

ORBIT: Benchmarking SfM in the Wild with 360° Video

Supplementary Material

A. Method Parameter Details

Metric Scale Recalibration. When computing metric scale for each reconstructed sequence in our dataset (Section 3.2.2 of the main paper), for the sake of computational efficiency, we subsample the frames by $10\times$. Furthermore, for each cube face, if there are more than 100 visible points, we only consider the first 100 visible points visible to that face for the purpose of estimating metric scale. For depth estimation, we project the 360° frame into 90° FOV cube faces, resized to 512×512 . We use the default configuration for Depth Pro [1] as follows:

- patch encoder preset: “dinov2116 384”
- image encoder preset: “dinov2116 384”
- decoder features: 256
- use fov head: True
- fov encoder preset: “dinov2116 384”.

Ground Truth Cross-Validation. When cross-validating the data, we use as ground truth (Section 3.2.3), in order to account for different speeds of motion in different cameras, we define a per clip threshold measurement as the median of translational distances between nearby frames in the ground truth (rig-based) trajectory. As such, we compare the ATE not only with a fixed threshold based on meters but also with its median speed moment as well. We define the per clip threshold τ as:

$$\tau(g) = \text{med}(\|g_i - g_{i+3}\|_2^2) \quad (4)$$

In order for a rig-based 360° trajectory g to be considered verified against a per-cube-face trajectory e , g should satisfy all of the following:

$$\text{ATE}(g, e) < 2 * \tau(g) \quad (5a)$$

$$\text{RPE}_r(g, e) < 0.2 \quad (5b)$$

$$\text{RPE}_t^*(g, e) < 1.0, \quad (5c)$$

where $\text{RPE}_t^*(g, e)$ is a normalized version of Eq. 2 where we divide by the corresponding translational differences in g .

Test Trajectory Viewpoint Selection: Way Points. When creating test trajectories from our panoramic video, we sample rotational angles are sampled every 30th frame from a normal distribution $\mathbf{l}_i \sim N([0, 0, 0], [1, 20, 2])$ (interpolated between keyframes using spherical linear interpolation), and add jitter noise to the rotation sampled per-frame according to a smaller normal distribution: $\mathbf{j}_i \sim N([0, 0, 0], [.1, .1, .1])$. We sum these samples to form into \mathbf{c}_i and finally use spherical linear interpolation calculated

for all frames. Furthermore, we set the initial rotational (view) matrix B either to identity or to the challenging cube face as described in Sec. 3.3 in the main paper.

The final rotational matrix we use for frame i is:

$$\mathbf{c}_i = \sum_0^i \mathbf{l}_i \quad (6)$$

$$R_i = J_i C_i B \quad (7)$$

J_i and C_i are rotational matrix equivalents of the rotational degrees \mathbf{j}_i and \mathbf{c}_i correspondingly.

B. Extended Benchmark: ORBIT 2

We prepare an extended version of ORBIT called ORBIT 2 which has 390 clips, with FOVs varying between 30 to 120 degrees (Fig. 8). The basic ORBIT has fewer clips with the FOV fixed around 120 degrees. Furthermore, ORBIT 2 has different rotational patterns (Fig. 9), mainly cinematic, scanning, orbit, panning, way points, role and original.

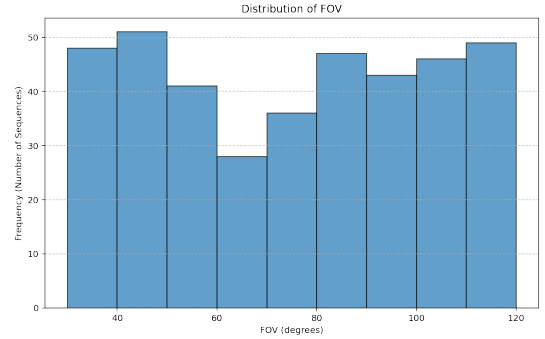


Figure 8. **ORBIT 2 FOV Distribution** – The distribution of fields of view for the clips in ORBIT 2.

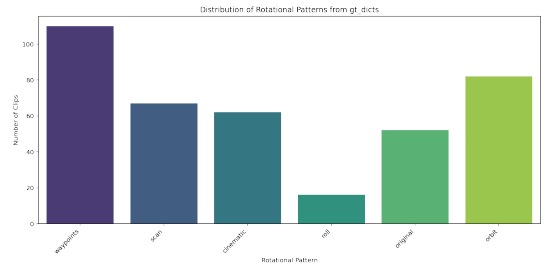


Figure 9. **ORBIT 2 Rotation Pattern Distribution** – The number of clips in the benchmark for each pattern.

