poses (e.g.,

the last frame of A is the first frame of $A+B$; the last frame of B is the first frame of $B+C$). This design allows per-sub-scene pose estimates to be chained via matrix multiplication into a consistent global trajectory.

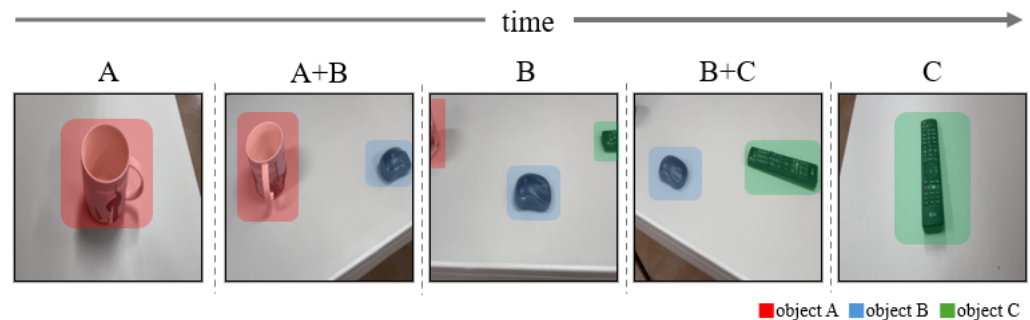


Figure 4. Example sequence in our dataset: Sample frames from a single image sequence. Single-object and multi-object scenes appear repetitively; colored masks indicate object areas.

3.2. Pose Estimation of Single-Object Sub-Scenes

For single-object sub-scenes (A , B , and C), we estimate camera poses using Ray Diffusion. The model takes as input an image sequence and an object bounding box for each frame. To handle coordinate system ambiguity, Ray Diffusion rotates the world coordinates such that the first camera has identity rotation and rescales the scene so that the first camera translation has unit norm. As a result, the estimated translations are expressed in a non-metric scale. The subsequent frames within each sub-scene are then estimated relative to this reference, yielding camera extrinsic parameters $[\mathbf{R} \mid \mathbf{t}]$. In our implementation, we use the official Ray Diffusion model released by the authors and perform inference only, without additional training.

3.3. Pose Estimation of Multi-Object Sub-Scenes

For the entire video sequence, we first estimate both relative and metric depth maps for all frames using VDA. Although the metric depth output provides an absolute scale, we found that its per-pixel accuracy is generally lower than that of the relative depth prediction. In contrast, the relative depth is more spatially consistent, but lacks an absolute scale. To combine the advantages of both outputs, we estimate a scene-level linear mapping that aligns the relative depth to the metric depth, thereby obtaining metric-aligned depth maps. These depth maps are subsequently used for point cloud registration in multi-object sub-scenes and for scale alignment in single-object sub-scenes.

$$D_t^{\text{metric}}(u, v) \approx \alpha D_t^{\text{rel}}(u, v) + \beta, \quad (1)$$

where $D_t^{\text{rel}}(u, v)$ and $D_t^{\text{metric}}(u, v)$ denote the relative and metric depth at pixel (u, v) of frame t , and α, β are a global scale and shift shared throughout the scene. Parameters (α, β) are estimated by minimizing the squared error between the two depth maps on all pixels of all frames:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta} \sum_{t \in \mathcal{T}} \sum_{(u, v)} \left(\alpha D_t^{\text{rel}}(u, v) + \beta - D_t^{\text{metric}}(u, v) \right)^2 \quad (2)$$

where $\mathcal{T}$ denotes the set of frames in the scene. After applying $(\hat{\alpha}, \hat{\beta})$, the resulting metric-aligned depth maps are used to generate point clouds on a consistent metric scale.

Next, SAM2 is applied to segment object masks, and only the depth values inside the mask regions are backprojected into 3D points using the camera intrinsic parameters. This yields object-centric point clouds for each frame, restricting the registration to the object

regions and reducing background influence. Figure 5 illustrates this pipeline, from RGB input to metric depth estimation and subsequent generation of cropped object-centric point clouds. We estimate the rigid transformation between consecutive frames by aligning the object-centric point clouds using GICP. Given two consecutive frames n and $n + 1$, GICP estimates a transformation $\mathbf{T}_{n \rightarrow n+1}$ that maps 3D points expressed in the coordinate system of frame n to that of frame $n + 1$. This transformation represents the relative camera motion between the two frames.

By recursively composing these relative transformations, we obtain the camera pose of each frame with respect to the first frame of the corresponding multi-object sub-scene. However, these translations are not directly comparable to those estimated by Ray Diffusion in single-object sub-scenes, where the trajectory is expressed in non-metric scale. As a result, the two trajectories cannot be directly concatenated without additional scale alignment.

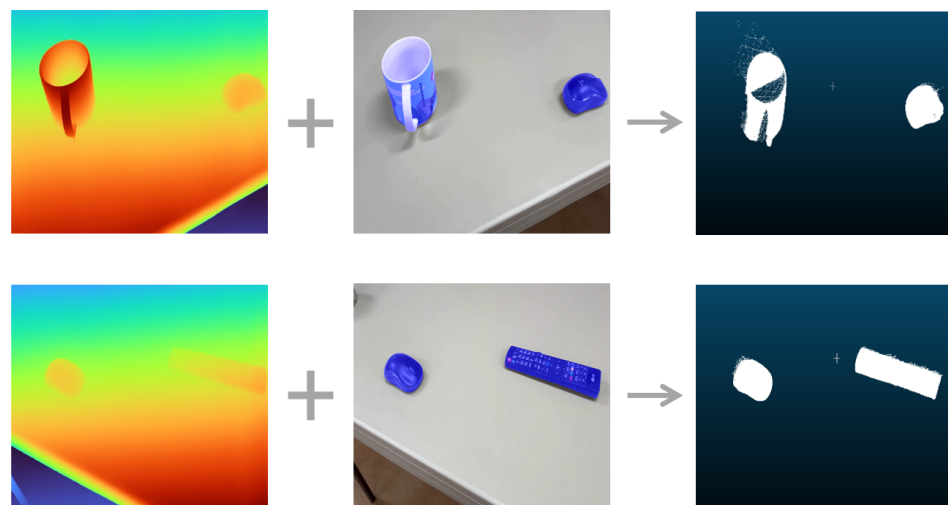


Figure 5. Pipeline from RGB image to object-centric point cloud. An RGB frame is converted into a metric depth map using VDA, after which SAM2 masks are applied to crop object regions. The masked depth is backprojected into 3D using camera intrinsics to generate object-focused point clouds.

3.4. Scale Alignment Between Sub-Scenes

The pose estimates obtained from single-object and multi-object sub-scenes are expressed in different scale domains. In single-object sub-scenes, Ray Diffusion produces translation vectors defined in an arbitrarily normalized coordinate system, whereas the point-cloud-based transformations in multi-object sub-scenes are expressed in the scale induced by monocular depth estimates. As a result, directly concatenating these pose sequences leads to a camera trajectory with inconsistent translation scales across sub-scenes. To reconcile this discrepancy, we estimate a scale factor S_a by minimizing the reprojection error defined in (7). For clarity, we describe the scale estimation procedure using scenes A and $A+B$, as a representative example. The same procedure is applied identically to scenes B and C .

