

LASER: Layer-wise Scale Alignment for Training-Free Streaming 4D Reconstruction

Supplementary Material

7. More Details about Submap Registration in Sim(3) Space

3D reconstruction in each window \mathcal{W}_i yields a local submap $\mathcal{S}_i = \{\mathbf{T}_t^{(i)} \mathbf{P}_t^{(i)}\}_{t \in \mathcal{W}_i}$ in the window's own coordinate system. We then estimate a similarity transform $(s_i^w, \mathbf{R}_i^w, \mathbf{t}_i^w) \in \text{Sim}(3)$ between \mathcal{S}_i to \mathcal{G}_{i-1} , which are defined in the world coordinate system (in our case, the first temporal window's coordinate system, based on the estimated point maps of the overlapping region. The induced camera pose in the world space for a frame $\mathbf{I}_{t \in \mathcal{W}_i}$ is $\mathbf{T}_t^w = (\mathbf{R}_i^w \mathbf{R}_t^{(i)} | s_i^w \mathbf{R}_i^w \mathbf{t}_t^{(i)} + \mathbf{t}_i^w)$. The global map \mathcal{G}_i is then updated progressively as $\mathcal{G}_i = \mathcal{G}_{i-1} \cup \{\mathbf{T}_t^w \mathbf{P}_t^{(i)}\}$, where $\mathcal{G}_0 = \emptyset$ in the initialization.

To estimate the Sim(3) transform, we first estimate the global scale factor s_i^w via a robust IRLS (Iteratively Reweighted Least Squares) optimization, enforcing a shared metric across two adjacent windows. Rotation and translation $(\mathbf{R}_i^w, \mathbf{t}_i^w)$ are then optimized via the Kabsch algorithm [22] under that metric using the *scaled* camera anchors based on the estimated s_i^w .

Scale estimation via IRLS based on point correspondences. We estimate the per-window scale s_i^w from point-wise correspondences in the intersection of two consecutive windows. Specially, for overlapping frames that share the same timestamp t in \mathcal{W}_{i-1} and \mathcal{W}_i , we extract 3D points for every pixel x in the intersection of the two windows $\mathbf{p}(x) = \mathbf{P}_t^{(i-1)}(x)$, $\mathbf{q}(x) = \mathbf{P}_t^{(i)}(x)$, and their associated confidences $c_p(x) = \mathbf{C}_t^{(i-1)}(x)$, $c_q(x) = \mathbf{C}_t^{(i)}(x)$. The set of *mutually confident correspondences* is then defined as:²

$$\mathcal{C} = \{(\mathbf{p}, \mathbf{q}) \mid c_p > g(\mathbf{C}_t^{(i-1)}), c_q > g(\mathbf{C}_t^{(i)})\}, \quad (3)$$

where g denotes the median function. Each pair $(\mathbf{p}, \mathbf{q}) \in \mathcal{C}$ represents the same 3D point in two coordinates of submaps, with both predictions considered reliable. We estimate the optimal scale s_i^w by solving the Huber-robust objective:

$$s_i^w = \arg \min_{s > 0} \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{C}} \rho(\|s \mathbf{p} - \mathbf{q}\|_2), \quad (4)$$

where $\rho(\cdot)$ is the Huber loss with parameter δ .

Rotation and translation based on scaled camera anchors. After estimating the global scale s_i^w from confident point correspondences, we scale the submap \mathcal{S}_i first and

then estimate the rigid transformation. We define canonical camera axes in each camera's coordinate system as the *up* $\mathbf{u} = (0, 1, 0)$ and *view* $\mathbf{v} = (0, 0, -1)$. Let $\mathcal{O}_i = \mathcal{W}_{i-1} \cap \mathcal{W}_i$ be the set of overlapping timestamps. Using the camera center $\mathbf{t}_t^{(i)}$ and normalized axes $(\mathbf{v}_t, \mathbf{u}_t)$, we form two scaled camera anchor sets $\{\mathbf{x}_t\}_{t \in \mathcal{O}_i}$ and $\{\mathbf{y}_t\}_{t \in \mathcal{O}_i}$, where:

