INFORMATION-DRIVEN CONTROL OF MULTI-ROBOT NETWORKS FOR DYNAMIC TARGET TRACKING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Cong Liu

August 2023

INFORMATION-DRIVEN CONTROL OF MULTI-ROBOT NETWORKS FOR DYNAMIC TARGET TRACKING

Cong Liu, M.S.

Cornell University 2023

Recently, research on multi-robot networks has attracted increasing attention due to their higher efficiency and robustness over single-robot systems. This work focuses on the control of multi-robot networks for tracking multiple human targets, which has promising applications in security and surveillance. Existing literature has shown that the information gain can guide sensors to make informative measurements. With this inspiration, an information gain based control algorithm was developed to optimize the tracking performance of multi-robot networks. The proposed control algorithm considered target tracking, target exploration, and collision avoidance. In addition, this work validated the network control through physical experiments involving Unmanned Ground Vehicles (UGVs) and real human targets, for which four online sensing algorithms including UGV localization, target detection, target localization, and target classification were implemented.

BIOGRAPHICAL SKETCH

Cong Liu is an M.S. student in the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University. He received his bachelor's degree in Mechanical Engineering from the joint school of University of Leeds and Southwest Jiaotong University. His research interests include robotics, computer vision, path planning, and control.

This document is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

I am very grateful to my advisor, Prof. Silvia Ferrari. I appreciate it for her instruction and support along my academic journey. I would also like to thank my special committee member, Prof.Elizabeth Farrell Helbling, who kindly give me great help and suggestion in my thesis. I also want to say thank you to all of our lab members, especially Junyi and Sushrut, who help me a lot in my two years study.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND MOTIVATION

1.1 Multi-Robot Network

A multi-robot network(MRN) is a system that consists of multiple robots and each robot has its own sensors, actuators, and processing capabilities. Robots can communicate with each other through the network and coordinate actions to achieve a common objective. Multi-robot networks are widely used in search and rescue [4, 9], surveillance [17, 25], exploration [5, 45], transportation [33, 34], and other fields. Compared with single robot systems, MRNs show strength in scalability, redundancy, adaptability, and task distribution [7, 16, 36]. As for scalability, MRNs can be scaled up by adding more robots into networks which increases the capabilities of MRNs to handle more complex missions. The redundancy is that if one robot fails to finish some tasks, other robots in MRNs can compensate and finish the task to guarantee the system continues to function. The adaptability of MRNs is shown in that MRNs can be easily adapted to different tasks and environments by incorporating different robots with different capabilities in MRNs. MRNs can distribute tasks to different robots so that MRNs can work on different tasks simultaneously. This parallelism achieved by task distribution results in high efficiency of task completion. Inspired by the significant strength of MRNs, this work proposes a multi-target tracking approach by using MRN.

1.2 Multi-Target Tracking with MRN

Target tracking involves estimating the trajectories of moving targets in a given scene which can be applied in different fields such as traffic monitoring, sports analytics, and surveillance. In the beginning, research in this area focused on using static sensors to collect data for target estimation to achieve tracking [37, 39, 44]. The main shortcomings of target tracking with static sensors include limited coverage, occlusions, and inefficient resource utilization. The limited coverage results from the fixed field of view of the sensors. The fixed field of view may not cover the whole area of interest, which means there could be blind spots where the targets cannot be tracked. Static sensors are vulnerable to occlusion as objects in the area of interest may block the view of the sensor. It leads to a loss of tracking information. As for the inefficient resource utilization, in some conditions, some sensors may be overloaded while some sensors are not utilized because sensors are static. To resolve these issues, research turns to utilizing MRNs for target tracking. The robots in MRNs are mobile robots equipped with sensors and the tracking performance can be optimized by cooperatively determining the robot control. By using mobile robots, robots can change their field of view to avoid occlusion and cover the whole area of interest [40]. Moreover, MRNs with mobile sensors can dynamically distribute tracking tasks among available sensors, achieving the improvement in efficiency of resources utilization [32]. With the strength of MRN in mind, this work proposes a target tracking approach using MRN.

The literature on multi-target tracking with MRN presents a wide range of algorithms, which can be broadly classified into two categories: centralized algorithm and decentralized algorithm. Centralized algorithms use a central coordinator to collect data from all robots and compute optimal control for each robot. The central coordinator controls MRNs by sending commands to each individual robot. As the central coordinator makes decisions based on global information, the calculated controls are optimal. On the other hand, decentralized algorithms enable each robot to decide its control by itself or based on communication with its neighbor robots. Decisions made from decentralized algorithms are based on local or partial information so that the calculated controls may be suboptimal. The centralized and decentralized algorithms can be compared in the following aspects: scalability, robustness, communication overhead, and computational complexity. The scalability of centralized algorithms is limited as the central coordinator needs to process all the information and the increased number of robots increases the computation load which results in degrading the tracking performance. Decentralized algorithms have better scalability because there is no central coordinator and each robot only processes local information. The tracking performance will not decrease by adding the number of robots. The robustness of the centralized algorithms is weak as the system purely relies on the central coordinator. If the central coordinator fails, the whole system will not function. Decentralized algorithms are more robust to failures, if one robot fails, the rest of the robots can still function. Centralized algorithms have high communication overhead because robots need to share information with the central coordinator and get instructions from it. Decentralized algorithms have relatively lower communication overhead as robots only share information with their neighbors. As for computational complexity, centralized algorithms have higher complexity as all the computation is completed by the central coordinator. While decentralized algorithms distribute computation to every robot so that the computational complexity for each robot is relatively low.

The comparison shows that decentralized algorithms have strength in better scalability, better robustness, lower communication overhead, and lower computational competitively. Inspired by this, this work develops a decentralized algorithm for target tracking with MRN.

1.3 Information-Driven Approach

This work develops an information-driven approach for controlling MRN to track multi-targets. The information-driven approach for target tracking with MRN focuses on controlling MRN to maximize the information gain. The information gain is a metric that evaluates the tracking performance, which is calculated based on the sensor measurements. This work chooses to utilize an information-driven approach for the following reasons. To develop tracking algorithms, multiple objectives need to be considered, such as minimization of track loss, maximization of probability of target detection, and identification accuracy [22]. Research indicates that information-driven approaches can simultaneously address multiple objectives for managing MRNs for tracking [10, 26]. The trade-off between objectives can be managed by various approaches. Besides this, another strength of the information-driven approach is that it can be implemented in a decentralized manner, which is suitable for our proposed decentralized architecture's tracking algorithm. In the decentralized information-driven approach, each robot has its own information gain and each robot locally makes the decision to control the action of itself to maximize the information gain. Above strengths of the information-driven approach makes it suitable for our work.

In summary, this work develops a decentralized information-driven approach to control multi-robot networks for dynamic target tracking. Chapter 2 shows the problem formulation that builds mathematical models of the tracking problem of this work. The methodology is explained in Chapter 3, which includes the online sensing and the tracking utility function. Chapter 4 is the results of the physical experiments which can be used to validate the proposed algorithm. The future work and conclusion are in Chapter 5 and Chapter 6 respectively.

CHAPTER 2

PROBLEM FORMULATION

This thesis attempts to develop algorithms for controlling multi-robot networks(MRN) to track multiple moving human targets . Developing mathematical models for the multi-robot networks and human targets is the first step to solve this problem. Then the online sensing will be resolved for robot localization, target detection, classification, and localization. On the basis of robot and target information obtained from online sensing, a novel tracking utility function will be developed and deployed to control the robots to track targets. The details are shown in the following sections.

2.1 Models of Robot and Target

This thesis will use Unmanned Ground Vehicles (UGV) as robots to track moving targets. The multi robot network(MRN) is represented by $\mathcal{N} = \{1, \ldots, N\}$, which is the index of the robot in MRN and N is the total number of robots. Fig.2.1 shows the configuration of a robot. Robots are assumed to work in a two-dimensional workspace $\mathcal{W} = [0, L_x] \times [0, L_y]$, and $L_x, L_y \in \mathbb{R}^+$ represent the boundary of the workspace. The initial frame $\mathcal{F}_{\mathcal{W}}$ is embedded in $\mathcal{W}$, where $O_{\mathcal{W}}$ is the origin and $x_I y_I$-plane aligns with the ground plane. There is a body frame $\mathcal{F}_{\mathcal{A}_i}$ embedded in each robot $i, i \in \mathcal{N}$, with its origin $O_{\mathcal{A}_i}$ at the pinhole of the robot camera. The $axis - x_B$ aligns with the robot's heading direction v_i and $axis - y_B$ is perpendicular to $axis - x_B$ in the ground plane. The state of each robot in the initial frame is described by a state vector $\mathbf{s}_i = [x_i \quad y_i \quad \theta_i]^T$. The x_i and y_i represent the position of the robot which are the 2D coordinates of $O_{\mathcal{A}_i}$ in $\mathcal{F}_{\mathcal{W}}$

while the orientation of the robot θ_i is defined as the angle between $axis - x_B$ and $axis - x_I$. The state of each robot can be estimated based on the Dead Reckoning with embedded odometry sensors including IMU and encoder. External motion sensors such as Motion Capture Systems also can be applied to localize robots. The details of the localization of the robot will be discussed in the Section 3.2.1.