(1) Reference 3D point selection. From the first image frame of scene A , we select the 2D centroid of the object mask, denoted as $\mathbf{u}_0 = (u_0, v_0)$. Let $\tilde{\mathbf{u}}_0 = (u_0, v_0, 1)$ denote its homogeneous representation. The corresponding 3D point $\mathbf{P}_0$ in the camera coordinate system is obtained by backprojection using the depth value estimated in Section 3.3 and the camera intrinsics $\mathbf{K}$ recovered from Ray Diffusion:

$$\mathbf{P}_0 = D_0(\mathbf{u}_0)\mathbf{K}^{-1}\tilde{\mathbf{u}}_0, \quad (3)$$

where $D_0(\mathbf{u}_0)$ denotes the depth value in pixel $\mathbf{u}_0$.

(2) Pose chain with scaled translation in the single-object scene. From the pose estimation results of each sub-scene, we obtain frame-to-frame transformations across scene A and scene $A+B$. We introduce an unknown scale factor S_a for translations estimated in scene A , initialized within a predefined range. Each transformation $\mathbf{T}_i$ is written as

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & s_i \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad s_i = \begin{cases} S_a, & i \in \text{scene } A \\ 1, & i \in \text{scene } A+B \end{cases} \quad (4)$$

and the composed transformation from frame 0 to frame t is given by

$$\mathbf{T}_{0 \rightarrow t} = \mathbf{T}_{t-1 \rightarrow t} \cdots \mathbf{T}_{1 \rightarrow 2} \mathbf{T}_{0 \rightarrow 1}, \quad (5)$$

Given the relative transformation $\mathbf{T}_{0 \rightarrow t}$, the reference point $\mathbf{P}_0$ is propagated to $\mathbf{P}_t$ for frames $t \in \mathcal{I}$, where $\mathcal{I}$ denotes the set of frames covering both the scene A and scene $A+B$. We then project $\mathbf{P}_t$ onto the image plane as

$$\hat{\mathbf{u}}_t = \pi(\mathbf{K} \mathbf{P}_t), \quad (6)$$

where $\pi(\cdot)$ denotes projection onto the image plane.

(3) Scale estimation by reprojection-error minimization. Finally, we estimate the optimal scale $\hat{S}_a$ by minimizing the reprojection error between the projected mask centers $\hat{\mathbf{u}}_t$ and the observed mask centers $\mathbf{u}_t$ over $t \in \mathcal{I}$:

$$\hat{S}_a = \arg \min_{S_a} \sum_{t \in \mathcal{I}} \|\hat{\mathbf{u}}_t - \mathbf{u}_t\|^2. \quad (7)$$

The estimated $\hat{S}_a$ is then uniformly applied to all translation parameters in the corresponding single-object scene. Figure 6 summarizes the overall scale alignment procedure, illustrated for an example where scene A consists of 40 frames and scene $A+B$ consists of 10 frames.

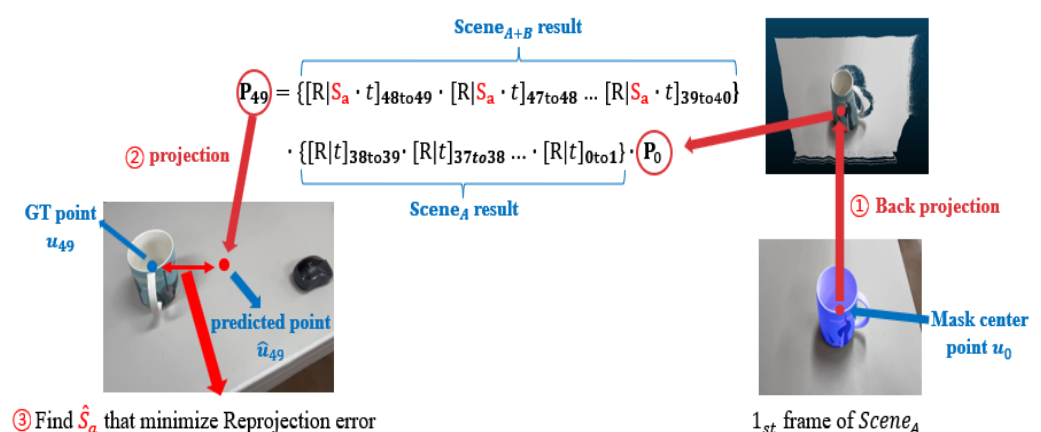


Figure 6. Overview of the proposed scale alignment procedure. For illustration, scene A is assumed to contain 40 frames and scene $A+B$ 10 frames. A reference 3D point is first selected in scene A . The translations in scene A are scaled by S_a , after which frame-to-frame transformations are chained across scenes A and $A+B$. The scale factor is optimized by minimizing the reprojection error of the propagated reference point over the shared frame set.

We repeat the same procedure for scene B and C , aligning each sub-scene to its neighboring sub-scene. Figure 7 illustrates the effect of the proposed scale alignment on the estimated camera trajectory. Without scale alignment, pose estimates from adjacent sub-scenes are chained with inconsistent translation scales, which leads to distortions in

the global trajectory and causes partial overlaps near sub-scene boundaries. After applying the reprojection-error-based scale optimization, these inconsistencies are alleviated, and the resulting trajectory becomes geometrically coherent across the entire sequence.

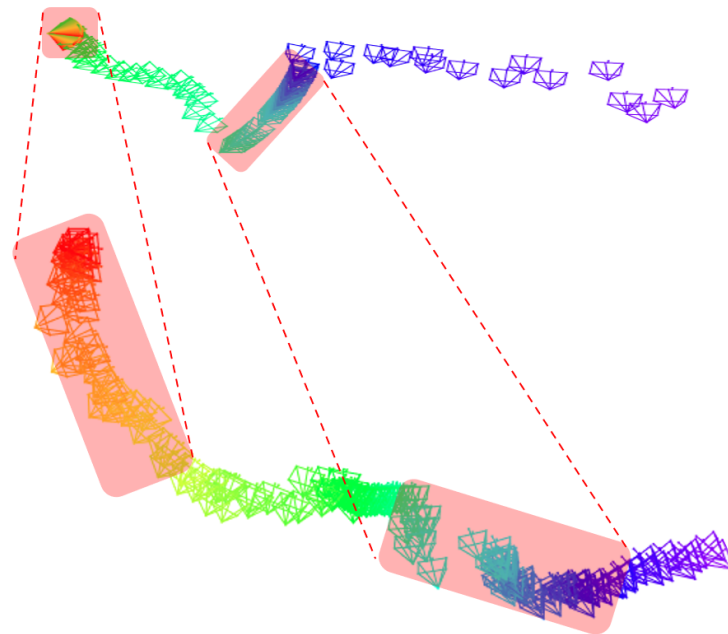


Figure 7. Camera trajectory comparison before and after the proposed scale alignment on a representative sequence. The top trajectory visualizes pose estimates obtained without scale alignment, where inconsistent translation scales across adjacent sub-scenes lead to trajectory distortions and overlapping regions, indicated in red. The bottom trajectory shows the result after converting relative depth to metric depth and applying the reprojection-error-based scale alignment, yielding a geometrically coherent and globally consistent camera trajectory.

3.5. Final Trajectory Integration

After scale alignment, the pose estimates from all sub-scenes are integrated into a single camera trajectory. Adjacent sub-scenes share a boundary frame, such that the last frame of a preceding sub-scene coincides with the first frame of the following one. This shared frame serves as a common reference that enables direct concatenation of pose estimates across sub-scenes. Consequently, scale-corrected poses from scenes A , $A+B$, B , $B+C$, and C are chained in order by matrix multiplication. Thus, the first frame of scene A is fixed as the global origin, and camera poses are sequentially composed along the temporal order of the sequence. Through this process, we obtain a globally consistent set of camera extrinsic parameters for all frames in the sequence.

