Paparazzo: Active Mapping of Moving 3D Objects

Davide Allegro[1] Shiyao Li[2] Stefano Ghidoni[1] Vincent Lepetit[2]

[1]University of Padova
[2]LIGM, École Nationale des Ponts et Chaussées, IP Paris, Univ Gustave Eiffel, CNRS

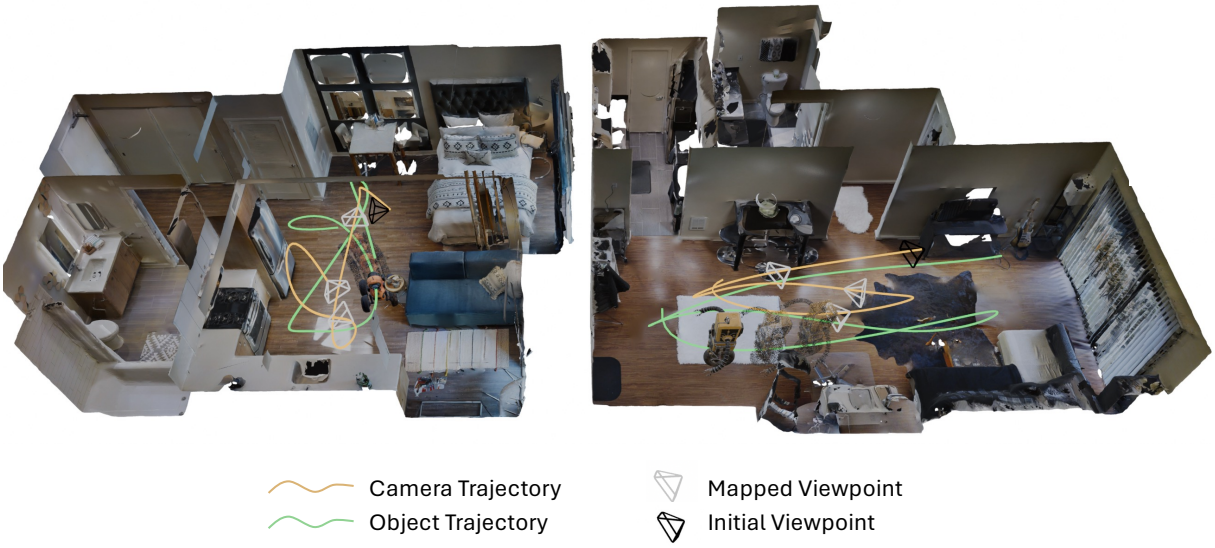

Figure 1. We introduce the novel task of active mapping of moving objects, requiring agents to plan observation trajectories while compensating for target motion. We also propose a method to solve this task and a benchmark for evaluation.

Abstract

Current 3D mapping pipelines generally assume static environments, which limits their ability to accurately capture and reconstruct moving objects. To address this limitation, we introduce the novel task of active mapping of moving objects, in which a mapping agent must plan its trajectory while compensating for the object's motion. Our approach, Paparazzo, provides a learning-free solution that robustly predicts the target's trajectory and identifies the most informative viewpoints from which to observe it, to plan its own path. We also contribute a comprehensive benchmark designed for this new task. Through extensive experiments, we show that Paparazzo significantly improves 3D reconstruction completeness and accuracy compared to several strong baselines, marking an important step toward dynamic scene understanding. Project page: https://davidea97.github.io/paparazzo-page/

1. Introduction

Scene exploration and mapping have been extensively studied in computer vision and robotics [1, 2, 31], with renewed interest driven by autonomous systems like drones and potential applications such as automated digital-twin generation. Existing exploration methods, however, rely on the assumption that the scene is static. This assumption fails in many real settings where moving objects constitute essential components of the environment. For example, in a construction site, trucks and mobile equipment are key elements of the scene that continuously reshape the workspace and accurately modeling them is important for maintaining up-to-date digital twins. However, the site cannot be halted to capture them in static conditions.

This is why we introduce a new task: active mapping of moving objects. An autonomous agent must reconstruct the 3D geometry of a non-cooperative object that moves independently of the mapping activity. This task is challenging

because the agent must gather views that reveal new parts of the object while compensating for the object's future motion during its own navigation. As a result, viewpoint quality depends jointly on geometric informativeness and the feasibility of reaching that view at the right time.

To solve this new task, we propose "Paparazzo", a learning-free framework for active 3D reconstruction of dynamic objects. Paparazzo considers a set of viewpoints distributed in a foveal configuration around the target object and moving with it over time. To select the most informative viewpoints, we rely on Fisher Information computed from a 3D Gaussian Splatting model [10], while, to predict the object trajectory and the future positions of these viewpoints, we leverage an Extended Kalman Filter [24]. Crucially, the viewpoint with the highest expected information gain is not always the optimal choice: because viewpoints move with the target object, a view with slightly lower information gain but significantly shorter travel time may be more beneficial and efficient. For this reason, we explore multiple strategies that jointly account for information gain and motion feasibility when selecting the next best viewpoint. Since Paparazzo requires no training data, it generalizes to new scenes and previously unseen objects.

A first advantage of the Extended Kalman Filter (EKF) is that it can combine past observations of the object to accurately predict its trajectory. We also use it to detect when the trajectory prediction is not reliable, e.g. when the object changes moving direction abruptly. When the EKF novelty indicates unreliable prediction, Paparazzo switches to a mode where it continuously adjusts the agent position to keep the object centered in the camera's field of view and within an optimal range, prioritizing observations that enhance motion estimation.

