Relative Robot Localization and Frame Alignment for Multi-Robot Collaboration

by

Mason B. Peterson

B.S., Brigham Young University (2022)

Submitted to the Massachusetts Institute of Technology
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2024

Authored by:	Mason B. Peterson Department of Aeronatuics and Astronautics May 10, 2024
Certified by:	Jonathan P. How R. C. Maclaurin Professor of Aeronautics and Astronautics, Thesis Supervisor
Accepted by:	Jonathan P. How R. C. Maclaurin Professor of Aeronautics and Astronautics Chair, Graduate Program Committee

Relative Robot Localization and Frame Alignment for Multi-Robot Collaboration

by

Mason B. Peterson

Submitted to the Massachusetts Institute of Technology
on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

ABSTRACT

The growing field of collaborative robotics has the potential to enable and improve the execution of many challenging robot applications. For instance, with teamwork between multiple agents, dynamic object tracking can more completely cover an environment and trajectory planning becomes safer. However, for robots to share the quickly changing spatial information involved in these tasks, robots need to be able to express information originally sensed or planned in their own frame into the frame of neighboring agents. This can be challenging in cases where robots have no global pose information resulting in steady accumulation of error, or drift, in their local pose estimates. To mitigate the effects of drift, neighboring agents must make up-to-date estimates of the alignment between their frames, which can be difficult due to ambiguous alignments and the presence of outlier measurements. To address these issues, the first contribution of this thesis is a method for performing fast incremental frame alignment between pairs of robots, enabling collaborative multiple object tracking (MOT), the task of monitoring the locations of dynamic objects in an environment. To perform frame alignment, robots build up maps of recently seen static objects and use these maps and the detections of tracked dynamic objects to correct for frame drift. Using frame alignment estimates, agents share object detection information and account for additional uncertainty associated with the alignment estimate. The second contribution of this thesis presents a method to perform frame alignment with no initial guess. Many potential frame alignments are computed and we develop a filter that uses temporal consistency to reject outlier alignments and only accept a series of alignments that are consistent over time. We demonstrate in hardware experiments our ability to perform frame alignment in difficult scenarios and improve the quality of collaborative object tracking onboard real robots.

Thesis supervisor: Jonathan P. How
Title: R. C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

I am grateful for the countless individuals who have played a part both in inspiring me to pursue a Master's degree and in supporting me during this undertaking.

First, a deep thank you to my advisor, Professor Jonathan How, who, despite being a world-class researcher whose time is in high demand, goes out of his way to sit down and help guide a student in finding the optimal sensor suite for a team of robots. I have been fortunate to benefit from his keen intuition in robotics problems. Oftentimes, I have found that implementing an idea from one of his offhand comments results in a solution to the particular problem I was trying to solve. I am grateful for his coaching and mentorship and his interest in solving hard problems.

I have been incredibly grateful for the people in the Aerospace Controls Laboratory (ACL), who have made this degree a truly enjoyable experience. Thanks to Parker Lusk, who spent an enormous amount of time mentoring me and helping me gain a solid foundation in robotics fundamentals, including the joys and pains of hardware experiments. Thanks to Yulun Tian for sharing your expertise in SLAM with me and Kota Kondo, Andrea Tagliabue, and Tong Zhao for giving me the chance to fly and crash drones with you. Thanks to Annika Thomas, Lucas Jia, Jouko Kinnari, Jacqueline Ankenbauer, Nick Rober, and Lakshay Sharma for building neat technologies with me in research and class projects. Thanks to Paul Kohler, Eve Ruff, Naman Aggarwal, Hannah Shafferman, and Kyle Sonandres for sharing a great desk pod, I'm grateful for our many conversations. Thanks to my many other ACL friends, Jeremy Cai, Andrew Fishberg, Lena Downes, Stewart Jamieson, Jesus

Tordesillas, Aneesa Sonawalla, Donggun Lee, Dong-Ki Kim, Anubhav Guha, Jo Nikolova, Jimmy Queeney, and Alex Meredith. Thanks to Antonio Avila for your excellent help as a UROP.

I am also so thankful for many other technical mentors in my life who helped me develop a love for robotics. Thank you Dr. Randy Beard, Dr. Josh Mangelson, Rolf Rysdyk, and Jasmine Minteer-Levine.

A special thanks goes to my family, whose love and support makes the hard days easier and the good days a celebration. Thank you to my dad for encouraging me to be curious and to my mom for believing that I could accomplish anything. Thank you to Brice, Kylie, Dani, Keni, and Jens for inspiring me to be my best. Lastly, thank you to my wife Lorena who has been there for me every step of the way through graduate school, from soldering robot parts to lifting my spirits when research seems stuck in a rut. You make every day better. Thank you.

This work was supported by the Ford Motor Company and ARL DCIST under Cooperative Agreement Number W911NF-17-2-0181.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The current capabilities of robotic systems are expanding rapidly with the development of methods for multi-robot collaboration. Intuitively, robotic tasks like monitoring the locations of nearby dynamic objects for surveillance or collision avoidance [1], planning safe trajectories around other traveling neighbors [2], and creating a globally consistent map of an environment [3] can be completed more effectively when performed by a team of collaborating robots. However, by involving multiple decision-making agents with limited communication bandwidth, additional complications must be algorithmically addressed to achieve the enhanced performance that cooperative robotics promises. Among these challenges is the problem of communicating precise spatial information (e.g., current locations of dynamic objects or positions of planned trajectory waypoints) accurately to neighbors.

Accurately sharing geometric information is difficult because, in many instances, communicating robots do not share the same coordinate frame, which means a robot needs a way to transform coordinates and rotations from its own frame into the frame of its neighbors. This transformation from one robot frame to another can be achieved through *frame alignment*, or estimating the rotation and translation between the origins of a pair of robots' coordi-

nate frames. Frame alignment is further complicated by the fact that robot localization is susceptible to drift if no global information is available [4] (e.g., from GPS). Because of drift, the true alignment between two frames may be constantly changing, so collaborative tasks require frame alignment estimates to be continuously updated so that time-sensitive information can be communicated in real-time.

This thesis introduces a method to iteratively correct frame alignment as drift is accumulated in local frames by creating object-level maps of the environment and aligning collections of recently observed objects. Additionally, we introduce an approach that considers multiple potential map alignments to perform frame alignment when no initial localization information is available. We demonstrate the use of these frame alignment methods in a system for performing collaborative multiple object tracking (MOT), the task of estimating the locations of dynamic objects in a scene.

The challenges involved in performing frame alignment for collaborative robotics are discussed in Section 1.2. Section 1.3 outlines the contributions of this thesis, Section 1.4 lists relevant publications, and Section 1.5 gives an overview of the remaining chapters.

1.2 Problem Statement

In mobile robotics, estimating ego state, including position and rotation, is necessary for fundamental tasks like path following and obstacle location estimation. Local state estimation approaches vary depending on available sensors, but state-of-the-art methods commonly fuse noisy IMU acceleration and angular rate measurements with wheel odometry, visual odometry, or lidar odometry. Local state estimation methods generally use the robot's 'wake-up' pose (i.e., initial position and rotation) as the origin of the world frame, $\mathcal{F}_{\text{world}}$. Then, small incremental changes in the vehicle's position and rotation are estimated and chained together over time to estimate the vehicle's current pose [5]. This incremental method is known as *dead reckoning*, and in the absence of global pose information (e.g., from GPS or a motion

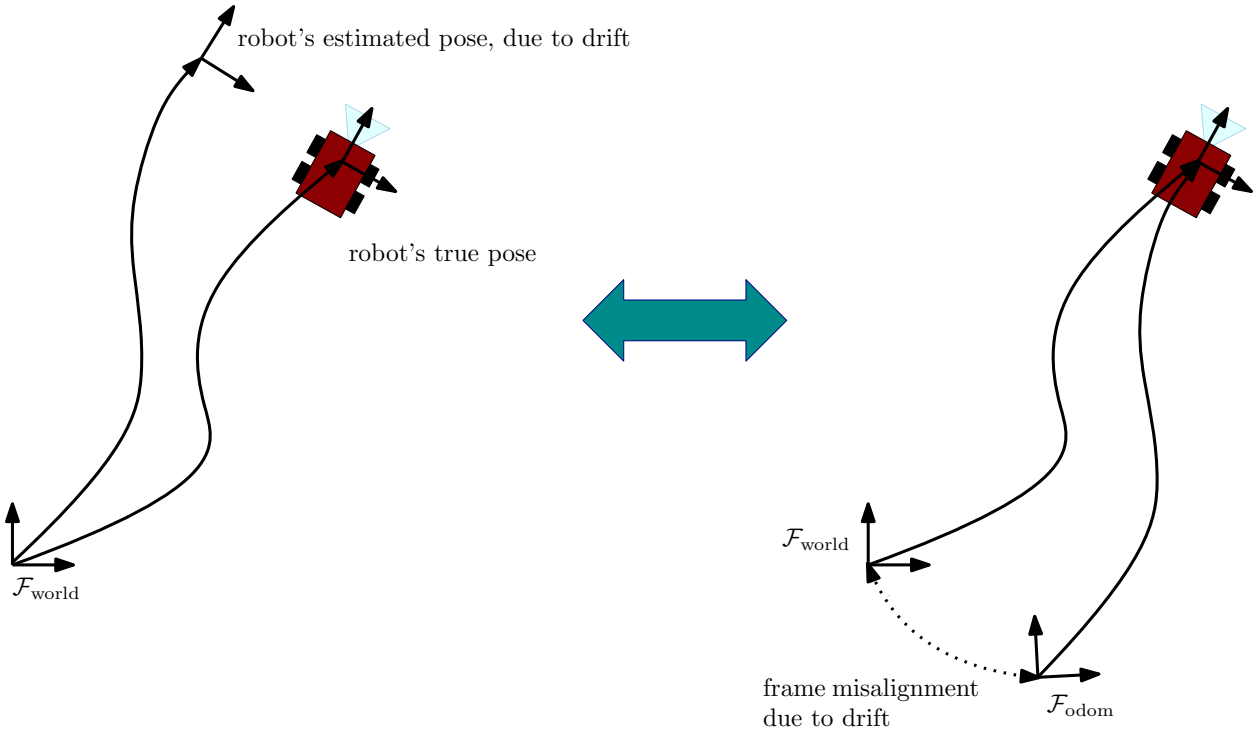


Figure 1.1: Two alternative views of drift. Drift can be visualized as causing errors in a robot's pose estimate in the world frame (left) and it can be visualized as causing a frame misalignment between the world frame $\mathcal{F}_{\text{world}}$ and the robot's odometry frame $\mathcal{F}_{\text{odom}}$.

capture system), small incremental errors in odometry chain together to grow into larger and larger errors, a phenomenon that is called *drift* and is visualized in Fig. 1.1.

Drift is essentially a frame alignment problem—while a robot's local odometry frame initially corresponds with the world frame, as drift accumulates, the odometry frame develops errors separating it from the world frame. The classic approach to eliminating drift is to perform simultaneous localization and mapping (SLAM) [6]. SLAM attempts to minimize the effects of drift by using measurements between the robots' pose and landmarks in the world (e.g., pixel location of visual features or 3D position of LiDAR features) to inform robot state estimation. SLAM further introduces the concept of *loop closure*, which is an instantaneous frame alignment that involves recognizing when a robot revisits a previously perceived location, allowing its state estimate to snap back into place and mitigate the effects of drift. In visual SLAM, loop closures are typically found using visual place recognition (VPR) technology [7] by attempting to match the currently observed image with a set of

previously seen images.

In this thesis, we are primarily concerned with relating quantities (e.g., planned trajectories or perceived object locations) between multiple robots' drifted odometry frames. To accurately share geometric information, frame alignment must be constantly updated to overcome the effects of drift, Performing frame alignment between two different robots' frames is particularly difficult since multi-robot scenarios often do not offer any prior information about the locations of other robots, meaning incorrect frame alignments are difficult to recognize and reject. Additionally, solving collaborative SLAM problems involves solving a large optimization problem and which can require communicating large amounts of data, making collaborative SLAM problems difficult to solve quickly. This means we need something more than collaborative SLAM for robots to communicate about time-sensitive and dynamic elements of collaborative tasks in real-time. Furthermore, VPR for loop closure tends to fail when a scene is viewed from large viewpoint differences.

To this end, this thesis seeks to develop frame alignment strategies that enable collaborative multi-robot tasks by:

- Performing frame alignment in real-time, thus mitigating the effects of drift when communicating about dynamic information.
- Operating without depending on initial frame alignment information.
- Aligning frames in difficult scenarios, in particular instances when robots perceive a scene from opposite views.
- Incorporating uncertainty associated with frame alignment when sharing information with neighboring robots.

