

Articulated Inter-robot 6D Pose Estimation for Planetary Multi-agent SLAM

Fusing intrinsic constraints with visual refinement
under occlusion

MSc Thesis

Taneshwar Pranav Parankusam

Articulated Inter-robot 6D Pose Estimation for Planetary Multi-agent SLAM

Fusing intrinsic constraints with visual
refinement under occlusion

by

Taneshwar Pranav Parankusam

MSc Thesis research

External Supervisor:	Dr. R. Giubilato
Internal Supervisor:	Prof. Dr. R. Babuška
Thesis Committee:	Dr. C. Pek, Dr. A. Dabiri
Project Duration:	December 1st, 2025 - July 17th, 2026
Faculty:	Faculty of Mechanical Engineering, Delft

Cover:	LRU in the LUNA Hall by DLR under CC BY-NC-ND 3.0 (Modified)
Style:	TU Delft Report Style, with modifications by Taneshwar Pranav Parankusam



Articulated Inter-robot 6D Pose Estimation for Planetary Multi-agent SLAM

Taneshwar Pranav Parankusam

Abstract—Planetary exploration robot teams rely on robust exploration pipelines, such as multi-agent SLAM, to perform complex collaborative tasks. Multi-agent SLAM methods rely on accurate relative transformations between agents to collaboratively merge maps, predominantly using indirect associations between keyframes to compute transforms. Recently, direct association methods that use onboard perception modules with pose estimation pipelines have been shown to provide more accurate, long-range relative transformations. However, current methods consider the robot as a single static rigid body and struggle greatly during dynamic movements, complex robot morphologies, and under harsh environmental occlusions. As a potential solution, we represent the robot as an ensemble of articulated components and investigate the use of articulated pose estimation methods with RGB images and intrinsic measurements. Specifically, we introduce novel post-processing steps based on factor graphs and differentiable rendering to refine upstream poses. First, pose estimates of individual components are visually refined using differentiable rendering and a segmentation mask to ensure visual plausibility. Second, by using a factor-graph formulation, we fuse estimated poses with partially available intrinsic measurements to preserve relative robot constraints and reconstruct missing estimates under occlusions. Third, we extend the factor-graph refinement to estimate over multiple frames and use visual odometry data (VO) to mitigate rotational symmetry issues. Experimental results on synthetic planetary terrains demonstrate that our method outperforms single-body pose estimation with a decrease in translational and rotational error by 55% and 61%, respectively. Additionally, our novel factor-graph refinement method is able to recover undetected articulated components under heavy occlusions and provides robust performance under dynamic movements in indoor lab data. [2]

I. INTRODUCTION

Planetary exploration robots enable new scientific discoveries by traversing extreme environments that are difficult or impossible for humans to explore. A core aspect of this is exploration and mapping, which helps identify risks, secure topological data, and herald human missions [19]. Multi-robot systems expand exploration capabilities by simultaneously exploring different terrains while executing mission objectives [39]. In GNSS-denied environments, such as planetary bodies, modern-day multi-robot systems rely on Simultaneous Localization and Mapping (SLAM) to operate. In a multi-agent setting, collaborative SLAM ties together various local maps constructed by individual robots in a combined global representation of the environment. It provides the foundations of several operations, including planetary terrain reconstruction, autonomous navigation, and multi-agent collaboration [16, 40, 39]. A key requirement for such multi-agent systems is accurate relative pose estimations between agents, which

directly translates to pose-graph constraints for consistent global map fusion. These inter-robot detections are typically obtained through a variety of sensor modalities ranging from visual observations to LiDAR point clouds, allowing robots to detect and localize each other by matching relevant sensor data. However, existing approaches largely rely on rigid-body transformations and indirect feature-based matching, restricting their capacity to capture more complex articulated structures.

Furthermore, planetary environments provide strict constraints for robotic exploration ranging from harsh environmental conditions to limitations on physical robot hardware. Weight and size restrictions drastically limit the onboard sensor suite, including visual sensors, range sensors, scientific equipment, and computational power [40]. As such, inter-robot data exchange is limited to sporadic communication requiring compact representations for global features. Moreover, partial observability due to occlusions or environmental noise such as dust introduces ambiguity in mapping and localization. Additionally, the large distances between planetary bodies confine communication bandwidth while also adding significant delays ranging from seconds to minutes between operational centers and robot teams, requiring high-level autonomy for efficient operation. Together, these constraints define the design of multi-agent SLAM systems.

Early rovers such as the Spirit and Opportunity rovers used stereo cameras for local obstacle avoidance and autonomous navigation. Though revolutionary at the time, they were lacking global localization, path planning, and mapping capabilities due to feasibility constraints [31], key elements required for complex collaborative missions. Iterating on these early systems, modern-day robots such as the Lightweight Rover Unit (LRU) include robust SLAM pipelines to enable high-level task autonomy under strict planetary constraints [40].

Despite advances in multi-agent SLAM for planetary scenarios, the requirement for accurate and robust pose constraints is hard-fixed. Direct inter-robot detection methods such as fiducial markers such as AprilTags [48] provide centimeter-level accurate pose estimation and are used for modern robot teams [39]. However, harsh lighting conditions, slight occlusions, and the limited range reduce practicality for large-scale missions. Alternatively, traditional pose estimation pipelines have shown their applicability in multi-agent scenarios for long-range estimation [37], but are limited by generalizability, harsh occlusions, and complex robot morphologies.

In parallel, articulated pose estimation has become a grow-

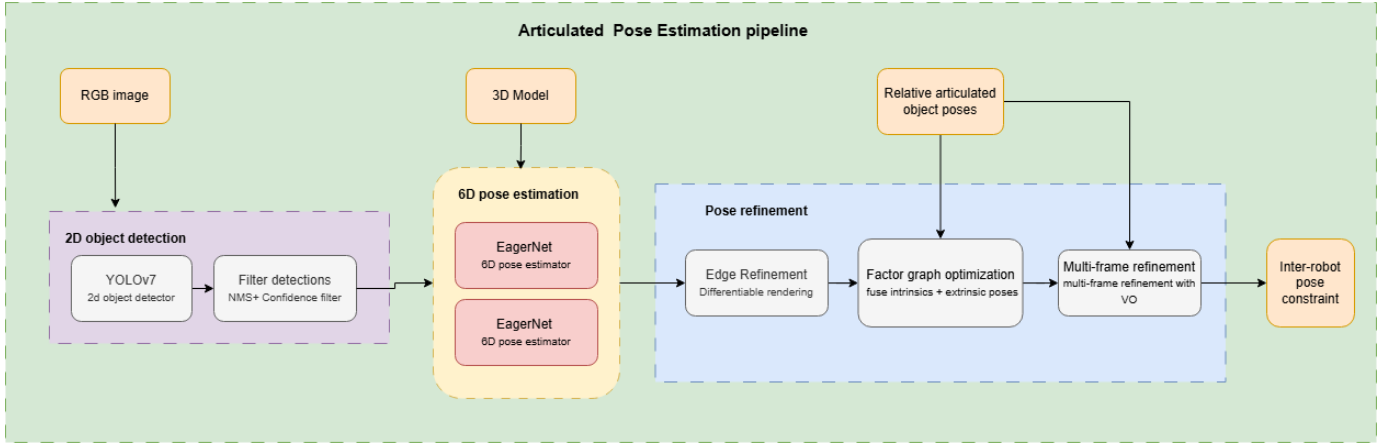


Fig. 1: Our proposed articulated pose estimation pipeline consisting of 3D main stages: 2D Detector, 6D Pose estimators, and Pose refinement. Our novel approach implements edge-based refinement using differential rendering, factor-graph optimization for fusing intrinsic and extrinsic measurements, and multi-frame tracking for occlusions.

ing field with applications in robotics, task automation, and human pose estimation. By considering the object as a system of linked articulated bodies, and fusing external pose estimates, accurate end-effector poses, or individual part poses can be retrieved, even under occlusions. As such, articulated pose estimation may provide the occlusion robustness required for accurate, long-range estimation.

Therefore, in this thesis, we aim to analyze the gap between articulated pose estimation and its applicability towards direct inter-robot 6D associations within the multi-agent SLAM framework. We provide insightful background, dig into key challenges arising from both domains, and explore the following research question: *How can articulated 6D pose estimation be integrated into Multi-agent SLAM to improve Inter-robot associations?* Through rigorous experimentation and verification, we formulate a new method for pose estimation in planetary scenarios and discuss how it compares with existing methods. The key contributions of this work are as follows:

- A novel factor graph pose refinement step for articulated pose estimation. Upstream articulated pose estimates are fused with intrinsic pose constraints from kinematic trees using factor graphs.
- A multi-frame extension of our factor graph refinement method using visual odometry. Visual odometry provides temporal constraints which are fused with per-frame spatial constraints over a fixed frame buffer.
- An edge refinement formulation using differentiable rendering to optimize individual bodies. We expand this further using segmentation masks to filter out target edges for robustness.
- A comprehensive empirical study across 5 refinement methods and 3 datasets. We introduce edge-based and segmentation-guided refinement steps and compare with our baselines using synthetic planetary terrain data and real-world indoor multi-robot lab data.

The remainder of this paper is organized as follows. In

Section II we explore the research gap within Inter-robot detections in SLAM and how articulated pose estimation may provide a solution. In Section III and Section IV we formulate the problem and assumption and provide a clear overview of our system. In Section V we discuss our proposed solution to the problem of articulated pose estimation for direct inter-robot detections. We highlight our experimental methodology in Section VI and state our results in Section VII. In Section VIII, we evaluate and discuss the data. Our concluding remarks on the topic are presented in Section IX.

II. RELATED WORK

A. Inter-robot detections

Loop closures are central to ensuring that accumulated drift is corrected to ensure consistency within the global map. In a multi-agent setting, this is achieved through either indirect associations, which rely on shared features and similarity metrics, or direct associations, which estimate rigid-body transformations from onboard perception.

Indirect methods use place recognition algorithms to perform loop closure detection using shared features among agents or a centralized server. Classical methods use Bag-of-words (BoW) techniques to extract handcrafted image features such as ORB [36] or BRIEF [5] to generate BoW vectors for each keyframe. Such vector descriptors are shared among agents and compared for similarity using metrics such as L1 Score and Cosine similarity. Centralized methods, including CCM-SLAM [38] and CVI-SLAM [23], select candidate frames using BoW search and further refine matches downstream using geometric verification methods such as RANSAC. Recent approaches have highlighted the potential of learned global descriptors, such as NetVLAD [2], which are robust against viewpoint and illumination changes. Distributed approaches including Kimera-Multi [6] and DOOR-SLAM [24] combine descriptor matching with Pairwise Consistency Maximization (PCM) for outlier rejections and are suitable

for restricted communication ranges. LiDAR-based approaches follow similar association pipelines to their visual counterparts, replacing image descriptors with compact spatial descriptors and use lightweight coarse descriptor search followed by fine-grained scan-matching for accurate transformations, as demonstrated in LAMP2.0 [7] and DiSCo-SLAM [18]. Indirect methods, though robust to viewpoint changes, large motions, and varying lighting conditions, suffer from severe communication bottlenecks requiring compact representations or multi-stage pipelines. Moreover, these methods struggle especially under low-texture environments such as planetary terrain [4].

Direct methods focus on explicitly measuring inter-robot rigid body transformations with perception stacks rather than place recognition algorithms. A popular class of robot pose estimation algorithms uses fiducial markers, such as AprilTags [33, 48] and Shen et al. method [41], which enable highly accurate 6D pose estimation in close proximity but struggle with planar rotations and require physical changes to robot structure. 3D April tags [44] overcome viewpoint limitations by using 6-faced cubes, allowing robust performance for underwater tasks; however, they face the same range limitation as distance is cubically proportional to the physical marker size. Active signal-based methods, including UVDAR and CREPES [34, 51], emit unique identifiable signals using UV or infrared LEDs to encode IDs and enable relative pose estimation. Though active sensing allows for much greater ranges [51], it requires specialized hardware, which for constrained robotic solutions such as rovers or small-scale drones may not be suitable. Recent works use deep learning techniques to estimate poses via bounding boxes or keypoints and are tracked over time [35, 20, 42]. To overcome this limitation, Rueggeberg et al. [37] proposed a long-range markerless approach that generalizes well to arbitrary robot bodies. However, as with other methods, single-body pose estimation struggles with occlusions, varying articulated configurations of a robot, and harsh environmental conditions. As such, we delve deeper into articulated pose estimation methods as a potential solution to these limitations.