For evaluation, we introduce a comprehensive benchmark and protocol for active mapping in dynamic environments, measuring reconstruction fidelity, spatial coverage, and temporal consistency across several baselines. We assume access to the target object's mask whenever it is visible. In practice, this can be achieved by background subtraction when the static scene is known, or alternatively by using a moving-object segmentation method when it is not [23, 26]. For this benchmark, we developed a simulator based on the Habitat simulator [22] generating complex motions of target objects within different environments.

Our experiments show that Paparazzo significantly improves 3D reconstruction fidelity and mapping efficiency compared to several baselines, marking a key step toward intelligent scene understanding in dynamic environments.

Our key contributions can be summarized as:

- We introduce the novel task of active mapping of moving objects, where an agent must efficiently reconstruct the 3D geometry of non-cooperative, independently moving targets.
- We propose Paparazzo, a learning-free dual-mode framework combining 3D Gaussian Splatting-based information gain with EKF-based motion prediction.
- We present the first benchmark for this task and demonstrate large performance gains across multiple dynamic scenarios.

2. Related Work

2.1. Active Mapping for Static Scenes

The goal of active mapping is usually to determine how an agent should move to efficiently explore and reconstruct an unknown 3D environment. Exploration must be exhaustive: by the end of the task, the agent should have covered the entire scene while keeping its trajectory as short as possible.

Works on active mapping can be broadly categorized into traditional and learning-based approaches. Traditional methods primarily rely on heuristic strategies, such as frontier-based exploration [6, 31] and next-best-view (NBV) selection [20, 21], or a combination of both [3, 5], to guide the robot's exploration process. They often employ voxel grids or point clouds to represent the scene.

Learning-based approaches have recently emerged to leverage deep neural networks and more expressive scene representations. For example, MACARONS [8] uses neural networks to predict the coverage gain of candidate camera poses, effectively guiding NBV selection. NextBestPath [17] learns to predict the piece of trajectories that maximizes the cumulative coverage gain along the path.

With the advent of NeRF [19] and 3D Gaussian representations [15], recent works emerged, such as ANM [32], NARUTO [7], and ActiveGS [11], that train such models as the intermediate scene state, using measures such as confidence or Fisher information to determine the next-best pose. Combined with traditional path planning algorithms, these methods achieve impressive performance in producing high-quality 3D reconstructions.

To the best of our knowledge, all active mapping works consider a static scene. In this paper, we are interested in mapping a non-cooperative mobile object, which is much more challenging as we need to estimate the object motion and compensate for it when planning the next move of our agent.

2.2. Mobile Object Passive Reconstruction

Our work is related to the reconstruction of mobile objects, such as in-hand scanning, where a target object is moved in front of one or several cameras [9, 12, 25, 27–29]. Like us, they aim to reconstruct an object while estimating its motion within a scene. In particular, [12] also uses Gaussian primitives to represent the object as we do. The key difference is that in the case of in-hand scanning, a user moves the object aiming to improve the reconstruction, which means

the object motion is intended to support the task, so it is "cooperative". In our case, we need to plan how to move the agent in the environment to capture new relative poses between the object and the agent, in addition to track and reconstruct the object. In such a scenario, the object moves in a "non-cooperative" manner, meaning that the object does not move in a way that facilitates its reconstruction.

3. Paparazzo

As shown in Figure 2, our "Paparazzo" method alternates between two operating phases depending on the confidence of its estimate of the target object's motion:

- *Object Tracking Mode* (Section 3.4): Paparazzo switches to this mode when its estimate of the object's motion is uncertain. It then keeps the target object in its field of view to improve this estimate.
- *Object Mapping Mode* (Section 3.5): Paparazzo switches to this mode when it is confident enough of the object's motion. It then plans motions to informative viewpoints for efficient reconstruction and executes them.

3.1. Problem Formulation

Let an agent equipped with a fixed, front-facing RGB-D camera C operate in a 3D world frame W containing a dynamic object O. We denote $T_A^B \in SE(3)$ as the rigid transformation from frame A to frame B, represented as a 4×4 homogeneous matrix. At each discrete time step k, the camera pose $T_{C_k}^W$ is assumed to be known from the agent's localization system, while the object pose $T_{O_k}^W$ is unknown and must be estimated. When the object is detected, its segmentation mask $\mathcal{M}_k$ allows extracting the corresponding 3D points $\mathcal{P}_{O_k}^{C_k} = \{p_j^{C_k}\}_{j=1}^{P}$ in the camera frame. Each 3D point $p_j^{C_k} \in \mathbb{R}^3$ is obtained by back-projecting the pixels within $\mathcal{M}_k$ using the available depth information and the camera intrinsic parameters.

The objective is to determine informative viewpoints that can be reached by the agent and enable it to efficiently observe and reconstruct the complete surface of the moving object with minimal views, producing a consistent 3D model in the object's local reference frame while predicting and adapting to its motion.

3.2. Initialization

At the first detection time t_d, we initialize the object pose from the 3D object points $\mathcal{P}_{O_{t_d}}^{C_{t_d}}$. The object translation $t_{O_{t_d}}^{C_{t_d}} \in \mathbb{R}^3$ is defined as the centroid of these points. The object rotation $R_{O_{t_d}}^{C_{t_d}} \in SO(3)$ is constructed by aligning its z-axis with the world vertical direction, while the x–y axes are obtained by performing PCA on the points projected onto the ground plane. The initial object pose expressed in the world coordinate system therefore is:

$$T_{O_{t_d}}^W = T_{C_{t_d}}^W T_{O_{t_d}}^{C_{t_d}}, \quad \text{with} \quad T_{O_{t_d}}^{C_{t_d}} = \begin{bmatrix} R_{O_{t_d}}^{C_{t_d}} & t_{O_{t_d}}^{C_{t_d}} \\ 0 & 1 \end{bmatrix}. \tag{1}$$

We initialize Gaussian primitives $\mathcal{G}_O$ from the object's segmented RGB-D observation, as in the SplaTAM backbone [14]. Although SplaTAM was originally designed for static scenes, we expressed $\mathcal{G}_O$ in the object reference frame by means of the estimated object pose that remains consistent across time as the object moves.