4. Experiment

4.1. Experimental Setup

4.1.1. Dataset

To evaluate the performance of the proposed method, we have generated total 40 indoor video sequences using the 26 mm focal length rear camera of an iPhone 13 mini at 10 fps. To capture each video sequence, we place three different objects on the table and record the video by moving the camera around the objects. As the experimental objects, we select total 11 small table-top object types, book, cup, vase, statue, mouse, laptop, teddy, bowl, phone, keyboard, and remote. Sample images of 11 object types are shown in Figure 8. The resolution of original captured image is 1920×1080 and the same resolution is used in all evaluations. All 40 video sequences are publicly available in

<https://github.com/Y-C-JEONG/Pose-Estimation-in-Multi-Object-Scenes> (accessed on 10 June 2026).



Figure 8. Sample images of all object types for our evaluation. From the top-left, book, cup, vase, statue, mouse, laptop, teddy, bowl, phone, keyboard, and remote.

Each video sequence contains approximately 110–150 frames and is intentionally structured to include alternating single-object and multi-object sub-scenes. In each video, three objects appear sequentially, forming five contiguous sub-scenes: $A \rightarrow A+B \rightarrow B \rightarrow B+C \rightarrow C$. Single-object sub-scenes consist of 30–40 frames, while multi-object sub-scenes consist of 10–15 frames. To enable pose integration, adjacent sub-scenes share a one-frame overlap, which serves as a boundary frame for trajectory concatenation.

The dataset includes objects from both categories seen during Ray Diffusion training and unseen categories, allowing evaluation of generalization performance in multi-object settings.

4.1.2. Baselines

We compare the proposed method with the following baselines:

- **Reference (COLMAP+SPSG):** We use a COLMAP-based Structure-from-Motion pipeline, where local features are extracted using SuperPoint and matched using SuperGlue (SPSG). Only sequences for which COLMAP successfully reconstructs the entire camera trajectory are used as reference. The resulting camera poses are treated as reference trajectories for quantitative evaluation.
- **Ray Diffusion:** The baseline model, Ray Diffusion, is applied to the full image sequence without sub-scene partitioning. In the original implementation of Ray Diffusion, camera trajectory is estimated for only a single object in each video sequence. However, the test sequences utilized in this study feature multiple object types that appear and disappear dynamically over time, preventing the definition of a consistent, object-specific bounding box across the entire sequence. Consequently, the full image region is utilized as the input for the baseline model.
- **DUST3R:** One of the SOTA (State of The Art) visual-SLAM methods, DUST3R is applied under two inference modes: center reference frame and sliding window. In the center reference frame mode, denoted DUST3R^a, the middle of the sequence is selected as the reference image and matched with other frames. For the sliding window mode, denoted DUST3R^b, pose estimation was performed with a sliding window size of 5. In both modes, global alignment is subsequently applied to recover the full scene structure.
- **ORB-SLAM2:** As a second SOTA visual SLAM method, ORB-SLAM2 is applied. In the default configuration, the algorithm frequently failed to initialize or produced

only a limited number of valid pose estimation, because our dataset has insufficient image features. Therefore, the maximum number of ORB features was increased from 1000 to 2000 to improve feature detection and tracking robustness. Since ORB-SLAM2 estimates camera poses only after successful initialization, we additionally introduce a completeness metric which represents the proportion of input frames for which valid pose estimates were successfully obtained.

4.2. Evaluation Metrics

Following the evaluation protocol of Ray Diffusion, we evaluate camera pose accuracy using rotation and translation metrics:

- **Camera Rotation Accuracy (RA):** Rotation accuracy measures the proportion of frame pairs whose relative rotation error is below a specified threshold. For both predicted and reference poses, we compute pairwise relative rotations between camera frames and measure the angular difference between them. We report RA at thresholds of 10° and 15° .
- **Camera Center Accuracy (CCA):** Translation accuracy is evaluated using camera center accuracy, defined as the proportion of frames whose Euclidean distance between predicted and reference camera centers is below a given threshold. Before evaluation, predicted and reference trajectories are first aligned using Umeyama similarity alignment [26] before comparison. The error threshold is defined relative to the scene scale, which is calculated as the maximum distance from the trajectory centroid to any camera center, as illustrated in Figure 9. We report CCA at thresholds of 0.1 and 0.2.
- **Mean Rotation Error (MRE):** Mean rotation error measures the average angular difference between predicted and reference relative camera rotations. Following the evaluation protocol of Ray Diffusion, pairwise relative rotations are computed for all ordered frame pairs from the predicted and reference camera poses. The angular distance between the corresponding relative rotations is then measured and the final MRE is obtained by averaging these angular errors over all frame pairs.
- **Mean Translation Error (MTE):** Mean translation error measures the average normalized camera center error between predicted and reference trajectories. The predicted camera centers are first aligned to the reference trajectory using Umeyama similarity alignment. After alignment, the Euclidean distance between corresponding camera centers is computed and normalized by the reference scene scale. The final MTE is obtained by averaging these normalized camera center errors over all frames.

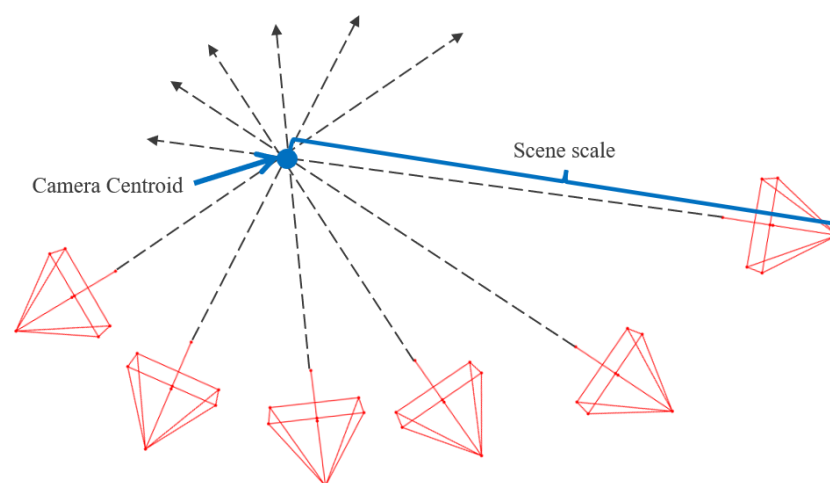


Figure 9. Illustration of the scene scale definition. The trajectory centroid is first computed from all camera centers, and the scene scale is defined as the maximum distance from the centroid to any camera center.

4.3. Evaluation Results

Figure 10 shows qualitative visualization of the camera trajectory in each sequence. In each row of the figure, sample images captured from a moving camera are shown in order, and the last column is the estimated camera trajectory. All estimated camera poses in each sequence are shown in different colors. As shown in the figure, the estimated camera trajectories visually follow the real camera motion without any outlying one.

Quantitative evaluations are conducted using reference trajectories obtained by COLMAP+SPSG. Initially, COLMAP+SPSG was executed to generate the reference trajectories for all test sequences. Subsequently, we visually inspect the trajectories of the COLMAP results and divide the test datasets to two subsets. The first set contains trajectories that visually match the original camera motion, whereas the second set includes those with noticeable deviations. Consequently, the evaluation results for the first dataset are presented in this section, and those for the second set are provided in Appendix A.

Evaluation results across the first set of 20 test video sequences are presented at Tables 1–5. Each table evaluates four metrics: rotation accuracy (RA), camera center accuracy (CCA), mean relative error (MRE), and mean trajectory error (MTE), with the final row of each table indicating the average values. All metrics are computed relative to the reference trajectories generated using COLMAP+SPSG.