B. Articulated pose estimation

Articulated pose estimation can be broadly categorized into three main families: robotic pose estimation, category-level pose estimation, and human pose estimation. In robotic pose estimation, the kinematic configuration is fused with external pose estimation methods to retrieve poses of the end-effector or each articulated component. A popular technique used by both DREAM [25] and PoseFusion [14], is to use belief maps of 2D sparse keypoints and fuse them with 3D keypoint estimates from forward kinematics and camera intrinsics before passing them to a PnP solver. RoboPEPP [13] uses a self-supervised pre-training strategy to learn the inherent physical structure by masking specific joints during pre-training. The pre-trained encoder and predictor networks are fine-tuned with subsequent 2D keypoint and joint angle networks to build sparse 2D-3D correspondences, which are refined with PnP.

While these methods have been shown to be more robust to occlusions than pure feature-based methods, they remain sensitive to symmetry ambiguity, missing keypoints, and struggle with the many pitfalls of traditional keypoint-based methods.

Category-level articulated object pose estimation (CAPE) methods predict poses for interconnected unseen parts which belong to general categories rather than specific instances, such as 3D CAD models of the object. Li et al. [27] tackle this issue by introducing the Articulation-aware Normalized Coordinate Space Hierarchy (ANCSH) which represents articulated objects in a two-stage canonical space consisting of object-level spaces with joint parameters and global information, and part spaces with normalized coordinate spaces (NPCS). By doing so, they are able to predict articulated object instances within known categories from pure depth images. Huang et al. [17] use a single-stage unified network for real-time performance to directly predict object centroids, semantic labels, and NPCS from point cloud data. Liu et al. [30] introduce the CAPER framework, which expands CAPE methods to multiple instances with unknown kinematic structures and unpredictable joint configurations from a single RGB-D image. Category-level methods are strong against variations in object fidelity due to their ability to generalize; however, they struggle under harsh lighting conditions, bad initial detections, and are still currently limited to 1D revolute and prismatic joints.

Although limited work has tackled articulated robot pose estimation, there exists a significant body of literature for Human Pose Estimation (HPE). By nature, HPE is premised on articulated structures, representing the human body as a 3D, rather than 6D, skeletal structure consisting of a kinematic graph of joints and segments. Graph and transformer-based methods model joint dependencies via learned relations and assume that 2D HPE keypoints are given. Approaches, such as PoseGTAC [52] and DGFormer [8], capture multi-scale spatial and global relationships while encoding global physical constraints and interactions between joints. Alternatively, Pose-oriented Transformers [26] explicitly encode skeletal topology into their attention mechanisms. Other methods utilize deep grammar networks [50, 12] or holistic triangulation [46] to encode kinematics, symmetry, and motor coordination of joints using bidirectional RNNs for prediction of single-view or multi-view poses. In contrast with other articulated methods, HPE-based methods effectively use geometrical constraints within their learned representations; however, they struggle with self-occlusions, depth ambiguity, and temporal inconsistency in monocular settings.

III. PROBLEM FORMULATION

Within multi-agent SLAM, inter-robot detections enable the fusion of local maps to generate a consistent global map. As such, direct association methods provide more accurate transformations between robots than place recognition methods due to their robustness in low-texture environments, but struggle with range and occlusions. Since map fusion requires precise position and orientation of a detected robot, pose estimation

is a natural extension of direct methods that can address some of these limitations.

The pose estimation problem for a single body is formally defined as follows. Given an RGB image, defined as $I \in \mathbb{R}^{H \times W \times 3}$ where H, W are height and width, respectively, we aim to predict the 6D pose of an object relative to the observing robot's camera frame. We use the homogeneous transformation matrix representation of the 6D pose throughout most of our methods as defined in equation 1. Additionally, we assume a calibrated camera-to-body transformation T_{body}^{cam} is available, which we use to generate the SLAM pose constraint T_{body}^{object} .

$$T_{cam}^{object} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3), \quad (1)$$

$$\mathbf{R} \in SO(3), \\ \mathbf{t} \in \mathbb{R}^3$$

$$T_{body}^{object} = T_{body}^{cam} \cdot T_{cam}^{object} \quad (2)$$

The single-body assumption does not consider the complex morphologies of robots as they move and interact with the environment. Rueggeberg et al. [37] attempted to use single-body pose estimation algorithms with RGB cameras and demonstrated their effectiveness in increasing detection range, but struggled immensely with rotational symmetry issues and detection errors under occlusion. This can be attributed to their assumption that the observed robot remains in a fixed static configuration as a single rigid body. Articulated methods relax the single-body assumption, modeling the robot as a set of interconnected rigid bodies rather than a fixed geometry. This allows them to capture diverse robot configurations and remain robust to occlusions. We formalize this by extending the single-body definition to a graph of interlinked poses. We define the set of body-to-object transformations between the *observing* robot and each *observed* articulated component as follows:

$$\hat{\mathcal{T}} = \{T_{body}^i\}_{i=1}^N \quad (3)$$

We additionally assume that the relative poses between connected components are available. Specifically, we denote the articulated body as a rooted kinematic tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node represents an articulated component and each directed edge $(a, b) \in \mathcal{E}$ represents a transformation $T_a^b \in SE(3)$ between components. The pose of any component relative to another by a composition of transformations can be obtained by a unique path in the tree, denoted by an ordered sequence $P(a, b) = \{(u, v) | (u, v) \in \mathcal{E} \text{ and } u, v \in \mathcal{V}\}$, as follows:

$$T_a^b = \prod_{(u, v) \in P(a, b)} T_u^v \quad (4)$$

The SLAM pose constraint between the observed articulated system and the observing robot is instead formulated as a transformation to the root of the kinematic tree.

IV. SYSTEM OVERVIEW

Within this section, we cover the core aspects of our system, including hardware components in subsection IV-A, and general software architecture in subsection IV-B.

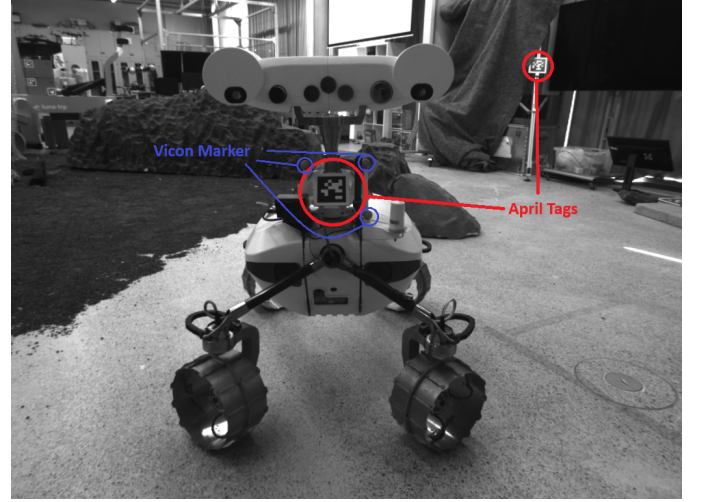


Fig. 2: Lightweight Rover Unit (LRU) in indoor lab with AprilTags and VICON marker system.

A. Hardware

Testing a multi-robot system accurately requires a range of hardware components from compute to physical robots and ground-truth measurement systems. Below we describe the hardware used along with some technical specifications of our setup.

- *Lightweight Rover Units (LRU)*: Developed by the Institute of Robotics and Mechatronics at DLR, these rovers are built for researching autonomous planetary exploration systems. There exist two variants of the LRU, LRU1 and LRU2, with the former containing additional cameras as part of its head segment. Each unit is equipped with an RGB camera and stereo cameras (for depth). Within our experiments, we use both units to record lab data.
- *VICON Motion capture*: Providing ground-truth data is crucial for assessing the quality of motion planning and pose estimation pipelines. To this end, we use an existing VICON motion capture system within the Indoor Lab which accurately records the position of VICON markers onboard the LRUs. The system requires careful calibration in order to maintain accuracy during maneuvers.
- *April Tags*: In addition to the VICON motion capture system, we use April Tags [48] for calibration. Unlike VICON markers, which only detect position, AprilTags provide accurate orientation information of the marker as well. Each LRU is equipped with several AprilTags

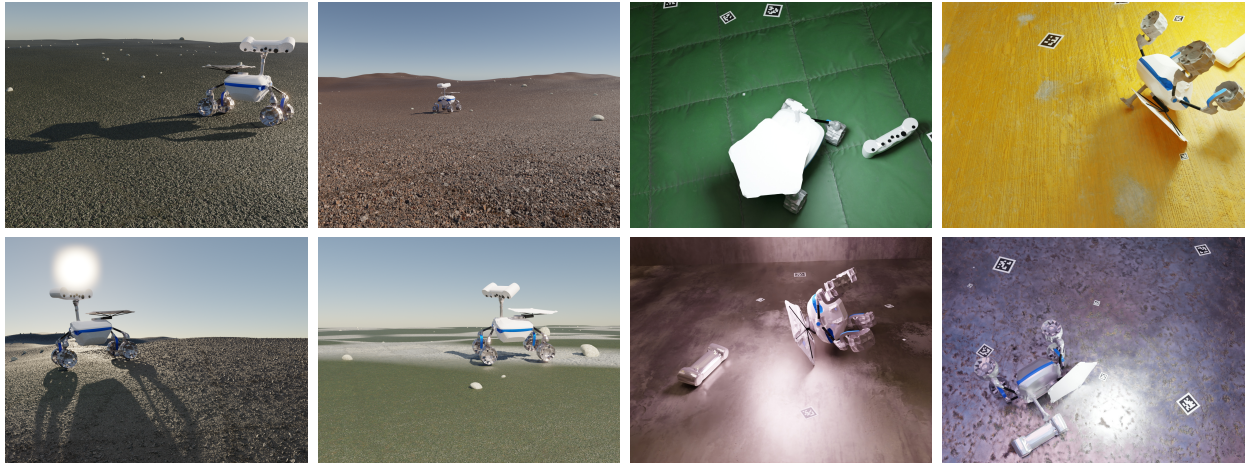


Fig. 3: Generated training data of Lightweight Rover Unit (LRU). **Left:** OASYS images with unstructured planetary terrain and joint constraints. **Right:** BlenderProc2 data with different textures and backgrounds, and no joint constraints.

placed at a variety of points across the robot body, ensuring visibility of at least one AprilTag from any perspective.

- *Workstations:* The lightweight rover units require ground station computers to maintain inter-robot communications as they contain limited compute. As such, we use a high-spec workstation consisting of an Intel Core i7 processor, 32 GB of DDR4 RAM, and an NVIDIA RTX 3080 to run all applications and our pipeline simultaneously.
- *Server clusters:* In addition to the workstations, model training and dataset generation use in-house server clusters using the SLURM workload manager. As thousands of high-quality planetary terrain images are required, the cluster makes this approach more feasible.

B. Software

Our system software consists of three main components: ROS2, our middleware for enabling interactions between different applications; our own pipeline, which is a ROS2 node; and other subsystems which are also ROS2 nodes which provide the greater Multi-robot SLAM pipeline. We detail the components below.

- *ROS 2:* Robot Operating System 2 (ROS 2), is an industry-standard framework used for developing robot applications. Both the LRU's and Vicon marker system utilize ROS2 for all operations, including communication, motion planning, SLAM and more. As such, each application is processed as a ROS 2 node, including our pose estimation pipeline.
- *External ROS2 nodes:* Within the system, several dozen ROS2 nodes are required for the LRU's to be operational, and were developed in-house. The main components our pipeline interacted with are the multi-robot SLAM component, RGB Image frames from the camera component, and finally the rigid-body transformation trees from the TF frames component.

- *Pose estimation pipeline:* Our pose estimation pipeline is primarily written in Python to take advantage of machine learning libraries such as PyTorch and PyTorch3D. We developed an offline pure Python version and an online ROS2 version, consisting of the same core components, to validate across a variety of datasets (see Section VI).

V. PROPOSED SOLUTION

Within this section, we highlight the different components of our proposed solution. The training procedure and data gathering are described in Section V-A. We describe the upstream pose estimation pipeline in Sections V-B and V-C. We detail the filtering required to remove noisy estimates in Section V-D. In Section V-E we describe our novel factor-graph based pose estimation approach including single and multi-frame methods. We extend our estimation pipeline with edge-based refinement methods in Section V-G and segmentation-guided refinement in Section V-G2.

A. Training data

Learning-based pose estimation tasks for specific robot configurations require custom datasets for training. Moreover, considering the context of planetary robotics, building datasets from real-world examples becomes infeasible. As such, we opted to use simulations, using approximate 3D models of the LRU in order to generate training data.