3.3. EKF-Based Motion Prediction

We rely on an Extended Kalman Filter (EKF) defined on $SE(3)$ to estimate the object state, composed of the object pose $T_{O_k}^W$ and its linear and angular velocities, together with its associated covariance matrix P_k.

We quantify our confidence in the estimated object state with two complementary metrics. The first is $U_k = \mathrm{tr}(P_k)$, which provides a compact measure of the state uncertainty. The second is the Normalized Innovation Squared (NIS), which quantifies the consistency of a new measurement $T_{O_k}^{W,\text{meas}}$ of the target object pose with the current predicted object pose $T_{O_{k|k-1}}^W$:

$$\mathrm{NIS}_k = y_k^\top S_k^{-1} y_k, \tag{2}$$

where $y_k = \log((T_{O_{k|k-1}}^W)^{-1} T_{O_k}^{W,\text{meas}})^\vee$ is the innovation on $SE(3)$ and $S_k = H P_{k|k-1} H^\top + R$ is the corresponding innovation covariance.

We consider the EKF estimate reliable when $U_k < \tau_u$ and $\mathrm{NIS}_k < \tau_n$ for N_s consecutive steps. If this condition is met, Paparazzo switches to *Object Mapping Mode* to perform information-driven active exploration and refine the reconstruction. Otherwise, the system transitions to the *Object Tracking Mode* to re-localize the object and stabilize the EKF.

3.4. Object Tracking Mode

The goal of this mode is to prioritize frequent observations of the target object in order to refine motion estimates. To this end, the agent actively keeps the object within the camera's field of view while continuously updating its reconstruction and motion estimate. At each time step, the agent rotates to move the segmentation mask toward the image center, and translates to adjust its distance to the object so that the object's apparent size remains approximately half of the image.

We also estimate the object pose $T_{O_k}^{W,\text{meas}}$. This is done by aligning the segmented point cloud $\mathcal{P}_{O_k}^{C_k}$ with the object reconstruction accumulated up to time $k-1$. To this end, we first use KISS-Matcher [18] to obtain a coarse but globally

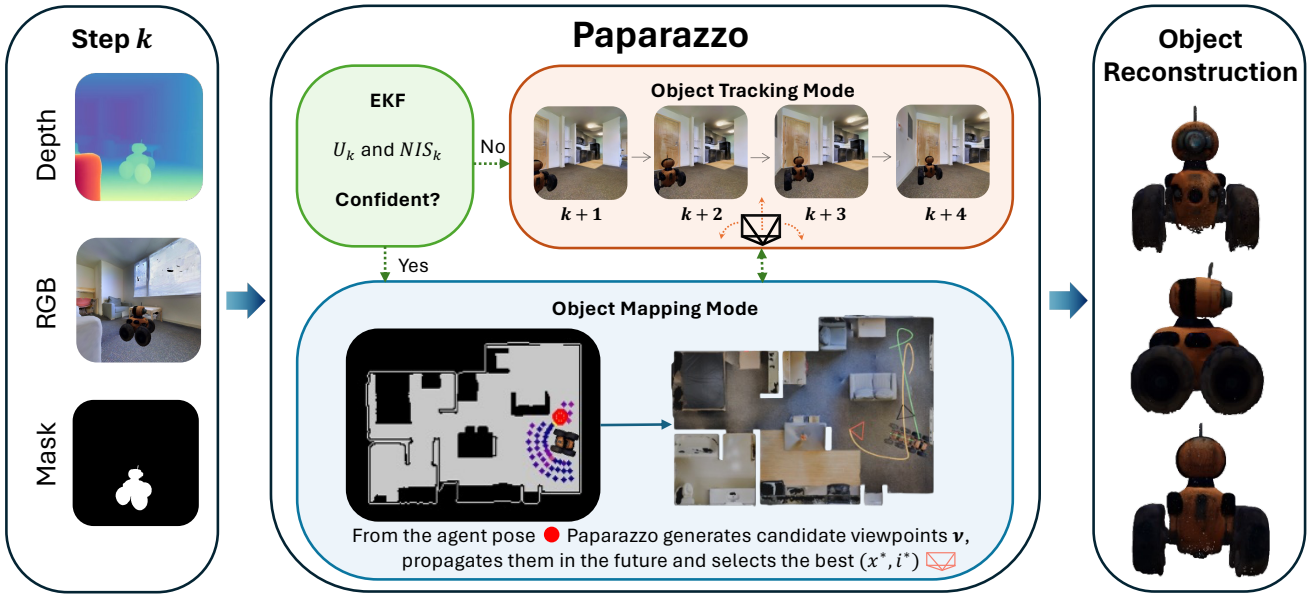

Figure 2. Paparazzo alternates between Object Tracking Mode and Object Mapping Mode based on the confidence of the EKF motion estimate. When the filter is uncertain, the agent prioritizes acquiring stabilizing observations; once confident, it predicts future object motion, generates and propagates candidate viewpoints, and selects the optimal one $(\mathbf{x}^*, i^*)$ that minimizes the final cost function.

consistent registration, robust to outliers and large displacement, and then refining this transformation using Colored ICP [16].