1.3 Contributions

An overview of the main technical contributions of this thesis are now given. The main objective of this thesis is to build algorithms for performing frame alignment between pairs of robots enabling the accurate communication of geometric information used in real-time, collaborative tasks. The key principle that guides these algorithms is that frame alignment can be computed by matching the perceived geometry of an environment between two robots' distinct maps. Additionally, this thesis asserts that coarse abstractions of both static and dynamic objects in the world can be used to create sparse and low-size world representations for performing frame alignment accurately.

1.3.1 Contribution 1: Collaborative MOT Enabled by Incremental Frame Alignment Corrections

Multi-Object Tracking with Localization Error Elimination (MOTLEE) [8] addresses the problem of performing frame alignment in real-time to mitigate the effects of drift for collaborative object tracking. In this contribution, we give a complete, distributed system for incrementally realigning drifting frames and performing collaborative MOT. Frame alignment is performed by creating maps of static objects in the environment and aligning these maps using the iterative closest point (ICP) method [9]. Additionally, we show that given an initial frame alignment with low enough errors, tracked dynamic objects (e.g., pedestrians) can be used to assist in frame alignment. Finally, we use our frame alignment method to properly account for measurement uncertainty due to localization and frame alignment errors, preventing failed data association and over-confident estimation.

1.3.2 Contribution 2: Frame Alignment without an Initial Guess Leveraging Temporal Consistency

This contribution [10] relaxes the assumption that robots begin with initial frame alignment knowledge. To accomplish this, robots use open-set image segmentation [11] to map generic objects (i.e., objects that are not recognized using a detector pre-trained on specific classes of objects) in their environment. Our algorithm, Temporally Consistent Alignment of Frames Filter (TCAFF), extracts multiple likely frame alignments from aligning mapped objects leveraging a no-initial-guess data association algorithm [12]. A frame alignment estimate is only made when a series of alignments over time demonstrate high temporal consistency. Using this method, incorrect alignments are rejected, and robots are able to align frames in challenging scenarios in which no initial alignment information is available.

1.4 Related Publications

This work of this thesis is based on the following publications:

- M. B. Peterson, P. C. Lusk, and J. P. How, "Motlee: Distributed mobile multi-object tracking with localization error elimination," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 719–726
- M. B. Peterson, P. C. Lusk, A. Avila, and J. P. How, "Motlee: Collaborative multi-object tracking using temporal consistency for neighboring robot frame alignment," *arXiv preprint arXiv:2405.05210*, 2024
- K. Kondo, C. T. Tewari, M. B. Peterson, A. Thomas, J. Kinnari, A. Tagliabue, and J. P. How, "Puma: Fully decentralized uncertainty-aware multiagent trajectory planner with real-time image segmentation-based frame alignment," *arXiv preprint arXiv:2311.03655*, 2023

1.5 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 gives a review of related work in multi-robot localization and collaborative object tracking. Contribution 1, collaborative MOT enabled by incremental frame alignment corrections, is detailed in Chapter 3. Contribution 2, frame alignment without an initial guess using temporal consistency is described in Chapter 4. Finally, Chapter 5 concludes this thesis and gives directions for future work.

Chapter 2

Related Work

This thesis is primarily concerned with performing frame alignment in collaborative multi-robot tasks, in particular, collaborative MOT. In this chapter, we review the literature on these topics. In Section 2.1, we discuss work related to first performing single robot visual localization and then performing multi-robot localization including visual frame alignment. Section 2.2 gives an overview of work in MOT, in particular, works addressing the additional challenges of fusing information from collaborating agents.

2.1 Visual Localization

Often detached from the MOT community, SLAM approaches the problem of robot localization by building a map of the robot's static surroundings and then using mapped elements to estimate the motion of the robot [4]. SLAM approaches are often differentiated by map representation used by the algorithm. Maps can be represented by dense voxel or surfel maps [14], 3D Gaussian splats [15], sparse visual-feature-based landmarks [16], [17], abstract objects [18]–[22], hierarchical scene graphs [23], or implicit neural radiance fields (NERFs) [24]. While dense SLAM representations can often produce detailed maps, the computation and memory burden of these methods is often ill-suited for platforms with low

compute or bandwidth constraints when collaboratively mapping.

A key component of SLAM that differentiates it from visual intertial odometry (VIO) is the concept of loop closures or frame alignment. Loop closure involves first recognizing that a robot is perceiving a scene it has already visited and then finding the transformation that relates the robot's current pose to its pose when the scene was first observed. Visual loop closure is enabled by VPR technology [7], which determines whether scenes captured in two images overlap by finding images with highly similar visual features [16], [17]. Single-agent loop closure detection benefits from the ability to validate loop closures by checking whether a robot's estimated current location geometrically agrees with the pose of the robot associated with the original image of the scene. If the robot is confident that these two matched locations are in fact very far away, the robot can use this information to reject this loop closure as incorrect.

Unlike single-agent SLAM, in collaborative SLAM (CSLAM), robots often have no prior information about the locations of other robots. This makes the problem of loop closure detection and verification much more challenging. To overcome this hurdle, recent CSLAM research has focused on developing algorithms for robustly rejecting outlier loop closures based on consistency with odometry measurements [3], [25], [26]. Complimentary to the problem of rejecting incorrect loop closures, CSLAM is also faced with the unsolved challenge of detecting loop closures in visually different scenarios, including matching images that view a scene from very different viewpoints [27], [28].

To this end, this thesis focuses on developing robot frame alignment methods that are viewpoint invariant by creating abstract and sparse representations of environments (e.g., object-level representations) that are not dependent on the viewpoint from which the scene is perceived. Additionally, this thesis proposes a method to reject incorrect frame alignments by measuring the temporal consistency of series of alignments and only accepting alignment sequences with high temporal consistency.

2.2 Multiple Object Tracking

Multiple object tracking (MOT) or multiple target tracking is the dynamic counterpart of SLAM. Where SLAM seeks to map static objects, MOT is the task of monitoring the locations of dynamic objects through time and predicting their future paths. While much of the classic MOT work focused on range and bearing sensors like radar, as image processing techniques improved, a great deal of literature shifted to revolve around the task of tracking dynamic objects seen by a single camera. Motivated by the need for robots to operate autonomously in environments alongside people, much of recent literature focuses specifically on tracking pedestrians [29]. An early work in this field, [30] emphasized that pedestrian motion is tightly influenced by the motion of other nearby people. More recent works like SimFuse [31], Deep-SORT [32], and Centertrack [33] incorporate recent learning advancements to make improvements to trajectory prediction and object-to-measurement data association.

Data association between existing object tracks and newly received measurements remains one of the fundamental challenges of MOT. Cues of data association often include comparing the newly detected position of an object with its predicted location from previously filtered measurements. Data association can be especially challenging in cluttered environments or in scenarios with high noise levels. A classic method for dealing with the data association problem is Multiple Hypothesis Tracking, which was developed and improved in early works including [34], [35]. Multiple Hypothesis Tracking delays making hard data association decisions by considering many different possible associations. These are managed as a tree, with each branch representing a different combination of combined measurements and states. By delaying hard associations until further information makes data association clearer, tracking improvement can be improved greatly at the cost of extra computation. To address the combinatorial nature of creating hypothesis trees, [36] and [37] develop systematic algorithms for empirically shown methods to save computation while still

maintaining the benefits of considering multiple hypotheses. In contrast to these methods that use multiple-hypothesis tracking for performing MOT, in this thesis we show that similar concepts can be used to reject outlier frame alignments when performing relative robot localization. Furthermore, while these works focus on single-agent MOT, the focus of this thesis is on performing tasks like MOT collaboratively with a team of robots.

2.2.1 Collaborative MOT

Collaborative MOT can be centralized [38], [39] or distributed [40]–[44]. In a centralized system, all measurement information is sent from robots to a central server, which fuses the measurement information and handles data association with access to all measurement and track information at once. In distributed systems, agents share measurement and estimation information directly with neighbors in their communication network and track estimates are computed collaboratively and in a distributed way.

Because tracks' estimated state information is often one of the biggest informers in cross-view data association, many works consider performing collaborative MOT with known relative sensor poses. The Multi-Target Multi-Camera Tracking (MTMCT) community often focuses on solving a maximum *a posteriori* optimization problem on a set of recorded videos offline to jointly optimize track estimates over all time [38], [39], [45], [46]. Methods for fusing static sensor information include tracking in the 2D image plane and then associating 2D tracks across views [38], [45], associating detections based on projected 3D locations [42], [46], and using learning-based ReID features [38], [42].

Conversely, in a mobile robot scenario, robots require real-time knowledge about the whereabouts of moving objects in their environment (i.e., optimizing objects' tracks using recorded videos does not suffice). Real-time object tracking onboard moving sensors is commonly approached by incrementally estimating objects' states as new detections are made. Ong et al. [47] proposed a decentralized particle filtering approach for tracking onboard multiple flight vehicles using global localization information from GPS for vehicles. Shorinwa et

al. [40] presented a distributed target tracking method based on consensus ADMM for use on autonomous cars with ground truth localization.

2.2.2 Collaborative MOT with Unknown Localization

Because teams of robots are necessarily going to experience localization drift in their ego-pose estimates, robots cannot accurately share track information with their neighbors in a common coordinate frame without some method to compensate for drift. To address this, recent work has begun to emerge that tackles both simultaneous localization and object tracking [22], [48]–[51]. Tian et al. [22] introduce a method for performing simultaneous SLAM and MOT on mobile agents with LiDAR sensors by performing data association on objects using similarity scoring on a sliding window of object tracks. Ahmad et al. [52] formulated a joint, collaborative MOT and localization problem as a pose graph optimization for robot soccer. They make the limiting assumption of known data association of measurements with static landmarks and tracked dynamic objects. Taghavi et al. [50] introduced a method for estimating static sensor bias in a multisensor MOT framework in a centralized system. Dames [51] proposed a distributed MOT algorithm based on random finite sets (RFS) and the probability hypothesis density (PHD) with the assumption that localization uncertainty is constant everywhere along the robot's path. In contrast, this thesis introduces TCAFF, a multiple hypothesis frame alignment filter, to determine the correct alignment of robot coordinate frames using temporally consistent frame alignment measurements used for communicating information in our distributed MOT system.

Chapter 3

Collaborative Multiple Object Tracking Under Localization Uncertainty

In this chapter, we outline a system for performing collaborative MOT while performing frame alignment to iteratively correct for the effects of drift. First, the objective of frame alignment is described in Section 3.1. We then describe our full MOT system in Section 3.2. This importantly includes a method to incorporate the uncertainty from frame alignment when sharing information with neighbors. In Section 3.3, our iterative frame realignment is described and finally, experimental results are shown in Section 3.4.

3.1 Background

Coordinated, multi-robot tasks require each robot to know the transformations to their neighboring robots' coordinate frames for collaboration about perceived information. Additionally, a robot must estimate its own pose within a local frame, which is usually referred to as the odometry frame. We write the i-th robot's pose in its own odometry frame $\mathcal{F}_{\text{odom}_i}$ at time k as $\mathbf{T}_{\text{r}_i}^{\text{odom}_i}(k)$. Since the robot's pose estimate can be susceptible to drift, we also consider the robot's pose in a world frame, $\mathbf{T}_{\text{r}_i}^{\text{world}}(k)$. The two frames $\mathcal{F}_{\text{odom}_i}$ and $\mathcal{F}_{\text{world}_i}$ are illustrated in Fig. 1.1. Because of drift, the transform between the two coordinate frames,

$\mathbf{T}_{\text{world}}^{\text{odom}_i}(k)$, may not remain static. We formulate frame alignment as the problem of computing the relative transform between two robots' odometry frames $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}$, which can be used to express spatial information in a neighboring robot's frame.

3.2 Collaborative MOT

The objective of MOT is to estimate $\mathbf{x}(k)$, the state of each dynamic object in an environment at time k, using measurements $\mathbf{z}(k)$ and modeled by a discrete-time linear dynamic system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k), \quad \mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k), \tag{3.1}$$

where $\mathbf{A}$ and $\mathbf{H}$ are the transition and measurement matrices, respectively, and $\mathbf{w}(k) \sim \mathcal{N}(0, \mathbf{Q}(k))$ and $\mathbf{v}(k) \sim \mathcal{N}(0, \mathbf{R}(k))$ are zero-mean independent Gaussian process and measurement noise with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Each robot keeps a local bank of state estimates of dynamic objects, $\hat{\mathbf{x}}(k)$, also referred to as *tracks*.

We adopt the distributed track management system of [42] to handle sharing track information between robots, and shared track information is incorporated using the Kalman Consensus Filter (KCF) [53], [54], as described in the remainder of this section.

At each timestep, each robot obtains dynamic object measurements $\mathbf{z}(k)$ from sensory input and then executes three tasks to collaboratively track objects: (1) perform local data association between new measurements and the tracks in robot i's local bank, (2) share updated KCF information with neighboring robots, and (3) perform KCF updates.