The results in Tab. 3 show similar success rate as V1 in Tab. 1 with MegaSAM as the best performing method. Additional masking with RoMo gives a slight boost to

Category	COLMAP		MegaSaM		MonST3R		RoMo+MegaSaM		ORB-SLAM2		VGGT-Long	
	ATE	RPE-R	ATE	RPE-R	ATE	RPE-R	ATE	RPE-R	ATE	RPE-R	ATE	RPE-R
Low Texture	4.2544	2.5611	2.9133	1.0022	8.2462	1.5075	3.7133	1.0778	6.0343	2.1243	3.1911	1.7378
Low Light	4.0622	3.6561	2.3144	0.4839	3.6259	0.9376	1.3583	0.5550	4.4873	1.1764	1.9422	0.7322
Crowd	6.2549	2.1338	4.7566	0.6330	5.2545	1.1289	4.1454	0.5989	5.2389	1.1185	2.8232	1.1607
Parallel Object (PO)	9.0892	2.0881	8.1703	0.7258	5.4722	1.1303	6.7292	0.7050	5.6681	1.0429	3.6456	1.1331
Fluid	0.8233	1.7875	0.3983	0.4383	1.2908	1.0792	0.4400	0.4233	1.4000	0.6543	0.6983	0.9833
1st Challenge	PO	Light	PO	Light	Texture	Texture	PO	Texture	Texture	Texture	PO	Texture
2nd Challenge	Crowd	Texture	Crowd	PO	PO	PO	Crowd	PO	PO	Light	Texture	Crowd

Table 2. **Categorized performance of methods.** Each row in the top section of the table reports ATE and RPE-R for each method on a subset of clips in ORBIT that present a challenge listed in Sec. 4.1. The best performing method in each category is boldfaced. The final two rows note the two most challenging categories for that method according to each metric. We observe that each method tends to have sensitivity to distinct kinds of challenges. Also, apart from the presence of fluids, all categories appear at least several times among the most challenging categories. The most challenging categories of challenge we observe include low-textured scenes and videos containing objects moving in parallel with respect to the camera.

Method	ATE ↓	ATE Median ↓	RPE-R ↓	RPE-T ↓	Success Rate % ↑
COLMAP	1.00 ± 1.81	0.19	1.26 ± 2.35	0.14 ± 0.34	41.38
MegaSAM	0.65 ± 1.41	0.06	0.58 ± 1.80	0.06 ± 0.15	61.69
RoMo+MegaSaM	0.53 ± 1.16	0.05	0.51 ± 1.45	0.06 ± 0.15	66.32
OrbSlam2	2.00 ± 2.48	1.19	1.53 ± 1.99	0.17 ± 0.31	3.34
VGGT-Long	1.04 ± 1.36	0.54	1.82 ± 3.08	0.15 ± 0.23	0.77
Monst3r	1.40 ± 1.77	0.78	2.20 ± 3.12	0.19 ± 0.40	1.54

Table 3. **ORBIT-2 Benchmark result.** Here we report the mean and standard deviation of ATE, RPE-T, and RPE-R as well as success rate percentages for a strict (Success) and the median of ATEs.

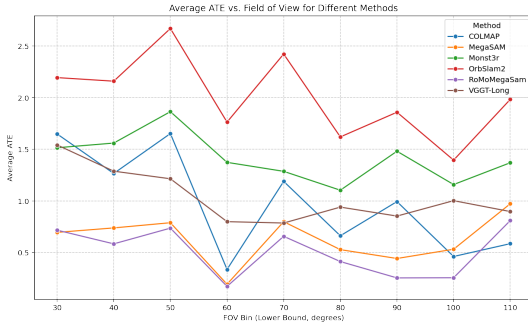


Figure 10. **Average ATE for FOV bins in ORBIT 2 –**

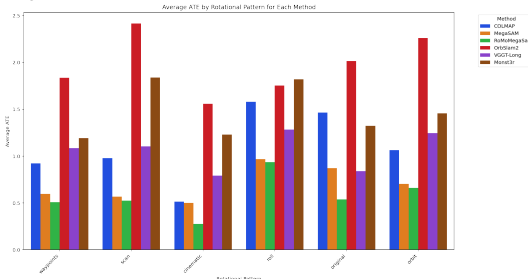


Figure 11. **Average ATE based on Rotation Pattern in ORBIT 2 –**

MegaSaM which is more pronounced in v2. The feed-forward methods such as VGGT-Long and Monst3r show