We then use $T_{O_k}^{W,\text{meas}}$ to update the EKF and improve the estimate of the object state, and to integrate the newly observed object point cloud into the object reconstruction. The Gaussian Splatting model $\mathcal{G}_O$ is concurrently refined using the SplaTAM optimization process, which incrementally densifies and updates the dynamic object representation using the new RGB-D observation transformed into the object frame.

Once the state uncertainty stabilizes, Paparazzo shifts its focus from re-localization to information-driven exploration, leveraging the learned object GS model to guide active reconstruction.

3.5. Object Mapping Mode

When the EKF stabilizes, Paparazzo transitions to the *Object Mapping Mode*. The goal of this mode is to move the agent to poses that will significantly improve the object reconstruction, while taking into account the object motion predicted by the EKF.

As shown in Figure 2, we sample candidate viewpoints $\mathcal{V}$ relative to the object reference frame, so that they move together with it. The camera centers corresponding to these viewpoints are distributed around the object in a foveated configuration, and the cameras point toward the object.

If the object were static, we could simply select the most informative viewpoints in $\mathcal{V}$ according to FisherRF [10] applied to the object GS model $\mathcal{G}_O$. However, since the object is moving, we must trade off between (i) the informativeness of a viewpoint and (ii) the temporal synchronization between the agent and the moving object. To quantify this trade-off, we introduce the following criterion:

$$B(\mathbf{x}, i) = -w_{\text{eig}}\text{EIG}(\mathbf{x}) + w_{\text{sync}}C_{\text{sync}}(\mathbf{x}, i)\,, \qquad (3)$$

where $\text{EIG}(\mathbf{x})$ is the FisherRF informativeness associated with the candidate viewpoint $\mathbf{x} \in \mathcal{V}$, and $C_{\text{sync}}(\mathbf{x}, i)$ is a criterion we introduce to measure how well the agent can synchronize with the motion predicted for the object when attempting to observe the object from viewpoint $\mathbf{x}$. Weights w_{sync} and w_{eig} balance the contribution of the two terms.

The FisherRF criterion quantifies how much a new viewpoint contributes to refining the parameters $\boldsymbol{\theta}$ of the current Gaussian Splatting representation $\mathcal{G}_O$ of the object. It can be computed analytically and efficiently from $\mathcal{G}_O$. More details can be found in [10].

The term $C_{\text{sync}}(\mathbf{x}, i)$ is defined as:

$$C_{\text{sync}}(\mathbf{x}, i) = \left|\hat{s}_{\text{agent}}(\mathbf{x}, i) - (i - k)\right|. \qquad (4)$$

Here, $\hat{s}_{\text{agent}}(\mathbf{x}, i)$ denotes the number of motion steps required for the agent to reach the camera pose $T_{O_i}^W \cdot \mathbf{x}$, where $T_{O_i}^W$ is the object pose predicted by the EKF for future time step i. We compute $\hat{s}_{\text{agent}}(\mathbf{x}, i)$ using an A* motion planner [13]. The term $i - k$ denotes instead the number of predicted time steps required for the object to evolve from its current pose $T_{O_k}^W$ to the predicted pose $T_{O_i}^W$. Thus,

$C_{\text{sync}}(\mathbf{x}, i)$ measures the temporal mismatch between the agent reaching the viewpoint associated with the object at time step i and the object itself reaching its predicted pose at that same time. We finally select the viewpoint that yields the best trade-off over a horizon of N_h future time steps:

$$(\mathbf{x}^*, i^*) = \underset{\mathbf{x}\in\mathcal{V},(i-k)\leq N_h}{\arg\min} B(\mathbf{x}, i)\,. \tag{5}$$

While moving the agent to pose $T_{O_i^*}^W \cdot \mathbf{x}^*$, Paparazzo continuously integrates new RGB-D observations into the object point cloud $\mathcal{P}$, updates $\mathcal{G}_O$, and monitors the EKF consistency. If the NIS or state uncertainty exceeds their confidence thresholds, mapping is halted, and the system reverts to *Object Tracking Mode*, maintaining a reactive loop between mapping and tracking. This dynamic coupling of information-driven mapping and motion-aware prediction constitutes the core novelty of Paparazzo, enabling 3D reconstruction of moving objects without assuming static scenes. Notably, Paparazzo runs online at 8 FPS. Full runtime and memory details are reported in the supplementary material.

4. Experimental Results

To evaluate our Paparazzo method, we introduce a dedicated benchmark and evaluation protocol designed to assess both reconstruction fidelity and spatial coverage over time. Experiments are conducted within Habitat 3.0 [22], a high-performance 3D simulator that provides realistic indoor environments and robot displacements.

We selected six photorealistic indoor scenes—three from the Matterport3D dataset (M) [4] and three from the Gibson dataset (G) [30]—commonly used for static active mapping [32]. To extend these static scenes to dynamic scenarios, we introduce a synthetic moving target object into each environment. We consider the four target objects shown in Figure 3. Each object is evaluated independently across all scenes in separate runs. This setup ensures statistical diversity across both objects and environments. The agent is equipped with an RGB-D sensor and initialized in a navigable pose, with the target object placed in front of it in a random position and orientation.

Object Motion Protocol. To comprehensively assess reconstruction performance under diverse object motion dynamics, we consider four motion patterns for the target:

- **Bouncing Ball:** upon collision, the object randomly changes orientation and continues in the new direction.
- **Forward & Backward:** the object moves along a straight line without changing orientation, moving forward until collision and then reversing direction.
- **Stop & Go:** similar to *Bouncing Ball*, but with intermittent stops—pausing every S steps and resuming after G steps—to simulate non-uniform velocity.