3.2.1 Local Data Association

Local Data Association (LDA) is the process of determining whether each measurement should be assigned to an existing track or whether it belongs to a new object that is not being tracked. This association decision is made by evaluating the negative logarithm of the

matching likelihood (NLML) [55] between each measurement $\mathbf{z}(k)$ and each existing track prediction as

$$\text{NLML}(\hat{\mathbf{x}}^+(k), \mathbf{z}(k+1)) = \|\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}^+(k)\|_{\mathbf{S}}^2 + d_{\mathbf{z}} \log 2\pi + \log\left(|\mathbf{S}|\right). \tag{3.2}$$

where $\hat{\mathbf{x}}^+(k) = \mathbf{A}\hat{\mathbf{x}}(k)$, $d_{\mathbf{z}}$ is the dimension of the measurement vector, and $\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{R}$ is the innovation covariance. We solve the LDA using a global nearest neighbor (GNN) approach [1] where the measurement-to-track association is formulated as a linear assignment problem and the NLML of each association is minimized using the Hungarian algorithm [56]. We apply a gate, τ, such that a track $\hat{\mathbf{x}}(k)$ cannot be matched with a $\mathbf{z}(k)$ if $\text{NLML}(\hat{\mathbf{x}}^+(k), \mathbf{z}(k+1)) > \tau$. Any unmatched measurements are set aside as part of a trial database where after ν sequential measurements, the trial track is accepted as a new track.

3.2.2 Information Sharing

Once measurements and tracks have been associated, robot i sends the message containing KCF update information to each neighboring robot j. It is important to note that because the robots' coordinate frames differ, quantities expressed in $\mathcal{F}_{\text{odom}_i}$ must be transformed into $\mathcal{F}_{\text{odom}_j}$ and the uncertainty associated with that transformation must also be reflected in the uncertainty associated with shared measurements and state estimates. We use the notation p_i^j to designate a quantity p that originated from robot i, and is now expressed in robot j's odometry frame $\mathcal{F}_{\text{odom}_j}$ using the estimated frame transformation $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}$. For each track, the information $(\texttt{ID}, \hat{\mathbf{x}}_i^{j+}, \mathbf{u}_i^j, \mathbf{U}_i^j)$ is shared, where $\texttt{ID}$ is a unique ID for the track, $\mathbf{u}$ and $\mathbf{U}$ are the information vector and information matrix associated with the track respectively. The information vector and matrix are formed with

$$\mathbf{u}_i^j = \mathbf{H}^\top (\mathbf{R}_i^j)^{-1} \mathbf{z}_i^j, \quad \mathbf{U}_i^j = \mathbf{H}^\top (\mathbf{R}_i^j)^{-1} \mathbf{H}, \tag{3.3}$$

where $\mathbf{R}_i^j$ is computed by propagating the measurement covariance through the uncertain frame transformation $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}(k)$, which will be discussed in Section 3.2.4.

3.2.3 Kalman Consensus Filter

Once robot i has received messages from each of its neighbors, the information is aggregated as

$$\mathbf{y}^i = \sum_{j \in \mathcal{N}_i \cup i} \mathbf{u}_j^i, \qquad \mathbf{Y}^i = \sum_{j \in \mathcal{N}_i \cup i} \mathbf{U}_j^i, \tag{3.4}$$

where $\mathcal{N}_i$ represents the set of robot i's neighbors. Track estimates are then obtained by fusing together all shared information using the KCF update,

$$\begin{aligned} \hat{\mathbf{x}}^i(k+1) = \hat{\mathbf{x}}_i^{i+}(k) + \mathbf{M}^i(k)\big[\mathbf{y}^i(k) - \mathbf{Y}^i(k)\hat{\mathbf{x}}_i^{i+}(k)\big] \\ + \frac{\mathbf{M}^i(k)}{1 + \|\mathbf{M}^i(k)\|} \sum_{j \in \mathcal{N}_i} \big(\hat{\mathbf{x}}_j^{i+}(k) - \hat{\mathbf{x}}_i^{i+}(k)\big), \end{aligned} \tag{3.5}$$

where $\mathbf{M}^i(k) = (\mathbf{P}^i(k)^{-1} + \mathbf{Y}^i(k))^{-1}$ is the Kalman gain in information form, and $\mathbf{P}^i(k)$ is the estimation covariance. Finally, $\mathbf{P}^i(k)$ is updated with $\mathbf{A}\mathbf{M}^i(k)\mathbf{A}^\top + \mathbf{Q}^i(k)$.

3.2.4 Uncertainty Propagation

To share the measurement $\mathbf{z}_i^i$, robot i must transform $\mathbf{z}_i^i$ into the frame $\mathcal{F}_{\text{odom}_j}$, and the frame alignment uncertainty $\mathbf{P_s}$ must be properly incorporated with the initial measurement uncertainty $\mathbf{R}_i^i$. We parameterize the frame alignment and its associated uncertainty using Euler angles, following Smith, Self, and Cheeseman (SSC) [57]. Then, $\mathbf{R}_i^j$ can be found by combining the uncertainty in the object measurement and the frame alignment with

$$\mathbf{R}_i^j = \mathbf{I}_{d_\mathbf{z} \times d_\mathbf{y}} \mathbf{J} \mathbf{P_s} \mathbf{J}^\top \mathbf{I}_{d_\mathbf{z} \times d_\mathbf{y}}^\top + \mathbf{C}_{\text{odom}_j}^{\text{odom}_i} \mathbf{R}_i^i (\mathbf{C}_{\text{odom}_j}^{\text{odom}_i})^\top. \tag{3.6}$$

where $\mathbf{J}$ is the left half of the Jacobian of the compounding (relative head-to-tail) relation, $\mathbf{P_s}$ is the covariance of the current frame alignment estimate, and $\mathbf{C}_{\text{odom}_j}^{\text{odom}_i}$ is the right half of

the Jacobian of the compounding relation, which is also the rotation matrix component of the frame alignment estimate.

3.3 Incremental Frame Alignment

In this section, we present a distributed method for eliminating error between robot frames using local maps of static landmarks and past dynamic object detections. Aligning these values, allows robots to compute an estimate $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k)$, correcting for drift in real-time.

3.3.1 Realignment with Static Landmarks

The key concept is that frame alignment drift can be accounted for and eliminated by aligning objects observed by pairs of robots. Because drift can accumulate and change quickly, our method aims to promptly correct for drift online by creating small, local maps of recently observed static landmarks. With two local maps in hand, performing frame realignment with each neighbor involves three steps: first, associating landmarks between maps; second, applying a weighting to pairs of landmarks; and finally, performing point registration to compute $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k)$.

To perform data association between two maps, we use iterative closest point (ICP) registration [9]. As ICP is a local method that requires a good initial guess, we use the previous estimate $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k-1)$ as a starting solution. Weights are applied to the associations found by ICP using the following weighting function to prioritize recent detections over old detections,

$$W(\ell_i, \ell_j) = (\ell_i \ell_j)^{-1}, \tag{3.7}$$

where a larger weight corresponds to a greater influence in the point registration and ℓ_i and ℓ_j are the number of frames since the corresponding static landmarks were last detected by robots i or j respectively. This is done to help account for drift that may have accumulated in older parts of the static local map.

The final step of the algorithm is to compute $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k)$ using Arun's method [58] on the associated landmarks and respective weights. The transformation $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k)$ should be applied to all outgoing detections sent to robot j to place the measurements properly in robot j's frame.

3.3.2 Realignment with Tracked Objects

When a large enough number of dynamic objects are co-visible, robots can use these already tracked objects and their measurements to perform frame alignment. To determine whether dynamic object detections can be used to perform frame realignment, we define η to be the number of concurrent, same-object detections between two agents and τ_η to be a threshold such that frame realignment is only performed with dynamic object detections when $\eta \geq \tau_\eta$. If $\eta < \tau_\eta$, then, static landmark realignment should be performed. Thus, with enough co-visibility for robots i and j to achieve a large η, determining the realignment transformation can be done without performing any additional mapping and by only exchanging information about tracked objects.

One important difference between aligning static landmarks and tracked object measurements is that measurements from tracked objects have already been associated during the KCF information exchange. So, given two pairs of already-associated dynamic object measurements, $\mathcal{Z}^i_i$ and $\mathcal{Z}^i_j$, we only need to find all pairs of detections $\mathbf{z}^i_i(k)$, $\mathbf{z}^i_j(k)$ that occurred at the same time k to perform realignment.

To realign frames with greater accuracy, we assign pairs of detections a weight that reflects the knowledge the agents have about the accuracy of the detection. We define the weighting function

$$W(\hat{\mathbf{x}}^i_i, \mathbf{z}^i_i, \mathbf{z}^i_j) = ((\mathbf{H}\hat{\mathbf{x}}^i_i - \mathbf{z}^i_i)^\top (\mathbf{H}\hat{\mathbf{x}}^i_i - \mathbf{z}^i_j))^{-1}, \tag{3.8}$$

which prioritizes aligning pairs of detections where both are consistent with the estimated state of the object, thus rejecting noisy detections and even detections that may have been

associated incorrectly.

Once this step has been performed, a transformation is found using Arun's method for registration [58]. Since aligned measurements $\mathbf{z}_i^i(k)$, $\mathbf{z}_j^i(k)$ were already placed into robot i's odometry frame, alignment of these measurements results in an intermediate correction transformation, $\mathbf{T}_{\text{realign}}$. So, to conclude frame realignment, $\hat{\mathbf{T}}_{\text{odom}_j}^{\text{odom}_i}(k)$ is updated with $\hat{\mathbf{T}}_{\text{odom}_j}^{\text{odom}_i}(k) = \mathbf{T}_{\text{realign}}\hat{\mathbf{T}}_{\text{odom}_j}^{\text{odom}_i}(k-1)$.

One problem that may arise with this algorithm is that it assumes that frame alignment between two robots is accurate enough that robots can associate detections of common objects. This will not be the case under severe frame misalignment. To address this issue and to give robots the ability to make correct data associations when frame error is large, we make τ (see Section 3.2.1) reactive to the amount of detected frame misalignment, indicated by the magnitude of $\mathbf{T}_{\text{realign}}$. The data association tolerance τ is scaled up when frame realignment yields a large $\mathbf{T}_{\text{realign}}$ and τ is returned back to its original value when frame realignment begins to yield a small $\mathbf{T}_{\text{realign}}$.

3.4 Experiments

We evaluate our distributed mobile MOT system using two self-collected datasets generated using a team of ground robots in a $10 \times 10\,\text{m}$ room. Each robot was equipped with an Intel RealSense T265 Tracking Camera and an Intel RealSense L515 LiDAR Camera. A VICON motion capture system was used to collect ground truth pose information of robots, moving objects (pedestrians), and static landmarks (cones). For simplicity, we assume objects evolve according to (3.1) with a constant velocity motion model—thus the state of an object is defined to be $\mathbf{x}(k) \overset{\text{def}}{=} [p_x, p_y, v_x, v_y]^\top \in \mathbb{R}^4$. Each robot makes noisy 2D position observations in the ground plane so that the measurement model is defined as $\mathbf{z}(k) \overset{\text{def}}{=} [\tilde{p}_x, \tilde{p}_y]^\top \in \mathbb{R}^2$, where object measurements are made by processing CenterTrack [59] trained on the JRDB dataset [60], [61] on T265 left fisheye images. Although CenterTrack additionally provides

Figure 3.1: MOT hardware experiment in our motion capture room. We use this setup to demonstrate our MOTLEE algorithm's ability to perform distributed MOT onboard moving robots with localization uncertainty.

IDs associated with detections (i.e., the CenterTrack network attempts to infer data association), we find these to be too noisy for practical use.

In addition to providing images used for person detection, the T265 stereo camera provides each robot with ego-motion estimation based on stereo visual odometry. Although each robot begins with knowledge of relative frame alignments (e.g., by initializing $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k_0) = \mathbf{T}^{\text{odom}_i}_{\text{odom}_j}(k_0)$), odometry drift quickly grows resulting in frame misalignments that cause failure of distributed data association and fusion of incoherent data in the KCF. Using our MOTLEE algorithm, we address these challenges by performing frame alignment and appropriately incorporating odometry and frame alignment uncertainty into shared measurement uncertainty.

We use MOTA [62] as the performance metric, given by

$$\text{MOTA} = 1 - \frac{\Sigma_k(m(k) + fp(k) + mme(k))}{\Sigma_k g(k)}, \tag{3.9}$$

where $m(k)$ is the number of ground truth objects missed in frame k, $fp(k)$ is the number of false positives (i.e., perceived objects that are not found in frame k), $mme(k)$ is the number of mismatches, or objects that were reported under a different ID at time k than the ID previously used for that object, and $g(k)$ is the number of ground-truth objects in the scene. The components of MOTA are highly sensitive to the system's ability to correctly perform data association and report the correct location of objects. This makes it a good candidate for measuring the effects of frame misalignment which causes data association errors and corrupts shared object detection measurements.