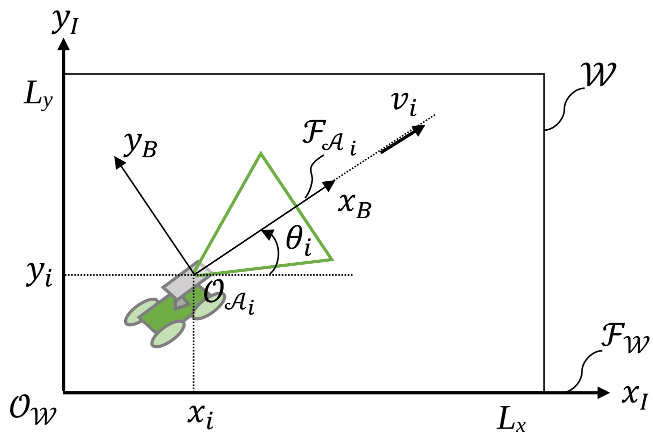

Figure 2.1: The configuration of a robot.

The dynamic model of the robots in this thesis is the unicycle model [12, 31], which is described as

$$\dot{\mathbf{s}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix} = \mathbf{f}(\mathbf{s}_i, \mathbf{u}_i), \quad \forall i \in \mathcal{N} \tag{2.1}$$

in which the vector $\mathbf{u}_i$ is the control vector that consists of the linear velocity v_i and angular velocity w_i, so that the control vector is expressed as $\mathbf{u}_i = [v_i \quad w_i]^T \in \mathbb{R}^2$. The sampling interval $\Delta t \in \mathbb{R}^+$ is assumed to be constant all the time, then the robot state and control vector at any discrete time k can be written as

$$\mathbf{s}_i(k) = \mathbf{s}_i(k\Delta t) \tag{2.2}$$

$$\mathbf{u}_i(k) = \mathbf{u}_i(k\Delta t) \tag{2.3}$$

The state and control of the MRN can be represented by matrices consisting of every robot's state and control vector so that the state and control can be represented as $\mathbf{s}(k) = [\mathbf{s}_1^T(k) \quad \dots \quad \mathbf{s}_N^T(k)]^T \in \mathbb{R}^{3N}$ and $\mathbf{u}(k) = [\mathbf{u}_1^T(k) \quad \dots \quad \mathbf{u}_N^T(k)]^T \in \mathbb{R}^{2N}$.

The moving targets in this thesis are human targets which are represented by $\mathcal{M} = \{1, \dots, M\}$, where M is the number of targets and $j \in \mathcal{M}$ is the index of each target. The index j will also be used as a unique identity for each target. The state of each target at the discrete time k is defined as $\mathbf{x}_j(k) = [x_j(k) \quad y_j(k) \quad v_{x,j}(k) \quad v_{y,j}(k)]^T \in \mathbb{R}^4$, where the four variables denote the position and velocity in x_I and y_I direction of the initial frame $\mathcal{F}_W$. By combing all the targets' state vectors, the states of all the targets can be expressed as $\mathbf{x}(k) = [\mathbf{x}_1^T(k) \quad \dots \quad \mathbf{x}_M^T(k)]^T \in \mathbb{R}^{4M}$. The targets are assumed to move in constant velocity with additive Gaussian process noise $\mathbf{w}(k)$. Then the motion model of each target at each discrete time k is denoted as

$$\mathbf{x}_j(k) = \mathbf{F}\mathbf{x}_j(k-1) + \mathbf{w}(k), \quad \mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2.4}$$

in which $\mathbf{F} \in \mathbb{R}^{4\times 4}$ is the state transition matrix [3], and $\mathbf{Q}$ represents the covariance matrix of the process noise $\mathbf{w}(k)$.

2.2 Target Detection, Classification, and Localization

To track targets with robots, the localization of each target is essential as robots need to do path planning based on the trajectory of targets. Target detection is the premise of target localization. As there are multiple targets, the target classification is needed to enable the robot to distinguish each target.

This thesis considers the targets as human targets and the robots have onboard RGBD cameras. Robots can observe the targets which are in the field-of-view(FOV) of the robot with onboard cameras. The robot's FOV is the range of the visual area of the workspace. The FOV of each robot is represented by $\mathcal{S}_i \subset \mathcal{W}, \forall i \in \mathcal{N}$. There are many well-developed computer vision algorithms that can detect human targets from images and videos [14, 20, 47]. One of those algorithms is adopted for target detection in this thesis by using the images streamed from robots' onboard cameras. Each robot will be given a set of targets to track. When tracking targets, one robot's path planning should take into account the states of the targets that are assigned to it. It is necessary that robots can distinguish different targets so that they can plan their path based on the targets assigned to them. In order to help robots tell apart different targets, the target classification algorithm is adopted. After the target is detected, the target is named a special ID, and based on the ID, the targets can be assigned to different robots to be tracked and the robot can update the state of each target accordingly. The target classification is done by a deep neural network. The details of the target detection and classification algorithm are discussed in the Section 3.2.

As for target localization, this thesis measures the position of the target in the global frame $\mathcal{F}_W$ based on the RGBD images and robot localization. The details of this target localization method are also shown in the Section 3.2. The measurement function is expressed as

$$\mathbf{z}_{i,j}(k) = \mathbf{H}\mathbf{x}_j(k) + \mathbf{v}(k) \quad \text{if} \quad \mathbf{x}_j(k) \in \mathcal{S}_i(k) \tag{2.5}$$

in which $\mathbf{z}_{i,j}$ represents the i^{th} robot's measurement on the j^{th} target, and $\mathbf{H} = [\mathbf{I}_2 \quad \mathbf{0}_2]$ and $\mathbf{v}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is the additive Gaussian noise for the measurement function. Kalman filter is used to recursively estimate the state of targets by combining the target dynamic model (2.4) and measurement model (2.5) as follows

$$\hat{\mathbf{x}}_{i,j}(k) = \begin{cases} \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) + \mathbf{K}(k)\mathbf{e}_{i,j}(k) & \text{if } \mathbf{x}_j(k) \in \mathcal{S}_i(k) \\ \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) & \text{if } \mathbf{x}_j(k) \notin \mathcal{S}_i(k) \end{cases} \tag{2.6}$$

where $\hat{\mathbf{x}}_{i,j}(k)$ is the estimated state of the j^{th} target by the i^{th} robot at time k, $\hat{\mathbf{x}}_{i,j}(k-1)$ is the estimated state at time $k-1$, $\mathbf{e}_{i,j}(k)$ is the innovation term of the Kalman filter which is $\mathbf{e}_{i,j}(k) = \mathbf{z}_{i,j}(k) - \mathbf{H}\mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1)$, and $\mathbf{K}(k)$ is the well known Kalman gain matrix. When the target is not in the FOV of the robot, the robot will predict the target state based on the target's dynamics model, and when the robot detects the target within its FOV, it will do the state estimation by combing the prediction from dynamics model and measurement with Kalman filter.

2.3 Target Tracking

Target tracking aims at controlling the MRN to get the most informative measurements of the targets. Before controlling the robots to track the targets, target assignment is required in order to specify which targets should be tracked by which robots. In each discrete time k, each robot $i \in \mathcal{N}$, $\mathcal{N} = \{1, \ldots, N\}$, should be assigned a subset of the targets $P_i(k) \in \mathcal{M}$, $\mathcal{M} = \{1, \ldots, M\}$, where i is the index of robots. The target assignment for the MRN is expressed as the collection of each subset, $P(k) \triangleq \{P_1(k), \ldots, P_N(k)\}$. A valid targets assignment is that all the targets are assigned to robots and each target is only assigned to one robot. Then all the targets will be tracked and no targets will be tracked by more than one robot.

The valid target assignments can be expressed as

$$P_i(k) \cap P_{i'}(k) = \emptyset, \text{ if } i \neq i', \text{ and } \bigcup_{i=1}^{N} P_i(k) = \mathcal{M} \tag{2.7}$$

The collection of all the valid target assignments is defined as the target assignment space, which is represented by $\mathcal{P}$. There are several existing methods that can achieve valid target assignments [13, 38, 43]. For simplicity, this thesis assumes that the valid targets assignment $\mathcal{P}$ to be known a *priori*, and this thesis focus on developing an algorithm to control each robot to track targets assigned to it.