$$\mathbf{x}_t = (s_i^w \mathbf{t}_t^{(i)}, s_i^w \mathbf{t}_t^{(i)} + \mathbf{v}_t^{(i)}, s_i^w \mathbf{t}_t^{(i)} + \mathbf{u}_t^{(i)}), \quad (5)$$

$$\mathbf{y}_t = (\mathbf{t}_t^{(i-1)}, \mathbf{t}_t^{(i-1)} + \mathbf{v}_t^{(i-1)}, \mathbf{t}_t^{(i-1)} + \mathbf{u}_t^{(i-1)}). \quad (6)$$

We then estimate the window-level rigid transform $(\mathbf{R}_i^w, \mathbf{t}_i^w)$ by minimizing the alignment error between the two anchor sets via the Kabsch algorithm [22]:

$$\mathbf{R}_i^w, \mathbf{t}_i^w = \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3} \sum_{t \in \mathcal{O}_i} \|\mathbf{R} \mathbf{x}_t + \mathbf{t} - \mathbf{y}_t\|_2^2. \quad (7)$$

Differences from existing approaches. Although VGGT-Long [9] also adopts a sliding-window strategy for streaming inputs, our method differs in how the registration Sim(3) is estimated from overlapping windows. VGGT-Long applies IRLS to jointly optimize a closed-form scale s together with \mathbf{R} and \mathbf{t} computed via the Kabsch algorithm. In contrast, we first estimate the scale using point-cloud correspondences within matched camera coordinate systems, and then estimate \mathbf{R} and \mathbf{t} using the scaled inputs. This two-stage procedure yields more stable and robust registration.

Furthermore, our SE(3) registration is obtained from minimal camera anchors derived directly from camera poses. These anchors avoid the artifacts introduced by point-map predictions and additionally preserve trajectory consistency, particularly in small-scale scenes.

We conduct ablation studies on these registration strategies in Sec. 9.

8. Implementation Details

We instantiate LASER using either VGGT [59] or π^3 [64] as the offline 3D reconstruction backbone. For video depth estimation, small-scale camera pose estimation, and indoor multi-view point map estimation, we evaluate both variants. For large-scale camera pose estimation on KITTI Odometry [16] and outdoor point map estimation on Waymo [54], we use π^3 as the backbone for its stronger geometric prior. On the kilometer-scale KITTI Odometry, we additionally incorporate loop closure following the VGGT-Long [9] configuration for fairness.

²To avoid notation clutter, we omit the variable x from now on.

Table 7. Outdoor, Long-term Point Map Estimation on **Waymo** [54]. We report Accuracy (Acc, lower is better), Completeness (Comp, lower is better) and Chamfer Distance (Chamfer, lower is better). We show metrics for each segment ID; Avg. is the mean across segments.

Segment ID	Metric	Avg.	163453191	183829460	315615587	346181117	371159869	405841035	460471311	520018670	610454533
VGGT-Long [9]	Acc ↓	0.508	0.453	0.096	0.629	1.441	0.457	0.379	0.510	0.481	0.129
	Comp ↓	0.456	0.412	0.101	0.552	1.341	0.365	0.344	0.496	0.386	0.102
	Chamfer ↓	0.482	0.432	0.098	0.591	1.391	0.411	0.361	0.503	0.434	0.115
π^3 -Long	Acc ↓	1.043	0.912	0.160	1.209	0.837	1.728	0.228	0.545	3.101	0.668
	Comp ↓	0.745	0.738	0.145	0.502	0.362	1.085	0.155	0.208	3.149	0.356
	Chamfer ↓	0.894	0.825	0.153	0.856	0.600	1.406	0.192	0.376	3.125	0.512
Ours (π^3)	Acc ↓	0.560	0.422	0.176	1.151	0.651	0.896	0.127	0.541	0.247	0.832
	Comp ↓	0.266	0.315	0.158	0.194	0.223	0.459	0.106	0.326	0.232	0.385
	Chamfer ↓	0.413	0.368	0.167	0.673	0.437	0.677	0.116	0.434	0.240	0.608