Regarding translation accuracy evaluated via CCA, the proposed method achieves 55.48% at the 0.1 threshold and 95.04% at the 0.2 threshold, significantly outperforming the baseline method, which yields 25.35% and 68.47%, respectively. For RA, the proposed framework obtains 71.48% at 10° and 91.10% at 15°, marginally better than the baseline method, which records 67.40% and 86.04%. Furthermore, the proposed method demonstrates superior performance over the baseline in terms of both MRE and MTE.

The proposed framework is also evaluated against two widely adopted camera pose estimation paradigms: DUST3R, a learning-based visual SLAM framework, and ORB-SLAM2, a conventional feature-based visual SLAM system. Compared with DUST3R^a, the proposed method shows better performance in RA@15°, comparable results in RA@10°. In CCA, MRE, and MTE analysis, our method is much better than DUST3R^a. Compared with DUST3R^b, the proposed approach shows higher accuracy in RA@15° and CCA with threshold 0.2, but lower accuracy in RA@10°. However, it consistently maintains better accuracy in MRE and MTE.

Table 1. Evaluation results of the proposed method using the first set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_cup	95.2107	71.8243	65.8120	94.8718	7.5253	0.0941
book_bowl_statue	100.0	85.8602	84.7328	98.4733	6.2494	0.0759
book_cup_mouse	82.4096	57.2737	66.6667	98.2456	9.1277	0.0858
bottle_remote_keyboard	93.9001	71.8236	75.2212	100.0	7.9502	0.0755
bottle_teddy_cup	90.3922	65.7563	53.3333	100.0	8.7469	0.0958
bowl_cup_phone	87.1514	71.3793	42.2414	98.2759	8.0682	0.1102
bowl_mouse_cup	99.4710	83.6368	21.8750	92.9688	6.3363	0.1383
bowl_mouse_phone	97.9972	83.7137	47.3684	96.4912	6.6261	0.1085
cup_book_mouse	92.1629	50.5638	63.9456	97.2789	9.3875	0.0927
cup_bowl_book	78.9816	69.1323	50.3497	93.7063	8.6103	0.1118
cup_bowl_mouse	98.2900	81.7950	60.2740	95.8904	6.8884	0.0964
cup_bowl_phone	90.3830	74.6622	43.5374	93.1973	7.6740	0.1133
lap_book_bowl	93.9355	74.6581	41.6000	98.4000	7.3500	0.0985
lap_book_cup	82.1850	61.7618	22.6562	72.6562	9.0809	0.1588
lap_book_mouse	89.4835	72.6809	58.7156	95.4128	7.9321	0.0880
lap_book_phone	100.0	90.5182	58.3333	100.0	6.0360	0.0886

Table 1. Cont.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
lap_bowl_cup	97.9558	72.3265	39.1304	87.8261	7.4115	0.1265
lap_bowl_mouse	98.2971	86.2595	74.8092	94.6565	7.6952	0.0936
teddy_cup_keyboard	99.5768	71.8921	73.8318	100.0	7.6429	0.0772
teddy_cup_remote	54.4978	32.8350	54.0323	97.5806	14.7163	0.1004
Average	91.1141	71.5177	54.9233	95.2966	8.0528	0.1015

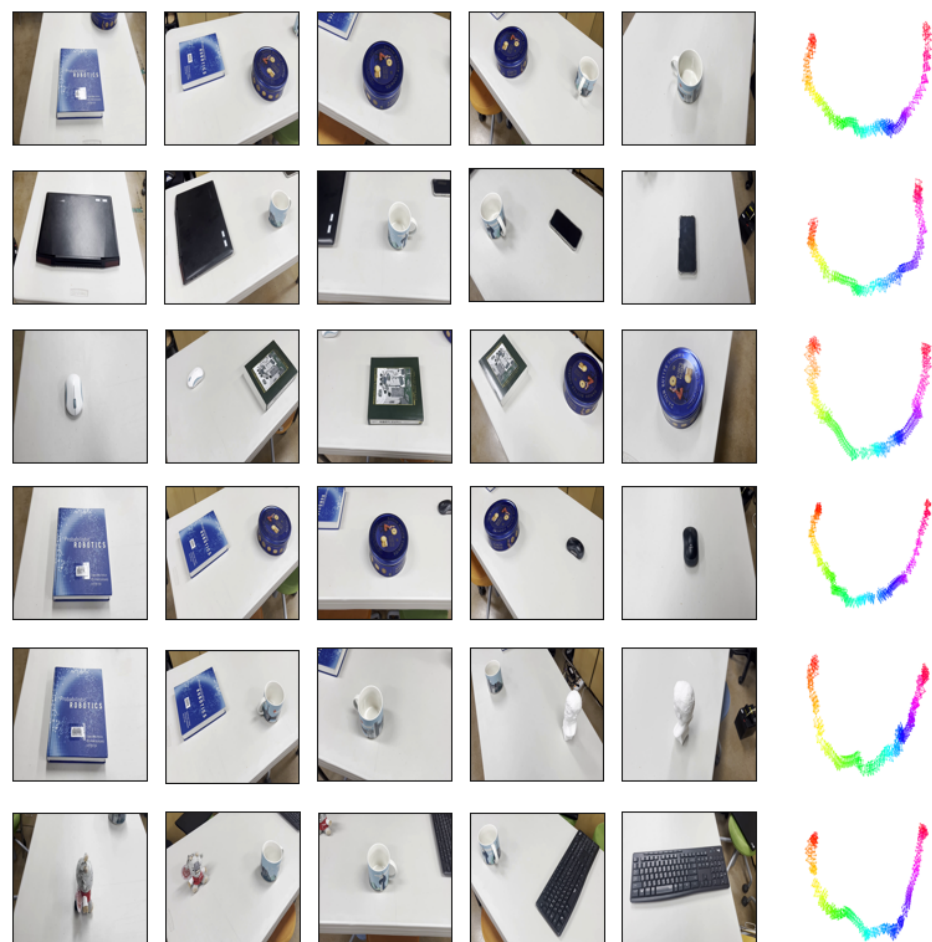


Figure 10. Experimental results: In each row, the left five images show sample frames in each sub-scene, and the right image shows the estimated camera trajectory.

Table 2. Evaluation results of Ray Diffusion using the first set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_cup	99.7937	89.1836	55.5556	87.1795	6.3626	0.1065
book_bowl_statue	96.5708	78.4146	35.8779	77.8626	7.4153	0.1400
book_cup_mouse	98.1835	78.8387	28.0702	73.6842	7.1753	0.1510
bottle_remote_keyboard	56.2895	46.2231	3.5398	27.4336	38.3683	0.2728
bottle_teddy_cup	92.7031	70.6443	19.1667	75.8333	8.2643	0.1560
bowl_cup_phone	95.1724	75.9070	25.8621	90.5172	7.6887	0.1293
bowl_mouse_cup	84.6211	52.6206	24.2188	51.5625	10.0349	0.1923
bowl_mouse_phone	67.7690	40.2577	17.5439	55.2632	12.6459	0.1890
cup_book_mouse	96.9434	75.7245	19.0476	76.8707	7.3391	0.1614
cup_bowl_book	87.6293	58.2783	30.0699	75.5245	9.5507	0.1545
cup_bowl_mouse	65.9329	47.2933	22.6027	57.5342	13.4031	0.2033