1) *OASYS:* OASYS [32] is an open-source unstructured outdoor environment simulator specifically designed for planetary terrain. The simulator was modified to enable 3D models with multiple bodies and joint constraints. By providing a .blend file with the contained 3D model and a .yaml configuration file containing specified joint limits, each sample gets a randomized individual pose for each articulated body under random environmental configurations. Additionally, for every sample, we were able to generate individual component masks, visibility masks, bounding boxes, and exact pose coordinates for each articulated component. The output dataset

was converted to the BOP format [43] to later train learned components of the pipeline.

2) **BlenderProc**: Along with OAISYS, we used BlenderProc2 [11] for data generation to add robustness in close-range detections. Onboard the LRU, there are several April tags, which were not present within the 3D model and significantly impact detection rate. As such, they were used as distractor objects during training to increase robustness. Similar to OAISYS, BlenderProc2 provides a variety of environmental conditions such as lighting and various backgrounds; however, it does not support articulated bodies. Instead, each robot component is treated as its own separate object, helping reduce the correlation effects during pose estimation, essential for improving performance under occlusions.

B. 2D Bounding Box detection

To facilitate our downstream pose estimator, we require a 2D bounding box detector to locate the object in image space. This drastically reduces the search space for 6D pose estimators and is typically integrated as part of the backbone [49, 3] or outsourced to upstream detectors [37]. For our task, we use YOLOv7 [47], an industry-standard object detector which we train using an in-house pipeline that augments our training set for increased robustness. The low computational cost, ease of training, and robustness made it an attractive option for our methods. As with the training data, each articulated component is treated as a separate object by the detector.

C. Robust 6D Pose estimator

The poses are detected using EagerNet [45], a network specifically designed for orbital scenarios with harsh lighting conditions and partial visibility. Additionally, EagerNet is capable of achieving state-of-the-art performance in orbital scenarios with approximate 3D models of the objects, a capability that is particularly valuable for our work, given that only approximate 3D models of the LRU are available. It achieves this robustness by predicting dense 2D-3D correspondences between pixels and 3D object points and uses several techniques, including foreground confidences, coordinate errors, and surface region class labels, to produce multiple hypotheses. These are iteratively refined, from which the most probable pose estimate is selected as the model's output.

Though accurate, EagerNet is capable of only predicting single object classes. For our articulated configurations, an individual EagerNet network is trained to predict a single component, requiring N networks for N Rigid body components. As such, during inference time, we use an ensemble of networks to make our pose estimates. In our experiments, we consider a two-body scenario consisting of the LRU head and the LRU body, as other articulated components have no ground-truth transformations for comparison in our robot configuration. A natural drawback of this approach is that constraints and relations between components become difficult to learn, as our estimators assume no knowledge of other robot

components. As such, in subsection V-E we introduce a potential solution to this approach using live intrinsic measurements to provide kinematic constraints.

D. Filtering

Iterative SLAM pipelines rely heavily on accurate constraints, as once the graph has been optimized, it cannot be undone. Therefore, false inter-robot detections that create inaccurate SLAM constraints must be avoided at all costs, and care must be taken in the pose estimation pipeline to ensure accuracy. As such, raw detections and pose estimations require filtering before being published. In this subsection, we highlight the core filters used by our system.

- **Confidence filter**: YOLOv7 detections provide confidence scores at inference time, which help identify which boxes the model thinks are best. These scores aid Non-Maximum Suppression (NMS) which filters out redundant bounding boxes which contain significant overlaps. The remaining detections are fed into a confidence filter which removes those with scores below a confidence threshold. Only best detection is assigned to the object since we assume there is only one object per class.
- **Border-contact filter**: From our experimentation, objects which are too close drastically impact pose estimation performance due to occlusions. Using a border-contact filter, we can detect if an object is too close by observing how many bounding box borders contact the image borders. We found that objects with 2 or more detection borders intersecting the image boundary are typically too close to get any meaningful estimates.
- **Distance filter**: Pose estimates that are beyond a certain distance have larger variance and translational errors as observed in our experiments. For practical scenarios, we limit the detection range to a maximum number of meters to reduce noise and false detections.
- **Pose change filter**: SLAM algorithms append a new set of parameters and constraints for every detected landmark, leading to a proportional increase in computational cost. Additionally, pose constraints published at high frequency can slow down graph optimization while providing little information. Therefore, we introduce a pose-change filter that evaluates whether new poses have changed in comparison to the previous pose and whether enough time has passed.

E. Factor Graph Pose refinement

6D Pose estimations from upstream methods provide extrinsic measurements of articulated components without assuming any kinematic links between them. As defined in Section III, we assume access to the observed articulated system's kinematic tree, giving us the relative poses between components. The available relative poses from the intrinsics can be fused with global pose estimates from the extrinsic measurements to further refine estimates, providing more accurate predictions under occlusions and harsh lighting.

As such, we treat the pose refinement problem as a fusion problem, applying similar strategies to graph-based SLAM methods [21, 22]. We formulate this as a Maximum A Posteriori (MAP) estimation problem of the set of unknown true camera-to-object poses of each component $\mathcal{X} = \{X_i \in SE(3) \mid i \in \mathcal{V}\}$ given a set of measurements for the camera-to-object poses from our estimator $\mathcal{Z} = \{Z_i \in SE(3) \mid i \in \mathcal{V}\}$ as follows:

$$\mathcal{X}^{MAP} = \arg \max_{\mathcal{X}} P(\mathcal{X} \mid \mathcal{Z}) \quad (5)$$

To find the probability distribution $P(\mathcal{X} \mid \mathcal{Z})$ and solve the MAP problem, we use factor graphs. A factor graph is a bipartite graph that factorizes a probability distribution into independent factors and variables. For our problem, we define the factor graph as follows:

$$\mathcal{F} = (\mathcal{X}, \mathcal{C}, \mathcal{L}) \quad (6)$$

where $\mathcal{C}$ is a set of all factors and $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{C}$ which links variables to their corresponding factors. Unary factors $\phi_i(X_i) \in \mathcal{C}$ represent the likelihood of the measurement $\phi_i(X_i) = P(Z_i \mid X_i)$ given the unknown pose, while binary factors $\phi_{i,j}(X_i, X_j) \in \mathcal{C}$ for $(i, j) \in \mathcal{V}$ represent the likelihood $P(Z_{ij} \mid X_i, X_j)$ of the relative transformations. This allows us to present the distribution $P(\mathcal{X} \mid \mathcal{Z})$ as a product of unary and binary factors as follows:

$$P(\mathcal{X} \mid \mathcal{Z}) = \prod_{i \in \mathcal{V}} \phi_i(X_i) \prod_{i,j \in \mathcal{V}} \phi(X_i, X_j) \quad (7)$$

Additionally, we also assume that the measurement noise is Gaussian in nature, giving us:

$$\phi_i(X_i) \propto \exp(-r_i^T \Sigma_i^{-1} r_i) \quad (8)$$

$$\phi_{i,j}(X_i, X_j) \propto \exp(-r_{ij}^T \Sigma_{ij}^{-1} r_{ij}) \quad (9)$$

where r_i and r_{ij} represent the residuals or measurement error, and the corresponding covariances. By substituting equations 8, 7 into Equation 5, we get a non-linear least squares optimization problem.

$$\mathcal{X}^{MAP} = \arg \min_{\mathcal{X}} \left(\sum_{i \in \mathcal{V}} \|r_i^T \Sigma_i^{-1} r_i\| + \sum_{i,j \in \mathcal{V}} \|r_{ij}^T \Sigma_{ij}^{-1} r_{ij}\| \right) \quad (10)$$

To perform this optimization, we use the GTSAM Library [9] which provides simplified functions and interfaces for factor graphs. Furthermore, it readily integrates robust cost functions, such as the Huber loss, which we use in our work, preventing outliers from dominating the optimization and does so by assuming residuals are non-Gaussian.

For our system, we have two articulated components, the LRU head and LRU body, along with the relative transformations between them. We list our factors in Equation 11 and provide a corresponding graphical overview in Figure 4. Each measurement $Z_i \in \mathcal{Z}$ is accompanied by a covariance matrix

for the measurement noise, which can be tuned depending on the system.

$$\begin{aligned} P(\mathcal{X} \mid \mathcal{Z}) &= \phi_H(X_H) \times \phi_B(X_B) \times \phi_{BH}(X_B, X_H) \quad (11) \\ \text{s.t. } \mathcal{X} &= \{X_H, X_B\} \\ \mathcal{Z} &= \{Z_H, Z_B, Z_{BH}\} \end{aligned}$$

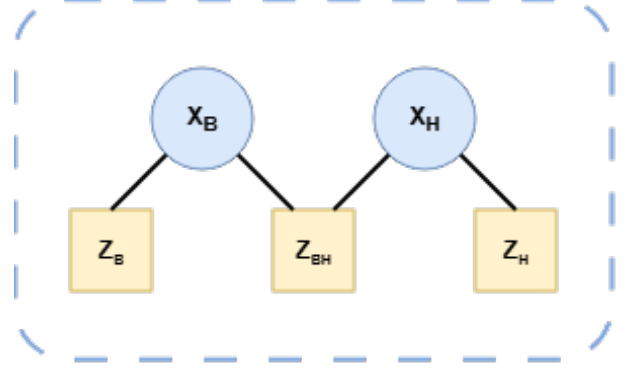


Fig. 4: Factor-graph formulation. Z_B, Z_H, Z_{BH} represent extrinsic body measurement, head measurement, and intrinsic body-to-head measurement respectively. X_B, X_H are camera-to-object poses that are optimized

Factor graph pose refinement assumes a unimodal distribution for measurement noise. However, as upstream pose estimates from EagerNet can be rotationally ambiguous, the noise distribution remains bimodal, leading to erratic pose estimates that diverge from the ground-truth. To this end, we introduce a simple flip filter, which uses the relative transformation between two components to assess whether one component should be rotated by 180° around a particular axis. In our case, we noticed symmetry issues around the z-axis and formulate the filter as follows:

$$T_{cam}^B = \begin{cases} T_{cam}^B & \text{if } |\psi| < r_{thresh} \\ \begin{bmatrix} R_z(180^\circ) & 0 \\ 0 & 1 \end{bmatrix} \cdot T_{cam}^B & \text{else } |\psi| \geq r_{thresh} \end{cases} \quad (12)$$

where r_{thresh} is a threshold for the yaw angle ψ between the LRU head and the LRU body and $R_z(180^\circ)$ is a z-axis rotational matrix for a 180° rotation. Though it may not resolve all rotational symmetry cases, it helps reduce the problem to a unimodal distribution, allowing for stability and consistency during factor graph refinement.