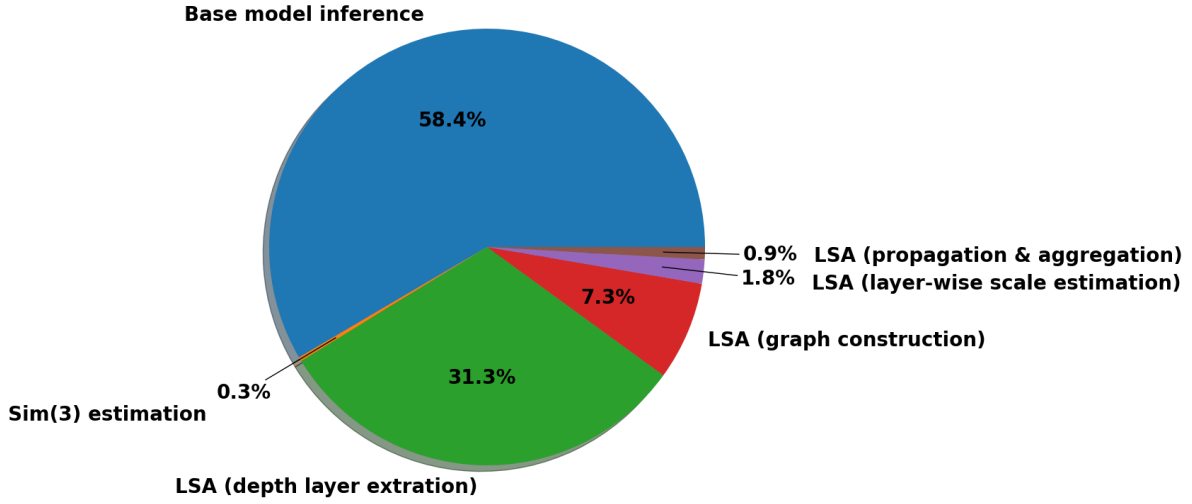


Figure 8. Runtime analysis of each module within the pipeline.

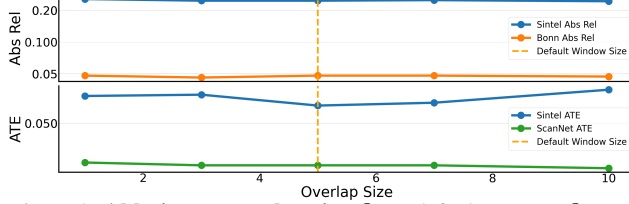


Figure 9. Ablation on overlap size O . In default, we use $O = 5$.

We use multi-threading to run model inference for each window and registration of adjacent window pairs with LSA refinement concurrently. At the beginning of depth graph construction, we try to assign each frame to separate available threading to achieve maximum parallelism.

9. Submap Registration (Sec. 7).

Fig. 9 examines the effect of selecting different overlap sizes O , our method is robust to a wide range of settings, the default choice ($O = 5$) maintains a good balance between accuracy and inference complexity. Fig. 10 and Tab. 8 compares alternative strategies for estimating the SE(3) trans-

form between submaps. Replacing IRLS with a closed-form solver degrades both depth and pose accuracy, confirming the importance of robust scale estimation in this stage. Replacing scaled camera anchors with scaled point maps produces similar depth metrics but noticeably weaker camera trajectories.

10. Time Analysis for the Registration Module

We also provide a detailed runtime analysis of each module in our framework, as shown in Fig. 8. Using a window size of 20 with an overlap of 5, the measured runtimes are as follows: π^3 single inference pass: 1.344 s; Sim(3) estimation: 0.007 s; depth-layer extraction: 0.719 s; graph construction: 0.168 s; scale initialization: 0.041 s; and propagation & aggregation: 0.021 s.