Figure 3. The four target objects used in our experiments, featuring different shapes and colors. Each object is evaluated independently across all environments.

- **Curved Bouncing Ball:** the object follows curved trajectories; upon collision, it randomly changes orientation and speed, then restarts along a new curved direction.

Each motion pattern introduces distinct challenges related to motion predictability, visibility continuity, and temporal occlusions. Each translation of the object is 5 cm per step, each rotation is 10° per frame.

Agent Motion Model. The agent navigates using a discrete action space made of standard motion primitives:

- **Forward:** move ahead by 15 cm,
- **Rotate Left / Right:** yaw rotation by 10°.

These control primitives follow the canonical Habitat discretization for active perception tasks, allowing reproducible and physically plausible exploration behavior.

4.1. Evaluation Protocol

We report both geometric accuracy and exploration efficiency across all experiments. Metrics are computed by comparing the reconstructed point cloud $\mathcal{P}$ against the ground-truth object model $\mathcal{P}_{GT}$ at the final time step and throughout exploration. We considered three metrics:

- **3D Coverage [%]**: the percentage of ground-truth points that have at least one reconstructed point within a distance threshold $\tau = 1$ cm (higher = more complete reconstruction);

- **Completeness [cm]**: the mean distance from each ground-truth point to its nearest reconstructed point (lower = fewer missing regions).
- **AUC**: the area-under-curve of coverage with respect to the agent steps, reflecting how efficiently coverage increases during exploration (higher = faster coverage).

All metrics are tested across all the scenes, objects, and motion patterns detailed above.

Baselines. We compare Paparazzo against three baselines designed to isolate the contributions of viewpoint selection, motion prediction, and temporal feasibility:

- Random Walk (RW): a classical baseline in active mapping for static scenes. The agent moves randomly across the environment, accumulating object point clouds whenever the object falls within its field of view, without considering the object motion.
- Random Informative Selection (RIS): an ablation of our method that selects, at each mapping iteration, a random feasible pose among the $N_h \times |\mathcal{V}|$ informative candidate viewpoints, ignoring both the synchronization cost and the predicted feasibility of observing the object from that position.
- Tracking-Only (TO): we keep the agent in Object Tracking Mode, a purely passive strategy that continuously tracks the object's motion using the EKF but performs no active viewpoint selection or mapping, serving as a lower bound for reconstruction completeness.

These baselines allow us to disentangle the impact of Paparazzo's key components—motion-aware viewpoint selection and temporal feasibility reasoning—on overall reconstruction quality and mapping efficiency.

4.2. Results

Table 1 gives the quantitative results across six different indoor scenes for our four dynamic motion types. All reported values are averaged over all test objects within each scene and across five runs, with each run consisting of 500 agent steps. For all experiments, we configured Paparazzo to predict up to $N_h = 60$ future steps using the EKF.

Over all configurations, Paparazzo consistently outperforms the baselines in terms of coverage, completeness, and AUC, demonstrating its superior efficiency in reconstructing dynamically moving objects. Its adaptive strategy, alternating between object tracking and mapping modes, allows it to handle different motion complexities more effectively than static or passive baselines.

Bouncing Ball (BB) motion. Although this case might be expected to be more challenging due to the object's unpredictable motion when bouncing, reconstruction is generally easier. The object's frequent bounces against walls allow it to be seen from multiple viewpoints, benefiting all methods. In this case, the Tracking-only (TO) baseline achieves reasonably good results, reaching an average coverage of 75%. However, Paparazzo consistently outperforms all baselines, achieving higher coverage than TO and surpassing 80%. The advantage is even more evident when considering the AUC, where Paparazzo outperforms every baseline across almost all scenes, confirming its efficiency in dynamic object reconstruction.

In contrast, the Random Walk (RW) baseline performs poorly, as it completely disregards the object's position. The Random Informative Selection (RIS) baseline achieves slightly better results, but still underperforms compared to TO and Paparazzo, since it lacks synchronization with the object's trajectory. Consequently, it frequently fails to reach feasible observation points in time, leading to incomplete reconstructions.

Curved Bouncing Ball (CBB) motion. This scenario reflects more real-world conditions, where the object follows curved trajectories and varies its speed, introducing unpredictability that makes trajectory estimation more challenging. As a result, Paparazzo's performance drops significantly across all metrics. Notably, in the Ribera scene, Paparazzo achieves a coverage approximately 5% lower than TO, likely due to the narrow environment, which limits the agent's ability to anticipate the motion and reposition effectively before the next bounce, especially when the object moves faster. However, when averaged across all scenes, Paparazzo remains clearly superior for the CBB motion.

Forward & Backward (FB) motion. This condition is more challenging because the object does not rotate, requiring the agent to actively reason about the scene in order to reposition and observe it from new viewpoints. Consequently, performance metrics are generally lower. Nevertheless, Paparazzo achieves higher coverage than all baselines and a higher AUC across all scenes. TO performs worse due to its passive policy, which limits its ability to observe the object from diverse viewpoints. Finally, RW and RIS perform poorly due to their lack of object awareness and temporal anticipation.

Stop & Go (SG) motion. This represents the most challenging scenario, as the object intermittently pauses during motion, introducing unpredictable interruptions that hinder motion estimation. This motion pattern best reflects realistic conditions, where objects may temporarily stop or slow down. Consequently, all methods exhibit a general performance drop compared to the BB motion.