F. Multi-frame pose refinement

Factor-graph pose refinement optimizes the articulated component poses only for a single frame, and as stated earlier, can potentially suffer from rotational symmetry issues. A potential solution would be to consider extending the framework to include multiple frames, helping spot inconsistencies across the time domain. As such, we extend the factor graph pose refinement framework to include multi-frame tracking, fusing

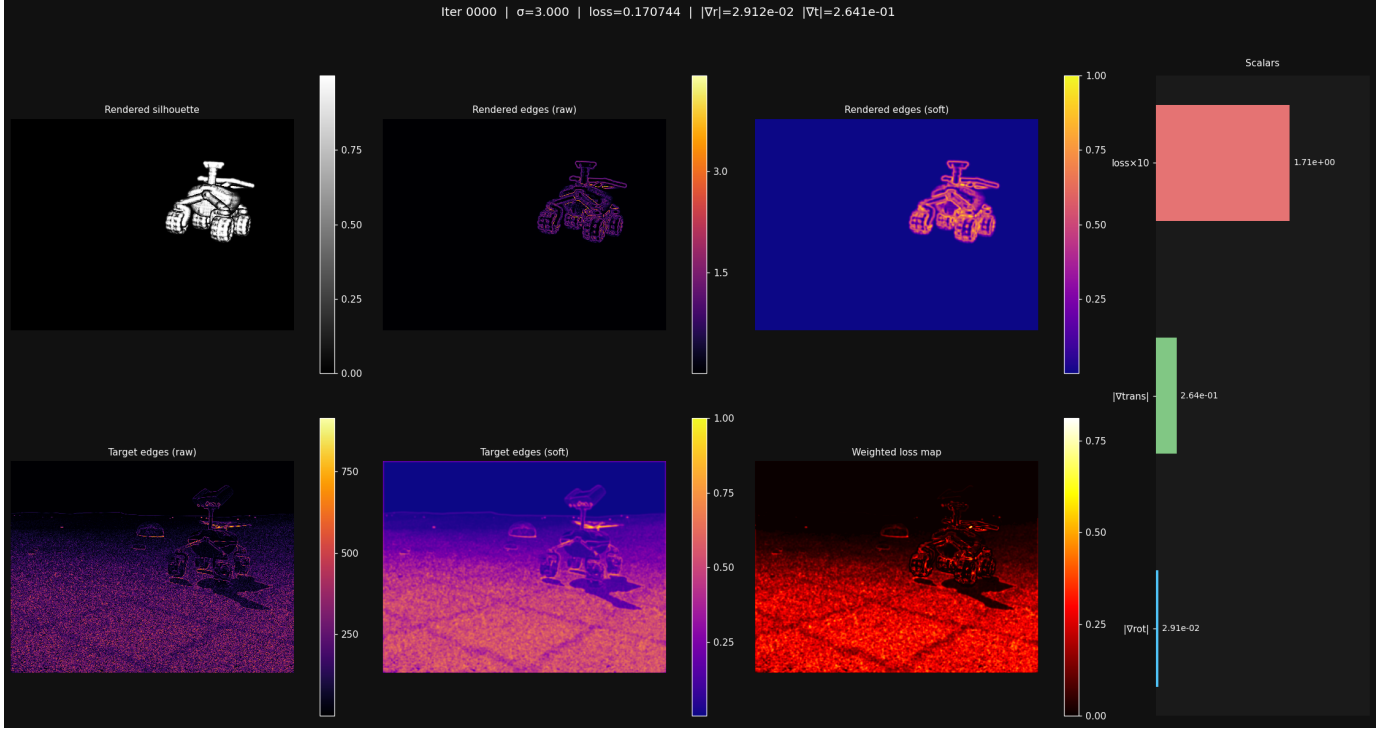


Fig. 5: Visualization of the edge extraction and refinement process. The top row shows the rendered silhouette, extracted rendered edges, and the softening. The bottom row shows extracted target edges, softening, and the contribution of each pixel to the loss.

additional measurements from Visual Odometry (VO) to get the motion between frames. Poses $Z_H = \hat{T}_{cam}^H$ and $Z_B = \hat{T}_{cam}^B$ along with a timestamp and relative transformation measurement $Z_{BO} = \hat{T}_{odom}^B$ between the odometry reference frame and the root frame of the articulated body (in this case Z_B), are stored within a buffer of size k . The buffer acts as a sliding window, only considering a set number of frames, rather than all previous time frames, to avoid the buildup of drift. In addition to considering only a set number of frames, we also implement a time filter to prevent any odometry drift from building up from stale detections. If the detections between detections is too long, then the previous detections are discarded, leaving only the latest measurements that are within the set boundaries. Therefore, we only perform multi-frame refinement once the buffer is full.

$$\begin{aligned}
 P(\mathcal{X} \mid \mathcal{Z}) &= \prod_{i=0}^k \phi_H(X_{H_i}) \times \phi_B(X_{B_i}) \times \phi_{BH}(X_{B_i}, X_{H_i}) \\
 &\quad \times \prod_{i=1}^k \phi_{BB}(X_{B_{i-1}}, X_{B_i}) \\
 \text{s.t. } \mathcal{X} &= \{X_{H_0}, \dots, X_{H_k}, X_{B_0}, \dots, X_{B_k}\} \\
 \mathcal{Z} &= \{Z_{H_0}, \dots, Z_{H_k}, Z_{B_0}, \dots, Z_{B_k}, \\
 &\quad Z_{BH_0}, \dots, Z_{BH_k}, Z_{B_i B_j} \forall i < j\}
 \end{aligned} \tag{13}$$

The tracking algorithm formally extends our factor graph

as shown in Equation 13. Each variable and measurement belonging to sets $\mathcal{X}$, $\mathcal{Z}$ is additionally indexed by time. Along with that, we introduce new temporal factors to indicate measured transformations between the root component of the kinematic chain across time.

G. Edge-based differential refinement

Images contain rich geometrical details that help identify differing sections of the robot. The simplest features of which are edges, which are easy to extract and preserve robot geometry. An intuitive refinement technique would be to align the edges of the observed image with the edges of the render of our estimated pose. However, traditional rendering processes lose 3D information when projecting points to the image plane, making pose optimization infeasible. Differential rendering approaches [29, 28] use soft rasterization to create smooth continuous gradients that enable optimization. Taking inspiration from Liang et al. [29], we optimize for edge overlap rather than segmentation mask overlap, as we suspect edges to contain richer information. We developed two methods: a purely geometrical approach for edge-based pose refinement, and a segmentation-guided approach which uses a learned mask to keep relevant target edges.

1) Edge-based pose refinement: Upstream poses are typically represented as homogeneous matrices $T \in \mathbb{R}^{4 \times 4}$ that consist of both a translational vector $t \in \mathbb{R}^{3 \times 1}$ and a rotation matrix $R \in \mathbb{R}^{3 \times 3}$. The rotation matrix is converted to the axis-angle representation to reduce the number of free

variables during optimization and guarantee orthonormality. The transformed pose is rendered using a soft rasterizer, which uses blurring to produce soft differentiable boundaries around the mesh, and a soft silhouette shader. Moreover, we place a point light at the camera origin, with additional ambient light, to sufficiently illuminate the scene.

Both the target image and the rendered image have their edges extracted using a differentiable Sobel edge detector. The Sobel edge detector emphasizes high spatial frequency regions that correspond to edges using two $\mathbb{Z}^{3 \times 3}$ convolutions Gx and Gy as shown below:

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, Gy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (14)$$

These kernels are convolved around each image, giving us the horizontal and vertical gradients at each pixel. The magnitude of these gradients is calculated per pixel, as the norm of the vertical and horizontal gradient components, as shown in equation 15, where i, j define pixel coordinates. It must be noted that values are clamped before the square root to avoid infinite gradients at 0.

$$\|G_{i,j}\| = \sqrt{\text{clamp}(Gx_{i,j}^2, 0, 1) + \text{clamp}(Gy_{i,j}^2, 0, 1)} \quad (15)$$

Unlike rendered images, target images contain heavy noise within their edge map caused by background noise, unstructured terrain, and objects within the scene. As such, we generate an influence mask, which decides the relevance of an edge pixel towards the final loss. To filter out weak and noisy edges, we apply a soft threshold using the sigmoid function, where E defines the edge at i, j , E_{th} is the threshold, and γ is a scaling factor:

$$P(i, j) = \sigma_{sig}(\gamma \cdot (E_{i,j} - E_{th})) \quad (16)$$

To further smooth out edges and reduce high-frequency noise, a circular isotropic Gaussian kernel is applied, which is defined in equation 17. As such, neighboring pixels of an edge also contribute towards the loss, creating a smooth circular region of influence, generating smoother gradient flows. In this case $(u, v) \in \mathbb{Z}^2$ describes the kernel coordinates, and σ_{size} is an odd kernel size.

$$K_{\sigma}(u, v) = \frac{\exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)}{\sum_{u,v} \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)} \quad (17)$$

$$(u, v) \in \left[-\frac{k}{2}, \frac{k}{2}\right], \text{ where } k = 2\sigma_{size} + 1 \quad (18)$$

With the edges cleanly extracted at each data point, we inflate target edges using a Gaussian blob kernel to soften the edges, similar to equation 17. A kernel with an odd size is initialized with a Gaussian distribution at its center. The standard deviation σ_k of the distribution specifies the radius of

influence, creating a softening circular region around the edge. Additionally, by normalizing and thresholding the values, we prevent vanishing and exploding gradients. We drive edges closer together during optimization by gradually annealing σ_k and reducing the softening to introduce more distinct geometric features.

Our goal was to select a loss function that contained strong convergence properties and down-weighted extreme outliers. As such, we chose the Mean Squared Error (MSE) loss between the target edge image defined in equation 19. The influence mask's weight $w_{i,j}$ is applied to the pixel error between the rendered edge $E_{i,j}$ and the target edge $T_{i,j}$.

$$\mathcal{L} = \frac{\sum_i \sum_j w_{i,j} \cdot (E_{i,j} - T_{i,j})^2}{\sum_i \sum_j w_{i,j}} \quad (19)$$

The calculated loss is back-propagated to optimize the initial pose, and the process is repeated for N iterations. We use the Adam optimizer, with a constant learning rate parameter specified individually for the translational component and the rotational component. We do not add learning rate decay, as the number of iterations N is small and the annealing of σ_k during the softening step ensures convergence.

2) **Segmentation-guided pose refinement:** Target filtering purely through geometrical approaches isn't enough to distinguish relevant robot component edges from background objects or terrain noise, especially in cluttered environments. As such, we used a learned approach to segment the image, extract an instance object mask, and select only edges within the mask, generating an accurate target image.

Instance segmentation is the process of identifying individual objects within an image and providing unique segmentation masks. We use the YOLO family of models, specifically YOLOv26, and train it on the same dataset as our upstream pose estimators. Though globally accurate, binary segmentation masks produced by learned methods struggle with finely matching the contours of an object. As such, it becomes important to inflate the segmented mask to prevent the relevant target edges from being masked out.

$$M_{i,j} = \max_{-k/2 < u < k/2 - k/2 < v < k/2} (p_{u,v}) \quad (20)$$

We use a form of morphological dilation which inflates the binary segmentation mask using max-pooling kernels. As each pixel is either 0 or 1, any pixel near the masked region is included within the newly generated mask, effectively inflating the original masked region. We use the kernel specified in equation 20 where $k \in \mathbb{Z}$ is the kernel size and $p_{u,v} \in \mathbb{R}$ is the binary pixel value. After the inflation process, the mask is applied directly to the target edges after Sobel edge extraction and before softening.

VI. EXPERIMENTAL PROCEDURE

A. Datasets

As part of our experimental procedure, we use a diverse range of data to evaluate our novel method. This includes

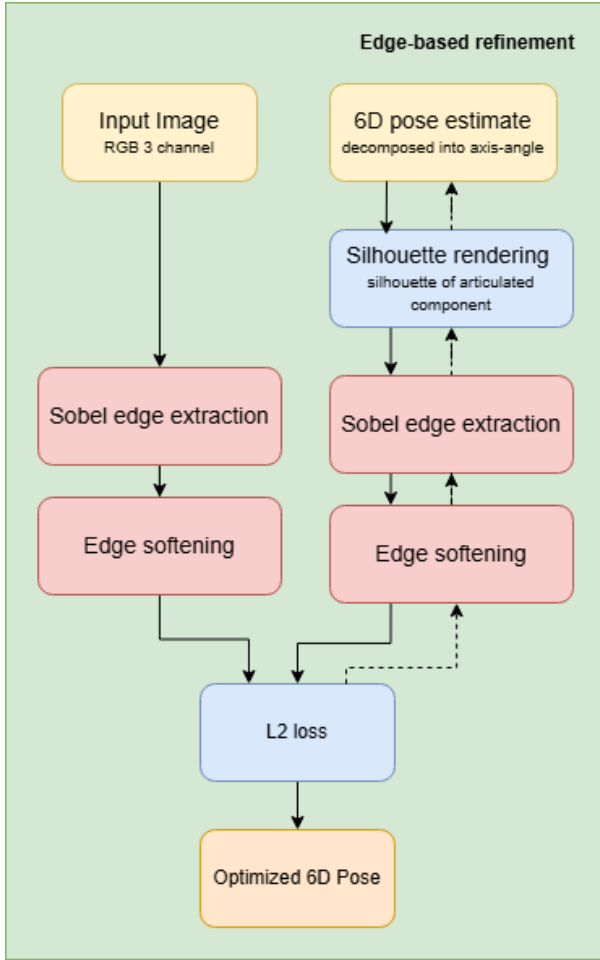


Fig. 6: Edge-based refinement pipeline which takes as input a 6D pose estimate, 3D model, and original image. By comparing the edges via an L2 loss, the gradients are backpropagated to optimize the 6D pose.

variations of synthetic data and indoor lab data for testing 6D pose estimation performance under occlusions, planetary terrain, and noisy environments.