A tracking utility function is developed which aims at evaluating the tracking performance of the MRN. The tracking utility function $U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k))$, reflects the tracking performance of j^{th} robot by i^{th} robot. The specifics of the tracking utility function are defined in the Section 3.4. Each target is assumed to move independently so that the tracking utility of MRN U_g can be expressed as the sum of the tracking utility of each target:

$$U_g \triangleq \sum_{i \in \mathcal{N}} \left[\sum_{j \in P_i^*(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \right] \tag{2.8}$$

where $P_i^*(k)$ represents the set of targets assigned to robot i which is assumed to be known a *priori*. The optimized control of MRN can be got by maximizing U_g, i.e.,

$$\max_{\mathbf{u}(k)} \quad \sum_{i \in \mathcal{N}} \left[\sum_{j \in P_i^*(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \right] \tag{2.9}$$

$$\text{s.t.} \quad \mathbf{s}_i(k+1) = \mathbf{f}(\mathbf{s}_i(k), \mathbf{u}_i(k)), \quad \forall i \in \mathcal{N} \tag{2.10}$$

$$\mathbf{a}_1 \leq \mathbf{s}_i(k) \leq \mathbf{a}_2, \quad \forall i \in \mathcal{N} \tag{2.11}$$

$$|\mathbf{u}_i(k)| \leq \mathbf{b}, \quad \forall i \in \mathcal{N} \tag{2.12}$$

in which $\leq$ is the elementwise inequalities, $\mathbf{a}_1 = [0 \quad 0 \quad 0]^T$, $\mathbf{a}_2 = [L_x \quad L_y \quad 2\pi]^T$, and $\mathbf{b}$ is the boundary of the control input. The one-step future state $\mathbf{s}_i(k+1)$ can be calculated based on the control $\mathbf{u}_i(k)$, so that they can be collectively expressed as

$$\chi_i(k) \triangleq [\mathbf{u}_i(k)^T \quad \mathbf{s}_i(k+1)^T]^T \tag{2.13}$$

The constraints in (2.10)-(2.12) can be rewritten as

$$\begin{aligned}\mathcal{X}_i \triangleq \{\chi_i(k) \in \mathbb{R}^5 | \ \mathbf{s}_i(k+1) = \mathbf{f}(\mathbf{s}_i(k), \mathbf{u}_i(k)),\\ \mathbf{D}\chi_i(k) \leq \mathbf{d}\}, \quad \forall i \in \mathcal{N}\end{aligned} \tag{2.14}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & -\mathbf{I}_3 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{b} & \mathbf{b} & \mathbf{a}_2 & \mathbf{a}_1 \end{bmatrix}^T \tag{2.15}$$

The state and control of the MRN can be represented as the collection of each robot's $\mathcal{X}_i$ as $\chi = [\chi_1^T(k) \ \ldots \ \chi_N^T(k)]^T$. $\chi(k)$ can be denoted as the decision variables of the optimization problem in (2.9). The search space of this optimization problem is $\mathcal{X}_1 \times \ldots \times \mathcal{X}_N$.

Directly addressing the optimization problem in (2.9) requires complete knowledge of robots and targets, which can be achieved by a central station that gathers data from all robots and optimizes control as a whole. The drawback of this kind of centralized algorithm is that the communication and computational load is high. This thesis proposes a decentralized approach to solve the problem (2.9) where each robot in MRN their own tracking utility $U_{i,j}$. The decentralized optimization approach is shown in section 3.1.

CHAPTER 3

METHODOLOGY

This chapter shows the methodology used in this thesis. In this chapter, a decentralized robot control optimization method is proposed to reduce the computational load of controlling MRN to track targets in section 3.1. Then the online sensing in section 3.2 shows the robot localization method and target estimation method which includes target detection, classification, and localization. Communication between robots is necessary as sharing information can improve tracking performance. Section 3.3 shows three types of communication used in this work. Then the target tracking utility function is specified in section 3.4. The tracking utility function contains three parts, which are information gain, target exploration, and collision avoidance. By optimizing the tracking utility function, the optimal control of each robot can be got.

3.1 Decentralized Target Tracking

To reduce the computational load of the optimization problem (2.9), a decentralized optimization approach is proposed. In this method, cooperative robots locally maximize the tracking utility function while satisfying physical constraints to achieve the optimization of the global tracking utility U_g. Each robot should solve the following problem,

$$\begin{aligned} \max_{\chi_i(k)} \quad & \sum_{j \in P_i^*(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \\ \text{s.t.} \quad & \chi_i(k) \in \mathcal{X}_i, \quad \forall i \in \mathcal{N} \end{aligned} \tag{3.1}$$

where $P_i^*(k)$ is the optimal target assignment which is assumed to be known a *priori*. The optimization result of (3.1) is represented by $\chi_i^*(k)$, containing the optimized control $\mathbf{u}_i^*(k)$ and future state $\mathbf{s}_i^*(k+1)$, based on which the robot can be controlled to track targets. The decentralized optimization approach is run concurrently on each robot in MRN to distribute the computation to enhance efficiency. The high-efficiency optimization approach makes it practical to run in real-time to control the robots to track targets.

Fig.3.1 shows the process of the decentralized optimization implemented on each robot, which has two parts, online sensing, and control. The online sensing parts contain the robot localization based on the motion sensor and the target estimation. Target detection, classification, and localization are done in the target estimation part using information from the robot's camera and the robot localization data. In the control part, each robot optimizes the local tracking utility with the given target assignment and target estimation information.

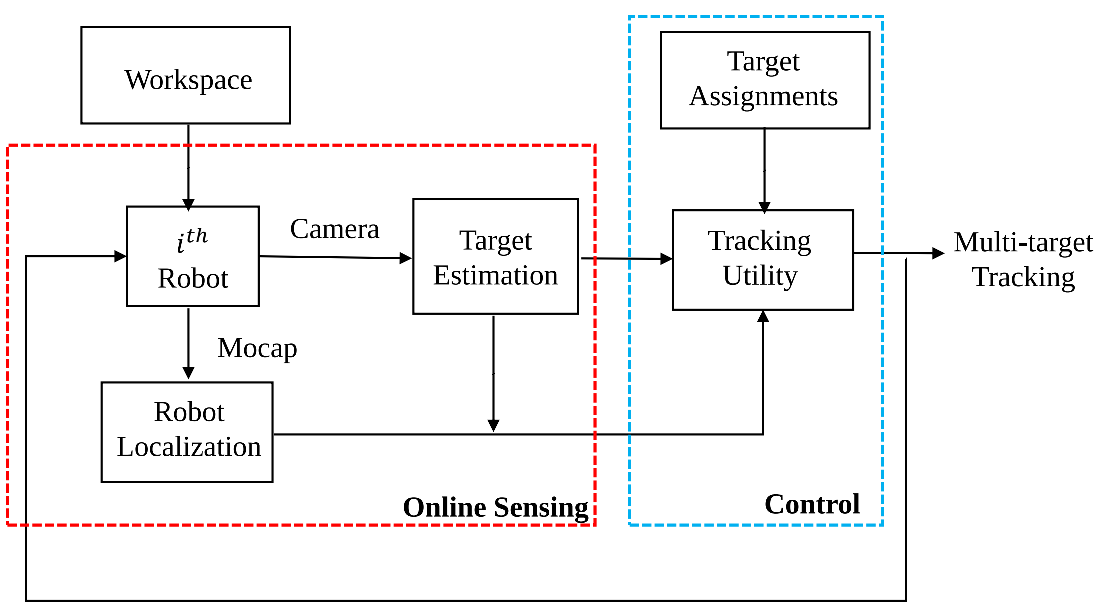

Figure 3.1: The decentralized optimization implemented on each robot.

3.2 Online Sensing

In this work, online sensing provides data for MRN control including robot localization, target detection, classification, and localization. In specific, there are two tasks for online sensing :

- Robot localization: Estimate the localization of each robot in real-time with a motion capture system (Mocap).
- Target detection, classification, and localization: Detect targets and recognize their IDs based on the images streamed from robots' onboard cameras with computer vision algorithms. Then estimate the localization of targets by combining depth data, RGB data, and robot localization data.

3.2.1 Rbobot Localization

Robot localization refers to the process of estimating the pose (position and orientation) of robot with respect to its environment. There are several well-developed methods for robot localization such as Global Positioning System (GPS) [8], Inertial Measurement Unit (IMU) [24], and Visual odometry method [1]. The GPS method is not suitable for indoor environments because the GPS signal will be blocked. IMUs are subject to drift errors that accumulate over time and result in significant pose estimation errors. The visual odometry method is sensitive to lighting conditions and errors from camera calibration. To localize robots indoors with high accuracy and robustness, a motion capture system (Mocap) OptiTrack is adopted.