3.4.1 Effects of Localization Error

We now demonstrate the performance of MOTLEE's resilience to frame misalignment and localization error. First, we present our results in the context of a dataset taken with 5 pedestrians walking around our motion capture room with 4 stationary robots located around the perimeter. We use this dataset to artificially alter the localization error of each robot by initializing robot frame alignments with incorrect $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k_0)$. In this way, we isolate the effects of localization error directly, without also introducing other error associated with performing mobile MOT, e.g., poor object detection from motion blur.

To represent the effects of inter-robot frame misalignment, artificial error is introduced into the system by adding a random, constant bias to the robot pose estimates at the start of a run by initializing $\hat{\mathbf{T}}^{\text{odom}_i}_{\text{odom}_j}(k_0) = \mathbf{T}_{\text{error}}\mathbf{T}^{\text{odom}_i}_{\text{odom}_j}(k_0)$, where $\mathbf{T}_{\text{error}}$ is composed of random heading error $\theta_{\text{error}} \sim \mathcal{N}(0, \sigma_\theta)$ and random translation error in the x, y plane with magnitude $t_{\text{error}} \sim \mathcal{N}(0, \sigma_t)$. We then capture the performance of the system for varying levels of σ_t and σ_θ representing varying levels of localization uncertainty. We correlate the standard deviations σ_θ and σ_t at a ratio of 8.12 deg heading error per 1 m translation error. This ratio is the amount of heading error that would produce 1 m of error in the estimated position of a tracked object that is a distance of 7 m away from the robot, the approximate average distance between tracked objects and each robot in this dataset. To produce the following

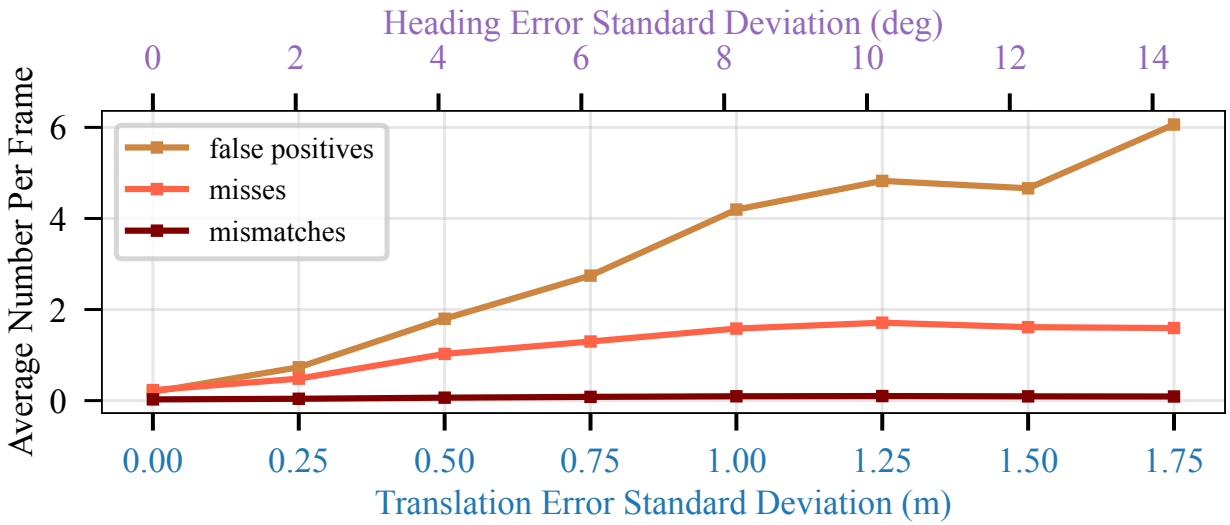


Figure 3.2: Performance of distributed MOT from Casao et al. [42] under varying levels of artificially introduced localization uncertainty. Each of the plotted values directly affects tracking accuracy as defined by MOTA (see (3.9)). Without accounting for localization uncertainty, tracking accuracy degrades as error increases. Misses and particularly false positives are the dominating part of the degradation in performance. False positive tracks occur because each robot within the multi-view system shares incorrect information about tracked objects, causing the network to report many different tracks of objects that do not exist.

results, we run our full dataset through our MOT framework over 5 different samples at each uncertainty level to represent the average performance of the system as error is introduced.

Fig. 3.2 and Fig. 3.3 show studies on the isolated effects of localization error. Fig. 3.2 decomposes the different pieces of MOTA and shows how misses and particularly false positives break a system's MOT performance as localization error is introduced. Fig. 3.3 demonstrates that while Casao et al.'s [42] MOT algorithm rapidly breaks down with frame alignment error in the system, our MOTLEE algorithm is robust to localization error. Frame realignment is performed here by only aligning detections of dynamic objects.

3.4.2 Mobile Experiment

In this section, we evaluate MOTLEE using a team of three mobile ground robots moving along pre-determined trajectories while four pedestrians walk among the robots. By nature

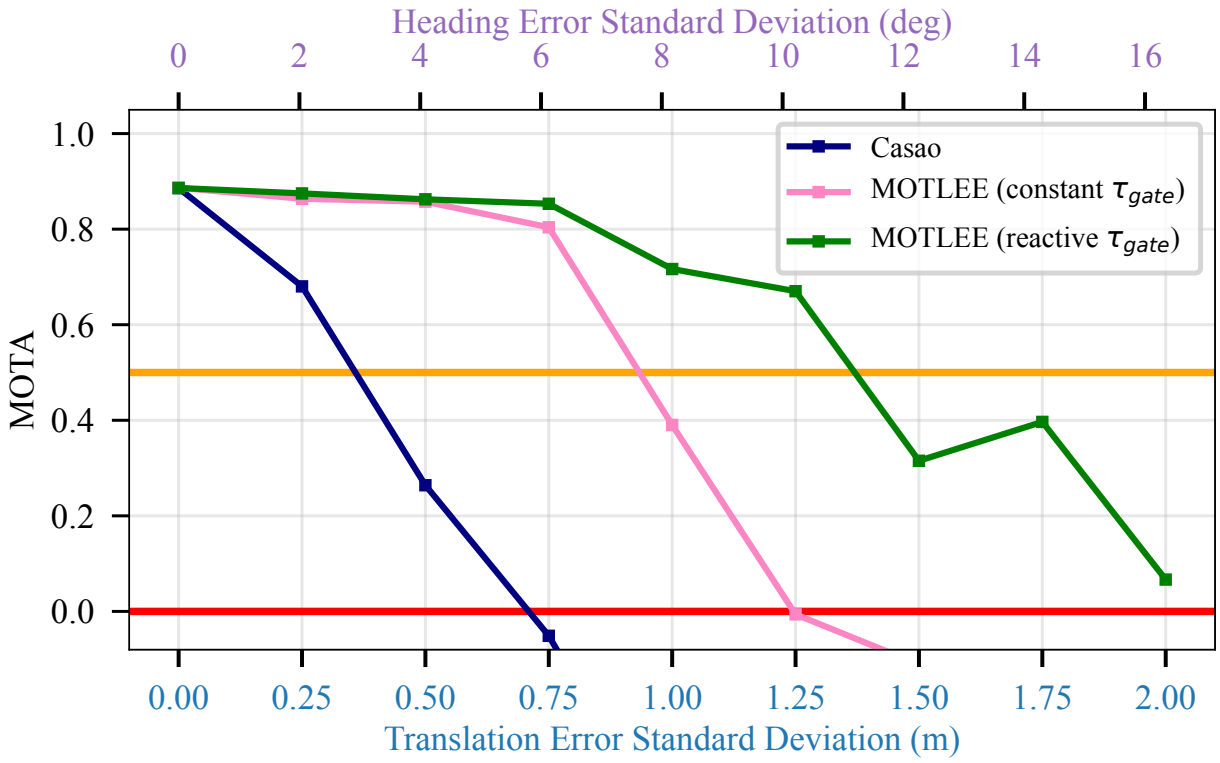


Figure 3.3: MOT performance results as localization error is introduced. Two lines at MOTA = 0.5 and MOTA = 0.0 are given as references for understanding the performance degradation. An example scenario that would earn a MOTA of 0.5 is tracking only half of the visible objects and missing the other half. For a MOTA of 0.0, a system could similarly miss half of all visible objects but additionally predict that each of those objects are in some other location. MOT performance rapidly degrades with the introduction of localization error (navy line). Performing realignment with dynamic object tracks (pink line) realigns frames successfully in low-uncertainty regimes, and performing realignment with a reactive data association tolerance τ_{gate} (green line) makes the system robust to even greater amounts of error.

of the small room that the experiment is run in, each robot follows a circular path in its assigned region of the room. Because of this, cameras have limited co-visibility of objects for approximately half of the time while they face the walls. In this experiment, there are too few shared dynamic detections to be used for frame realignment ($\eta < \tau_\eta$), so static landmarks are used to realign frames. These landmarks are detected and mapped by each robot using the L515 camera. Detections of cones are made using YOLOv7 [63] with weights from [64]. The corresponding 3D points within the 2D bounding box are then identified by color thresholding and a 3D cone position is estimated using the median of these points.

Cone detections are accumulated into local maps using the robot's odometry combined with 3D cone detections to chain pairwise associations. Local maps are shared with neighbors at a frequency of 1 Hz, and frame alignment is performed using these local maps as they arrive. We represent the frame alignment uncertainty, $\mathbf{P_s}$, as a diagonal covariance matrix with elements determined by a linear scale of the difference between the current and most recent frame alignments.

Fig. 3.4 shows that using only noisy odometry readings without realignment only works well while localization error is small, which only occurs at the beginning of the run. However, as more error is accumulated in each of the robot's frame alignments, $\hat{\mathbf{T}}_{\text{odom}_j}^{\text{odom}_i}(k)$, the system fails to perform collaborative object tracking accurately. In contrast, when using our MOTLEE framework for realignment, we are able to perform near the level of the system using ground truth localization. We achieve an average MOTA of 0.724, similar to the ground truth performance of 0.756, while Casao et al. [42] breaks down due to the static camera assumption and scores a MOTA of 0.141.

We also show the error between $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}(k)$ and $\hat{\mathbf{T}}_{\text{odom}_j}^{\text{odom}_i}(k)$ in Fig. 3.5. With our MOTLEE frame realignment, we get a median error of 1.83 deg heading error and 0.22 m translation error. Although we use the recorded data to run the system offline so that we can compare different localization scenarios, our algorithm can easily be run in real-time with each tracking and mapping cycle taking an average of 7.1 ms with a standard deviation of 2.9 ms and each frame alignment cycle taking an average of 76.7 ms with a standard deviation of 23.9 ms.