- **Synthetic data:** Our synthetic dataset is built from OAISYS, much in the same way that our training data is generated, and consists of 800 samples. As such, each sample is an I.I.D random variable, which allows for good single-shot evaluation, but cannot be used to assess tracking-based pose refinement. The dataset provides segmentation and visibility masks, giving the exact line-of-sight visibility percentage (%), enabling the comparison of model performance under varying levels of object visibility.
- **Indoor lab data (Live):** To evaluate under real-world conditions, we collected live data of DLR’s LRU systems navigating within the Mobile Robotics Laboratory, recording a total of 684 samples. Ground-truth data is collected using the indoor VICON motion capture system, with markers being placed on key robot sections.

The motion capture system is calibrated and publishes unified ROS2 TF Frames, which are unified with internal sensor data under a single TF Tree. This gives us access to both ground-truth and live odometry data. Unlike our synthetic data, our indoor data is recorded as ROS2 bag files, allowing for sequential image data and enabling us to evaluate our tracking-based method.

- **Indoor lab data (Occlusions):** To mimic occlusion scenarios found within our synthetic data, an additional dataset was recorded with obstacles that limit line-of-sight visibility of the LRU, consisting of 950 samples. Additionally, the dataset introduced camera movement of the observing robot, as previous datasets only had static camera positions.

B. metrics

To assess the performance of the model, we used common metrics found in the literature for the task of pose estimation, as described below:

- **Translational error:** The translational error is defined as the Euclidean distance between the ground-truth (x, y, z) and the estimated poses $(\hat{x}, \hat{y}, \hat{z})$. Additionally, we also compute the translational error per individual axis, especially to observe the model’s localization capabilities in the depth domain. It is computed as follows:

$$\|\mathbf{t}_{gt} - \hat{\mathbf{t}}_{est}\|_2 \quad (21)$$

- **Rotational error:** The rotational error is defined as the geodesic distance, which is the shortest path between two orientations in on a rotational manifold $SO(3)$. I.e., it is the smallest rotation required to go from the estimated orientation to the ground truth. Specifically, we use the quaternion representation and the following equation to calculate this error:

$$e_{\theta} = 2 \arccos(|q_{gt} \cdot q_{est}|) \quad (22)$$

- **Visibility:** Visibility is defined as the percentage of the object’s mask that is visible or not under occlusion. With this information, we are able to correlate estimation performance against occlusion robustness.
- **Distance:** Similar to visibility, we compare how the model performance changes across a variety of distances. The distance, in this case, is the Euclidean distance between the ground-truth pose of the *observing* robot and the estimated pose of the *observed* robot.

C. Hyperparameter tuning

Hyperparameter tuning is a critical component of ensuring optimal performance of the system. As such, we tune the hyperparameters of each method separately on a subset of the synthetic dataset, using Optuna [1]. This hyperparameter optimizer works by using a specified cost function and efficient sampling strategies. In our case, we used the ADD metric typically found in the BOP benchmark [15], which unifies the translational error and rotational error into one metric. For graph-based refinement, we tuned using 100 runs, while

TABLE I: Average translational error in [m] and rotational error in [deg] of different methods across 3 datasets. Each method adds additional steps to the baseline pose detector, a re-implementation of [37]. Highlighted values indicate the best-performing method for each dataset.

Method	Synthetic Dataset	Indoor Lab (No Occlusion)	Indoor Lab (Occlusion)
<i>Translational Error in [m] $\pm$ Std.</i>			
B	0.2703 ± 0.4233	0.4133 ± 0.3139	0.6064 ± 0.4435
B + GR (1-frame)	0.1230 ± 0.1450	0.4112 ± 0.2663	0.5699 ± 0.4443
B + GR (3-frame)	N/A	0.4121 ± 0.2205	0.5566 ± 0.4212
B + ER	0.2703 ± 0.4232	0.4156 ± 0.3165	0.6081 ± 0.4472
B + SR	0.2703 ± 0.4233	0.4149 ± 0.3155	0.6079 ± 0.4471
B + GR (1-frame) + SR	0.1231 ± 0.1452	0.4154 ± 0.2884	0.5715 ± 0.4510
<i>Rotational Error in [deg] $\pm$ Std.</i>			
B	21.5187 ± 38.7497	54.3190 ± 66.3145	69.4447 ± 66.4046
B + GR (1-frame)	8.2539 ± 20.4923	43.6404 ± 60.1585	39.7060 ± 50.2395
B + GR (3-frame)	N/A	50.0436 ± 47.6259	39.0371 ± 46.7934
B + ER	21.5146 ± 38.7535	54.3231 ± 66.3384	71.5022 ± 67.2184
B + SR	21.5163 ± 38.7529	54.3831 ± 66.3421	70.9840 ± 66.9936
B + GR (1-frame) + SR	8.2553 ± 20.5131	43.6416 ± 60.1613	39.7262 ± 50.5265

B = Baseline, GR = Graph refinement, ER = Edge refinement, SR = Segmentation-guided refinement.

for edge-based refinement, we only ran 20 runs because of the difference in computational complexity between the two methods. All our experiments use the best possible parameters for each method to maximize performance.

VII. RESULTS

Our experiment tested a total of 6 methods, including the baseline, against 3 different datasets, which we detail within this section. In these experiments, the baseline is a re-implementation of the markerless pose estimation method from Rueggeberg et al. [37]. In Section VII-B we provide a general overview of the performance of each method. Section VII-D evaluates how method performance scales with distance, while Section VII-E provides an analysis of qualitative results. Finally, we discuss the impact of each component in our ablation study in Section VII-F.

A. Tuned hyperparameters

Our tuned hyperparameters are described in Table II. It must be noted that segmentation-guided refinement does not introduce any new parameters. Furthermore, multi-frame graph refinement could not be tuned on the synthetic dataset because samples are not sequential. We observe that with tuned edge-refinement, the learning rates for both translation and rotation are near zero, with a low number of iterations N . Additionally, max and min values for softening indicate that annealing did not occur.

B. Overall performance

In Table I, we report the average translational and rotational errors for all methods against every dataset. The table summarizes performance across different levels of occlusion

TABLE II: Tuned hyperparameters for edge-based and graph-based pose refinement on subset of synthetic data. Graph-refinement rows report the diagonal noise values for each factor: body pose X_B , head pose X_H , and the intrinsic between-factor $\phi_{BH}(X_B, X_H)$ (Eq. 11).

Method	Param.	Value		
Edge refine.	lr (transl.)	3.42e-05		
	lr (rot.)	3.53e-05		
	iter. N	10		
	max σ_k	1		
	min σ_k	1		
Graph refine.		x	y	z
	ϕ_B transl.	6.36e-01	5.99e-01	2.76
	ϕ_B rot. (r,p,y)	3.46e-02	1.31e-01	3.66e-01
	ϕ_H transl.	5.52e-02	1.11	9.79e-02
	ϕ_H rot. (r,p,y)	2.86	1.10e-03	3.61e-01
	ϕ_{BH} transl.	2.48e-01	1.11e-05	4.22e-05
	ϕ_{BH} rot. (r,p,y)	3.91	2.37	2.91e-02

and varying distances. Moreover, the results show the average error across all articulated components. For synthetic datasets, the multi-frame graph refinement could not be evaluated because samples are not sequential and are extracted from a random set of environmental conditions. We observe that the best-performing method within the synthetic dataset is the baseline + graph refinement, which obtained a 55% decrease in translational error and 61% decrease in rotational error. For the indoor lab dataset without occlusions, single-frame graph

refinement performs significantly better in terms of rotational error but shows negligible performance for translational error. Multi-frame graph refinement performs the best by a marginal difference for translational error but substantial improvement on rotational error statistics. It must be noted that edge refinement and segmentation-guided refinement showed no observable difference from the baseline.

TABLE III: Average translational error in [m] and rotational error in [deg] of different methods for each occlusion level within the Synthetic dataset. Large outliers were removed using IQR, and bold values indicate the best-performing method.

Method	Occlusion levels			
	None 0–5%	Low 5–35%	Med. 35–65%	High >65%
<i>Translational Error [m] $\pm$ Std.</i>				
Baseline	0.17 $\pm$ 0.17	0.18 $\pm$ 0.16	0.19 $\pm$ 0.19	0.24 $\pm$ 0.11
GR	0.09 $\pm$ 0.07	0.11 $\pm$ 0.08	0.13 $\pm$ 0.09	0.12 $\pm$ 0.07
ER	0.17 $\pm$ 0.17	0.18 $\pm$ 0.16	0.19 $\pm$ 0.19	0.24 $\pm$ 0.11
SR	0.17 $\pm$ 0.17	0.18 $\pm$ 0.16	0.19 $\pm$ 0.19	0.24 $\pm$ 0.11
GR+SR	0.09$\pm$0.07	0.11$\pm$0.08	0.13$\pm$0.09	0.12$\pm$0.07
<i>Rotational Error [deg] $\pm$ Std.</i>				
Baseline	12.1 $\pm$ 1.17	11.1 $\pm$ 12.4	15.2 $\pm$ 15.3	14.8 $\pm$ 12.2
GR	5.59 $\pm$ 6.08	5.43 $\pm$ 4.71	4.89 $\pm$ 3.48	7.91 $\pm$ 16.1
ER	12.1 $\pm$ 1.17	11.1 $\pm$ 12.4	15.2 $\pm$ 15.3	14.8 $\pm$ 12.2
SR	12.1 $\pm$ 1.17	11.1 $\pm$ 12.4	15.2 $\pm$ 15.3	14.8 $\pm$ 12.2
GR+SR	5.59$\pm$6.08	5.43$\pm$4.71	4.89$\pm$3.48	7.91$\pm$16.1

C. Performance under occlusion

Understanding the performance of each method under different visibility conditions provides an overview of the robustness of the pose estimates. In Table III, the errors corresponding to 4 levels of occlusion are specified. As observed, at a general level, higher occlusion levels lead to larger estimation errors, with the full pipeline consisting of Graph refinement and Segmentation-guided refinement performing the best under occlusion. Similar to overall performance, ER and SR provide virtually no improvements over the baseline; as such, GR+SR has the same score as GR, as the SR component does not contribute.

D. Effect of distance

Model performance can be influenced by distance, as the number of pixels occupied by the object varies proportionally. In Figure 7, we illustrate model performance in relation to distance for each method and dataset. Within the synthetic dataset, we observe that model performance stays relatively consistent across a range of 20 meters. The Baseline + Graph refinement method showed the lowest translational and rotational error with minimal variance of only 0.2m. Among both Indoor lab datasets, all methods showed a proportional

increase in translational error with distance, whereas rotational error showed an inverse relationship.

E. Qualitative results

Along with quantitative metrics, we perform qualitative analysis by visually inspecting the pose estimates of each method. In Figures 8, 9, 10, we cover method differences, occlusion robustness, and failure cases in the synthetic and indoor lab datasets.

Figure 9 compares estimated poses detected by each method for the articulated head and body components of the LRU. The baseline pose noticeably exhibits a disconnect between components, with the head translated upwards along the z-axis (red). Both edge refinement and segmentation-guided refinement methods exhibit identical behavior, as supported by the quantitative results. However, the graph-based refinement method adjusts the head component to match the original rotation while anchoring the joint according to the constraints. We also observe the flip filter correcting body pose by 180° along the z-axis before refinement. Multi-frame graph refinement shows more deviation than the single-frame method.

As observed in Figure 8, the full pipeline is evaluated against varying levels of occlusion. Baseline pose estimates show an increase in errors proportional to the degree of occlusion, with limited error under no to medium levels. Under heavy occlusion, the baseline fails to detect all components. In contrast, the full pipeline is capable of correcting and even reconstructing component poses even under heavy occlusions.

Understanding the failure cases is crucial for assessing the applicability for different scenarios. Figure 10 highlights two frequent cases: partial inaccurate recovery and breaking rotational symmetry failure. In the first image, the estimated pose is drastically rotated along the body’s y-axis in comparison to the original position during an occluded maneuver in the indoor lab. The body is inferred from the head’s position, whose initial detection was inaccurate, leading to an inaccurate recovery. The second image illustrates a common 180° rotational error along the z-axis from the convergence of the model to an ambiguous rotational pose. The symmetric nature of the object body makes it difficult to consistently break symmetry under translational errors from relative constraints alone, even with an additional flip filter.

F. Ablation study

To understand the contribution of each component towards the final pose estimate, we incrementally add each component towards the pose detector as shown in Table I. Among all the methods, graph refinement (both single and multi-frame) alone was able to provide substantial improvements from the baseline methodology with a 55% decrease in translational error on the synthetic dataset. However, edge-refinement and segmentation-guided refinement showed no improvement over the baseline. Therefore, the contribution of edge-based refinement methods towards the full pipeline can be considered negligible.