less than 2%. Since ORBIT is an extremely challenging dataset with a great degree of variation in terms of challenges each models faces it results on methods failing to estimate a close enough trajectory in majority of the cases. As such the standard deviation of average metrics are often quite high. Therefore, we also report the median of the ATE over clips to further show case the skew of the result’s distribution. While the average ATE of VGGT-Long and Colmap are close the ATE median of Colmap is 5 times smaller than VGGT-Long. This difference is more bold when comparing MegaSam and feedforward methods, which further highlights our conclusions that while feed forward methods might be successful in terms of average ATE when compared with methods incorporating optimization techniques, they are more than 90% of the time imprecise and therefore not applicable for many applications such as 3D reconstruction. The results per FOV and rotational pattern are depicted in Fig. 10 and Fig. B.

C. Benchmark Method Configurations

For the sake of aligning the timestamps between ground truth and trajectories if some of the frames are dropped, we use the last previously estimated camera position for the dropped frames.

COLMAP [18]. We set the COLMAP settings as follows:

- Matcher method: EXHAUSTIVE MATCHER
- Target percent images initial SfM pass: 1.0
- Min images initial SfM pass: 500
- Max images initial SfM pass: 1024
- Mapper bundle adjustment global function tolerance: $1e-6$
- Bundle Adjustment function tolerance: $1e-6$
- Camera model: SIMPLE RADIAL

MegaSaM [13]. We use Depth Pro for monocular depth estimation, as well as relative depth for sky prediction and metric depth aligning. We use the default setting for all other hyperparameters.

RoMo [5]+MegaSaM [13]. We first run RoMo [5] with its default settings, with two iterations using SAM 2 [16] features as input as well as RAFT [21] optical flow estimates and a final SAM 2 [16] refinement. We use a batch size of 16 and a learning rate of $2e-2$ and a ratio threshold of 0.5 for finding trusted frames.

MonST3R [31]. In contrast to multi-frame feedforward methods (*e.g.*, VGGT [27] and VGGT-Long [3]), MonST3R is designed for two-frame inputs rather than sequences exceeding a thousand frames. To adapt the model for longer sequences, we first resize the input images such that the shorter dimension is 224 pixels. We then employ window-based inference with a window size of 50 and an overlap ratio of 0.3. For all remaining hyperparameters, we follow to the original configuration.

VGGT-Long [3]. We use the default base config of VGGT-Long in their official release GitHub repository with a chunk size of 60, overlap of 30, and loop chunk size of 20. This method uses dense alignment, and we use the following IRLS parameters of $\delta = 0.1$, $\text{max_iters} = 5$, and $\text{tolerance} = 1e-9$.

ORB-SLAM2 [15]. We use the default settings of ORB-SLAM2 with a maximum of 4,000 features per frame and a camera scale of 1.0. A sequence was counted a failure if fewer than 30 frames were tracked or if 2 consecutive frames were dropped. As a result, ORB-SLAM2 only outputs camera estimations for 65 clips and fails on the rest.

D. Challenge sub-categorization

To further analyze the statistics of ORBIT in terms of types of challenge scenarios, we label each clip based on the presence of each of the types challenges in Sec. 4.1 in the main paper, *i.e.*, low texture, low light, crowds (camera-independent dynamic objects), the presence objects moving in parallel with the camera, and the presence of large bodies of water/fluids. Overall, we have 9, 18, 74, 36, and 12 clips respectively in each category. Table 2 reports the ATE and RPE-R for each method on subsets of clips corresponding to each challenge category. Based on the results, we ob-

served that the most challenging category is the presence of an object moving alongside the camera, which significantly degrade the performance of MegaSaM and COLMAP. On the other hand, MonST3R and ORB-SLAM2, struggle most when faced with low-texture scenes. VGGT-Long struggles most with low texture and the presence of moving objects either independent or parallel to camera. We also observe that using RoMo masking for MegaSaM usually improves the rotational estimate and ATE of MegaSaM significantly in cases with parallel-motion and low light, but worsens results on low-texture scenes, which is in line with our expectations for a motion-masking method.