OptiTrack system tracks targets with reflective markers, which are attached to the targets being tracked. These markers' surfaces are covered by a special material that can reflect infrared light emitted by the cameras in the OptiTrack system. The cameras of the OptiTrack system are placed around to cover the workspace, The cameras emit pulses of infrared light that reflect off the markers and bounce back to the cameras. The system can determine the distance between the camera and the marker by measuring the time it takes for light to travel from the camera to the marker and back. The position of each marker in 3D space is then triangulated by the OptiTrack program based on this distance information. By measuring the angles of the reflective markers with respect to the camera, the system can also determine the orientation of the markers. The OptiTrack system can precisely estimate the pose of targets in 3D space by tracking the markers on that object or subject.

In this work, ten OptiTrack Prime^{x}41 cameras are calibrated to cover the workspace, and markers are attached to robots. The pose estimation of the robot obtained from OptiTrack software is published to a Robot Operating System (ROS) topic through an OptiTrack ROS package. The robots can subscribe to the topic to get their pose in real-time.

3.2.2 Target Estimation

Fig.3.2 shows the pipeline of target detection, classification, and localization on each robot. Target detection is used to recognize human beings in images streamed from robots' onboard cameras. Then the target classification algorithm should be used to distinguish each target by recognizing each target's unique ID. Target localization estimates the position of the detected targets in

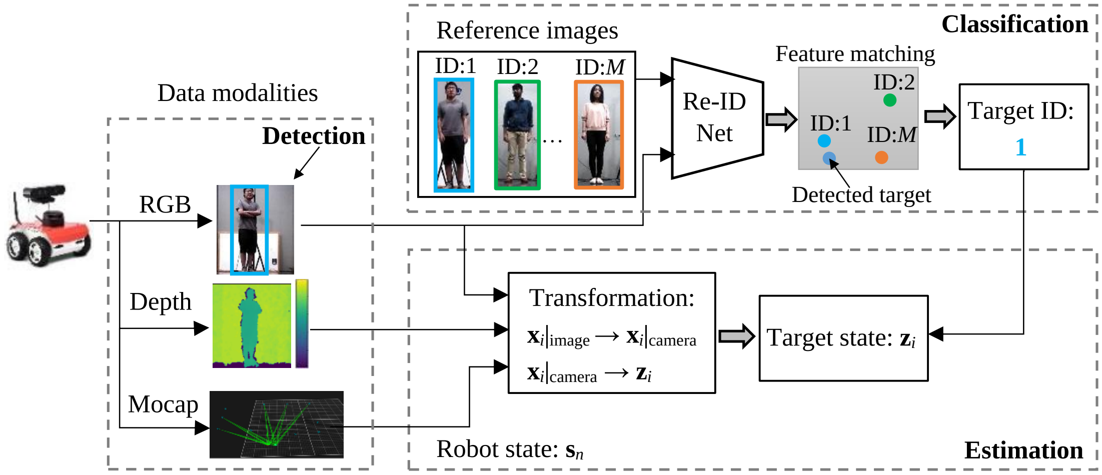

Figure 3.2: Online sensing pipeline for integrated targets detection, classification, and localization.

the initial frame.

CNN-based (Convolutional Neural Network-based) human target detection is one of the most famous approaches to detect humans from images and videos with the help of deep-learning neural networks. CNNs are able to automatically learn to recognize features from visual data. There are several famous CCN-based object detection algorithms including YOLO, Faster R-CNN, Mask R-CNN, and Cascade R-CNN [6, 19, 21, 27]. To implement human detection, CNNs can be trained using a sizable dataset of annotated human photos or videos, including both images with and without humans. The CNN gains the ability to identify visual features of people, such as body size, shape, and movement patterns, during training. CNNs can identify human targets in fresh photos or movies once it has been trained. To achieve this, the image or video is processed through CNNs, and then a probability map is created to show the presence of humans in that image or video. The position of human targets can be indicated by a binary mask created through thresholding the probability map. In this work, a Mask R-CNN detector is used to detect human targets by drawing bounding boxes on images to represent the position of the human in the image frame.

In this work, each robot needs to be able to distinguish between the targets it is tracking and those it is not since this study attempts to track many targets, with each robot only tracking a subset of targets. The aim of target classification is to enable robots to differentiate targets based on robot RGB imagery. Each target is known to each robot by a reference image with a pre-specified unique ID. Target classification works by matching IDs to the detected targets by comparing the detected target to reference images. The ID of the reference that most resemble the detected target will be matched to that detected target. The viewpoint from which robots observe targets shifts over time as the robots and targets move. Target classification needs to be robust so that it can avoid ID-switching as viewpoint changes.

To achieve this, a type of deep learning architecture used for person re-identification, Re-ID Net is adopted for target classification [30, 41]. Re-ID Net consists of two main components, which are a feature extraction network and a metric learning network. The feature extraction network is used to extract informative features of targets from images. It utilizes a convolutional neural network (CNN) to produce a vector of features based on images of targets. These extracted features are emblematic of the person's characteristics, which are robust to changes in viewpoint, pose, and other factors. The metric learning network aims at learning a distance function that can be used to differentiate different people. The distance metric learned by the metric learning metric network can evaluate the similarity between the feature vectors extracted from images. The extracted features from the feature extraction network are the input of the metric learning network. The output of the metric learning network is a score ranging from 0 to 1, which is called ReID score that describes the similarity between the features from low to high.

Fig.3.3 shows the target classification process implemented on each robot. There is a gallery folder that contains reference images for each target-of-interest with pre-specified IDs. The Re-ID Net extracts features from each detected target and compares those features with each gallery reference image's features. The ReID-Net then generates a ReID score array which consists of the ReID score for the detected target and each reference image. The highest ReID score and corresponding ID of that reference image are output. If the highest ReID score exceeds a critical value, it means the similarity between the detected target and one of targets-of-interest is high. Then the ID of that target-of-interest is matched to this detected target. If the highest ReID score is smaller than the critical value, that means the detected target does not match any targets-of-interests. This detected target may be an intruder that does not need to be tracked and no ID will be associated with this target.

After a detected target is associated with an ID, it needs to be localized by robots' RGBD cameras. In existing research on vision-based target estimation, targets positions are usually measured using a monocular camera in the image frame or virtual image plane, leading to complicated non-linear sensor measurement models [15, 26, 42]. This work adopts the ray tracing method to directly measure the position of targets in the global frame. The center pixel of the detected target's bounding box got from the detection algorithm in the image coordinate can be used to represent the position of the target in the image ref-

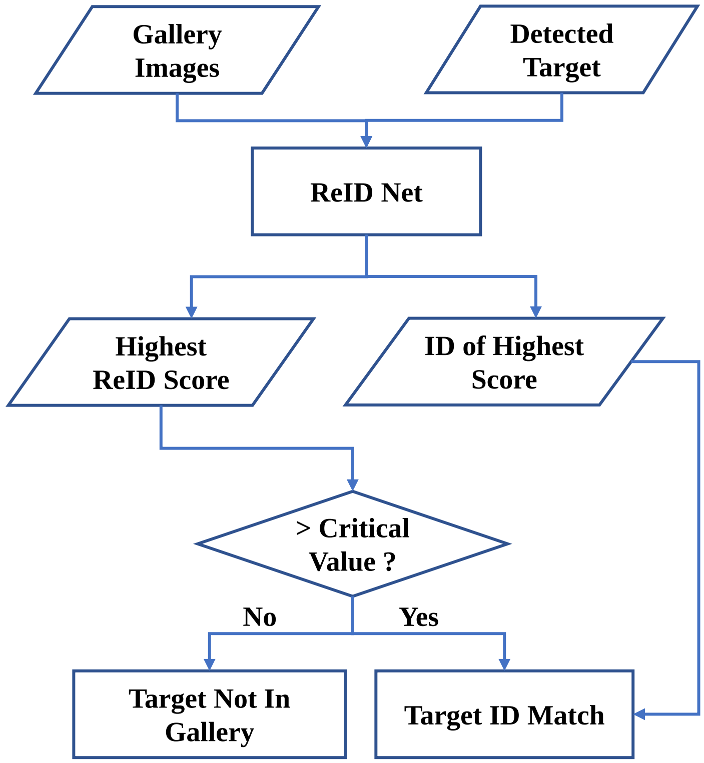

Figure 3.3: Pipeline for target classification implemented on each robot

erence frame. The position of the j^{th} target in image reference frame is denoted by $x_j|_{\text{image}}(k) \in \mathbb{R}^{2\times1}$. As the RGB image and the depth image share the same image reference frame, the target depth $d_j(k)$ can be got by the value of the point $x_j|_{\text{image}}(k)$ in the depth image. Let $\mathcal{F}_{\mathcal{A}_i}$ be the i^{th} robot's camera frame, the target position in the camera is estimated by

$$\mathbf{x}_j|_{\text{camera}}(k) = d_j(k)K^{-1}[\mathbf{x}_j|_{\text{image}}(k) \quad 1]^T \tag{3.2}$$

where $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix.