Table 2. Cont.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
cup_bowl_phone	62.7435	34.7684	15.6463	46.9388	14.5252	0.2105
lap_book_bowl	99.3290	84.4645	27.2000	77.6000	6.5088	0.1499
lap_book_cup	96.5674	77.5221	11.7188	67.1875	7.2326	0.1889
lap_book_mouse	100.0	96.1774	44.9541	84.4037	5.3772	0.1279
lap_book_phone	97.2129	80.6583	45.0000	80.0	6.9503	0.1370
lap_bowl_cup	98.6575	79.6644	29.5652	77.3913	7.1047	0.1470
lap_bowl_mouse	94.8679	62.3136	29.7710	80.1527	8.7585	0.1437
teddy_cup_keyboard	61.4530	57.6794	11.2150	47.6636	30.6961	0.1978
teddy_cup_remote	68.4500	61.5132	10.4839	58.8710	25.5739	0.2090
Average	86.0445	67.4074	25.3555	68.4737	12.0488	0.1684

Table 3. Evaluation results of DUS3R^a using the first set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_cup	51.1936	36.1922	8.5470	17.9487	64.7924	0.5144
book_bowl_statue	100.0	98.5672	97.7099	100.0	2.8472	0.0375
book_cup_mouse	67.8466	47.4616	2.6316	14.9123	59.4206	0.5280
bottle_remote_keyboard	86.7257	83.4703	65.4867	93.8053	22.6780	0.1154
bottle_teddy_cup	87.2269	84.5938	17.5000	49.1667	25.1373	0.2786
bowl_cup_phone	83.7481	71.8441	22.4138	42.2414	28.6232	0.2918
bowl_mouse_cup	85.3100	80.9424	10.9375	39.0625	29.0250	0.2945
bowl_mouse_phone	52.5229	34.2028	11.4035	42.9825	37.8514	0.3595
cup_book_mouse	69.8164	66.6760	19.7279	40.1361	27.9206	0.2988
cup_bowl_book	84.4479	79.9271	24.4755	57.3427	21.7332	0.2455
cup_bowl_mouse	68.1342	53.1034	13.0137	36.3014	33.5294	0.3370
cup_bowl_phone	97.7914	97.2882	94.5578	98.6395	3.5592	0.0530
lap_book_bowl	87.6000	87.1097	82.4000	93.6000	9.1158	0.1093
lap_book_cup	64.9975	62.4508	12.5000	38.2813	30.7633	0.3036
lap_book_mouse	82.3819	78.0666	14.6789	32.1101	15.5518	0.3144
lap_book_phone	73.9916	73.7255	30.8333	80.8333	22.5389	0.2348
lap_bowl_cup	78.4287	66.9108	13.0435	24.3478	40.7591	0.4262
lap_bowl_mouse	100.0	99.9413	100.0	100.0	3.7777	0.0338
teddy_cup_keyboard	95.2389	94.4631	96.2617	98.1308	3.7317	0.0463
teddy_cup_remote	75.8196	56.9761	8.0645	25.0000	38.5217	0.3628
Average	79.6611	72.6956	37.3093	56.2421	26.0939	0.2593

Table 4. Evaluation results of DUS3R^b using the first set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_cup	50.1916	33.0091	5.9829	12.8205	56.5577	0.5038
book_bowl_statue	100.0	99.3423	98.4733	100.0	3.8211	0.0317
book_cup_mouse	75.6560	62.3971	1.7544	16.6667	46.7523	0.4568
bottle_remote_keyboard	88.6062	75.7427	92.9204	100.0	6.6044	0.0512
bottle_teddy_cup	52.5070	45.8824	4.1667	9.1667	81.3028	0.4952
bowl_cup_phone	88.5607	84.2879	24.1379	45.6897	24.5374	0.2628
bowl_mouse_cup	92.4336	91.9168	37.5000	90.6250	17.0438	0.1733
bowl_mouse_phone	82.1146	71.0449	7.0175	41.2281	36.2142	0.3531
cup_book_mouse	64.6165	49.8742	2.7211	20.4082	33.2375	0.3658
cup_bowl_book	100.0	93.1744	95.8042	100.0	5.2071	0.0508
cup_bowl_mouse	90.8077	89.0316	33.5616	75.3425	17.7203	0.1990
cup_bowl_phone	100.0	96.5800	99.3197	100.0	4.3682	0.0509
lap_book_bowl	100.0	96.9290	100.0	100.0	3.6110	0.0392

Table 4. *Cont.*

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
lap_book_cup	100.0	100.0	92.9688	100.0	3.2129	0.0524
lap_book_mouse	100.0	99.8131	95.4128	100.0	4.2595	0.0486
lap_book_phone	100.0	100.0	95.0000	100.0	2.6338	0.0526
lap_bowl_cup	100.0	100.0	88.6957	100.0	2.5529	0.0592
lap_bowl_mouse	100.0	93.4351	87.0229	100.0	3.7286	0.0650
teddy_cup_keyboard	100.0	100.0	100.0	100.0	2.0765	0.0397
teddy_cup_remote	97.6397	86.2182	95.1613	100.0	5.2220	0.0581
Average	89.1567	83.4339	62.8811	75.5974	18.0332	0.1705

Table 5. Evaluation results of ORB-SLAM2 using the first set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE	Completeness (%)
book_bowl_cup	97.6477	97.6477	97.6190	97.6190	1.0085	0.0487	71.7949
book_bowl_statue	100.0	100.0	100.0	100.0	1.2715	0.0233	7.6336
book_cup_mouse	99.6582	90.0547	98.7013	98.7013	4.2807	0.0279	67.5439
bottle_remote_keyboard	100.0	100.0	17.3913	30.4348	1.8081	0.3367	20.3540
bottle_teddy_cup	100.0	100.0	100.0	100.0	2.2145	0.0057	73.3333
bowl_cup_phone	100.0	100.0	100.0	100.0	1.3361	0.0038	68.1034
bowl_mouse_cup	—	—	—	—	—	—	0.0
bowl_mouse_phone	100.0	100.0	97.0588	100.0	4.0044	0.0244	29.8246
cup_book_mouse	—	—	—	—	—	—	0.0
cup_bowl_book	100.0	100.0	100.0	100.0	2.3621	0.0048	80.4196
cup_bowl_mouse	—	—	—	—	—	—	0.0
cup_bowl_phone	—	—	—	—	—	—	0.0
lap_book_bowl	100.0	95.3333	8.0000	20.0	4.6406	0.4121	20.0
lap_book_cup	100.0	99.8566	100.0	100.0	3.5862	0.0051	65.6250
lap_book_mouse	—	—	—	—	—	—	0.0
lap_book_phone	95.9111	81.6841	100.0	100.0	5.8924	0.0060	86.6667
lap_bowl_cup	100.0	100.0	98.4848	98.4848	3.4497	0.0639	57.3913
lap_bowl_mouse	100.0	100.0	100.0	100.0	2.5246	0.0060	76.3359
teddy_cup_keyboard	—	—	—	—	—	—	0.0
teddy_cup_remote	84.5421	67.2161	100.0	100.0	7.4623	0.0028	84.6774
Average	98.4114	95.1280	86.9468	88.9457	3.2744	0.0694	40.4840

Finally, the proposed method is compared with ORB-SLAM2, a benchmark conventional visual SLAM method and one of the top-performing methods even compared to the learning-based methods. While ORB-SLAM2 outperforms the proposed framework across most metrics—excluding CCA at 0.2 threshold, it exhibits severe robustness limitations. As shown in Table 5, ORB-SLAM2 fails to run in multiple sequences. A total of 6 sequences fail among 20 test sequences. This is because of the inherent problem of pose initialization, which is typical in monocular visual-SLAM methods such as ORB-SLAM2 [27]. In addition, even though ORB-SLAM2 successfully runs in the other sequences, it can track only a fractional segment of the trajectory because monocular visual-SLAM can provide pose estimation results only after initialization, in the middle of sequences. To quantify this behavior, a completeness score is introduced in the final column of Table 5. This completeness score is defined as the ratio of successfully tracked frames to the total number of frames in each sequence. The average completeness score of ORB-SLAM2 is only 40.48%, while all the others are 100%. A comprehensive summary of the evaluation across all five methods is shown in Table 6.