Overall, Table 2 shows that ORBIT exposes a diverse set of challenges and is a valuable tool for diagnosing and analyzing current state-of-the-art methods by highlighting their failure modes.

E. Trajectory Comparisons

Please see the accompanying video folders as well as the provided webpage for better visualizations of our test clips and of the qualitative performance of each benchmark method. Fig. 12, Fig. 13, Fig. 14 and Fig. 15 show sample frames and trajectory graphs comparing the ground truth trajectory and each method’s output on clips from ORBIT. We observe in these trajectory comparisons that feed-forward methods like MonST3R and VGGT-Long almost never produce a perfectly matching estimate, but are robust in that they are usually never further than a certain tolerance from ground truth. On the other hand, bundle adjustment-based methods like COLMAP and MegaSaM are a perfect match for many frames but they can go completely out of bounds on other frames.

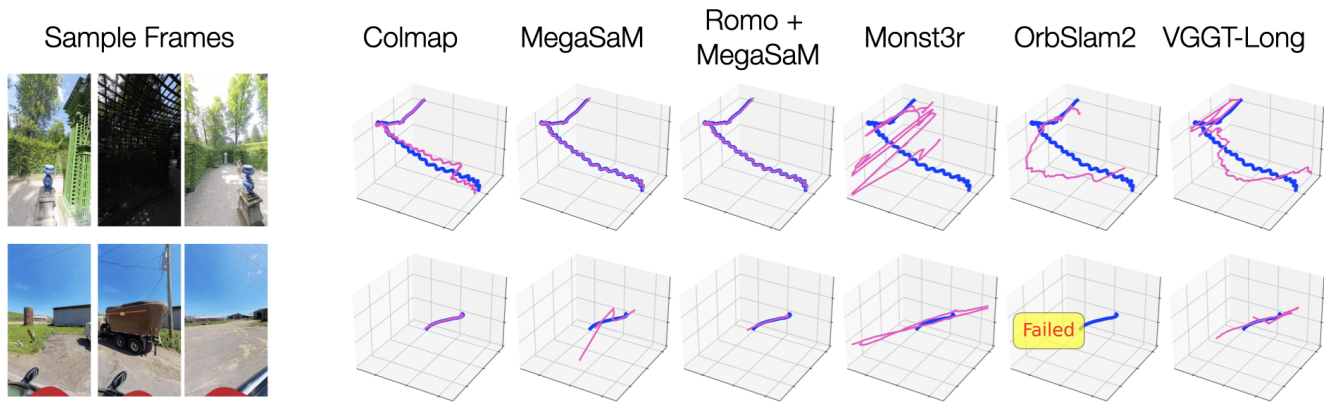


Figure 12. **Trajectory Comparisons.** Sample frames from ORBIT clips alongside trajectory comparisons between ground truth and each method’s estimated trajectory. Ground truth is shown in blue.