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

in which f_x and f_y are the focal length, s is the skew, and (c_x, c_y) represents the principal point. The measurement of the j^{th} target position by i^{th} robot $\mathbf{z}_{i,j}(k)$, is given by transforming $\mathbf{x}_j|_{\text{camera}}(k)$ to initial frame $\mathcal{F}_{\mathcal{W}}$

$$\mathbf{z}_{i,j}(k) = \mathbf{R}_i(k)\mathbf{x}_j|^T_{\text{camera}}(k) + \mathbf{t}_i^T(k) \tag{3.4}$$

where $\mathbf{R}_i(k)$ and $\mathbf{t}_i^T(k)$ denote camera extrinsic parameters got based on robot state $\mathbf{s}_i(k) = [x_i(k) \quad y_i(k) \quad \theta_i(k)]^T$,

$$\mathbf{R}_i(k) = \begin{bmatrix} cos(\theta_i(k)) & -sin(\theta_i(k)) & 0 \\ sin(\theta_i(k)) & cos(\theta_i(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{t}_i(k) = [x_i(k) \quad y_i(k) \quad 0]^T \tag{3.5}$$

It can be found that there are no velocity terms in the measurement model $\mathbf{z}_{i,j}(k)$. There is an assumption that the measurement model is subjected to additive Gaussian noise $\mathbf{v}(k)$ so that the complete measurement model is

$$\mathbf{z}_{i,j}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_j(k) + \mathbf{v}(k) \tag{3.6}$$

which completes the measurement model specified in (2.5)

3.3 Communication

Communication is essential for this work as it enables the robots to share information about environments and their observations, which helps achieve the tracking objective more efficiently and accurately. The communication in this work is assumed to be no time delay and the communication range is unlimited. The configuration of the communication is shown in Fig.3.4. There are three types of communication in this work:

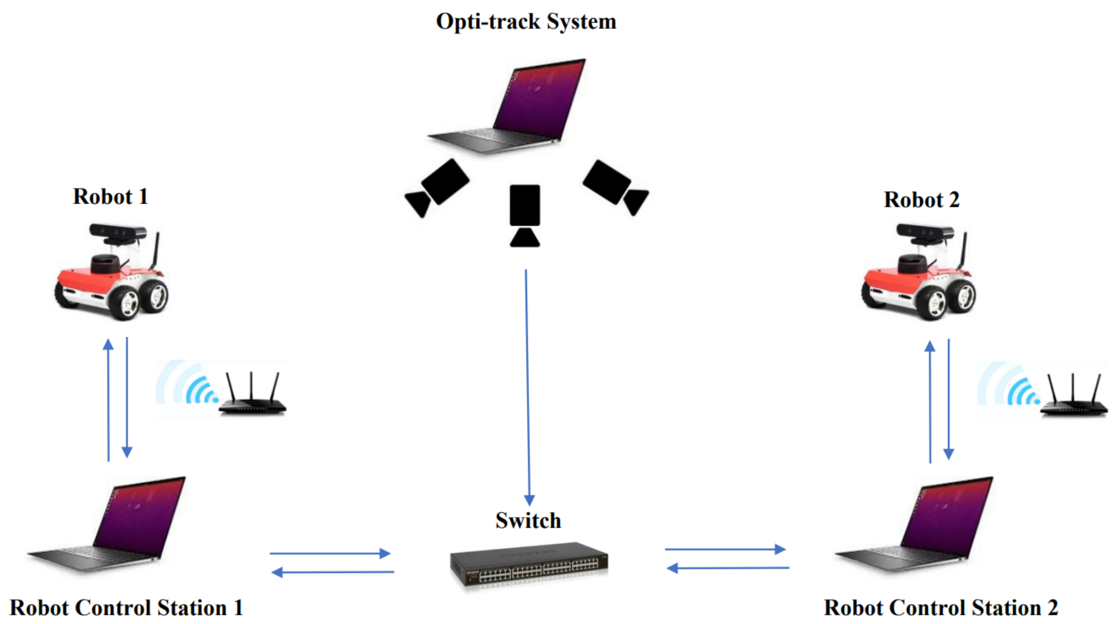

Figure 3.4: Configuration of Communication

- Communication between OptiTrack system and robot control stations: The OptiTrack system shares the robots' pose estimation through a switch with each robot control station.
- Communication between robots and robot control stations: Each robot has a control station, and robots communicate with their control stations by Robot Operating System(ROS) in a local area network (LAN). Robots stream the RGBD images to control stations for processing. Robot control stations send optimal control commands to their robots, which are obtained from robot control stations.
- Communication between robot control stations: Robot control stations finish target states estimation based on the RGBD images and share the target states with other control stations. With robot localization and target states estimation information, the robot control stations obtain the optimal control of robots by optimizing the tracking utility function which is shown

in section 3.4.

3.4 Target Tracking Utility Function

Once the measurements of targets are obtained, a target assignment algorithm should be applied to assign targets to each robot to track. This work focuses on controlling robots to track targets, the target assignment is assumed to be known a *priori*. As mentioned in section 3.1, each robot maximizes its local tracking utility to get optimal control of MRN. The existing study shows the strengths of the information-driven approach in target tracking which are real-time performance, high, accuracy, and wide adaptability [10, 23, 26, 29]. Information-driven approaches can simultaneously address multiple performance criteria for managing sensors for target tracking. With this idea, this work develops an information gain which is the expected entropy reduction(EER) in the target state to control MRN to track targets. During target tracking, collision avoidance with obstacles and target exploration for lesser-tracked targets are also in consideration. This work develops a tracking utility function that solves the trade off between information gain for tracking, target exploration, and collision avoidance.

3.4.1 Information gain (Expected Entropy Reduction)

This work takes the expected entropy reduction(EER) as the information gain to control MRN to track targets. Entropy evaluates the amount of uncertainty or randomness of a system. The entropy used in target tracking reflects the uncertainty of the estimation of the target's pose. The entropy increases with the increasing uncertainty of the robot in the target's estimation. The EER approach works by predicting the entropy reduction if the robots move to a particular position. By calculating the entropy reduction of each possible pose of the robot, the pose that results in the most signification entropy reduction can be found. That possible pose will be chosen as the next waypoint of the robot. The EER is calculated based on the prior distribution and posterior distribution of the target estimation, which are the uncertainty about the target's localization before and after the observation. The observation is the target's position estimation from online sensing, which can be used to update the prior distribution and obtain a posterior distribution.

Let $Z_{i,j}(k+1)$ denote the one-step future measurement of the j^{th} target by the i^{th} robot, which is modeled as Bernoulli random finite set (RFS). If the estimated future state of target ($\hat{\mathbf{x}}_{i,j}(k+1)$) is within the planned filed-of-view of the robot ($\mathcal{S}_i(k+1)$) , $Z_{i,j}(k+1)$ will contain a measurement $\mathbf{z}_{i,j}(k+1)$. Otherwise, $Z_{i,j}(k+1)$ will be an empty set.

$$Z_{i,j}(k+1) = \begin{cases} Z_{i,j}(k+1) = \{\mathbf{z}_{i,j}(k+1)\} & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \in \mathcal{S}_i(k+1) \\ Z_{i,j}(k+1) = \emptyset & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \notin \mathcal{S}_i(k+1) \end{cases} \tag{3.7}$$

The probability density function of one-step future measurement is

$$f(Z_{i,j}(k+1)) = \begin{cases} p_D \cdot g(\mathbf{z}_{i,j}(k+1)) & \text{if } Z_{i,j}(k+1) = \{\mathbf{z}_{i,j}(k+1)\} \\ 1 - p_D & \text{if } Z_{i,j}(k+1) = \emptyset \end{cases} \tag{3.8}$$

in which $g(\cdot)$ is a Gaussian distribution got based on measurement model (2.5). p_D represents the target detection probability which obeys the Bernoulli probability distribution. This work assumes that there is no missed detection if the estimated future target state is within the planned field of view of the robot.

$$p_D = \begin{cases} 1 & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \in \mathcal{S}_i(k+1) \\ 0 & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \notin \mathcal{S}_i(k+1) \end{cases} \tag{3.9}$$