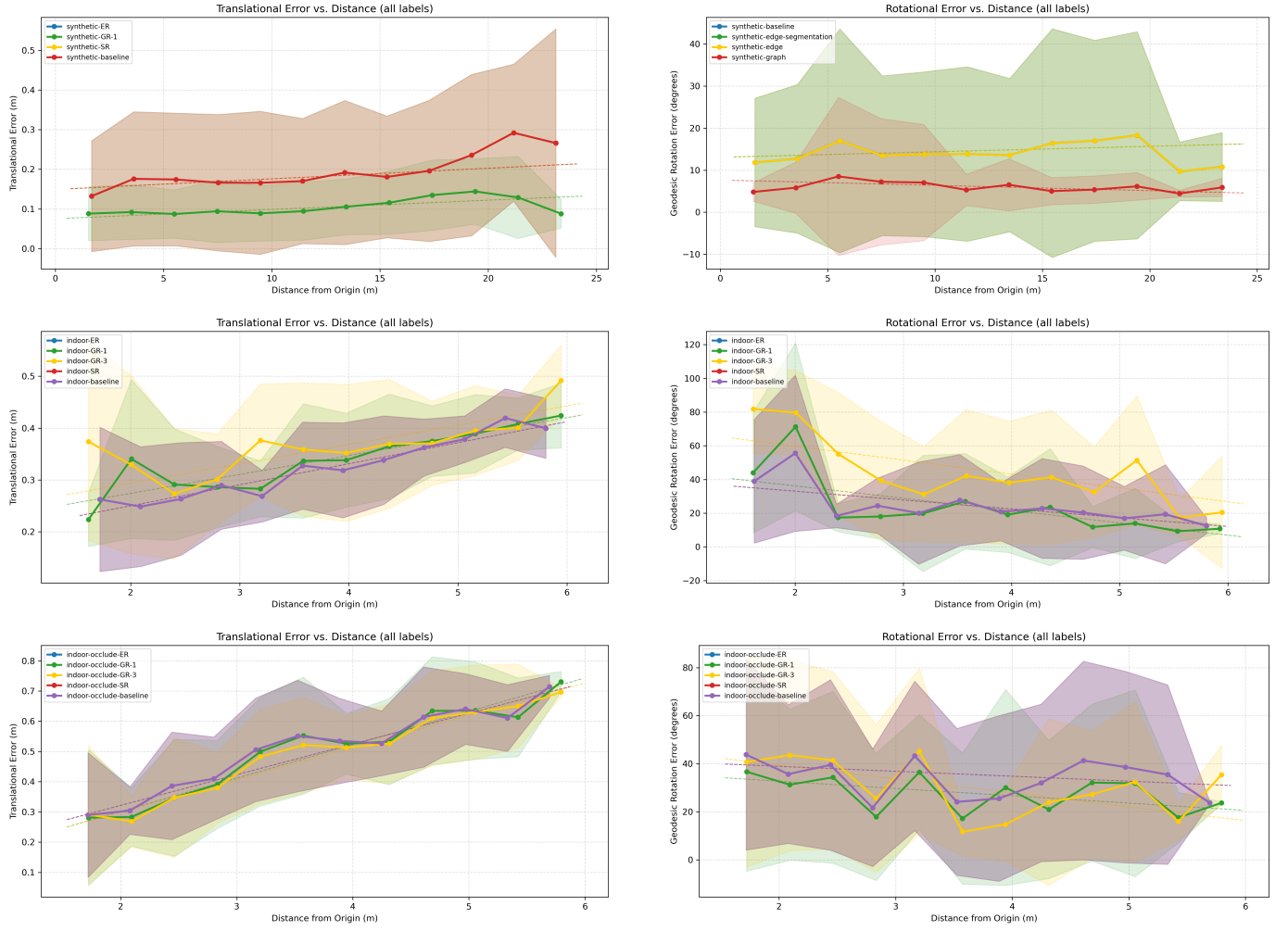


Fig. 7: Translational and Rotational errors versus distance for each dataset and method. The columns are translational errors [m] vs distance [m] and rotational errors [deg] versus distance [m] respectively. The rows are organized by dataset. From top to bottom: Synthetic dataset, Indoor lab dataset without occlusions, and Indoor lab dataset with occlusions. For each increment of 0.5m we display the line of best fit and ± 1 std residual band.

TABLE IV: Wilcoxon T-test results on the synthetic dataset across all occlusion levels. Mean diff = baseline – method; positive means the method reduced error. Sig. represents significance.

Method	Metric	Mean Diff	$p < 0.001$	Sig.
ER	Trans. [m]	−0.0000	0.171	No Sig.
	Rot. [deg]	0.0046	1.08×10^{-4}	Sig.
SR	Trans. [m]	−0.0000	0.014	No Sig.
	Rot. [deg]	0.0031	0.026	No Sig.
GR	Trans. [m]	0.0378	2.86×10^{-20}	High Sig.
	Rot. [deg]	2.5532	1.07×10^{-30}	High Sig.

VIII. DISCUSSION

A. ER and SR validity

Both Edge refinement (ER) and Segmentation-guided refinement (SR) provided almost identical results to the baseline, as shown in Table I, which can directly be correlated to the near-zero learning rates for both translational and rotational error chosen from hyperparameter tuning (see Table II). In Appendix A, we investigated the impact of learning rate using a few qualitative samples from our tuning samples and found that higher learning rates led to lower alignment. Additionally, we noticed that the convergence location for higher learning-rate refinement methods is away from the original body, indicating that the target edges do not properly capture the object geometry and pose information. Therefore, further processing of target images for both the ER and SR methods is required to ensure accurate convergence to the predicted poses.

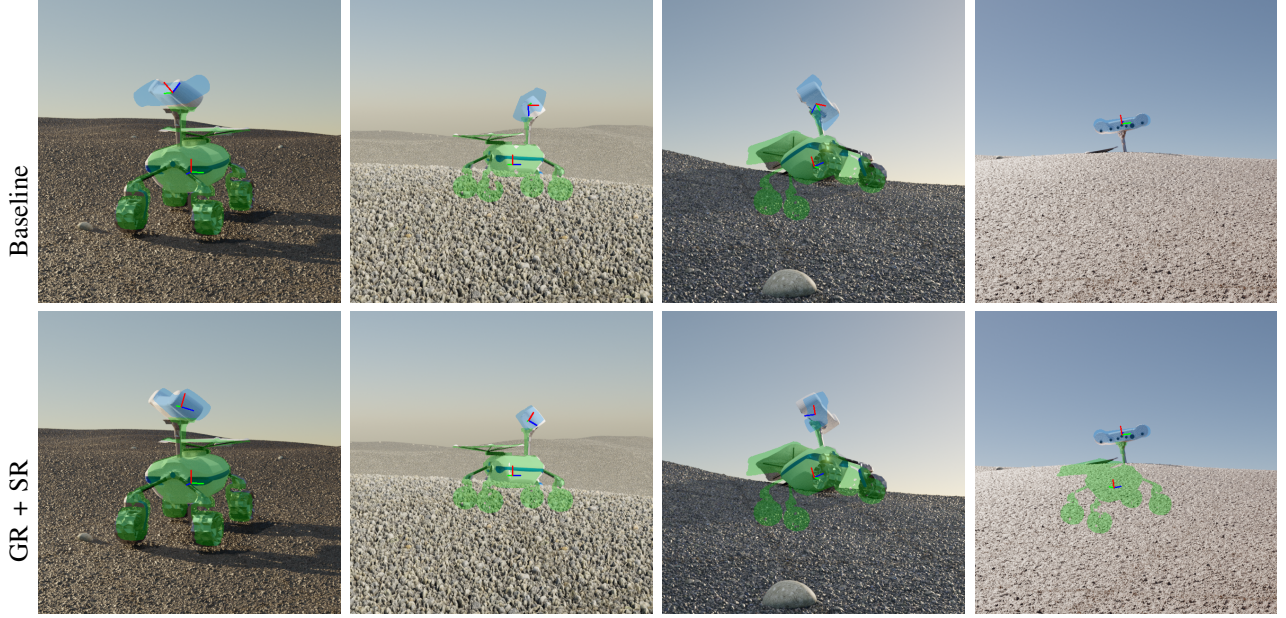


Fig. 8: Qualitative pose estimation results across increasing occlusion severity (none, light, medium, heavy). Top row: baseline. Bottom row: Graph refinement (GR) + Segmentation-guided refinement (SR).

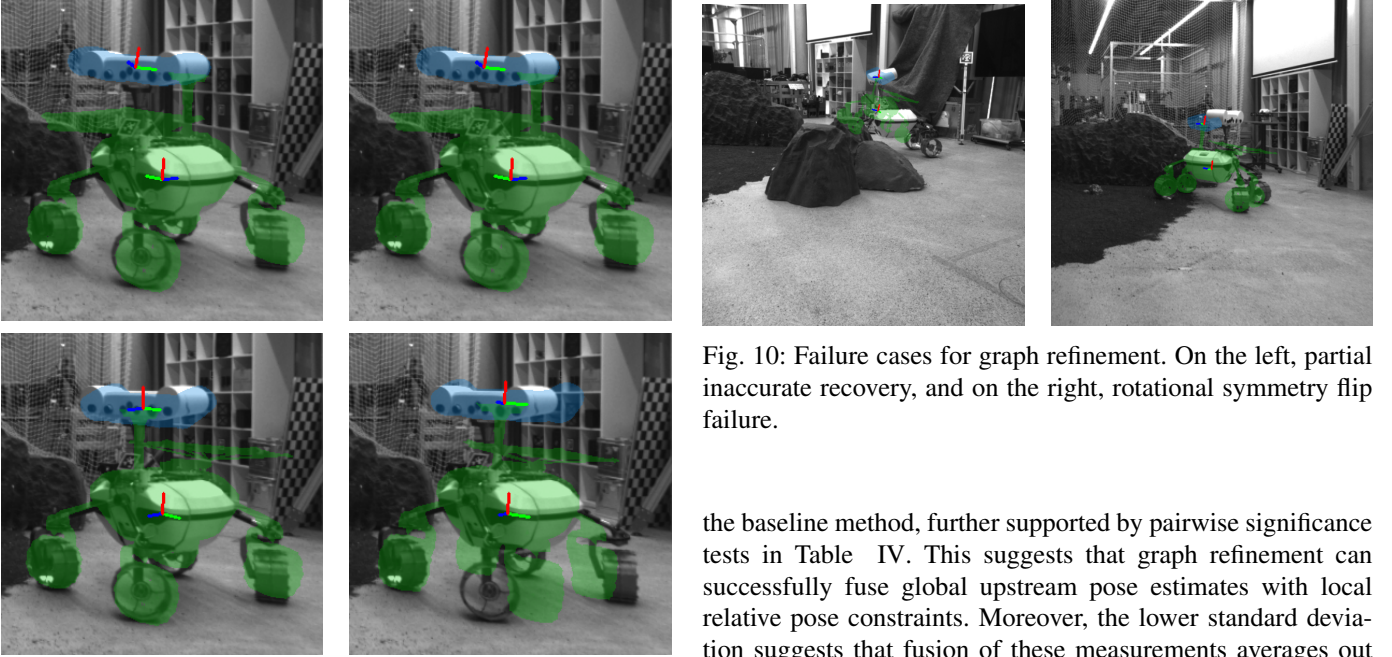


Fig. 9: Side-by-side comparison of 4 methods of the articulated Lightweight Rover Unit robot in an indoor lab. **Top left:** Baseline, **Top Right:** Baseline + Edge refinement, **Bottom left:** Baseline + Graph refinement, **Bottom right:** Baseline + Multi-frame Graph refinement (3 frames).

B. Graph refinement performance

As observed in Section VII-B, graph refinement showed a substantial improvement in the synthetic dataset compared to



Fig. 10: Failure cases for graph refinement. On the left, partial inaccurate recovery, and on the right, rotational symmetry flip failure.

the baseline method, further supported by pairwise significance tests in Table IV. This suggests that graph refinement can successfully fuse global upstream pose estimates with local relative pose constraints. Moreover, the lower standard deviation suggests that fusion of these measurements averages out the deviation between the articulated components.

Additionally, we observe from our qualitative results in Section VII-E that graph refinement enables poses to be recovered when upstream pose estimation fails drastically. Our factor graph formulation enables us to optimize for unknown poses even when extrinsic measurements $Z_i \in \mathcal{Z}$ lead to unary factors being dropped from the graph topology. As long as variables are present between factors in the connected topology, the variables $X_i \in \mathcal{X}$ will be present in the optimization and can be recovered [10]. This is also the result we observe across our results.