For Paparazzo, difficulties arise when the agent has already planned to move toward an informative pose where the object is expected to appear, but the object stops earlier in a non-visible region. In such cases, the agent continues

Table 1. Quantitative results across scenes for the four dynamic motion types (Bouncing Ball (BB), Curved Bouncing Ball (CBB), Forward & Backward (FB), and Stop & Go (SG)). Reported values are averaged over all test objects and runs. Each entry shows Coverage (%), Completeness (cm), and AUC.

Motion / Method		Denmark (G)			Ribera (G)			Greigsville (G)			PuKPg4mmafe (M)			GdvgFV5R1Z5 (M)			pLe4wQe7qrG (M)			Average		
		Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC
BB	RW	57.21	1.79	0.56	55.93	1.81	0.55	49.01	2.20	0.48	51.10	2.16	0.50	51.46	2.02	0.50	44.31	2.49	0.44	51.50	2.08	0.51
	RIS	74.32	0.80	0.65	64.37	1.27	0.58	65.60	1.13	0.61	63.73	1.24	0.59	70.83	0.96	0.61	63.55	1.32	0.57	67.07	1.12	0.60
	TO	83.08	0.66	0.77	76.10	0.87	0.69	71.20	1.06	0.66	75.71	0.89	0.68	75.85	0.90	0.71	73.41	1.03	0.69	75.89	0.90	0.70
	Paparazzo	**86.93**	**0.61**	**0.81**	**80.28**	**0.81**	**0.73**	**77.13**	**0.88**	**0.72**	**79.30**	**0.83**	**0.73**	**83.10**	**0.76**	**0.77**	**82.31**	**0.71**	**0.74**	**81.51**	**0.77**	**0.75**
CBB	RW	49.32	2.19	0.49	58.17	1.76	0.57	48.48	2.31	0.48	51.55	2.18	0.51	52.85	2.00	0.52	47.14	2.27	0.47	51.25	2.12	0.50
	RIS	67.69	1.00	0.63	62.24	1.28	0.55	56.40	1.52	0.54	65.06	1.13	0.60	58.60	1.47	0.56	53.34	1.72	0.51	60.56	1.35	0.56
	TO	70.46	1.01	0.67	**70.84**	**1.05**	0.62	68.88	1.05	0.66	68.62	1.19	0.64	68.76	**1.04**	0.65	61.33	1.32	0.59	68.15	**1.11**	0.64
	Paparazzo	**74.99**	**0.96**	**0.71**	67.52	1.28	**0.63**	**72.30**	**1.01**	**0.70**	**72.39**	**1.08**	**0.68**	**71.48**	1.05	**0.68**	**65.66**	**1.28**	**0.62**	**70.73**	**1.11**	**0.67**
FB	RW	50.03	2.16	0.49	53.21	1.91	0.53	47.44	2.35	0.47	53.25	1.98	0.52	51.49	2.02	0.51	46.01	2.35	0.46	50.24	2.13	0.50
	RIS	48.06	2.17	0.46	54.61	1.70	0.49	59.65	1.51	0.53	58.49	1.56	0.53	52.07	1.80	0.49	54.85	1.73	0.51	54.62	1.75	0.50
	TO	53.34	1.93	0.52	66.45	1.23	0.61	59.44	1.49	0.56	66.27	1.22	0.60	64.36	1.35	0.59	62.44	1.47	0.58	62.05	1.45	0.58
	Paparazzo	**60.83**	**1.53**	**0.57**	**71.13**	**1.11**	**0.63**	**72.45**	**1.04**	**0.66**	**67.01**	**1.20**	**0.63**	**69.60**	**1.18**	**0.62**	**65.34**	**1.35**	**0.61**	**67.73**	**1.23**	**0.62**
SG	RW	56.59	1.82	0.56	56.53	1.83	0.55	49.09	2.24	0.48	52.42	2.13	0.51	51.90	1.98	0.51	44.28	2.48	0.44	51.80	2.08	0.51
	RIS	56.98	1.58	0.56	44.68	2.33	0.44	49.78	2.16	0.49	45.61	2.33	0.45	48.09	2.10	0.47	46.31	2.33	0.46	48.58	2.14	0.48
	TO	68.94	1.05	0.67	58.16	1.54	0.56	60.90	1.42	0.60	55.47	1.71	0.54	63.97	1.25	0.62	57.29	1.51	0.56	60.79	1.41	0.59
	Paparazzo	**79.70**	**0.79**	**0.77**	**71.20**	**1.12**	**0.65**	**62.85**	**1.38**	**0.62**	**56.03**	**1.69**	**0.55**	**72.22**	**1.04**	**0.68**	**66.19**	**1.20**	**0.63**	**68.03**	**1.20**	**0.65**

Table 2. Quantitative comparison of different motion types averaged across all scenes. We evaluate four dynamic behaviors (Bouncing Ball (BB), Curved Bouncing Ball (CBB), Forward & Backward (FB), and Stop & Go (SG)) and report results for all test objects. Each cell shows Coverage (%), Completeness (cm), and AUC.