The prior and posterior distribution (covariance matrix) of target state estimation are denoted by $\mathbf{\Sigma}_{i,j}(k+1|k)$ and $\mathbf{\Sigma}_{i,j}(k+1|k+1)$, respectively. The prior distribution reflects the robot's uncertainty of the target's state before a future measurement and the posterior distribution is the uncertainty after a future measurement. These two distributions can be updated through

$$\mathbf{\Sigma}_{i,j}(k+1|k) = \mathbf{F}\mathbf{\Sigma}_{i,j}(k|k)\mathbf{F}^T + \mathbf{Q} \tag{3.10}$$

$$\begin{aligned} &\mathbf{\Sigma}_{i,j}(k+1|k+1) \\ &= \begin{cases} (\mathbf{I} - \mathbf{K}(k)\mathbf{H})\mathbf{\Sigma}_{i,j}(k+1|k) & \text{if } Z_{i,j}(k+1) = \{\mathbf{z}_{i,j}(k+1)\} \\ \mathbf{\Sigma}_{i,j}(k+1|k) & \text{if } Z_{i,j}(k+1) = \emptyset \end{cases} \end{aligned} \tag{3.11}$$

where the prior distribution is updated based on the dynamic model of the target. The posterior distribution is updated with Kalman filtering if the future measurement is non-empty, otherwise, it is equal to the prior distribution. The entropy reduction of the target state estimation could be calculated through [25, 28],

$$R_{i,j}(Z_{i,j}(k+1)) = \frac{1}{2}\log \frac{|\mathbf{\Sigma}_{i,j}(k|k)|}{|\mathbf{\Sigma}_{i,j}(k+1|k+1)|} \tag{3.12}$$

in which $|\cdot|$ is matrix determinant. The information gain, which is EER in this work, is the expectation of the future measurement, [26]:

$$I_{i,j} = \mathbb{E}_{Z_{i,j}(k+1)}[R_{i,j}(Z_{i,j}(k+1))] \tag{3.13}$$

the derivation is shown below, $Z_{i,j}(k+1)$ is denoted by $Z_{i,j}$ for brevity in this part:

$$
\begin{aligned}
I_{i,j} &= \mathbb{E}_{Z_{i,j}}[R_{i,j}(Z_{i,j})] \\
&= \int R_{i,j}(Z_{i,j})\; f(Z_{i,j})\; \delta Z_{i,j} \\
&= (1-p_D)\; R_{i,j}(Z_{i,j}=\emptyset) \; + p_D \;\int R_{i,j}(Z_{i,j}=\mathbf{z}_{i,j})\; g(Z_{i,j}=\mathbf{z}_{i,j}) d\mathbf{z}_{i,j} \\
&= (1-p_D)\; R_{i,j}(Z_{i,j}=\emptyset) + p_D\; R_{i,j}(Z_{i,j}=\mathbf{z}_{i,j})
\end{aligned}
\tag{3.14}
$$

3.4.2 Target Exploration and Collision Avoidance

This work also addresses target exploration and collision avoidance in addition to the information gain that drives robots to track targets. As each robot may be assigned more than one target, some targets may be lost in some time instant. Target exploration can attract robots to explore the lesser-tracked targets. There could be obstacles in the workspace, the collision avoidance aims at preventing collision between robots and obstacles.

To achieve target exploration, a navigation reward is developed considering the geometry of the bounded and directional robot field-of-view. The cumulative tracking time of the j^{th} target until time instant k can be obtained through

$$t_j(k) = \sum_{\kappa=1}^{k} \Big(\sum_{i=1}^{N} \mathbf{1}(\mathbf{x}_j(\kappa) \in \mathcal{S}_i(\kappa)) \cdot \mathbf{1}(j \in P_i(\kappa)) \Big) \tag{3.15}$$

Then the time of the j^{th} target is missed until time step k is

$$\tau_j(k) = k - t_j(k) \tag{3.16}$$

where the larger of $\tau_j(k)$ indicates the longer time the target is missed by robots. The navigation reward that can attract the i^{th} robot to track the j^{th} target is defined as

$$J_{i,j} = -\tau_j(k) \cdot \oint_{\mathcal{S}_i(k+1)} \|\mathbf{p}_{x,y} - \hat{\mathbf{x}}_{i,j}(k+1)\| \, dxdy \tag{3.17}$$

where $\mathbf{p}_{x,y} \in \mathcal{W}$ denotes the position of an arbitrary point within the workspace and the integration over the field-of-view of the i^{th} robot $\mathcal{S}_i(k+1)$ considers its boundary and direction. For brevity, the navigation reward is the missed tracked time multiplied by the sum of the distance between the arbitrary point in $\mathcal{S}_i(k+1)$. A negative sign is also added, which makes the higher the navigation reward, the more attractive to robots to explore the missed targets. The negative sign is to make the navigation reward has a maximization framework like the information gain.

Collision avoidance is implemented by defining a collision penalty. The obstacles in the workspace are denoted by $\mathcal{B} \subset \mathcal{W}$. There will be a penalty if the planned states of the robot will collide with the obstacles or the moving targets in the workspace:

$$C_{i,j} = \gamma\Big(\mathbf{1}(\|\mathbf{s}_i(k+1) - \hat{\mathbf{x}}_{i,j}(k+1)\| \leq \epsilon) + \mathbf{1}(\mathbf{s}_i(k+1) \in \mathcal{B})\Big) \tag{3.18}$$

where $\gamma \in \mathbb{R}^+$ is a constant, which reflects the penalty, $\epsilon \in \mathbb{R}^+$ is the minimum distance to ensure the robot will not collide with obstacles and moving targets. The value of ϵ can be decided based on the geometry of the robot.

By combing target exploration and collision avoidance with EER, the tracking utility function of the i^{th} robot about j^{th} target is developed as

$$U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) = I_{i,j} + J_{i,j} - C_{i,j} \tag{3.19}$$

Although the robot control $\mathbf{u}_i(k)$ does not appear directly in the tracking utility function, it impacts the planned robot state via the robot motion model (2.1), therefore influencing the tracking utility.

3.4.3 Optimization

The objective function of the decentralized control optimization algorithm for target tracking (3.1) is specified by (3.19). Each robot optimizes the local tracking utility(3.1) to get the optimal control of MRN. This decentralized optimization problem is discontinuous, non-convex, and multimodal. The discontinuity results from the collision avoidance ($C_{i,j}$) and the field-of-view model used in information gain ($I_{i,j}$). It is a multimodal function with non-convexity because it sums the tracking utility of multiple assigned targets ($|P_i^*(k)| \geq 1$). The traditional optimization methods have requirements for objective functions. For example, gradient-based methods require objective functions that are differentiable, and sub-gradient methods require the existence of sub-gradients at each point of the objective functions. The characteristics of this decentralized optimization problem indicate that traditional optimization methods are unsuitable for this problem. Modern metaheuristic algorithms can be used to solve complex problems that have no requirements for the characteristics of objective functions. This work uses the Genetic Algorithm (GA), a metaheuristic algorithm that draws inspiration from the ideas of genetics and natural selection. Existing research shows the good performance of GA in non-convex and discontinuous optimization problems [2, 11, 18, 46].

The GA solves the optimization problem by creating a population of candidate solutions first, which is represented by parameters or genes. Then a fitness function is used to evaluate these solutions in solving the problem. If there are no solutions that are good enough to solve the problem, GA will then create a new generation of solutions based on the present population by using some operations such as selection, crossing over, and mutation. In the selection process, the fittest solutions from the current population of solutions are selected as parents of the next generation's solutions. The crossover operation then creates a new offspring solution by recombining the genes of two parents. The mutation operation introduces new variations to solutions by randomly changing some genes of an individual solution. The fitness function is used each time when a new generation of solutions is created. This process is repeated until a good solution is found or a stopping criterion is met.

The fitness function in this work is the tracking utility function. The optimized solution is $\chi_i^*(k)$, which includes the one-step future state and the robot control. After the optimized solution is got, robots will be controlled based on the obtained optimal control to reach the obtained one-step future states. After robots reach the obtained future states are reached, GA will run again to generate new optimal solutions to control robots. By running GA recursively, robots can be continuously controlled to track targets while avoiding collision and explicating lesser-tracked targets.

3.4.4 Computational Complexity Analysis

The computational complexity of the optimization process is determined by the calculation of the tracking utility function and the number of iterations the GA algorithm runs. The number of iterations is represented by N_G, and the time each GA iteration takes is neglected. The tracking utility used in each GA iteration has three parts: information gain ($I_{i,j}$), target exploration navigation reward ($J_{i,j}$), and collision avoidance ($C_{i,j}$). The complexity analysis is done separately for these three parts. According to (3.13), the complexity of information gain calculation is $\mathcal{O}(M)$. The field-of-view of the robot is assumed to be discretized into L grids to approximate the integration in navigation reward ($J_{i,j}$) in (3.17), then the computational complexity of $J_{i,j}$ is $\mathcal{O}(LM)$. Collision avoidance ($C_{i,j}$) considers preventing the robot from colliding with both targets and obstacles as shown in (3.18). Then the time complexity of collision avoidance is $\mathcal{O}(M + |\mathcal{B}|)$. By summing up these three parts, the complexity of the entire process is

$$\mathcal{O}(N_G(M + LM + |\mathcal{B}|)) \tag{3.20}$$

It shows that the decentralized control optimization algorithm has polynomial time complexity as its running time can be expressed as a polynomial function of the size of the input. It is considered efficient and practical for this real-time target tracking problem, as its running time does not grow exponentially with the size of the input.