Table 6. Comparison of the average performance of the first set of 20 test sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE	Completeness (%)
Proposed	91.1076	71.4835	55.4861	95.0479	8.0522	0.1005	100.0
Proposed ¹	91.1141	71.5177	55.0808	93.2632	8.0528	0.1038	100.0
Ray Diffusion	86.0445	67.4074	25.3555	68.4737	12.0488	0.1684	100.0
DUSt3R ^a	79.6611	72.6956	37.3093	56.2421	26.0939	0.2593	100.0
DUSt3R ^b	89.1567	83.4339	62.8811	75.5974	18.0332	0.1705	100.0
ORB-SLAM2	98.4114	95.1280	86.9468	88.9457	3.2744	0.0694	40.48

¹ Proposed method without scale alignment (Results of ablation study 1).

4.4. Ablation Studies

To evaluate the performance of the proposed method under varying experimental conditions, two ablation studies were conducted. The first ablation study is done without scale alignment between the sub-scene, while the second one evaluates the impact of varying the image patch size utilized to select the reference 3D point for scale alignment.

As described in Section 3.4, the pose estimates obtained from single-object and multi-object sub-scenes are expressed in different scale domains. In single-object sub-scenes, Ray Diffusion generates translation vectors in an arbitrarily normalized coordinate system, whereas in multi-object sub-scenes, translation vectors are scaled according to monocular depth estimates. To evaluate the effect of the scale alignment, the scale parameter s_i in Equation (4) is intentionally set to 1.0 and the pose estimation results are obtained as shown in Table 7. As shown in the table, the translation accuracy degrades when the scale alignment parameter is omitted compared to when it is properly estimated.

In the second ablation study, the image patch size for selecting the reference 3D point P_0 is set to 1×1 , 3×3 and 5×5 . As noted in Section 3.4, the 2D reference point u_0 is originally defined as the 1×1 centroid of the object mask. For this ablation, the 2D reference points were expanded to include 3×3 and 5×5 local neighborhoods centered around the centroid. All neighboring points were back-projected into the 3D space using Equation (3), and the spatial average of these 3D points was selected as the final reference point. The corresponding ablation results are summarized in Table 8. This result indicates that varying the reference patch size yields marginal performance differences. This minimal variance is primarily due to the fact that the object depth is derived from a monocular depth estimation network, which inherently exhibits spatial consistency and is less susceptible to the high-frequency localized noise typically found in hardware depth sensors.

Table 7. Ablation study 1: Proposed method without scale alignment.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_cup	95.2107	71.8243	67.5214	95.7265	7.5253	0.0940
book_bowl_statue	100.0	85.8602	83.2061	98.4733	6.2494	0.0776
book_cup_mouse	82.4096	57.2737	65.7895	99.1228	9.1277	0.0846
bottle_remote_keyboard	93.9001	71.8236	82.3009	100.0	7.9502	0.0647
bottle_teddy_cup	90.3922	65.7563	50.0000	95.8333	8.7469	0.0996
bowl_cup_phone	87.1514	71.3793	38.7931	93.1034	8.0682	0.1158
bowl_mouse_cup	99.4710	83.6368	10.1562	73.4375	6.3363	0.1638
bowl_mouse_phone	97.9972	83.7137	47.3684	94.7368	6.6261	0.1132
cup_book_mouse	92.1629	50.5638	63.2653	97.2789	9.3875	0.0885
cup_bowl_book	78.9816	69.1323	50.3497	93.0070	8.6103	0.1135
cup_bowl_mouse	98.2900	81.7950	59.5890	95.8904	6.8884	0.0978
cup_bowl_phone	90.3830	74.6622	40.8163	91.1565	7.6740	0.1193
lap_book_bowl	93.9355	74.6581	36.8000	98.4000	7.3500	0.1015

Table 7. Cont.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
lap_book_cup	82.1850	61.7618	19.5312	75.7812	9.0809	0.1650
lap_book_mouse	89.4835	72.6809	59.6330	94.4954	7.9321	0.0902
lap_book_phone	100.0	90.5182	61.6667	100.0	6.0360	0.0863
lap_bowl_cup	97.9558	72.3265	32.1739	77.3913	7.4115	0.1421
lap_bowl_mouse	98.2971	86.2595	73.2824	94.6565	7.6952	0.0951
teddy_cup_keyboard	99.5768	71.8921	81.3084	100.0	7.6429	0.0653
teddy_cup_remote	54.4978	32.8350	58.0645	96.7742	14.7163	0.0977
Average	91.1141	71.5177	54.0808	93.2632	8.0528	0.1038

Table 8. Ablation study 2: Comparison of MTE across scale-alignment patch sizes (1×1 , 3×3 , 5×5). The bold font is the best case in each sequence.

Sequence	Patch Size		
	1×1	3×3	5×5
book_bowl_cup	0.094051	0.094054	0.094055
book_bowl_vase	0.075875	0.075869	0.075870
book_cup_mouse	0.085845	0.085838	0.085836
bottle_remote_keyboard	0.075548	0.075563	0.075575
bottle_teddy_cup	0.095785	0.095786	0.095780
bowl_cup_phone	0.110206	0.110206	0.110204
bowl_mouse_cup	0.138265	0.138296	0.138290
bowl_mouse_phone	0.108541	0.108537	0.108538
cup_book_mouse	0.092714	0.092731	0.092737
cup_bowl_book	0.111782	0.111780	0.111779
cup_bowl_mouse	0.096434	0.096431	0.096429
cup_bowl_phone	0.113250	0.113258	0.113248
lap_book_bowl	0.098465	0.098468	0.098468
lap_book_cup	0.158757	0.158748	0.158749
lap_book_mouse	0.088045	0.088027	0.088021
lap_book_phone	0.088562	0.088552	0.088546
lap_bowl_cup	0.126486	0.126491	0.126494
lap_bowl_mouse	0.093647	0.093635	0.093630
teddy_cup_keyboard	0.077170	0.077143	0.077119
teddy_cup_remote	0.100401	0.100380	0.100396
Average	0.101492	0.101490	0.101488

5. Conclusions

This paper presented a camera pose estimation framework for multi-object image sequences by integrating learning-based relative pose estimation with depth-guided geometric registration. Our method partitions videos into two types of sub-scenes, applying Ray Diffusion to single-object sub-scenes while estimating frame-to-frame motion in multi-object sub-scenes using monocular video depth, object masks, and point cloud registration via GICP. To solve the scale mismatch problem between the translations from Ray Diffusion and the depth-induced translations, we also introduce a reprojection-error-based scale alignment strategy using shared boundary frames, enabling consistent trajectory composition across sub-scenes.

Quantitative evaluation on our indoor dataset demonstrates that the proposed approach improves both rotation accuracy and camera center accuracy compared to directly applying Ray Diffusion to full sequences. In particular, camera center accuracy shows a clear gain under both evaluation thresholds, indicating that the proposed method helps produce more reliable global trajectories in multi-object settings. To evaluate the performance

of the proposed method under varying experimental conditions, two ablation studies are also conducted.