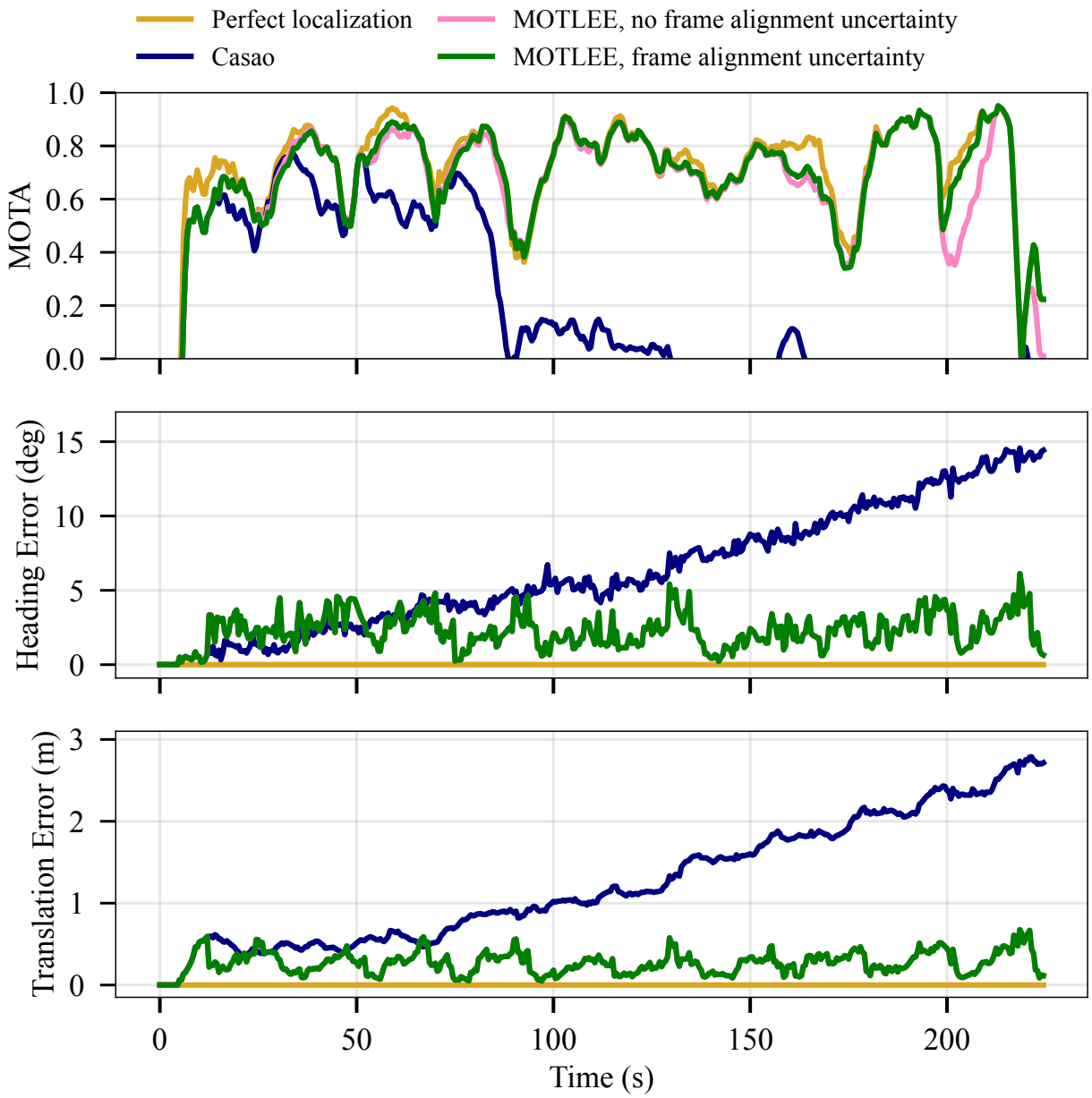


Figure 3.4: Four-minute long mobile MOT test results. MOTA is usually normalized by the sum of the number of objects in each frame over a whole run. However, to show the evolution of tracking over time, the MOTA score is shown over a sliding window of 10 seconds. Casao et al.'s algorithm [42], which does nothing to correct for localization error, quickly degrades as drift accumulates in the robots. We show that MOTLEE demonstrates improved robustness to frame misalignment and is able to achieve results similar to those of a system with ground truth localization. We also show that by incorporating frame alignment uncertainty into exchanged information in the KCF (green line), MOTLEE achieves a higher MOTA than when this uncertainty is ignored (pink line) even though frame alignment results do not change (the pink and green lines are on top of each other in the heading and translation error plots).

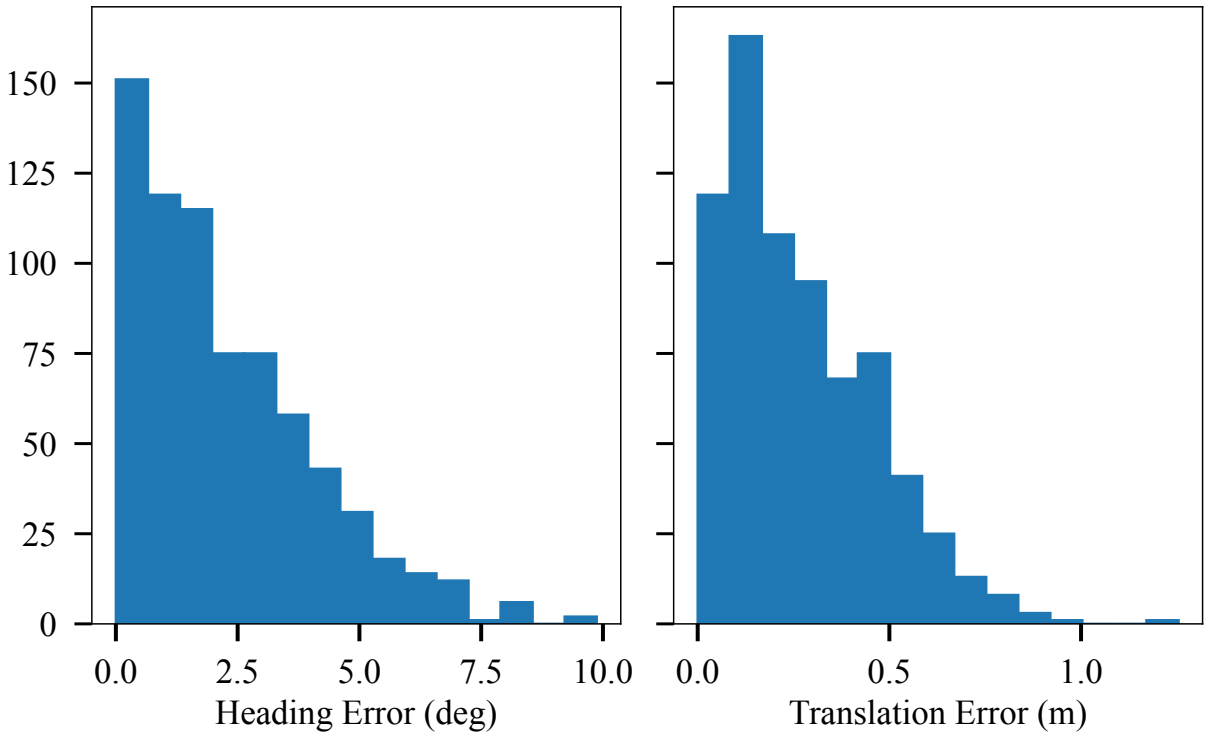


Figure 3.5: Histogram of error in frame realignment estimates during mobile MOT experiment.

Chapter 4

Frame Alignment without an Initial Guess

In chapter 3, we addressed the problem of iteratively correcting for frame alignment drift. In this chapter, we develop a method for performing frame alignment when no initial information on relative robot localization is available. This makes the problem much more challenging as the frame alignment problem must often deal with ambiguities—a potential frame alignment may look convincingly correct; however, geometric or visual aliasing may be present, and so a mechanism for rejecting incorrect potential alignments must be developed.

Section 4.1 presents our method for aligning robot frames without an initial guess by searching for sequences of temporally adjacent alignments with high consistency. Experimental results demonstrating our algorithm's ability to find frame alignments in challenging scenarios are then given in Section 4.2.

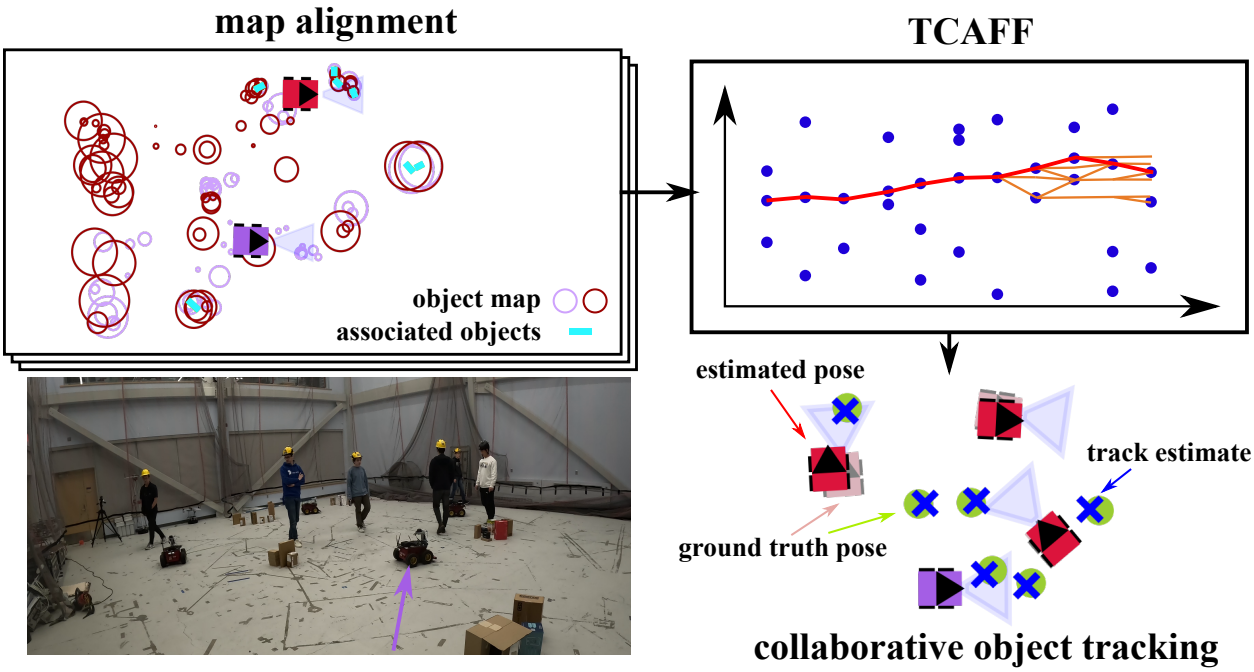


Figure 4.1: To perform collaborative object tracking with a team of mobile robots (bottom right), potential frame alignments are found by aligning maps of objects (top left) and filtered using a multiple-hypothesis-inspired frame alignment filter (top right). Relative robot poses are shown from the point of view of the purple robot.

4.1 Temporally Consistent Alignment of Frames Filter (TCAFF)

TCAFF, our method for no-initial-guess frame alignment, is depicted in Fig. 4.1. Robots create open-set object maps and then use TCAFF to filter potential alignments using temporal consistency. Once a frame alignment has been found, robots can communicate in collaborative tasks, like MOT.

4.1.1 Map Alignment

We perform frame alignment, finding $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}(k)$, by creating sparse maps of recently observed objects in the environment and then aligning the maps of pairs of robots as shown in Fig. 4.1. The map of robot i is denoted as $\mathcal{M}_i$ and is composed of objects represented by their width

w, height h, time since the object was last seen ℓ, and centroid position p^{odom_i} expressed in $\mathcal{F}_{\text{odom}_i}$. To ensure that drifted parts of robot i's map do not affect the estimated $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}(k)$, an object is only included in $\mathcal{M}_i$ if $\ell < \kappa$, where κ is a tuneable parameter based on how fast drift is expected to accumulate.

A long-standing problem with object-based mapping in robotics is that traditionally image detection networks must be trained to detect certain objects [18]–[20]. This can require significant overhead and restricts the mapping techniques to functioning successfully only in environments where those specific objects are present in sufficient numbers. We follow the works of [13], [65] in using the open-set image detector FastSAM [11] to segment open-set objects in real-time and use these generic object measurements to create object maps.

Maps of recently observed objects are shared with neighboring robots at a regular rate, and frame alignment is performed by treating the alignment of centroids as a point registration problem. As shown in Algorithm 1, we make modifications to the standard CLIPPER method [12], [66] to perform robust global data association between points. CLIPPER solves point registration by formulating the problem as a graph optimization and leveraging geometric consistency to reject outlier associations. First, a consistency graph $\mathcal{G}$ is formed where the nodes a_p are putative associations between a point in the first map p_i and a point in the second p_j. Weighted edges exist between nodes if two associations are consistent with each other. This is determined by using a distance measurement, $d(a_p, a_q) = |\,\|p_i - q_i\| - \|p_j - q_j\|\,|$. If $d(a_p, a_q) < \epsilon$, a weighted edge is added to the graph $\mathcal{E}_{p,q} = s(a_p, a_q)$, where s is a function that maps the similarity of two associations to a score $\in [0, 1]$. Finally, the edges of the graph are used to create a weighted affinity matrix $\mathbf{M}$ where $\mathbf{M}_{p,q} = s(a_p, a_q)$, and a continuous relaxation of the following problem is optimized

$$\begin{aligned} &\max_{\mathbf{u}\in\{0,1\}^n} \frac{\mathbf{u}^\top \mathbf{M}\mathbf{u}}{\mathbf{u}^\top \mathbf{u}}. \\ &\text{subject to} \quad u_p u_q = 0 \text{ if } \mathbf{M}_{p,q} = 0,\ \forall_{p,q}, \end{aligned} \tag{4.1}$$

where the elements of $\mathbf{u}$ indicate whether an association has been kept as an inlier association. See [12] for more details.

By creating a point cloud from centroids of mapped objects, CLIPPER can be leveraged to form associations between objects within each map, as done in [13], [65], without any initial frame alignment knowledge. We form the initial putative associations given to CLIPPER by only including correspondences that would associate objects with similar widths and heights in both maps to help keep the optimization small enough to be solved in real-time. After objects in each map have been associated, we align the two maps using a weighted Arun's method [58], where weights are assigned between two objects $W(\mathbf{o}_i, \mathbf{o}_j) = (\ell_i \ell_j)^{-1}$ where ℓ is the time since $\mathbf{o}$ was last seen. This allows objects that were seen more recently (i.e., can give more up-to-date information about relative robot pose) greater influence in the point registration.

4.1.2 Near Optimal Associations

In practice, these sparse maps often include ambiguities in how they should be aligned due to geometric aliasing (i.e., repetitive object structure), difficulty in representing the object's true centroid due to occlusion and partial observations, small overlap between the two maps, and high noise level from the coarseness of the map representation. These ambiguities can lead to registration problems that have many local optima whose objective values are numerically similar or even cases where the global optimum does not necessarily correspond to correct data associations. Additionally, an alignment between two object maps can still be found even if the maps belong to non-overlapping areas, so a method for rejecting incorrect frame alignments is needed. A heuristic approach can be taken, such as requiring a certain number of associated objects to consider a frame alignment [67], but this requires making assumptions about the expected abundance of objects to be mapped in the environment and may still result in finding incorrect frame alignments or rejecting correct alignments that do not meet the minimum number of associations.