11. Evaluation Details of Efficiency Benchmark

We report FPS and peak memory usage on the Sintel [2] benchmark for all methods on an A6000 GPU. The image

Table 8. Ablation Studies for Submap Registration (Sec. 7). *w/o IRLS* denotes estimating scale via closed-form solution instead of IRLS. *w/o Anchor* denotes estimating rigid transformation on scaled point maps instead of scaled camera anchors.

	Sintel		Bonn		Sintel		
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑	ATE ↓	RPE _{trans} ↓	RPE _{rot} ↓
Ours	0.247	68.8	0.048	97.4	0.061	0.028	0.249
w/o IRLS	0.328	51.4	0.123	85.6	0.107	0.035	0.249
w/o Anchor	0.247	68.8	0.048	97.4	0.081	0.039	0.742

resolution for DUST3R-based [63] methods is 512×288 except Spann3R, which only supports 224×224 , and VGGT-based [59] methods are 518×294 .

12. Outdoor Multi-view Point Map Estimation

Tab. 7 reports long-term multi-view point map estimation results. LASER using π^3 as backbone achieves the best overall performance among training-free methods, substantially outperforming both VGGT-Long [9] and π^3 -Long in Comp and Chamfer while maintaining comparable Acc.

For outdoor setting, we use the Waymo Open Dataset [54] on urban driving segments and report Acc, Comp, and Chamfer distance, following [9] (results in the supplementary material). To mitigate artifacts from sky and far-background regions, we uniformly filter out the lowest-confidence 40% of predicted points for *all* methods; these results serve as a comparative reference rather than a strict head-to-head benchmark.

13. Future Directions

Although our method demonstrates strong performance, it has room for improvements. We show some failure cases in Fig. 12, and list three directions that interest us most:

- **Different hyperparameters for indoor and outdoor scenes.** Our framework requires empirical hyperparameter tuning for diverse environments (e.g., window size, overlap ratio, and depth-layer confidence thresholds). While this manual tuning improves stability for each domain, it reduces the generality of our method and makes adaptive adjustment when transferred to new settings an interesting direction to explore.
- **Performance bounded by backbone reconstructors.** Because our system is built on top of offline 3D reconstructors, its performance is heavily dependent on the backbone submap prediction quality. For example, when using VGGT as backbone, our method inherits VGGT’s inability to handle dynamic or non-rigid scenes. As VGGT struggles to maintain reliable geometry and camera pose estimates in the presence of moving objects, our method also fails under such conditions. This dependency limits applicability to fully static or quasi-static scenes. We look forward to seeing how advancement on offline 3D reconstructors can boost our method as well.

- **LSA Sensitivity to complex scenes and object occlusions.** Our current depth layer graph construction used in LSA is based purely on predicted frame-wise depth maps, without persistent contexts of distinct objects, which results in multiple objects being assigned to the same depth layer within complex scenes, also unable to recover layer correspondences if an object is occluded entirely and later re-enters the scene. We seek to improve the robustness of our design and the temporal trackability of different objects by incorporating appearance information from raw images.