Despite these improvements, the proposed pipeline still depends on the quality of monocular depth prediction and segmentation masks, and point cloud registration may become less stable under severe occlusions or noisy depth estimates. Future work will explore more robust registration strategies and stronger temporal constraints for multi-object sub-scenes, as well as broader evaluations across diverse environments and capture conditions.

Author Contributions: Conceptualization, S.-Y.P.; methodology, software, validation, formal analysis, D.-U.S., K.P. and Y.J.; investigation, D.-U.S., K.P., Y.J. and S.-Y.P.; writing—original draft preparation, Y.J.; writing—review and editing, D.-U.S., K.P., Y.J. and S.-Y.P.; supervision, S.-Y.P.; project administration, S.-Y.P.; funding acquisition, S.-Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (No. RS-2024-00341886).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The video sequences used for the experiments of this article and documentations are publicly available in Github at: <https://github.com/Y-C-JEONG/Pose-Estimation-in-Multi-Object-Scenes> (accessed 10 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This appendix presents the experimental results and performance comparisons of the proposed method with Ray Diffusion, DUS3R, and ORB-SLAM2, evaluated on the second set of 20 test video sequences. The second dataset was recorded using table-top objects identical to those employed in the first dataset. To ensure a fair evaluation, all experimental configurations and environmental conditions for the second dataset were maintained consistent with those specified for the first dataset in Section 4. As shown in the average performance comparison, the evaluation scores for the second dataset are lower than those for the first set of 20 video sequences, while the proposed method performs still better in terms of the average error scores. Because COLMAP reconstructions were utilized as the reference trajectories, we consider this performance degradation to be due to the estimation errors in the COLMAP trajectories.

Table A1. Evaluation results of the proposed method using the second set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_mouse	93.2707	79.8591	71.7557	95.4198	6.7143	0.0890
book_cup_phone	99.9889	84.7651	59.2593	100.0	6.6097	0.0925
book_mouse_phone	97.8885	84.7539	29.8246	88.5965	6.0163	0.1301
bottle_cup_keyboard	70.5519	33.2922	50.4673	94.3925	11.8632	0.1014
bottle_cup_remote	43.1144	30.4611	45.6140	100.0	17.8250	0.1034
bottle_mouse_book	87.4108	41.6242	13.7615	74.3119	10.4543	0.1600
bottle_remote_bottle	73.4073	45.8655	22.3214	91.0714	10.8172	0.1381
bottle_teddy_keyboard	76.1744	44.1281	47.3214	95.5357	10.8073	0.1067
coffee_cup_doll	88.7295	61.3067	55.0847	100.0	9.0123	0.0952
coffee_cup_mouse	68.3319	40.8341	47.8632	95.7265	11.7433	0.1047
cup_book_mouse	83.5021	46.4692	52.7559	100.0	10.0091	0.0956
lap_cup_mouse	99.9845	75.7957	47.3684	94.7368	7.1931	0.1061

Table A1. *Cont.*

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
lap_cup_phone	99.6832	86.3223	57.8512	90.9091	6.7365	0.1070
laptop_teddy_bottle	61.4957	31.4456	34.4262	68.8525	13.3747	0.1714
laptop_teddy_cup	63.2079	41.8436	50.8929	92.8571	14.7750	0.1119
laptop_teddy_keyboard	71.2248	46.7546	44.7154	91.0569	11.6942	0.1141
laptop_teddy_remote	56.6016	41.6038	6.7227	53.7815	15.7314	0.2001
mouse_book_bowl	91.1565	74.1403	39.4558	94.5578	7.7104	0.1110
remote_bottle_book	82.2768	52.2178	9.7087	61.1650	9.9877	0.1815
teddy_keyboard_remote	50.7551	31.6705	41.7391	95.6522	17.0599	0.1156
Average	77.9378	53.7577	41.4455	88.9312	10.8067	0.1218

Table A2. Evaluation results of Ray Diffusion using the second set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_mouse	89.8062	67.4809	27.4809	70.2290	8.9997	0.1597
book_cup_phone	98.1537	78.1647	32.5926	84.4444	7.5946	0.1344
book_mouse_phone	89.5047	62.3506	29.8246	80.7018	9.0676	0.1433
bottle_cup_keyboard	67.2545	46.5350	10.2804	54.2056	19.5340	0.2046
bottle_cup_remote	66.3872	38.9846	15.7895	55.2632	22.5273	0.2351
bottle_mouse_book	99.1165	77.8457	11.0092	48.6239	7.4908	0.2073
bottle_remote_bottle	68.2754	39.7362	1.7857	15.1786	12.1607	0.3167
bottle_teddy_keyboard	55.3893	41.5058	9.8214	30.3571	30.8046	0.2263
coffee_cup_doll	69.7088	44.9660	10.1695	57.6271	11.5381	0.1987
coffee_cup_mouse	72.0307	49.2190	25.6410	70.9402	11.7848	0.1629
cup_book_mouse	65.9668	40.0075	21.2598	75.5906	13.4491	0.1587
lap_cup_mouse	97.3762	78.6524	34.2105	83.3333	7.4542	0.1388
lap_cup_phone	91.2672	61.6942	43.8017	90.9091	8.7998	0.1189
laptop_teddy_bottle	86.4788	50.5894	67.2131	93.4426	9.5560	0.0925
laptop_teddy_cup	92.3423	55.5341	59.8214	96.4286	9.2445	0.0979
laptop_teddy_keyboard	51.6327	34.4396	10.5691	26.0163	46.1369	0.2791
laptop_teddy_remote	78.9916	48.1413	63.0252	94.9580	13.2139	0.1133
mouse_book_bowl	95.3406	71.3633	17.6871	65.3061	8.0432	0.1751
remote_bottle_book	95.4693	74.6431	10.6796	39.8058	7.6536	0.2182
teddy_keyboard_remote	64.6224	51.1518	0.8696	26.0870	29.4764	0.2533
Average	79.7557	55.6503	25.1766	62.9724	14.7265	0.1817

Table A3. Evaluation results of DUS3R^a using the second set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
book_bowl_mouse	77.0875	76.9348	66.4122	87.0229	18.4370	0.1414
book_cup_phone	85.7159	78.4522	73.3333	88.1481	8.5868	0.1547
book_mouse_phone	69.5234	69.1508	22.8070	49.1228	21.2782	0.2802
bottle_cup_keyboard	93.0347	79.4393	99.0654	99.0654	7.1063	0.0454
bottle_cup_remote	79.8944	78.5437	35.9649	78.9474	15.9828	0.1934
bottle_mouse_book	76.0958	74.8556	23.8532	49.5413	21.5137	0.2747
bottle_remote_bottle	40.7175	39.0283	0.0	1.7857	80.7650	0.6596
bottle_teddy_keyboard	100.0	100.0	100.0	100.0	3.3138	0.0256
coffee_cup_doll	93.3942	88.9034	72.0339	88.1356	4.5992	0.1328
coffee_cup_mouse	89.3162	83.1712	55.5556	86.3248	5.7942	0.1408
cup_book_mouse	100.0	100.0	100.0	100.0	3.3322	0.0281
lap_cup_mouse	77.5501	74.4450	17.5439	44.7368	19.4057	0.2790
lap_cup_phone	100.0	99.3526	97.5207	98.3471	3.2847	0.0443
laptop_teddy_bottle	100.0	99.9865	100.0	100.0	3.8780	0.0250
laptop_teddy_cup	100.0	100.0	100.0	100.0	1.6979	0.0267