Instead, we propose a multi-hypothesis approach to address these ambiguities. Our method considers different possible alignments of object maps and finds the most likely alignment leveraging consistency of alignments over time. The first step in this process is finding object-to-object associations that are correct but that may be sub-optimal according to (4.1). To do this, we develop an algorithm to extract multiple near optima from CLIPPER, shown in Algorithm 1.

Algorithm 1 Multiple Near Optima (MNO) CLIPPER

Input affinity matrix $\mathbf{M} \in [0,1]^{m \times m}$ of consistency graph $\mathcal{G}$
Output $\mathcal{Y}$ Set of near-optimal transformation meas. of $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}$
$\mathcal{Y} = \{\}$
for $n \in 1 : N$ **do**
    `inlier_associations` $\leftarrow$ CLIPPER($\mathbf{M}$)
    $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i} \leftarrow$ `aruns_method(inlier_associations)`
    $\mathcal{Y} \leftarrow \mathcal{Y} + \{\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}\}$
    for $p \in$ `inlier_associations` **do**
        for $q \in$ `inlier_associations` **do**
            $\mathbf{M}_{p,q} \leftarrow 0$

To search for nearly optimal associations, we initially run the standard CLIPPER, which gives a single set of associations which is often the globally optimal solution to equation (4.1). Then, elements of the affinity matrix $\mathbf{M}$ that were selected by the previous solution are set to 0 to force CLIPPER to find a new set of associations that does not include any of the previously selected associations, which thus yields a new set of nearly optimal associations. This is repeated for a set number of iterations, N.

4.1.3 Multiple Hypothesis Formulation

We construct TCAFF, a frame alignment filter for finding map alignments that are consistent over time. This formulation considers different associations of incoming frame alignment measurements and is inspired by multiple hypothesis tracking (MHT) [34], [36] which is typically used for assisting with the data association of object detections to tracked object states in the MOT community. Instead, we apply similar concepts to determine when a

sequence of frame alignments represents correctly associated map objects between a pair of robots. The goal of TCAFF is at each timestep to take several potential frame alignments, each referred to as frame alignment measurement $\mathbf{y}_k \in \mathcal{Y}_k$, and produce a filtered frame alignment estimate $\mathbf{s}_k$.

For simplicity, we first assume an initial frame alignment guess, $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}(k_0)$. At each timestep k, robot i gets many potential frame alignments from MNO-CLIPPER. However, because of map alignment ambiguity or lack of overlapping information (i.e., robots i and j are not mapping any common objects), there exists only one or zero correct frame alignment measurements. By observing these measurements over time, the most likely sequence corresponding to the true frame alignment can be determined. The selected measurements are then filtered to obtain an accurate frame alignment estimate.

This problem can be expressed as a more general maximum *a posteriori* (MAP) estimation problem of selecting the measurement variables that best fit the model

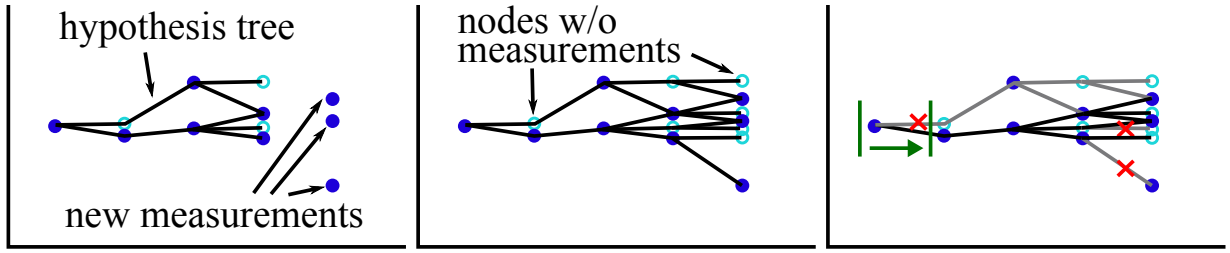
$$\begin{aligned} \underset{\mathbf{y}_1 \in \mathcal{Y}_1, \ldots, \mathbf{y}_K \in \mathcal{Y}_K}{\arg\max} \quad & p\left(\mathbf{s}_K \middle| \mathbf{y}_1, \ldots, \mathbf{y}_K\right) \\ \text{s.t.} \quad & \mathbf{s}_0 = \mathbf{s}(k_0), \\ & \mathbf{s}_{k+1} = \texttt{kalman\_update}(\mathbf{s}_k, \mathbf{y}_k), \end{aligned} \tag{4.2}$$

This formulation can be used in other applications where temporal consistency can be used to give extra information in ambiguous scenarios, but for our specific use case of estimating a frame alignment, we use $\mathbf{s}_k$ and $\mathbf{y}_k$ as parameterizations of 2D frame alignments $\mathbf{T}_{\text{odom}_j}^{\text{odom}_i}$ where $\mathbf{s}_k = [x, y, \theta]^\top$.

We represent $\mathbf{s}_k$ and $\mathbf{y}_k$ as Gaussian random variables, which allows us to rewrite equation (4.2) as

$$\underset{\mathbf{y}_{1:K}}{\arg\max} \quad \prod_{k=1}^{K} \frac{\exp\left(-\frac{1}{2}\|\mathbf{y}_k - \mathbf{H}\mathbf{s}_k\|_{\mathbf{S}_k}^2\right)}{\sqrt{(2\pi)^{d_\mathbf{y}}\,|\mathbf{S}_k|}}. \tag{4.3}$$

where $d_\mathbf{y}$ is the dimension of the measurement vector, $\mathbf{H}$ is the measurement matrix, $\mathbf{S}_k = \mathbf{H}\mathbf{P}_\mathbf{s}\mathbf{H}^\top + \mathbf{R}_\mathbf{y}$ is the innovation covariance matrix, $\mathbf{P}_\mathbf{s}$ is the estimate covariance resulting from



(a) (b) (c)

Figure 4.2: Visualization of TCAFF multiple hypothesis process. (a) A new set of measurements is computed. (b) Leaf nodes are extended by applying Kalman Filter updates with candidate measurements. (c) Window is slid forward and unlikely branches are pruned.

the Kalman Filter, and $\mathbf{R}_\mathbf{y}$ is the measurement covariance. Taking the negative logarithm of equation (4.3) yields

$$\underset{\mathbf{y}_{1:K}}{\arg\min} \quad \sum_{k=1}^{K} \frac{1}{2} \left(d_\mathbf{y} \log(2\pi) + \|\mathbf{y}_k - \mathbf{H}\mathbf{s}_k\|^2_{\mathbf{S}_k} + \log(|\mathbf{S}_k|) \right). \tag{4.4}$$

If there is no measurement at time k (i.e., $\mathbf{y}_k = \text{None}$), we use a probability of no measurement p_{NM} resulting in

$$\underset{\mathbf{y}_{1:K}}{\arg\min} \sum_{k=1}^{K} \begin{cases} \frac{1}{2} \left(\|\mathbf{y}_k - \mathbf{H}\mathbf{s}_k\|^2_{\mathbf{S}_k} + \log(|\mathbf{S}_k|) \right), & \mathbf{y}_k \in \mathcal{Y}_k \\ -\log(p_{\text{NM}}) - \frac{1}{2} d_\mathbf{y} \log(2\pi), & \mathbf{y}_k = \text{None}. \end{cases} \tag{4.5}$$

We solve this optimization problem with a multi-hypothesis approach to find the correct frame alignment from a set of potentially incorrect frame alignment measurements. Thus, the robots use TCAFF to delay hard data association decisions until adequate information is gained and frame alignment measurements can be evaluated for temporal consistency.

4.1.4 Frame Alignment Filter

Finding optimal frame alignments in (4.5) amounts to evaluating measurements as branches of a tree. At the root of the tree is the initial state estimate $\hat{\mathbf{s}}_0$ and covariance $\mathbf{P}_{\mathbf{s}_0}$. The root is connected to one or more children by different edges, each child representing an estimate

$\hat{\mathbf{s}}_1$ and each edge representing a selected measurement $\mathbf{y}_1$. As leaves of the tree $\hat{\mathbf{s}}_k$ are added, the optimal estimate $\hat{\mathbf{s}}_k^*$ can be found by selecting the sequence of measurements resulting in the minimum objective value (4.5). Because the objective value of a node's child is the sum of its own objective value and an additional cost, computation can be saved by reusing the node's pre-computed cost when adding children.

An update step is illustrated in Fig. 4.2. First, new measurements $\mathcal{Y}_k$ are obtained from MNO-CLIPPER. Next, a gating is performed for each leaf node $\hat{\mathbf{s}}_k$ and $\mathbf{y}_k$ that prohibits adding nodes with high cost values (i.e., highly unlikely measurements), helping keep computation requirements low. A Kalman filter update [68] is then performed between each associated node and measurement to compute $\hat{\mathbf{s}}_{k+1} = \texttt{kalman\_update}(\mathbf{s}_k, \mathbf{y}_k)$, and $\hat{\mathbf{s}}_{k+1}$ is added to the tree as a new leaf node.

Finally, because this hypothesis tree approach is exponential in complexity, pruning must be employed to keep computation manageable. We employ a "sliding window" and "max branches" pruning approach [37]. For a sliding window of length W, all branches $\mathbf{s}_{k-W}$ that do not have the leaf node $\mathbf{s}_k^*$ as a descendant are pruned, leaving only a single $\mathbf{s}_{k-W}$ node, which becomes the root of the new window-bound hypothesis tree. Then, for a maximum number of branches B, only the B most optimal leaves are kept and the rest are pruned.

To apply our TCAFF approach to a scenario where no initial $\mathbf{s}_0$ exists, we introduce a sliding window method for exploring possible initial frame alignments $\mathbf{s}_0$. At each timestep k, each measurement $\mathbf{y}_{k-W}$ is used to initialize the root of a new exploring tree. The measurements in $\mathcal{Y}_{k-W+1}$, ..., $\mathcal{Y}_k$ are all added to each of the exploring trees and the optimal $\mathbf{s}_k^*$ from all of the exploring trees is selected and compared against a threshold τ_{align} to determine whether to initialize a "main" tree with the corresponding root at $\mathbf{s}_{k-W}$. Until a main tree has been selected, the method declares that no frame alignment estimate can be found. Similarly, a mechanism is needed to remove a main tree and go back to the exploring phase if it becomes unlikely that the main tree is correct. We return to the exploring phase if enough time has passed during which no measurements have been added to the optimal

leaf node. This method allows robots to leverage *temporal consistency* (i.e., the fact that a correct frame alignment estimate should produce many consistent measurements over time) to reject incorrect measurements and align frames with no initial guess.

4.2 Experiments

We experimentally evaluate our TCAFF method for aligning coordinate frames in a temporally consistent manner in Sections 4.2.1 and 4.2.2. The results from our full collaborative MOTLEE system using TCAFF are shown in Section 4.2.3. Unless otherwise stated, the following parameter values were used for all experiments: $\kappa = 20.0$, $\tau = 10.0$, $\nu = 3$, $p_{\text{NM}} = 0.001$, $\tau_{\text{align}} = 8.0$, $N = 4$, $W = 8$, and $B = 200$.

4.2.1 Indoor Frame Alignment Experiment

First, we evaluate TCAFF's ability to estimate the frame alignment between two robots without any initial guess, which requires the ability to distinguish between maps that belong to the same area and maps that result from traveling in non-overlapping areas. Two robots are initially driven around a $10 \times 10\,\text{m}$ motion-capture room. The robots are equipped with an Intel RealSense T265 Tracking Camera whose onboard VIO is used for ego-pose estimation and an Intel RealSense L515 LiDAR Camera that is used for extracting depths of segments detected by FastSAM. In our experiments, object maps are updated at 10 Hz and are shared at 1 Hz. Each robot's TCAFF is updated upon the reception of a neighboring robot's map. Boxes are scattered around the room to represent generic objects that can be detected by the open-set segmentation of FastSAM [11]. Both robots start in the room, and then one robot leaves and accumulates odometry error out-of-view of the other robot before returning to the room for the remainder of the experiment.

In Fig. 4.3 the frame alignment results from this experiment demonstrate that the robots accurately estimate frame alignments when maps overlap at the start and end of the experi-

ment. Additionally, the robots correctly recognize that potential alignments in the middle of the run do not exhibit temporal consistency and should not be incorporated into the frame alignment estimate. During the times when an estimate is made, the robots estimate the relative frame alignments with average translation and rotation errors of 0.35 m and 1.1 deg respectively.