Motion / Method		Object 1			Object 2			Object 3			Object 4			Average		
		Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC
BB	RW	57.30	1.84	0.56	64.19	1.24	0.64	43.99	2.64	0.43	40.54	2.59	0.40	51.50	2.08	0.51
	RIS	75.84	0.78	0.68	63.74	1.14	0.60	68.24	1.10	0.59	60.45	1.45	0.53	67.07	1.12	0.60
	TO	86.94	0.60	0.80	71.89	0.95	0.69	78.25	0.86	0.71	66.49	1.19	0.59	75.89	0.90	0.70
	Paparazzo	**90.64**	**0.55**	**0.83**	**79.19**	**0.78**	**0.75**	**82.60**	**0.73**	**0.75**	**73.60**	**1.00**	**0.67**	**81.51**	**0.77**	**0.75**
CBB	RW	56.04	2.07	0.55	62.73	1.33	0.62	43.68	2.59	0.43	42.55	2.49	0.42	51.25	2.12	0.50
	RIS	66.90	1.07	0.62	62.87	1.18	0.61	58.26	1.61	0.53	54.20	1.55	0.51	60.56	1.35	0.56
	TO	76.13	0.87	0.71	67.98	1.09	0.66	67.36	**1.13**	0.61	61.12	1.34	0.57	68.15	**1.11**	0.64
	Paparazzo	**79.13**	**0.83**	**0.74**	**71.39**	**1.04**	**0.69**	**68.70**	1.24	**0.64**	**63.68**	**1.33**	**0.60**	**70.72**	**1.11**	**0.67**
FB	RW	53.81	1.87	0.53	65.02	1.22	0.64	41.28	2.84	0.41	40.84	2.58	0.41	50.24	2.13	0.50
	RIS	57.10	1.64	0.51	59.85	1.31	0.57	50.55	2.09	0.45	50.98	1.96	0.47	54.62	1.75	0.50
	TO	63.43	1.45	0.58	67.30	1.10	0.66	57.27	1.77	0.53	60.19	1.47	0.55	62.05	1.45	0.58
	Paparazzo	**73.40**	**1.03**	**0.65**	**73.24**	**0.97**	**0.70**	**62.14**	**1.53**	**0.57**	**62.13**	**1.41**	**0.56**	**67.73**	**1.23**	**0.62**
SG	RW	58.19	1.80	0.57	66.35	1.17	0.66	42.86	2.71	0.42	39.82	2.64	0.40	51.80	2.08	0.51
	RIS	51.49	2.09	0.50	57.23	1.40	0.57	44.36	2.71	0.43	41.23	2.34	0.41	48.58	2.14	0.48
	TO	65.42	1.31	0.63	67.27	1.09	0.66	57.66	1.65	0.56	52.82	1.60	0.52	60.79	1.41	0.59
	Paparazzo	**73.27**	**1.04**	**0.69**	**72.97**	**0.97**	**0.71**	**64.07**	**1.41**	**0.61**	**61.82**	**1.38**	**0.59**	**68.03**	**1.20**	**0.65**

its motion without visual confirmation, making subsequent re-localization significantly more challenging. Nevertheless, Paparazzo still achieves substantially higher coverage and completeness than all baselines.

Paparazzo's behavior allows it to effectively exploit stopping phases when the object remains visible, continuing the mapping and refining the reconstruction from different viewpoints utilizing the available observation time. This advantage is reflected in the AUC, which increases by at least 10% in almost all scenes compared to the best-performing baseline. In contrast, the Tracking-only (TO) mode remains idle, waiting for motion to resume, highlighting the limitations of a purely passive strategy, while RIS again performs poorly due to its lack of temporal synchronization.

Overall, these results highlight that Paparazzo's adaptive alternation between Object Tracking Mode and Object Mapping Mode enables it to robustly handle diverse dynamic scenarios and motion complexities. By effectively balancing tracking accuracy with active exploration, Paparazzo consistently outperforms all baselines across motion types, leveraging motion variations and stop phases to achieve superior reconstruction performance.

To further assess the robustness and generality of the proposed method with respect to object variability, we report in Table 2 the quantitative results obtained by Paparazzo and the baseline methods for each target object. This analysis is crucial to demonstrate that our approach generalizes well across diverse objects, especially since Paparazzo does

Figure 4. **Visualization of the 3D reconstruction of Object 3 and Object 4 under Stop & Go motion.** We compare the RW, RIS, and TO baselines against our Paparazzo method. Paparazzo produces significantly more complete and geometrically consistent reconstructions.

not rely on object-specific training or fine-tuning. On average, Paparazzo improves coverage by nearly 10% for Objects 1, 2, and 4. A smaller gain is observed for Object 3, where the improvement over the best-performing baseline (TO) is about 6.5% on average. In particular, under the Curved Bouncing Ball (CBB) motion, TO slightly outperforms Paparazzo in completeness while achieving comparable coverage of Object 3. This behavior is likely due to the object's front-facing high-frequency texture, which biases the informative viewpoint selection toward similar orientations, reducing the effective diversity of the captured views. Nevertheless, Paparazzo still maintains the best overall average performance across all objects and metrics. The improvement is particularly significant in the Stop & Go (SG) motion, where Paparazzo achieves more than 10% higher coverage and AUC than the best-performing baseline. This result highlights its ability to dynamically balance exploration and reconstruction under less predictable or partially observable object trajectories (see Fig. 4).

All these findings confirm that Paparazzo's policy, alternating between object tracking and active mapping, generalizes effectively across both shape and appearance variations. Its design allows robust handling of heterogeneous objects and motion behaviors without any task-specific training, demonstrating strong potential for deployment in real-world dynamic reconstruction scenarios.

5. Conclusion

In this work, we introduced the new task of active mapping of a rigid moving object, a setting that departs from the long-standing assumption of static scenes in exploration and reconstruction.

We proposed Paparazzo, a learning-free framework that integrates motion prediction, information-driven viewpoint selection, and behavioral adaptation to the target object's dynamics. Our experiments show that Paparazzo substantially surpasses existing baselines, demonstrating that explicitly reasoning about where to look, when to look, and how to adapt the agent's behavior to the object's motion is crucial for efficient and accurate reconstruction of moving objects. By continuously balancing viewpoint informativeness, motion feasibility, and field-of-view maintenance, Paparazzo enables a form of dynamic scene understanding that was previously unexplored.