Table A3. Cont.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE
laptop_teddy_keyboard	98.3740	97.9342	99.1870	99.1870	4.6723	0.0232
laptop_teddy_remote	79.5756	61.0027	59.6639	89.9160	21.4163	0.1550
mouse_book_bowl	79.6477	59.8174	86.3946	96.5986	9.5106	0.0627
remote_bottle_book	68.9891	64.7630	7.7670	26.2136	17.6742	0.3461
teddy_keyboard_remote	81.7391	73.2876	43.4783	77.3913	18.7019	0.1663
Average	84.5328	79.9534	63.0290	78.0242	14.5475	0.1603

Table A4. Evaluation results of ORB-SLAM2 using the second set of video sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE	Completeness (%)
book_bowl_mouse	100.0	100.0	100.0	100.0	1.2115	0.0045	58.0153
book_cup_phone	100.0	100.0	100.0	100.0	0.7762	0.0049	66.6667
book_mouse_phone	100.0	100.0	100.0	100.0	3.2702	0.0313	8.7719
bottle_cup_keyboard	–	–	–	–	–	–	0.0
bottle_cup_remote	100.0	100.0	100.0	100.0	1.3415	0.0036	69.2982
bottle_mouse_book	100.0	100.0	100.0	100.0	1.0909	0.0122	64.2202
bottle_remote_bottle	100.0	98.6216	100.0	100.0	4.6851	0.0135	91.0714
bottle_teddy_keyboard	76.3494	67.3012	100.0	100.0	7.6579	0.0062	84.8214
coffee_cup_doll	83.0467	71.4660	100.0	100.0	8.4335	0.0138	94.0678
coffee_cup_mouse	–	–	–	–	–	–	0.0
cup_book_mouse	100.0	80.6865	100.0	100.0	6.3192	0.0086	93.7008
lap_cup_mouse	100.0	100.0	33.3333	83.3333	4.1354	0.1325	5.2632
lap_cup_phone	–	–	–	–	–	–	0.0
laptop_teddy_bottle	69.4245	56.6472	100.0	100.0	12.2020	0.0069	90.1639
laptop_teddy_cup	–	–	–	–	–	–	0.0
laptop_teddy_keyboard	100.0	100.0	100.0	100.0	1.3966	0.0059	51.2195
laptop_teddy_remote	79.7919	57.2562	100.0	100.0	8.9377	0.0060	89.9160
mouse_book_bowl	100.0	100.0	100.0	100.0	2.5174	0.0085	29.2517
remote_bottle_book	100.0	100.0	100.0	100.0	1.4189	0.0080	35.9223
teddy_keyboard_remote	80.1147	57.1335	100.0	100.0	9.7997	0.0081	96.5217
Average	93.0454	86.8195	95.8333	98.9583	4.6996	0.0172	51.4446

Table A5. Comparison of the average performance of the second set of 20 test sequences.

Sequence	RA@15° (%)	RA@10° (%)	CCA@0.1 (%)	CCA@0.2 (%)	MRE (°)	MTE	Completeness (%)
Proposed	77.9378	53.7577	41.4455	88.9312	10.8067	0.1218	100.0
Ray Diffusion	79.7557	55.6503	25.1766	62.9724	14.7265	0.1817	100.0
DUST3R ^a	84.5328	79.9534	63.0290	78.0242	14.5475	0.1603	100.0
ORB-SLAM2	93.0454	86.8195	95.8333	98.9583	4.6996	0.0172	51.4446

References

1. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* **2021**, *65*, 99–106.
2. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 139. [[CrossRef](#)]
3. Durrant-Whyte, H.; Bailey, T. Simultaneous Localization and Mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
4. Schonberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2016; pp. 4104–4113.

5. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; IEEE: New York, NY, USA, 2018; pp. 224–236.
6. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 4938–4947.
7. Zhang, J.Y.; Lin, A.; Kumar, M.; Yang, T.H.; Ramanan, D.; Tulsiani, S. Cameras as Rays: Pose Estimation via Ray Diffusion. *arXiv* **2024**, arXiv:2402.14817.
8. Besl, P.J.; McKay, N.D. Method for Registration of 3-D Shapes. In *Proceedings of the Sensor Fusion IV: Control Paradigms and Data Structures*; SPIE: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
9. Jeong, Y.; Park, S.Y. Camera Pose Estimation in Multi-Object Scenes Using Ray Diffusion and Point Cloud Alignment. In *Proceedings of the 2025 40th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, Wellington, New Zealand, 19–21 November 2025.
10. Chen, S.; Guo, H.; Zhu, S.; Zhang, F.; Huang, Z.; Feng, J.; Kang, B. Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 22831–22840.
11. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. *arXiv* **2024**, arXiv:2408.00714.
12. Segal, A.; Haehnel, D.; Thrun, S. Generalized-ICP. *Robot. Sci. Syst. (RSS)* **2009**, *2*, 435.
13. Sarlin, P.E.; Cadena, C.; Siegwart, R.; Dymczyk, M. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2019; pp. 12716–12725.
14. Zhang, J.Y.; Ramanan, D.; Tulsiani, S. RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer Nature: Cham, Switzerland, 2022; pp. 592–611.
15. Lin, A.; Zhang, J.Y.; Ramanan, D.; Tulsiani, S. RelPose++: Recovering 6D Poses from Sparse-View Observations. In *Proceedings of the International Conference on 3D Vision (3DV)*; IEEE: New York, NY, USA, 2024; pp. 106–115.
16. Wang, J.; Rupperecht, C.; Novotny, D. PoseDiffusion: Solving Pose Estimation via Diffusion-Aided Bundle Adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 9773–9783.
17. Wang, P.; Tan, H.; Bi, S.; Xu, Y.; Luan, F.; Sunkavalli, K.; Wang, W.; Xu, Z.; Zhang, K. PF-LRM: Pose-Free Large Reconstruction Model for Joint Pose and Shape Prediction. *arXiv* **2023**, arXiv:2311.12024.
18. Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; Revaud, J. Dust3r: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 20697–20709.
19. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1623–1637. [[PubMed](#)]
20. Ke, B.; Lee, S.; Metzger, N.; Müller, T.; Welling, M.; Geiger, A. Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. *arXiv* **2023**, arXiv:2312.02145.
21. Shao, J.; Yang, Y.; Zhou, H.; Zhang, Y.; Shen, Y.; Guizilini, V.; Wang, Y.; Poggi, M.; Liao, Y. Learning Temporally Consistent Video Depth from Video Diffusion Priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 22841–22852.
22. Hu, W.; Gao, X.; Li, X.; Zhao, S.; Cun, X.; Zhang, Y.; Quan, L.; Shan, Y. DepthCrafter: Generating Consistent Long Depth Sequences for Open-World Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 2005–2015.
23. Wang, Y.; Solomon, J.M. Deep Closest Point: Learning Representations for Point Cloud Registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2019; pp. 3523–3532.
24. Zhu, L.; Guan, H.; Lin, C.; Han, R. Leveraging Inlier Correspondences Proportion for Point Cloud Registration. *arXiv* **2022**, arXiv:2201.12094.
25. Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; Ilic, S.; Hu, D.; Xu, K. GeoTransformer: Fast and Robust Point Cloud Registration with Geometric Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9806–9821. [[CrossRef](#)] [[PubMed](#)]
26. Umeyama, S. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [[CrossRef](#)]
27. Mu, X.; Chen, J.; Zhou, Z.; Leng, Z.; Fan, L. Accurate Initial State Estimation in a Monocular Visual-Inertial SLAM System. *Sensors* **2018**, *18*, 506. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.