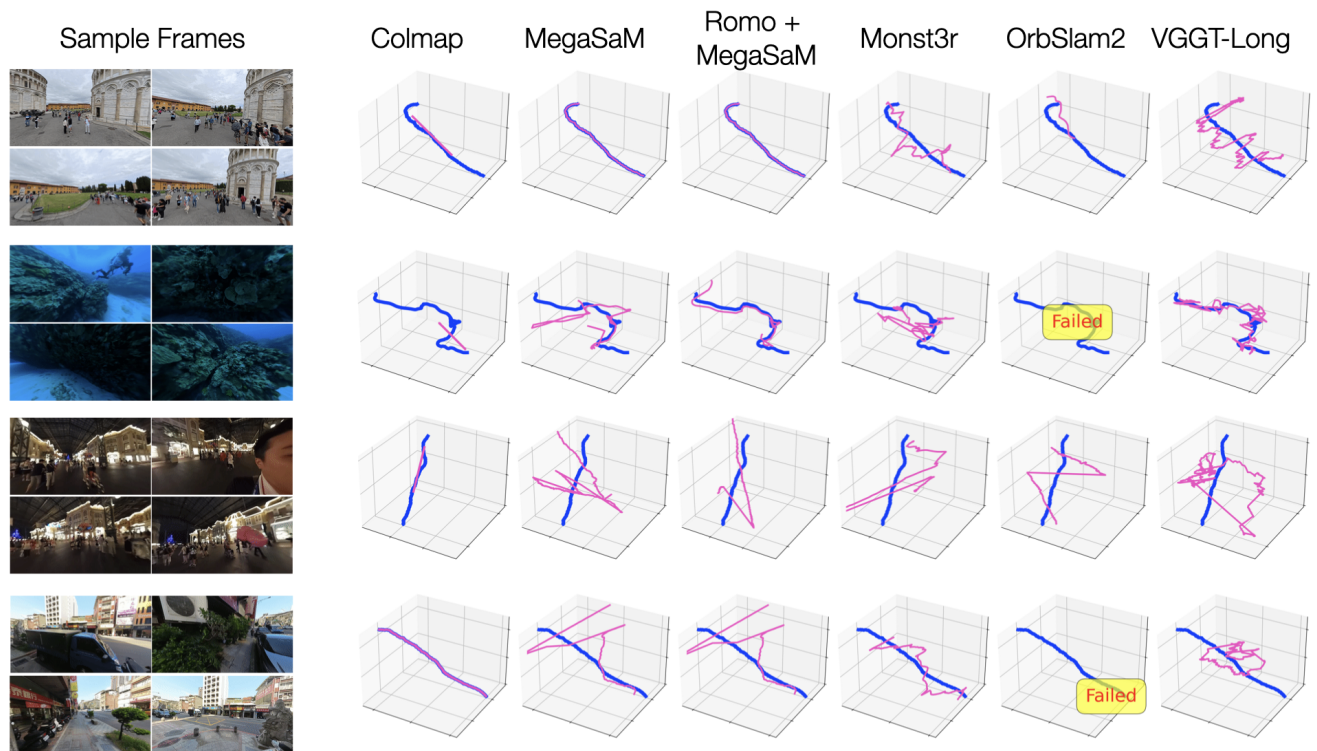


Figure 13. **Trajectory Comparisons.** Sample frames from ORBIT clips alongside trajectory comparisons between ground truth and each method’s estimated trajectory. Ground truth is shown in blue.