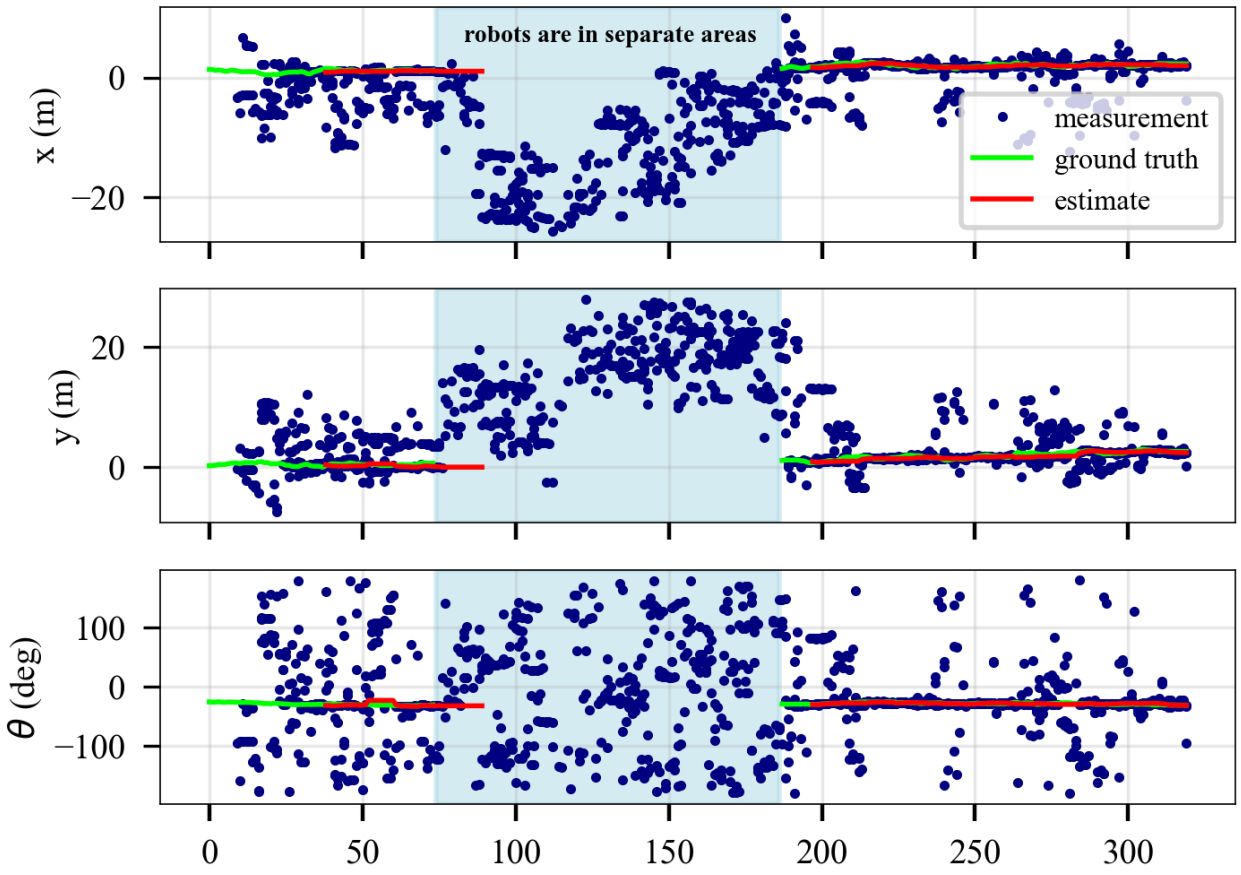


Figure 4.3: TCAFF is visualized by plotting the frame alignment measurements from MNO-CLIPPER in blue along with the ground truth and TCAFF frame alignment estimate. Each blue dot represents a frame alignment measurement $\mathbf{y}(k) = [x, y, \theta]^\top$, with the x, y, and θ axes shown in separate plots. TCAFF correctly recognizes when enough temporally consistent measurements are received to verify a correct frame alignment. The ground truth frame alignment disappears in the middle of the run when one robot leaves the VICON room and its ground truth pose is unavailable. While in separate areas, map information is still exchanged, and MNO-CLIPPER can be used to find alignments between the two maps, but TCAFF rejects these temporally inconsistent measurements.

Figure 4.4: Images from the two robots traveling in opposite directions in the Kimera-Multi dataset [69].

4.2.2 Outdoor Frame Alignment Experiment

The second experiment uses data collected in the Kimera-Multi outdoor dataset [69] to test our open-set object mapping and TCAFF system in natural outdoor environments. We test our frame alignment method working in two instances, one where robots run parallel to each other along the same path and the other where robots start 50 m away from each other and then cross paths. Traditional, image-based loop closure techniques fail to detect that robots cross paths when traveling opposite directions because of the large viewpoint difference as discussed in [69]. Images from the two robots' runs are shown in Fig. 4.4 and the average and standard deviation of frame alignment errors are shown in Table 4.1. The translation and rotation errors are computed as the difference between TCAFF's estimate and the ground truth frame alignment, noting that TCAFF's estimate only occurs after accepting an exploring hypothesis tree. The following parameters are changed in these outdoor experiments: $\kappa = 15.0$ and $W = 5$. This maintains objects in the map of recently seen objects for less time since odometry drift is worse in these scenarios. We show that our method correctly estimates frame alignments in challenging outdoor scenarios from the Kimera-Multi outdoor dataset, including a scenario where robots only observe the scene from opposite directions as they cross paths.

Table 4.1: Kimera-Multi Data Results

	Avg. Translation Error [m]	Avg. Heading Error [deg]
Same direction	1.01 ± 0.85	1.08 ± 0.82
Opposite directions	1.63 ± 0.67	1.11 ± 0.48

4.2.3 Full Collaborative MOT Experiment

Finally, we evaluate frame alignment from TCAFF used in conjunction with our MOTLEE system's collaborative object-tracking. We record and release a dataset of four robots driving autonomously around a motion capture room while six pedestrians walk around in the same space. This multi-robot MOT experiment presents many additional challenges when compared to the dataset recorded in [8], including more pedestrians and robots, robot trajectories that cover the whole motion capture space rather than non-overlapping areas, and the absence of domain-specific objects (e.g., cones) used for creating object maps.

We use the same sensors used in the 4.2.1 experiment with the substitution of the Intel RealSense T265 for an Intel RealSense D455 camera attached to the rear of the robot. Kimera-VIO [16] is used for ego-pose estimation and YOLOv7 [63] for person detection. We evaluate the MOT performance using the MOTA metric [62], as defined in Section 3.4.

In Fig. 4.5 we compare the MOTA results of our full MOTLEE system against robots that align frame using our ICP-based algorithm from Chapter 3 which is named here MOTLEE-ICP [8], CLIPPER [12], and ground truth frame alignment. Note that only MOTLEE-ICP is given the correct initial frame alignment. In the CLIPPER benchmarks, we accept the association solution from standard CLIPPER as long as a minimum number of objects are associated between the two maps and otherwise reject the alignment, and we show the CLIPPER method's sensitivity to this parameter. If the required number of associations is low, some incorrect associations are not rejected which hurts the performance of the inter-robot frame alignment and collaborative MOT. Alternatively, a high minimum number of associated objects may be set, but this makes a restrictive assumption about the expected quantity of overlapping objects in the two maps, and can result in many fewer accepted alignments.

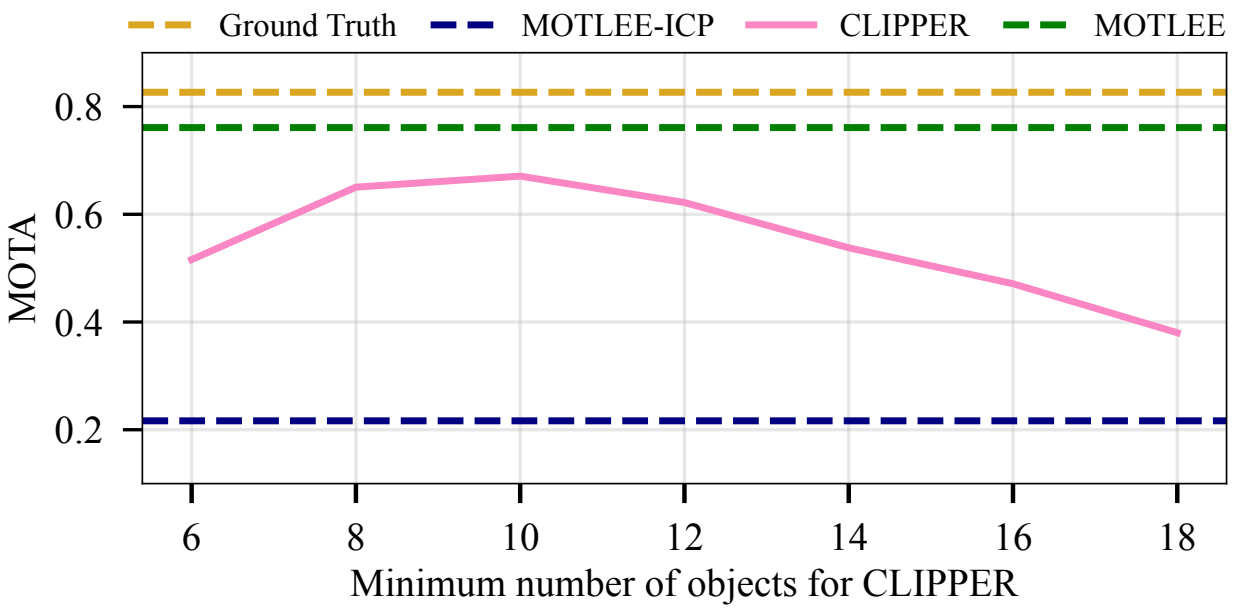


Figure 4.5: Comparison of MOTA results. Results for using a single CLIPPER solution that requires a set minimum number of associations are shown for a sweep of different parameter values. TCAFF is able to consider all potential alignments and reject incorrect frame alignments by leveraging temporal consistency, resulting in a higher object tracking accuracy.

Our method allows a system to benefit from the best of both worlds, additionally benefiting from alignments that can only be found when using our MNO variation of CLIPPER. With TCAFF, incorrect alignments are rejected and alignments can be found even when the maps have very little overlap.

Additionally, we show MOTA over a rolling window and frame alignment results in Fig. 4.6. MOTLEE achieves a total MOTA of 0.761, close to the ground-truth MOTA of 0.827, and achieves an average frame alignment error of 0.43 m and 2.3 deg.

4.2.4 Computation Time

We have broken up different pieces of the system to evaluate the computation time of the different elements where the mean and standard deviation of computation times are shown in Table 4.2. MNO-CLIPPER and TCAFF must both be performed for each neighboring robot. Each of the tasks listed in Table 4.2 can easily be run in parallel for real-time use.

Table 4.2: Pipeline Element Timing Analysis [ms]

Mapping (10 Hz)	MOT (10 Hz)	MNO-CLIPPER (1 Hz)	TCAFF (1 Hz)
13.5 ± 8.0	2.2 ± 0.9	150.2 ± 45.3	18.6 ± 20.5