C. Single-frame vs Multi-frame refinement

Multi-frame graph refinement surprisingly performs worse than the single-frame method within the Indoor Lab (No occlusion) dataset but better under the Indoor Lab (Occlusion) dataset. This demonstrates that multi-frame graph refinement provides slight improvements, conditioned on the presence of occlusions. Furthermore, looking at the standard deviations of the rotational error, we observe that multi-frame GR is more peaked than single-frame. As hypothesized in Section V-F, fusing visual odometry data helps spot inconsistencies across the time domain, one of which is rotational symmetries. The significantly lower rotational standard deviation suggests that pose rotations across multiple frames resolve closer to the true heading. We observe a similar relation across translations but not as significant.

D. Pose recovery and Accuracy under occlusion

Qualitative evaluation of the different levels of occlusion in Section VII-E shows the ability of the full pipeline to recover and estimate poses even under severe occlusion. By having at least one detected component and the set of relative constraints, the refinement method allows us to infer the position of other components, as detailed in the previous section (see Section VIII-B). However, these inferred components are prone to the same errors as the original detected component and can lead to inaccuracies.

E. Distance evaluation

Object distance has varying levels of influence on each method across the different datasets, as highlighted in Section VII-D. In the synthetic case, we observe that all methods show no correlation between the observed errors and object distance, likely due to the synthetic dataset being generated from a similar distribution as our training set. As such, our learned pose estimator EagerNet [45] provides stable performance across a wide range of distances, with our refinement methods providing only constant differences in error.

In contrast, we see a more natural trend within the indoor-lab datasets, particularly with translational error. Translational error increases with the object’s distance from the camera, which makes per-pixel localization for features more challenging. As objects move further away across the z-axis, small planar movements on the image plane correspond to much larger movements in 3D space and are constrained by pixel resolution limits. However, we notice relatively stable, if not slightly decreasing, performance for rotations, which could be attributed to truncation issues caused by image borders or the similarity with the distances within the training set.

F. Sim-to-real Gap

We strongly observe the sim-to-real gap for each refinement method, particularly visible in Table I between the synthetic dataset and the two indoor lab datasets. This can be attributed to the sim-to-real gap, as our live system contains significantly more geometric detail than the CAD models used for simulation and inference, requiring the model to extrapolate.

Additionally, all indoor lab data, with and without occlusions, provides 3-channel grayscale images, whereas the pose estimation model was trained on full color images. Therefore, the lack of color reduces the information available, requiring the model to rely more on geometric cues to estimate the position of each component. Similarly, within Edge refinement, we observe that target images in the indoor-live data contain different edges than the CAD model, which has a significant impact on convergence.

G. Limitations

While our proposed methods demonstrate clear improvements over single-body pose estimation, most notable under occlusions, we identify several limitations that constrain the generalizability of our findings. The failure cases described in Section VII-E highlight common limitations of single-frame and multi-frame graph refinement methods. Firstly, both articulated components, the LRU head and LRU body, are symmetric along at least one axis, which makes disambiguating the pose challenging. Even with a flip-filter to keep the distribution uni-modal, real-world runs struggle if both upstream pose estimates exhibit the same rotational flip, as seen in Figure 10. This is because single-frame graph refinement is only able to enforce local relative rotation between components, and not global rotation. Multi-frame methods also face similar issues under the same circumstances, but to a lesser degree, as errors are averaged across frames. Additionally, multi-frame methods, due to their larger number of variables, face higher risks of the optimization exploding if upstream detections are not uni-modal or are not within the defined error bounds of the noise model.

Our experiments have only considered an articulated system of two bodies due to the available intrinsic measurement. No joint information was available for other articulated components of the robot, such as the wheels, forcing us to consider them as static components of the LRU body. Moreover, the variance of our results could be attributed to the lack of robustness during model training of our pose estimator, indicating the need for larger datasets and possibly stronger augmentations.

Furthermore, EagerNet [45] is trained on each object individually, requiring an ensemble of networks to detect poses of multiple objects, which is not scalable for a large number of articulated components. The process also reduces any correlation information between articulated objects that can decrease robustness under partial visibility.

Our work purely evaluates the pose estimation performance of our methods and the baseline on a component-level. Though currently integrated into a Multi-agent SLAM system, we have yet to quantify the performance against system-level multi-agent SLAM metrics such as map consistency and trajectory errors. As a result, while our method states improved performance under occlusion for individual pose estimates, we cannot yet confirm if that translates to better SLAM performance, as other factors, including detection frequency, can alter the effects of our measurements.

H. Future work

Our work has shown the promise of articulated pose estimation methods as alternatives to pure single-body methods; however, there remains room for future work. Within this section, we highlight potential directions to improve upon our limitations, including evaluation on a multi-agent SLAM system, changes to our differentiable rendering methods, and increasing the robustness of articulated detections and upstream pose estimates.

- **Evaluation on Multi-agent SLAM data** - Our work has thoroughly explored our articulated methods against the baseline at the component level; however, it does not quantify downstream Multi-agent SLAM performance after integration into the SLAM system. Follow-up work would run system-level tests using localization and mapping metrics to verify the impact of articulated methods as the source of inter-robot constraints.
- **Unconstrained articulated components** - A potential solution to the lack of intrinsic measurements for some articulated components would be to use partial constraints. Consider the LRU wheels; they are capable of swiveling around the z-axis and are translational. Pose estimations could provide initial values for the graph optimization, with preset constraints restricting the factors only across certain dimensions, leaving the other's free.
- **Differentiable features** - Differentiable rendering requires differentiable functions, thereby limiting extracted features to edges, silhouettes, or masks. More robust features, including ORB [36], are lightweight but hand-crafted. Using differentiable feature extractors such as deep neural networks on rendered objects can improve performance.
- **Target edge filtering** - Clear target edges are fundamental in ensuring convergence to the true object pose, as detailed in Section VIII-A. Future work could explore better filtering strategies to remove environmental noise, such as planetary terrain, while preserving target edges within a localized region around the estimated pose. Another promising direction is using more descriptive loss functions, as Edge refinement and Segmentation-guided refinement are heavily dependent on informative gradients. As such, encoding scale, distance, and geometric features within the loss can improve performance [29].
- **Domain randomization** - The sim-to-real gap in our experiments was caused by our use of low-fidelity 3D models and availability of only grayscale images during inference. As such, randomizing the environments and domains of our training data while increasing the dataset size and augmentations, including grayscale, can boost out-of-distribution performance during inference. Additionally, training a single network, rather than an ensemble, can capture dependencies between articulated components as done with category-level articulated estimation methods [27, 30].

IX. CONCLUSION

Within this work, we investigated the use of articulated pose estimation techniques and their applicability towards direct inter-robot detection in multi-agent SLAM. We created 5 different methods and performed per-component analysis on synthetic planetary terrain data and indoor lab datasets. Our results showed that graph-based pose refinement, which enforces relative constraints between components, led to a 55% decrease in translational error and a 61% decrease in rotational error, in comparison to the baseline, showing clear improvements in inter-robot detection capabilities. Furthermore, the full pipeline showed its ability to recover undetected poses even under heavy occlusions with only a single detected component of the robot. In contrast, geometry-aware approaches such as edge-refinement and segmentation-guided refinement showed limited impact on estimates, likely as a result of insufficient filtering on target images and differences caused by the sim-to-real gap. Future work should evaluate system-level metrics on a Multi-agent SLAM system and prioritize improving target edge filtering to assist convergence to the true object pose. Our work provides a foundation for direct, markerless inter-robot detection methods in the planetary domain, demonstrating that representing robots as an ensemble of articulated components improves pose estimation robustness under occlusion and complex robot morphologies. This brings us a step closer to the accurate, long-range markerless detection that multi-agent SLAM depends on.

X. ACKNOWLEDGMENTS

This thesis would not have been possible without the contributions and support of many remarkable people, to whom I owe my sincerest gratitude. I want to thank Prof. Robert Babuška for invaluable guidance and critical feedback throughout this research. I am also thankful to Dr. Riccardo Giubilato from DLR-RMC for his daily supervision and many insights into modern-day rovers and planetary robotic systems. Additionally, I would like to thank DLR-RMC for hosting and providing me with the resources to conduct my research. Lastly, I am grateful to the friends and family who encouraged me to explore new horizons abroad to pursue my interests. Their support and belief have kept me fueled and motivated during the process.

REFERENCES

- [1] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.
- [2] Relja Arandjelović et al. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (June 2018), pp. 1437–1451. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2017.2711011. URL: <https://ieeexplore.ieee.org/document/7937898/> (visited on 07/12/2026).

- [3] Philipp Ausserlechner et al. *ZS6D: Zero-shot 6D Object Pose Estimation using Vision Transformers*. arXiv:2309.11986 [cs]. Sept. 2023. DOI: 10.48550/arXiv.2309.11986. URL: <http://arxiv.org/abs/2309.11986> (visited on 03/25/2026).
- [4] Andréa Macario Barros et al. “A Comprehensive Survey of Visual SLAM Algorithms”. In: *Robotics* 11.1 (Feb. 2022). ISSN: 2218-6581. DOI: 10.3390/robotics11010024. URL: <https://www.mdpi.com/2218-6581/11/1/24> (visited on 03/08/2026).
- [5] Michael Calonder et al. “BRIEF: Binary Robust Independent Elementary Features”. In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Berlin, Heidelberg: Springer, 2010, pp. 778–792. ISBN: 978-3-642-15561-1. DOI: 10.1007/978-3-642-15561-1_56.
- [6] Yun Chang et al. “Kimera-Multi: a System for Distributed Multi-Robot Metric-Semantic Simultaneous Localization and Mapping”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2021, pp. 11210–11218. DOI: 10.1109/ICRA48506.2021.9561090. URL: <https://ieeexplore.ieee.org/document/9561090> (visited on 03/22/2026).
- [7] Yun Chang et al. *LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments*. arXiv:2205.13135 [cs]. July 2022. DOI: 10.48550/arXiv.2205.13135. URL: <http://arxiv.org/abs/2205.13135> (visited on 03/22/2026).
- [8] Zhangmeng Chen et al. “DGFormer: Dynamic graph transformer for 3D human pose estimation”. In: *Pattern Recognition* 152 (Aug. 2024), p. 110446. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2024.110446. URL: <https://www.sciencedirect.com/science/article/pii/S0031320324001973> (visited on 03/25/2026).
- [9] Frank Dellaert and GTSAM Contributors. *borglab/gtsam*. DOI: 10.5281/zenodo.5794541. URL: <https://github.com/borglab/gtsam>.
- [10] Frank Dellaert and Michael Kaess. “Factor graphs for robot perception”. In: *Foundations and Trends in Robotics* 6.1-2 (2017), pp. 1–139. URL: <https://dellaert.github.io/files/Dellaert17fnt.pdf>.
- [11] Maximilian Denninger et al. “BlenderProc2: A Procedural Pipeline for Photorealistic Rendering”. In: *Journal of Open Source Software* 8 (Feb. 2023), p. 4901. DOI: 10.21105/joss.04901.
- [12] Haoshu Fang et al. *Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation*. arXiv:1710.06513 [cs]. Jan. 2018. DOI: 10.48550/arXiv.1710.06513. URL: <http://arxiv.org/abs/1710.06513> (visited on 03/25/2026).
- [13] Raktim Gautam Goswami et al. “RoboPEPP: Vision-Based Robot Pose and Joint Angle Estimation through Embedding Predictive Pre-Training”. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025, pp. 6930–6939. DOI: 10.1109/CVPR52734.2025.00650.
- [14] Xujun Han et al. “PoseFusion: Multi-Scale Keypoint Correspondence for Monocular Camera-to-Robot Pose Estimation in Robotic Manipulation”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 795–801. DOI: 10.1109/ICRA57147.2024.10610844. URL: <https://ieeexplore.ieee.org/document/10610844> (visited on 03/25/2026).
- [15] Tomáš Hodaň et al. “BOP: Benchmark for 6D Object Pose Estimation”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 19–35. ISBN: 978-3-030-01249-6.
- [16] Sungchul Hong et al. “Visual SLAM-Based Robotic Mapping Method for Planetary Construction”. In: *Sensors* 21.22 (Nov. 2021). ISSN: 1424-8220. DOI: 10.3390/s21227715. URL: <https://www.mdpi.com/1424-8220/21/22/7715> (visited on 03/08/2026).
- [17] Jingshun Huang et al. *You Only Estimate Once: Unified, One-stage, Real-Time Category-level Articulated Object 6D Pose Estimation for Robotic Grasping*. arXiv:2506.05719 [cs] version: 1. June 2025. DOI: 10.48550/arXiv.2506.05719. URL: <http://arxiv.org/abs/2506.05719> (visited on 03/25/2026).
- [18] Yewei Huang et al. “DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM With Two-Stage Global-Local Graph Optimization”. In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 1150–1157. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3138156. URL: <https://ieeexplore.ieee.org/document/9662965> (visited on 03/23/2026).
- [19] ISECG. *Global Exploration Roadmap*. Tech. rep. Cologne: The International Space Exploration Coordination Group, Aug. 2024, p. 65. URL: <https://www.globalspaceexploration.org/wp-content/isecg/GER2024.pdf>.
- [20] Ren Jin et al. “Drone Detection and Pose Estimation Using Relational Graph Networks”. In: *Sensors* 19.6 (Mar. 2019). ISSN: 1424-8220. DOI: 10.3390/s19061479. URL: <https://www.mdpi.com/1424-8220/19/6/1479> (visited on 03/26/2026).
- [21] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. “iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. Apr. 2007, pp. 1670–1677. DOI: 10.1109/ROBOT.2007.363563. URL: <https://ieeexplore.ieee.org/document/4209327> (visited on 03/11/2026).
- [22] Michael Kaess et al. “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering”. In: *2011 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. May 2011, pp. 3281–3288. DOI: 10.1109/ICRA.2011.