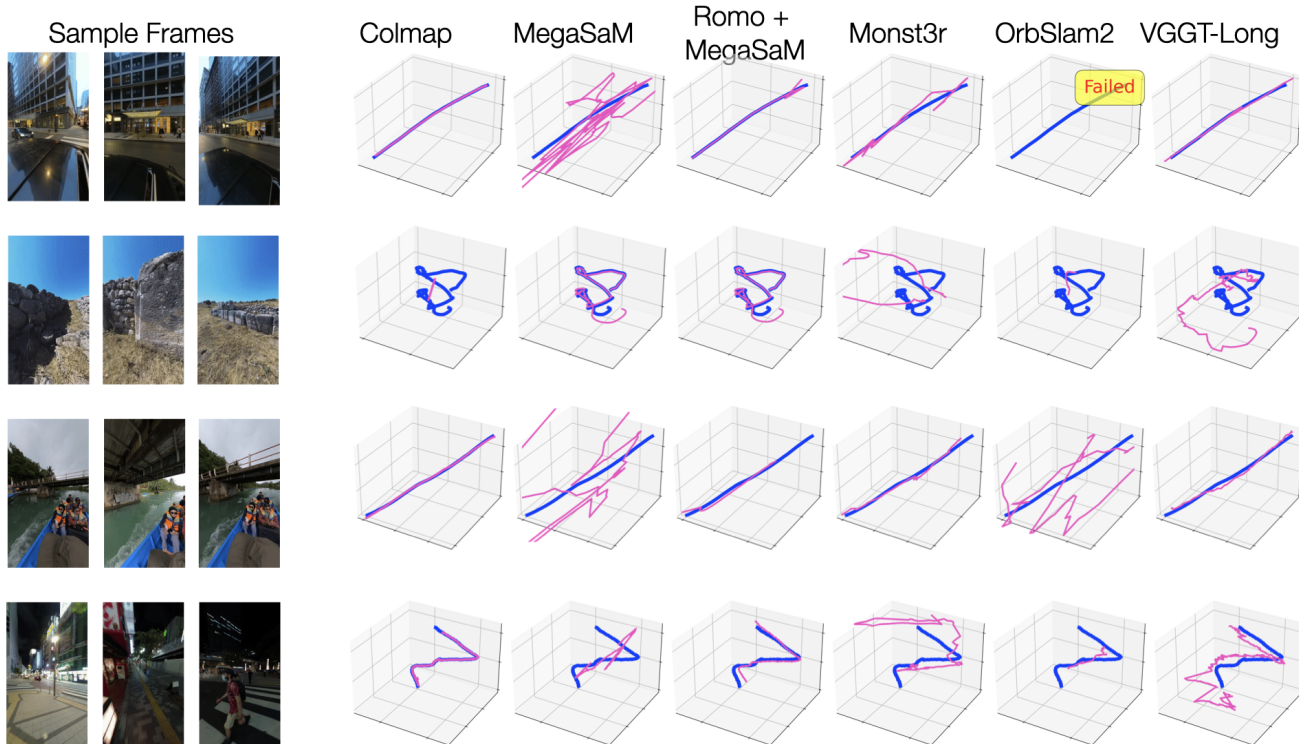


Figure 14. **Trajectory Comparisons.** Sample frames from ORBIT clips alongside trajectory comparisons between ground truth and each method’s estimated trajectory. Ground truth is shown in blue.

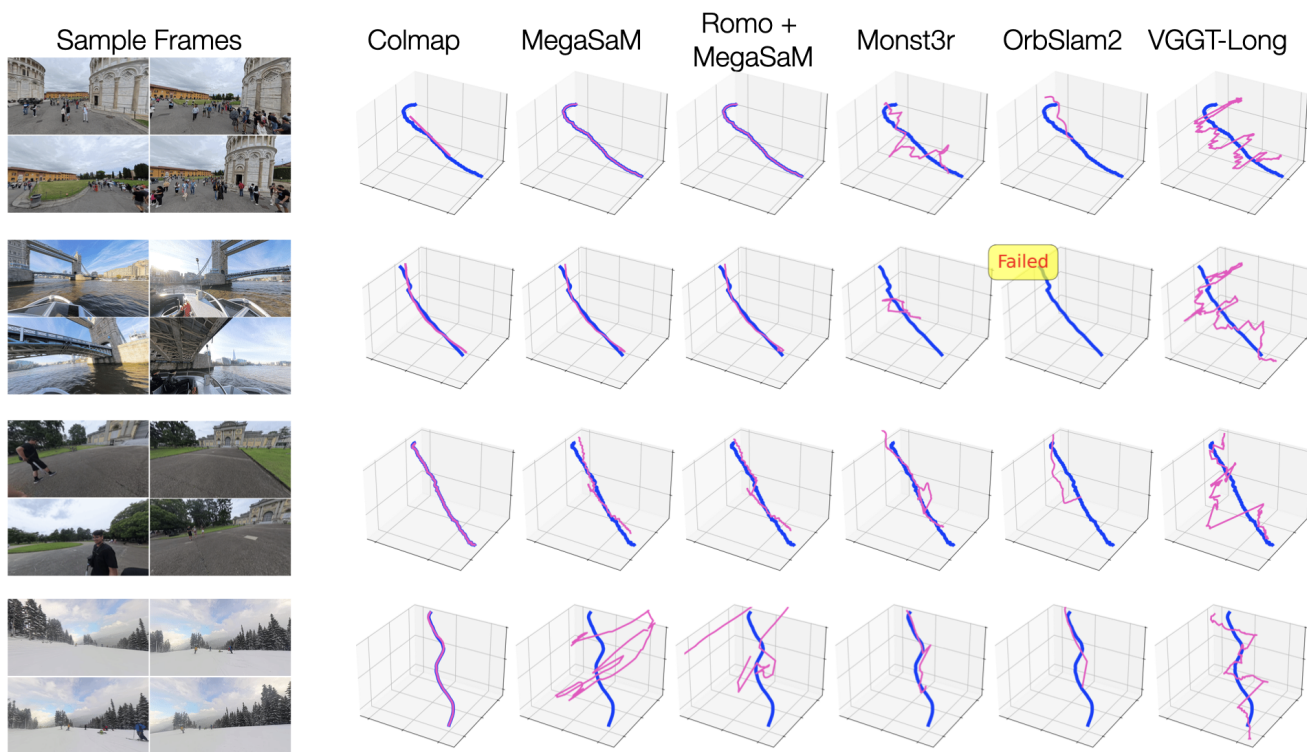


Figure 15. **Trajectory Comparisons.** Sample frames from ORBIT clips alongside trajectory comparisons between ground truth and each method’s estimated trajectory. Ground truth is shown in blue.