CHAPTER 4

PHYSICAL EXPERIMENTS AND RESULTS

To validate the proposed approach for target tracking, several physical experiments are designed. The experiment setup is shown in Section 4.1, which demonstrates the workspace and robots used in experiments. The online sensing algorithms are validated in Section 4.2 using a robot to observe a target that moves along a pre-defined trajectory. The proposed tracking algorithm is first tested in Section 4.3 by using a robot to track a single target. Then in Section 4.4, a target is tracked by a robot in the workspace with obstacles to show the tracking algorithm can avoid collision with obstacles. In Section 4.5, three targets are tracked by two robots to show the tracking algorithm can be used to control multiple targets with multiple robots.

4.1 Experiment Setup

The robots used in this work are Huasarion ROSBots as shown in Fig.4.1. The robots are equipped with Orbecc Astra RGBD cameras, which can provide both RGB and depth images for target classification and localization. Robots are armed with WiFi Antennas to achieve communications (Section 3.3). Robots also have odometry sensors including IMU and encoder which can be used to localize robots by the dead reckoning method. However, the localization accuracy of dead reckoning is low because the error will accumulate over the movement of robots. A motion capture system, OptiTrack, is then adopted for robot localization (Section 3.2.1). Robots are run in an indoor environment as shown in Fig.4.2. Ten OptiTrack Prime^{x}41 cameras are installed around the workspace for robot localization.

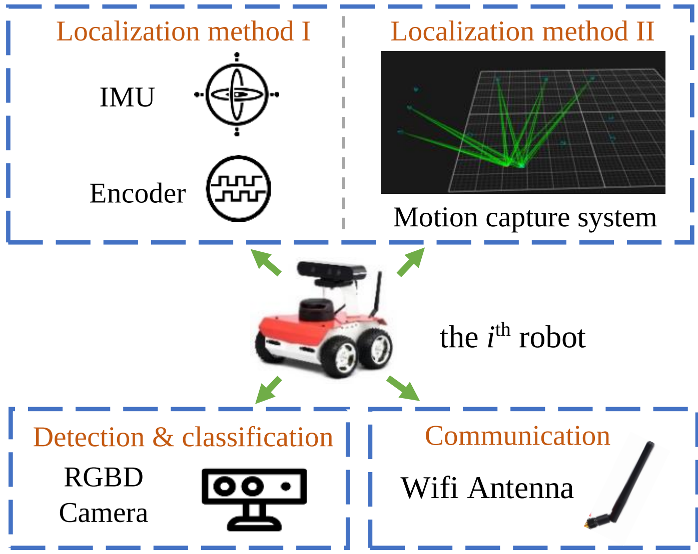

Figure 4.1: The Husarion ROSBots with various sensing capabilities.

Figure 4.2: The indoor workspace.

4.2 Validation of Online Sensing Algorithms

The aim of this section is to validate the online sensing algorithm (Section 3.2) including target detection, classification, and localization. In this experiment, one robot is used to observe a target move in a pre-defined planned trajectory. The robot stays static in this experiment and the target is moved within the field-of-view of the robot. Fig.4.3 shows several frames of images from the robot's onboard camera in one of the experiments.

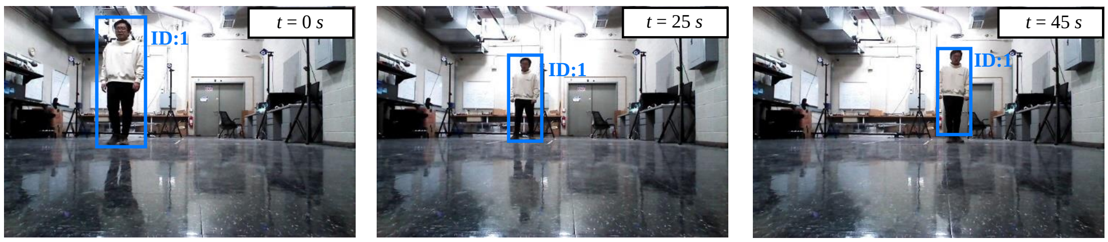

Figure 4.3: Demonstration of Online Sensing in real-time

Target detection is implemented with Mask R-CNN used by drawing bounding boxes on images to represent the position of the human in the image frame which is shown in Fig.4.3. During the experiment from $t = 0s$ to $t = 45s$, the target can be accurately detected as the target moves, so that the feasibility of the target detection algorithm is proved. The ReID Net implements the target classification to recognize the ID of the detected target. The reference images used in this experiment are shown in Fig.4.4, and the target in this experiment is the target whose ID is 1. The recognized ID of the detected target got from ReID Net is shown next to the bonding box in Fig4.3. The recognized ID is correct during the movement of the target which validates the target classification algorithm. The target localization is implemented based on RGB images, depth images, and robot localization by (3.2)-(3.5). Fig.4.5 shows the robot pose, the planned target trajectory, and the estimated target trajectory obtained from target localization. It shows that the estimated trajectory is very close to the planned trajectory, which shows the high accuracy of the target localization algorithm.

In summary, this experiment validates the online sensing algorithm in real-time. The target can be accurately detected, classified, and localized. With accurate online sensing results, robots can be controlled to track targets. The decentralized tracking algorithm is tested in the next section.

Figure 4.4: Reference images for target classification

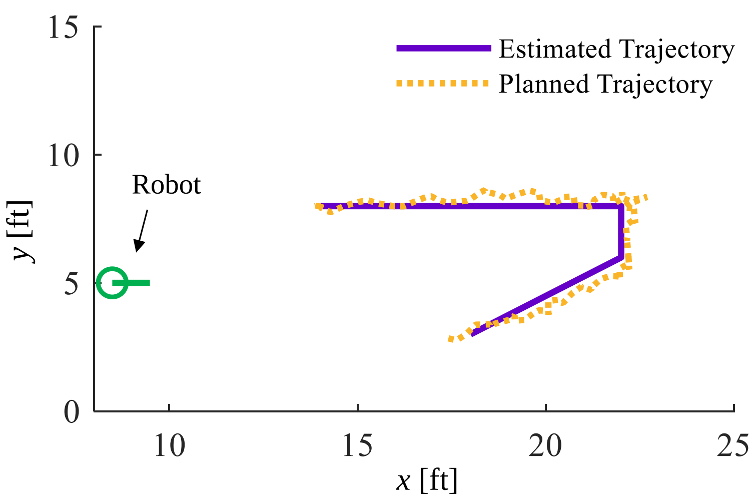

Figure 4.5: Estimation of Target Trajectory

4.3 Single Target Tracking

The aim of this experiment is to validate the target tracking ability. In this experiment, one target moves in the workspace, and one robot is used to track that target. The robot is controlled by optimizing the tracking utility function (3.19) to move to keep the target in the field-of-view of the robot. The initial state of the target and the reference images are given to the robot a-*priori*. The ID of the target in the experiment is 1. Fig.4.6 is a series of frames of the experiment from the robot's camera.

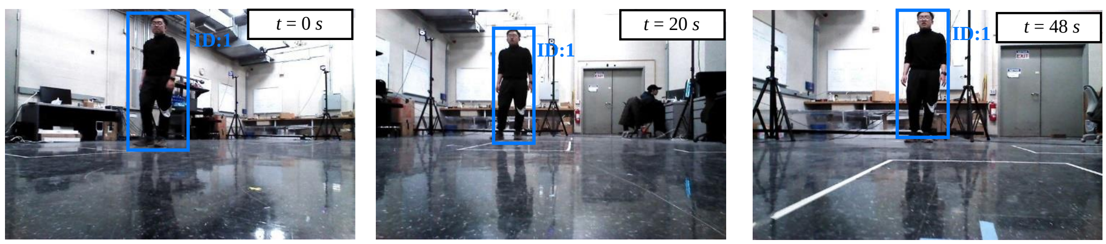

Figure 4.6: Demonstration of single tracking experiment

The target is detected and recognized correctly over the movement of the robot which shows the robustness of the online sensing to the viewpoint changes. During the experiment, The target is kept in the field-of-view of the robot as the target is always shown in images from the camera. The trajectories of the robot and the target are shown in Fig.4.7. It shows that the robot moves to follow the target in a desired distance and the target is in the field-of-view.

The single target tracking experiment was repeated successfully for more than 20 trials, which shows the feasibility and reliability of the proposed tracking algorithm. In this experiment, there is no obstacle in the workspace. To test the collision avoidance of the proposed algorithm, the single tracking experiment is repeated in the next section with an obstacle in the workspace.