5979641. URL: <https://ieeexplore.ieee.org/document/5979641> (visited on 03/11/2026).
- [23] Marco Karrer, Patrik Schmuck, and Margarita Chli. “CVI-SLAM—Collaborative Visual-Inertial SLAM”. In: *IEEE Robotics and Automation Letters* 3.4 (Oct. 2018), pp. 2762–2769. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2837226. URL: <https://ieeexplore.ieee.org/document/8360045> (visited on 03/23/2026).
- [24] Pierre-Yves Lajoie et al. “DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams”. In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 1656–1663. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2020.2967681. URL: <https://ieeexplore.ieee.org/document/8962967/> (visited on 03/23/2026).
- [25] Timothy E. Lee et al. “Camera-to-Robot Pose Estimation from a Single Image”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 9426–9432. DOI: 10.1109/ICRA40945.2020.9196596. URL: <https://ieeexplore.ieee.org/abstract/document/9196596> (visited on 03/25/2026).
- [26] Han Li et al. *Pose-Oriented Transformer with Uncertainty-Guided Refinement for 2D-to-3D Human Pose Estimation*. arXiv:2302.07408 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2302.07408. URL: <http://arxiv.org/abs/2302.07408> (visited on 03/25/2026).
- [27] Xiaolong Li et al. *Category-Level Articulated Object Pose Estimation*. arXiv:1912.11913 [cs]. Apr. 2020. DOI: 10.48550/arXiv.1912.11913. URL: <http://arxiv.org/abs/1912.11913> (visited on 03/25/2026).
- [28] Yi Li et al. “DeepIM: Deep Iterative Matching for 6D Pose Estimation”. In: *International Journal of Computer Vision* 128.3 (Mar. 2020). arXiv:1804.00175 [cs], pp. 657–678. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-019-01250-9. URL: <http://arxiv.org/abs/1804.00175> (visited on 03/25/2026).
- [29] Zekai Liang et al. “Differentiable Rendering-based Pose Estimation for Surgical Robotic Instruments”. In: *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2025, pp. 20898–20905. DOI: 10.1109/IROS60139.2025.11245870. URL: <https://ieeexplore.ieee.org/document/11245870/> (visited on 04/30/2026).
- [30] Liu Liu et al. “Toward Real-World Category-Level Articulation Pose Estimation”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 1072–1083. ISSN: 1941-0042. DOI: 10.1109/TIP.2021.3138644. URL: <https://ieeexplore.ieee.org/document/9670684> (visited on 03/25/2026).
- [31] Mark Maimone et al. “Autonomous Navigation Results from the Mars Exploration Rover (MER) Mission”. In: *Experimental Robotics IX*. Ed. by Bruno Siciliano et al. Vol. 21. Series Title: Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 3–13. ISBN: 978-3-540-28816-9 978-3-540-33014-1. DOI: 10.1007/11552246_1. URL: http://link.springer.com/10.1007/11552246_1 (visited on 03/06/2026).
- [32] Marcus Gerhard Müller et al. “Interactive OASYS: A photorealistic terrain simulation for robotics research”. In: *ICRA 2022 Workshop on Releasing Robots into the Wild: Simulations, Benchmarks, and Deployment* (May 2022).
- [33] Edwin Olson. “AprilTag: A robust and flexible visual fiducial system”. In: *2011 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. May 2011, pp. 3400–3407. DOI: 10.1109/ICRA.2011.5979561. URL: <https://ieeexplore.ieee.org/document/5979561> (visited on 07/12/2026).
- [34] Pavel Petracek et al. “Bio-Inspired Compact Swarms of Unmanned Aerial Vehicles without Communication and External Localization”. In: *Bioinspiration & Biomimetics* 16.2 (Mar. 2021). arXiv:2303.02989 [cs], p. 026009. ISSN: 1748-3182, 1748-3190. DOI: 10.1088/1748-3190/abc6b3. URL: <http://arxiv.org/abs/2303.02989> (visited on 03/26/2026).
- [35] Benjamin Ramtoul et al. *MSL-RAPTOR: A 6DoF Relative Pose Tracker for Onboard Robotic Perception*. arXiv:2012.09264 [cs]. Dec. 2020. DOI: 10.48550/arXiv.2012.09264. URL: <http://arxiv.org/abs/2012.09264> (visited on 03/26/2026).
- [36] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International Conference on Computer Vision*. ISSN: 2380-7504. Nov. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544. URL: <https://ieeexplore.ieee.org/document/6126544> (visited on 03/11/2026).
- [37] Markus Rüggeberg et al. *Markerless Robot Detection and 6D Pose Estimation for Multi-Agent SLAM*. arXiv:2602.16308 [cs] version: 1. Feb. 2026. DOI: 10.48550/arXiv.2602.16308. URL: <http://arxiv.org/abs/2602.16308> (visited on 03/22/2026).
- [38] Patrik Schmuck and Margarita Chli. “CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams”. In: *Journal of Field Robotics* 36.4 (2019). _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21854>, pp. 763–781. ISSN: 1556-4967. DOI: 10.1002/rob.21854. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21854> (visited on 07/12/2026).
- [39] Martin J. Schuster et al. “The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration”. In: *IEEE Robotics and Automation Letters* 5.4 (Oct. 2020), pp. 5315–5322. DOI: 10.1109/LRA.2020.3007468. URL: <https://ieeexplore.ieee.org/document/9134730>.
- [40] Martin J. Schuster et al. “Towards Autonomous Planetary Exploration”. In: *Journal of Intelligent & Robotic Systems* 93.3 (Mar. 2019), pp. 461–494. ISSN: 1573-

0409. DOI: 10.1007/s10846-017-0680-9. URL: <https://doi.org/10.1007/s10846-017-0680-9> (visited on 07/12/2026).
- [41] Qinqin Shen, Yi Fan, and Guoliang Wei. “A Novel Tag-Based Robust Pose Estimation and Tracking Algorithm via Binoculars Visual Measurement”. In: *IEEE Transactions on Instrumentation and Measurement* 74 (2025), pp. 1–12. ISSN: 1557-9662. DOI: 10.1109/TIM.2025.3554323. URL: <https://ieeexplore.ieee.org/abstract/document/10938684> (visited on 03/26/2026).
- [42] Florian Steidle et al. “A Temporal Perspective n-Point Problem with Model Uncertainties for Cooperative Pose Estimation in a Heterogeneous Robot Team”. In: *2023 European Conference on Mobile Robots (ECMR)*. ISSN: 2767-8733. Sept. 2023, pp. 1–7. DOI: 10.1109/ECMR59166.2023.10256287. URL: <https://ieeexplore.ieee.org/document/10256287> (visited on 03/26/2026).
- [43] Martin Sundermeyer et al. “BOP Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2023-June (Feb. 2023), pp. 2785–2794. URL: <http://arxiv.org/abs/2302.13075>.
- [44] Guoqiang Tang et al. “Relative Localization and Dynamic Tracking of Underwater Robots Based on 3D-AprilTag”. In: *Journal of Marine Science and Engineering* 13.5 (Apr. 2025). ISSN: 2077-1312. DOI: 10.3390/jmse13050833. URL: <https://www.mdpi.com/2077-1312/13/5/833> (visited on 03/26/2026).
- [45] Maximilian Ulmer et al. “6D Object Pose Estimation from Approximate 3D Models for Orbital Robotics”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2023, pp. 10749–10756. DOI: 10.1109/IROS55552.2023.10341511. URL: <https://ieeexplore.ieee.org/document/10341511/> (visited on 03/24/2026).
- [46] Xiaoyue Wan, Zhuo Chen, and Xu Zhao. *View Consistency Aware Holistic Triangulation for 3D Human Pose Estimation*. arXiv:2302.11301 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2302.11301. URL: <http://arxiv.org/abs/2302.11301> (visited on 03/25/2026).
- [47] Chien Yao Wang, Alexey Bochkovskiy, and Hong Yuan Mark Liao. “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2023-June (2023), pp. 7464–7475. ISSN: 9798350301298. DOI: 10.1109/CVPR52729.2023.00721. URL: <https://ieeexplore.ieee.org/document/10204762>.
- [48] John Wang and Edwin Olson. “AprilTag 2: Efficient and robust fiducial detection”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2016, pp. 4193–4198. DOI: 10.1109/IROS.2016.7759617. URL: <https://ieeexplore.ieee.org/document/7759617> (visited on 07/12/2026).
- [49] Bowen Wen et al. “FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), pp. 17868–17879. ISSN: 9798350353006. DOI: 10.1109/CVPR52733.2024.01692.
- [50] Yuanlu Xu et al. “Monocular 3D Pose Estimation via Pose Grammar and Data Augmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (Oct. 2022), pp. 6327–6344. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3087695. URL: <https://ieeexplore.ieee.org/document/9450016> (visited on 07/12/2026).
- [51] Zhiren Xun et al. *CREPES: Cooperative Relative Pose Estimation System*. arXiv:2302.01036 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2302.01036. URL: <http://arxiv.org/abs/2302.01036> (visited on 03/23/2026).
- [52] Yiran Zhu et al. “PoseGTAC: Graph Transformer Encoder-Decoder with Atrous Convolution for 3D Human Pose Estimation”. In: vol. 2. Aug. 2021, pp. 1359–1365. DOI: 10.24963/ijcai.2021/188. URL: <https://www.ijcai.org/proceedings/2021/188> (visited on 03/25/2026).

APPENDIX A HYPERPARAMETER TUNING DIAGNOSTICS

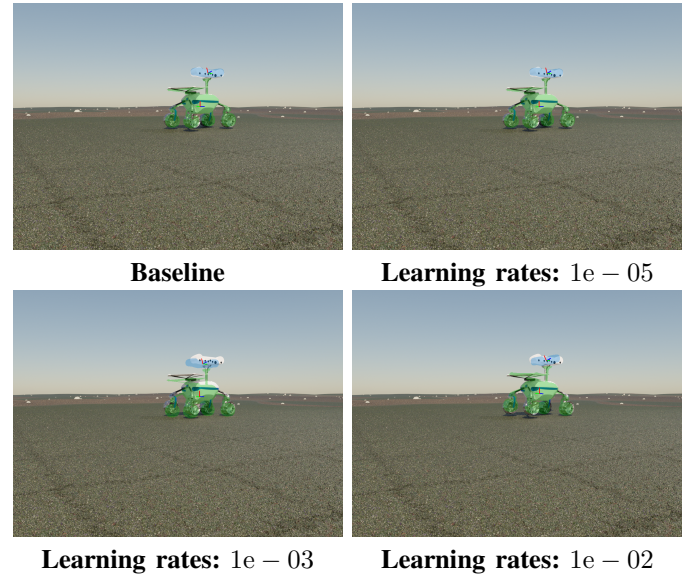


Fig. 11: Visualizations of the impact of translational and rotational learning rates on Edge refinement (ER). Translational and rotational learning rates are shared; higher learning rates show greater deviation. Top Left: Baseline method, Top Right: ER with learning rate 1e-5 (near-zero). Bottom Left: ER with learning rate 1e-03. Bottom right: ER with learning rate 1e-02.