We believe our work establishes an important foundation: active viewpoint planning coupled with motion-aware behavior is key for bringing 3D reconstruction beyond static scenes and toward real dynamic environments.

Acknowledgment

This work was in part supported by the European Union (ERC Advanced Grant explorer Funding ID #101097259)

References

[1] Joseph E. Banta, L. R. Wong, Christophe Dumont, and Mongi A. Abidi. A Next-Best-View System for Autonomous 3D Object Reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(5):589–598, 2000. 1

[2] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In *International Conference on Intelligent Robots and Systems*, pages 540–545, 2002. 1

[3] Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. *Robotics: Science and Systems*, 5:2, 2021. 2

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv Preprint*, 2017. 5

[5] Howie Choset, Sean Walker, Kunnayut Eiamsa-Ard, and Joel

Burdick. Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph. *International Journal of Robotics Research*, 19(2):126–148, 2000. 2

[6] Christian Dornhege and Alexander Kleiner. A Frontier-Void-Based Approach for Autonomous Exploration in 3D. *Advanced Robotics*, 27(6):459–468, 2013. 2

[7] Ziyue Feng, Huangying Zhan, Zheng Chen, Qingan Yan, Xiangyu Xu, Changjiang Cai, Bing Li, Qilun Zhu, and Yi Xu. Naruto: Neural Active Reconstruction from Uncertain Target Observations. In *Conference on Computer Vision and Pattern Recognition*, pages 21572–21583, 2024. 2

[8] Antoine Guédon, Tom Monnier, Pascal Monasse, and Vincent Lepetit. Macarons: Mapping and Coverage Anticipation with RGB Online Self-Supervision. In *Conference on Computer Vision and Pattern Recognition*, pages 940–951, 2023. 2

[9] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. In-Hand 3D Object Scanning from an RGB Sequence. In *Conference on Computer Vision and Pattern Recognition*, 2023. 2

[10] Wen Jiang, Boshu Lei, and Kostas Daniilidis. FisherRF: Active View Selection and Mapping with Radiance Fields Using Fisher Information. In *European Conference on Computer Vision*, pages 422–440, 2024. 2, 4

[11] Liren Jin, Xingguang Zhong, Yue Pan, Jens Behley, Cyrill Stachniss, and Marija Popović. Activegs: Active Scene Reconstruction Using Gaussian Splatting. *IEEE Robotics and Automation Letters*, 2025. 2

[12] Yufeng Jin, Vignesh Prasad, Snehal Jauhri, Mathias Franzius, and Georgia Chalvatzaki. 6DOPE-GS: Online 6D Object Pose Estimation Using Gaussian Splatting. In *International Conference on Computer Vision*, 2025. 2

[13] Chunyu Ju, Qinghua Luo, and Xiaozhen Yan. Path planning using an improved a-star algorithm. In *2020 11th international conference on prognostics and system health management (PHM-2020 Jinan)*, pages 23–26. IEEE, 2020. 4

[14] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat Track & Map 3D Gaussians for Dense RGB-D Slam. In *Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 3

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *IEEE Transactions on Robotics and Automation*, 42(4):139–1, 2023. 2

[16] Michael Korn, Martin Holzkothen, and Josef Pauli. Color Supported generalized-ICP. In *International Conference on Computer Vision*, pages 592–599, 2014. 4

[17] Shiyao Li, Antoine Guédon, Clémentin Boittiaux, Shizhe Chen, and Vincent Lepetit. NextBestPath: Efficient 3D Mapping of Unseen Environments. In *International Conference on Learning Representations*, 2025. 2

[18] Hyungtae Lim, Daebeom Kim, Gunhee Shin, Jingnan Shi, Ignacio Vizzo, Hyun Myung, Jaesik Park, and Luca Carlone. Kiss-Matcher: Fast and Robust Point Cloud Registration Revisited. In *International Conference on Robotics and Automation*, pages 11104–11111, 2025. 3

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*, 2020. 2

[20] Richard Pito. A Sensor-Based Solution to the "Next Best View" Problem. In *International Conference on Pattern Recognition*, pages 941–945, 1996. 2

[21] Richard Pito. A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 2002. 2

[22] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv Preprint*, 2023. 2, 5

[23] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 9302–9311. IEEE, 2025. 2

[24] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741, 2004. 2

[25] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-Time 3D Model Acquisition. In *ACM SIGGRAPH*, 2002. 2

[26] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. Neuraldiff: Segmenting 3d objects that move in egocentric videos. In *2021 International Conference on 3D Vision (3DV)*, pages 910–919. IEEE, 2021. 2

[27] Dimitrios Tzionas and Juergen Gall. 3D Object Reconstruction from Hand-Object Interactions. In *International Conference on Computer Vision*, 2015. 2

[28] Pengyuan Wang, Fabian Manhardt, Luca Minciullo, Lorenzo Garattoni, Sven Meie, Nassir Navab, and Benjamin Busam. DemoGrasp: Few-Shot Learning for Robotic Grasping with Human Demonstration. In *International Conference on Intelligent Robots and Systems*, pages 5733–5740, 2021.

[29] Thibaut Weise, Bastian Leibe, and Luc Van Gool. Accurate and Robust Registration for In-Hand Modeling. In *Conference on Computer Vision and Pattern Recognition*, 2008. 2

[30] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 5

[31] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997. 1, 2

[32] Zike Yan, Haoxiang Yang, and Hongbin Zha. Active Neural Mapping. In *International Conference on Computer Vision*, pages 10981–10992, 2023. 2, 5