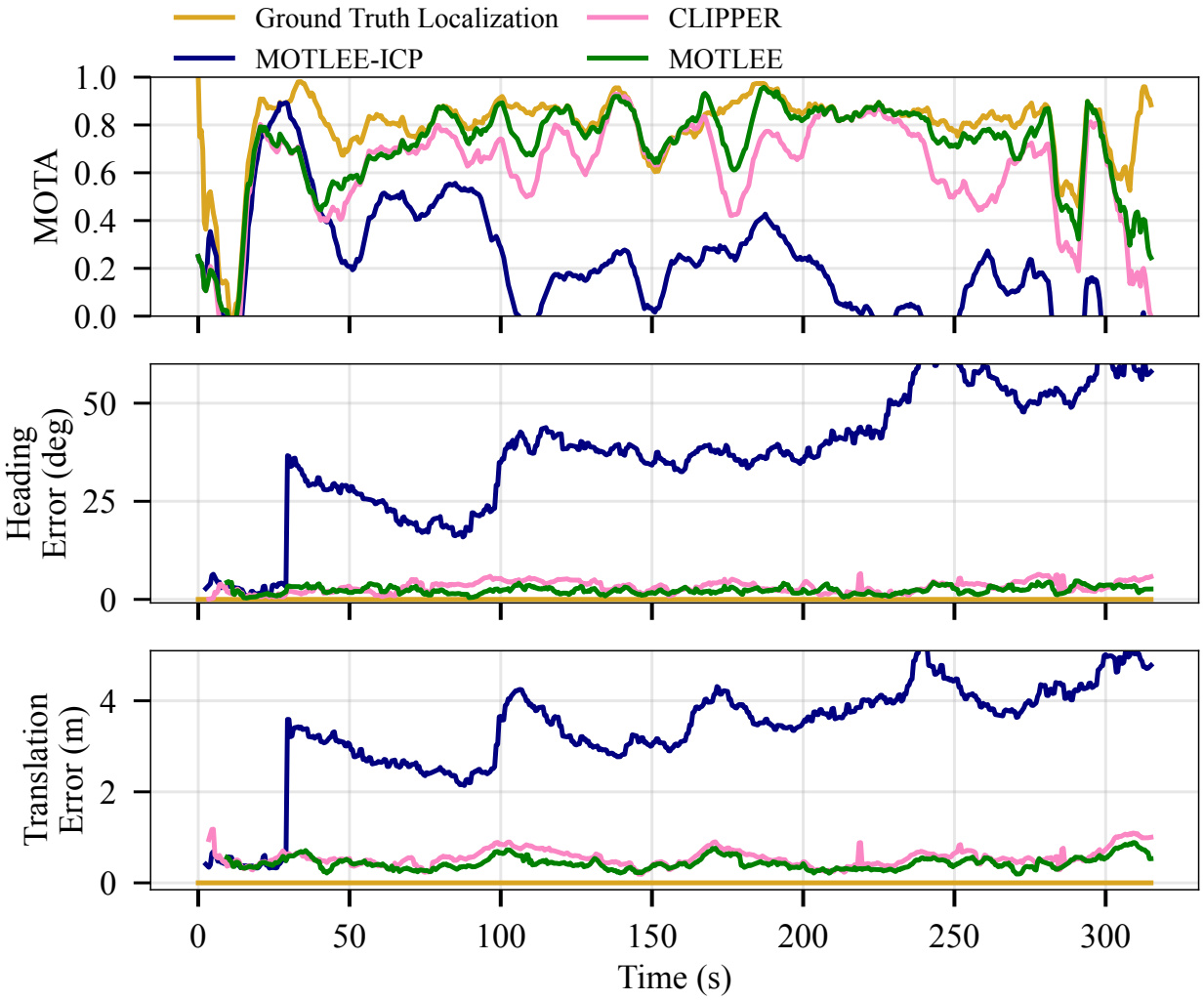


Figure 4.6: Comparison of object tracking accuracy and frame alignment accuracy in an experiment with four robots tracking six pedestrians. MOTA is computed over a rolling window of 10 s. MOTLEE using TCAFF for frame alignment is benchmarked against frame alignment from MOTLEE-ICP [8], CLIPPER [12], and ground truth frame alignment. Note that the CLIPPER benchmark rejects any alignments with fewer than 10 associations since this results in its best MOTA performance as seen in Fig. 4.5. MOTLEE is able to estimate frame alignments with no initial guess, use frame alignments with few associated objects by leveraging temporal consistency, and collaboratively track objects with accuracy similar to that of robots with ground truth localization.

Chapter 5

Conclusion

In this thesis, we have presented a framework for aligning robot coordinate frames for collaborative tasks in real-time. We first developed a system for incrementally correcting for frame drift by aligning tracked dynamic object measurements and sparse object maps, a frame alignment method that is view-invariant and communication efficient. We further extended this capability to performing frame alignment without any initial guess, which was accomplished by using open-set object maps and our frame alignment filter, TCAFF. We demonstrated that TCAFF allows robots to perform pairwise frame alignments in the presence of map ambiguity by considering multiple frame alignment hypotheses and determining correct alignments by searching for alignments with high temporal consistency. Although our implementation is specifically aimed at aligning coordinate frames, TCAFF could be used for other state estimation tasks that require rejecting temporally inconsistent false-positive measurements.

We demonstrated the use of our frame alignment pipeline as part of MOTLEE, our distributed system for performing collaborative dynamic object tracking. Included in this effort, we showed how to incorporate frame alignment uncertainty when sharing spatial information, including KCF state and measurement information. Using these methods, we showed through hardware experiments that MOTLEE can perform object tracking with

accuracy similar to that of a system with perfect localization.

5.1 Future Work

Multi-robot perception is a rapidly evolving field, and as such there remain many interesting directions to explore for future work. These include:

- **Open-set mapping with a consistent object-level understanding of an environment.** One limitation in our current use of FastSAM [11] is that we do not account for instances where objects are segmented differently depending on the specific image in which it is seen. For instance, depending on how large a car is in an image, the whole car may be included in a single segment, or each window, wheel, door, and headlight may be segmented separately. Higher-level semantics could be used to recognize when a single object has been segmented into multiple parts.

- **Incorporating additional object information when associating objects between maps.** Our work currently only uses object centroids to perform point registration with the small additional help of using object widths and heights to filter out clearly incorrect associations. Incorporating more information, including semantics, learned shape descriptors, and object orientation could disambiguate some challenging alignment problems.

- **Integrating object-based place recognition and loop closure into a factor graph SLAM formulation**. Our methods work well for aligning recently observed object maps for multi-robot systems in real-time. These concepts could be used to complement other SLAM methods, like visual loop closure, to improve visual SLAM systems' ability to accurately optimize a robot's complete trajectory and environment map.

References

[1] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*, vol. 19.

[2] R. Stern, "Multi-agent path finding–an overview," *Artificial Intelligence: 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures*, pp. 96–115, 2019.

[3] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[6] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[7] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *ieee transactions on robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[8] M. B. Peterson, P. C. Lusk, and J. P. How, "Motlee: Distributed mobile multi-object tracking with localization error elimination," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 719–726.

[9] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, Spie, vol. 1611, 1992, pp. 586–606.

[10] M. B. Peterson, P. C. Lusk, A. Avila, and J. P. How, "Motlee: Collaborative multi-object tracking using temporal consistency for neighboring robot frame alignment," *arXiv preprint arXiv:2405.05210*, 2024.

[11] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[12] P. C. Lusk and J. P. How, "CLIPPER: Robust Data Association without an Initial Guess," *IEEE Robotics and Automation Letters*, pp. 1–8, 2024. DOI: 10.1109/LRA.2024.3364842.

[13] K. Kondo, C. T. Tewari, M. B. Peterson, A. Thomas, J. Kinnari, A. Tagliabue, and J. P. How, "Puma: Fully decentralized uncertainty-aware multiagent trajectory planner with real-time image segmentation-based frame alignment," *arXiv preprint arXiv:2311.03655*, 2023.

[14] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph.," in *Robotics: science and systems*, Rome, Italy, vol. 11, 2015, p. 3.

[15] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *arXiv preprint arXiv:2312.02126*, 2023.

[16] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 1689–1696.

[17] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[18] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 1722–1729.

[19] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[20] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[21] H. Zhang, H. Uchiyama, S. Ono, and H. Kawasaki, "Motslam: Mot-assisted monocular dynamic slam using single-view depth estimation," in *IEEE/RSJ IROS*, 2022, pp. 4865–4872. DOI: 10.1109/IROS47612.2022.9982280.

[22] X. Tian, Z. Zhu, J. Zhao, G. Tian, and C. Ye, "Dl-slot: Dynamic lidar slam and object tracking based on collaborative graph optimization," *arXiv preprint arXiv:2212.02077*, 2022.

[23] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[24] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 3437–3444.

[25] H. Do, S. Hong, and J. Kim, "Robust loop closure method for multi-robot map fusion by integration of consistency and data similarity," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5701–5708, 2020.

[26] Z. Chen, J. Zhao, T. Feng, C. Ye, and L. Xiong, "Robust loop closure selection based on inter-robot and intra-robot consistency for multi-robot map fusion," *Remote Sensing*, vol. 15, no. 11, p. 2796, 2023.

[27] J. Ankenbauer, K. Fathian, and J. P. How, "View-invariant localization using semantic objects in changing environments," *arXiv preprint arXiv:2209.14426*, 2022.

[28] Y. Tian, Y. Chang, L. Quang, A. Schang, C. Nieto-Granda, J. P. How, and L. Carlone, "Resilient and distributed multi-robot visual slam: Datasets, experiments, and lessons learned," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 11 027–11 034.

[29] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103 448, 2021.

[30] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 261–268.

[31] G. Habibi and J. P. How, "Human trajectory prediction using similarity-based multi-model fusion," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 715–722, 2021.

[32] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, IEEE, 2017, pp. 3645–3649.

[33] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European conference on computer vision*, Springer, 2020, pp. 474–490.

[34] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[35] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood method for probabilistic multihypothesis tracking," in *Signal and data processing of small targets 1994*, SPIE, vol. 2235, 1994, pp. 394–405.

[36] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[37] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4696–4704.

[38] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *IEEE CVF/CVPR*, 2018.

[39] M. Hofmann, D. Wolf, and G. Rigoll, "Hypergraphs for joint multi-view reconstruction and multi-object tracking," in *IEEE/CVF CVPR*, 2013.

[40] O. Shorinwa, J. Yu, T. Halsted, A. Koufos, and M. Schwager, "Distributed multi-target tracking for autonomous vehicle fleets," in *IEEE ICRA*, 2020, pp. 3495–3501. DOI: 10.1109/ICRA40945.2020.9197241.

[41] L. He, G. Liu, G. Tian, J. Zhang, and Z. Ji, "Efficient multi-view multi-target tracking using a distributed camera network," *IEEE Sensors Journal*, vol. 20, no. 4, pp. 2056–2063, 2019.

[42] S. Casao, A. Naya, A. C. Murillo, and E. Montijano, "Distributed multi-target tracking in camera networks," in *IEEE ICRA*, 2021.

[43] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *IEEE CDC*, 2008.

[44] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed multi-target tracking and data association in vision networks," *IEEE T-PAMI*, vol. 38, no. 7, pp. 1397–1410, 2015.

[45] M. Bredereck, X. Jiang, M. Körner, and J. Denzler, "Data association for multi-object tracking-by-detection in multi-camera networks," in *IEEE ICDSC*, 2012.

[46] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE TPAMI*, 2007.

[47] L.-L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte, "A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles," in *IEEE/RSJ IROS*, 2006.

[48] O. Dagan, T. L. Cinquini, L. Morrissey, K. Such, N. R. Ahmed, and C. Heckman, "Towards decentralized heterogeneous multi-robot slam and target tracking," *arXiv preprint arXiv:2306.04570*, 2023.

[49] A. Ahmad, G. Lawless, and P. Lima, "An online scalable approach to unified multirobot cooperative localization and object tracking," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1184–1199, 2017.

[50] E. Taghavi, R. Tharmarasa, T. Kirubarajan, Y. Bar-Shalom, and M. Mcdonald, "A practical bias estimation algorithm for multisensor-multitarget tracking," *IEEE T-AES*, vol. 52, no. 1, pp. 2–19, 2016.

[51] P. M. Dames, "Distributed multi-target search and tracking using the phd filter," *Autonomous robots*, vol. 44, no. 3-4, pp. 673–689, 2020.

[52] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *IEEE ICRA*, 2013, pp. 5696–5701. DOI: 10.1109/ICRA.2013.6631396.

[53] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *IEEE CDC*, 2007, pp. 5492–5498.

[54] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *IEEE CDC*, 2009, pp. 7036–7042.

[55] J.-L. Blanco, J. González-Jiménez, and J.-A. Fernández-Madrigal, "An alternative to the mahalanobis distance for determining optimal correspondences in data association," *IEEE T-RO*, vol. 28, no. 4, 2012.

[56] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. DOI: https://doi.org/10.1002/nav.3800020109.

[57] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *IJRR*, vol. 5, no. 4, pp. 56–68, 1986.

[58] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE TPAMI*, no. 5, pp. 698–700, 1987.

[59] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *ECCV*, 2020.

[60] R. Martín-Martín, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese, "JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *IEEE TPAMI*, vol. 45, no. 6, pp. 6748–6765, 2023. DOI: 10.1109/TPAMI.2021.3070543.

[61] C. W. Anderson and J. P. How, "Implementation of vision-based navigation for pedestrian environments," M.S. thesis, MIT, 2022.

[62] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[63] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF CVPR*, 2023, pp. 7464–7475.

[64] M. Krupczak, *Coneslayer: Cone detection weights for yolov7*, version 1.0.0, [Online]. Available: https://github.com/mkrupczak3/Coneslayer, Mar. 1, 2023.

[65] A. Thomas, J. Kinnari, P. Lusk, K. Kondo, and J. P. How, *Sos-match: Segmentation for open-set robust correspondence search and robot localization in unstructured environments*, 2024. arXiv: 2401.04791 [cs.RO].

[66] P. C. Lusk, K. Fathian, and J. P. How, "CLIPPER: A graph-theoretic framework for robust data association," in *IEEE ICRA*, 2021.

[67] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, "Search and rescue under the forest canopy using multiple uavs," *IJRR*, vol. 39, no. 10-11, pp. 1201–1221, 2020.

[68] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," 1961.

[69] Y. Tian, Y. Chang, L. Quang, A. Schang, C. Nieto-Granda, J. How, and L. Carlone, "Resilient and distributed multi-robot visual SLAM: Datasets, experiments, and lessons learned," in *IEEE/RSJ IROS*, 2023.