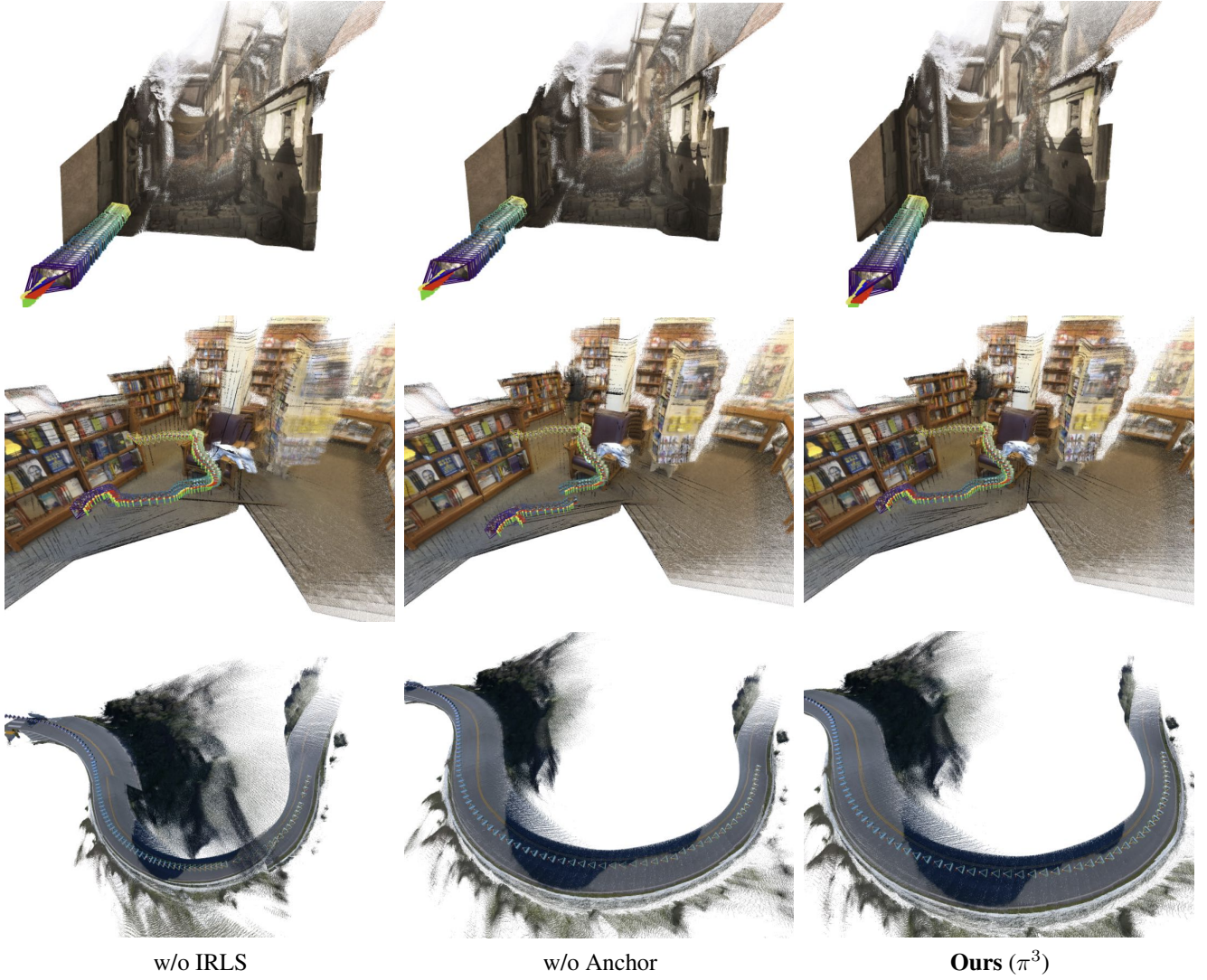


Figure 10. Ablation Studies for Submap Registration (Sec. 7). *w/o IRLS* denotes estimating scale via closed-form solution instead of IRLS. *w/o Anchor* denotes estimating rigid transformation on scaled point maps instead of scaled camera anchors.

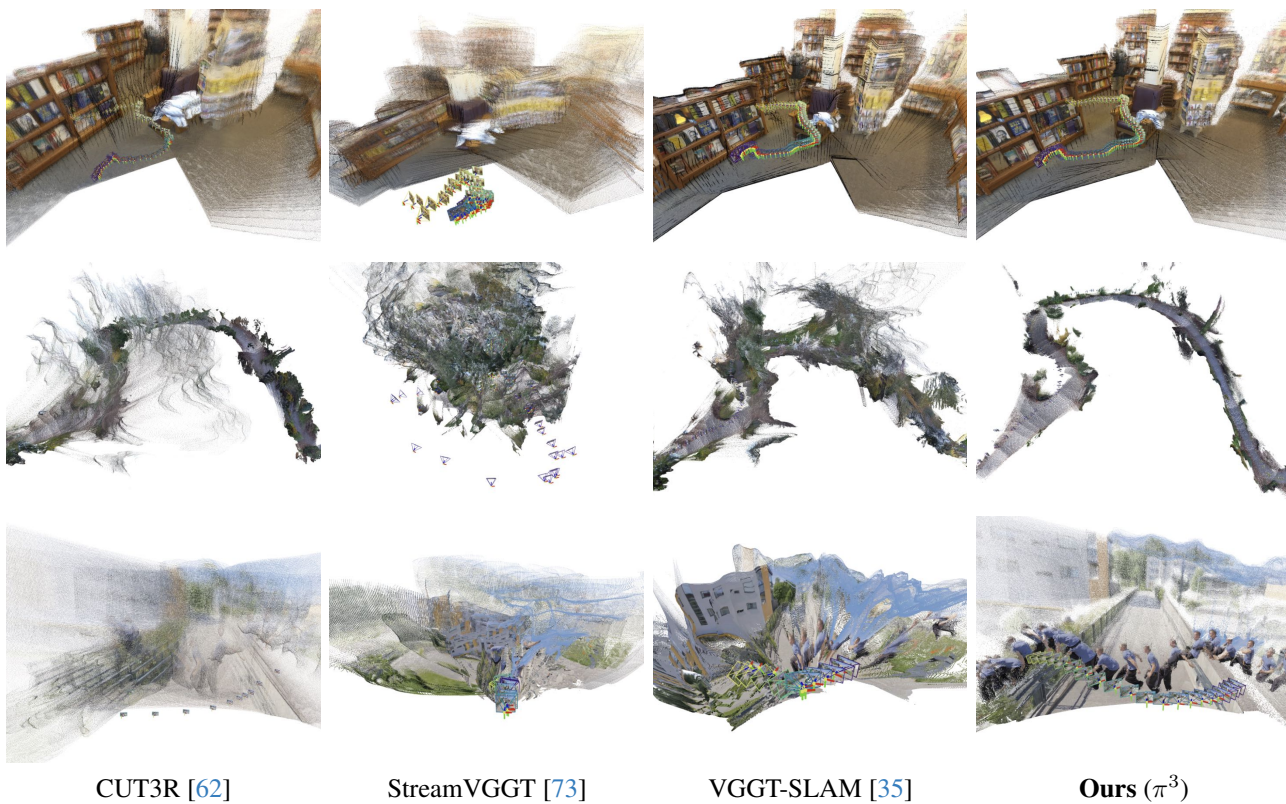


Figure 11. Qualitative comparison on different sequences.

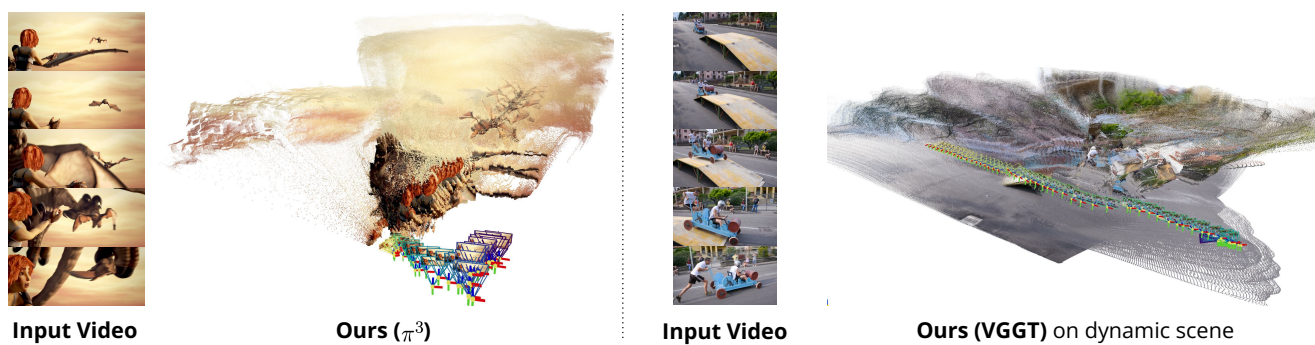


Figure 12. Failure Cases.