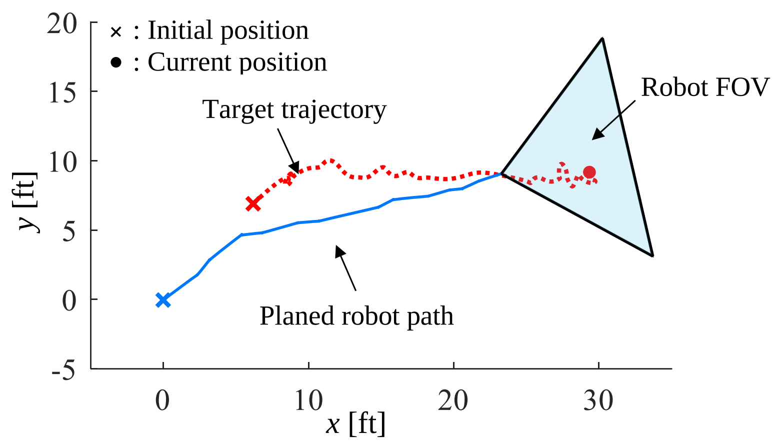

Figure 4.7: Trajectories of the target and robot for single target tracking

4.4 Single Target tracking with an obstacle in the workspace

Collision avoidance is essential to ensure safety in target tracking. The proposed target tracking algorithm considers collision avoidance by adding a collision penalty in the tracking utility function (3.18 - 3.19). In this experiment, there is still one single target to be tracked by a robot. The trajectory of the target is designed to be the same as the target trajectory in Section 4.3. The initial position of the robot is also the same as the initial position in Section 4.3. Thus, the planned robot trajectory would also be similar to the previous section if there is no obstacle in the workspace. This experiment adds an obstacle in the workspace which is in the planned path of the robot. The workspace with the obstacle is shown in Fig4.8. The proposed collision avoidance approach could be validated if the robot's trajectory is changed to bypass the obstacle. A series of frames of the experiment are shown in Fig.4.9.

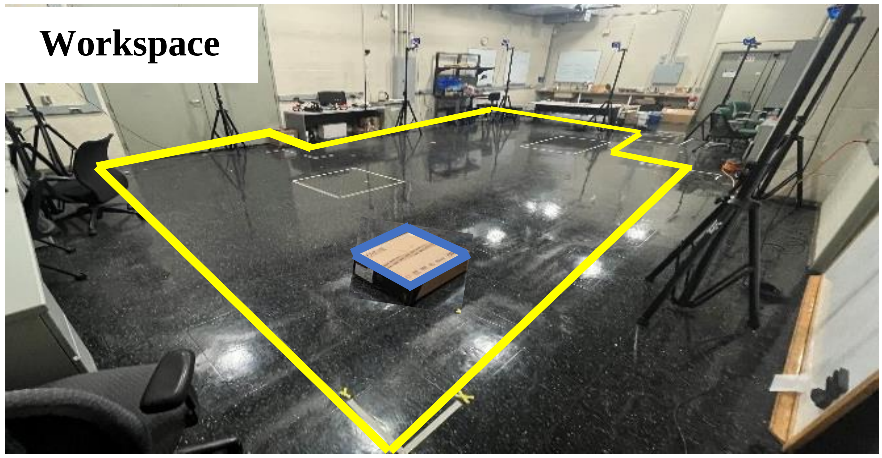

Figure 4.8: The workspace with an obstacle in it.

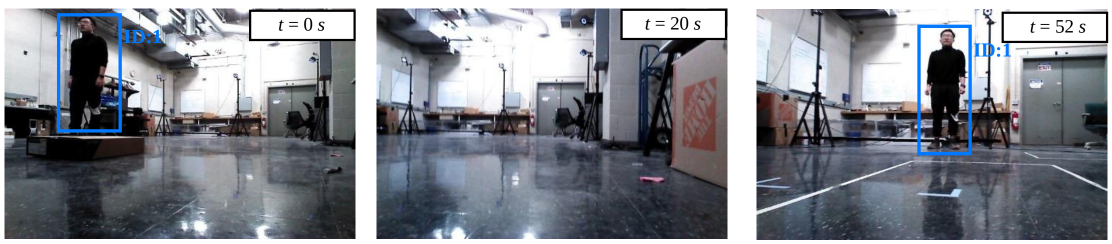

Figure 4.9: Demonstration of single tracking with obstacle

It can be found that at $t = 20s$, the target is not in the field-of-view, as the robot plans the path to avoid collision with the obstacle at that time. The robot tracks the target based on the prediction of the target dynamics (2.6) when the target is out of the field-of-view. Then the robot's field-of-view covers the target again and the robot implements the target tracking. Fig.4.10 shows the trajectories of the target and the robot which can be compared with Fig.4.7. In target tracking without obstacles, the robot gets close to the target directly and follows the trajectory of the target. As for target tracking with an obstacle in the workspace, the robot does not follow the target path directly. Instead, the planned path bypasses the obstacle and then gets close to and follows the target.

This experiment validates that the proposed decentralized target tracking algorithm can avoid collision with obstacles. From now on, the algorithm is tested on a single robot with a single target. In the next section, the algorithm is tested with multi-target tracked by multi-robot.

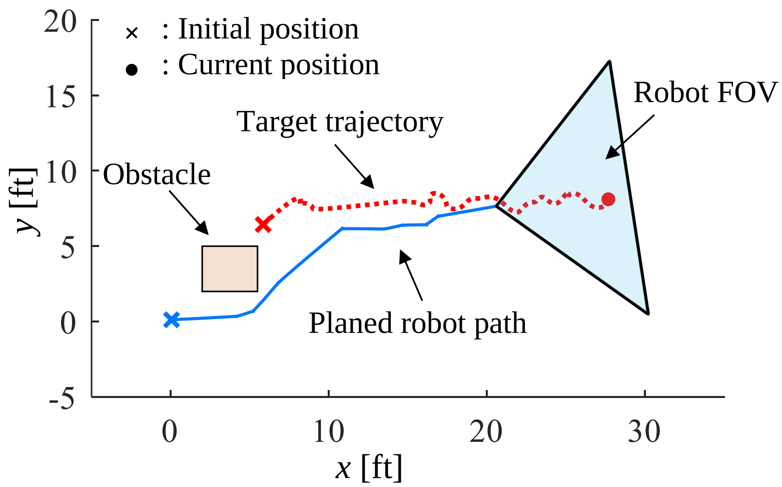

Figure 4.10: Trajectories of the target and robot for single target tracking with an obstacle in the workspace

4.5 Multiple targets tracked by multiple robots

This section aims at testing the decentralized tracking algorithm by using multi-robot to track multi-target. There are three targets and two robots in the obstacle-free workspace. The target assignment, initial states of targets, and their reference images with unique IDs were known to the robots a-*priori*. Consequently, the studies imitate a real-world tracking application in which certain preliminary locations and visual clues of the targets are provided to robots responsible for tracking them. Communication between robots is adopted for communicating the position of targets for collision avoidance (3.18). In this experiment, target 1 and target 3 are moving in the same direction and target 2 is moving in the opposite direction. The challenge of this experiment is that there is a time the targets are close to each, which challenges the target classification and has an influence on tracking performance. To demonstrate this experiment, a surveillance camera that covers the workspace was used to record this experiment. A series of frames from that camera is shown in Fig.4.11.

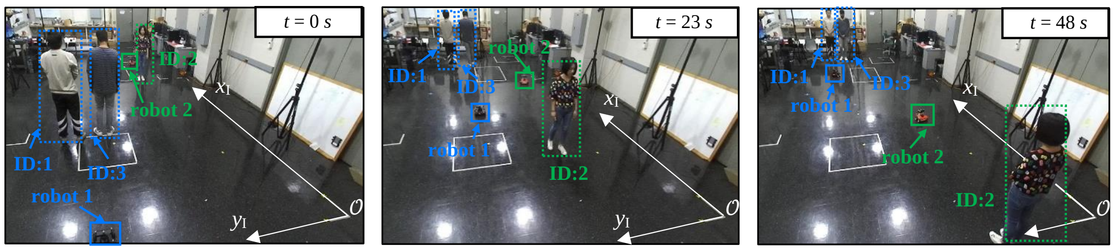

Figure 4.11: Demonstration of a robot network tracking three moving targets with the view of the recording camera.

As shown in Fig.4.11, robots are labeled with the same color as the targets assigned to them, which means target 1 and target 3 are assigned to robot 1, and target 2 is assigned to robot 2. It can be found that during the experiment, robots consistently track the targets assigned to them. Robots are not confused when three targets pass each other(from $t = 0s$ to $t = 23s$), which shows the robustness of the target classification. The trajectories of robots and targets are shown in Fig.4.12, where the colors of the robots' trajectories are the same as the colors of the trajectories of targets that are assigned to them. The trajectories also show that robots consistently tracked the targets assigned to them.

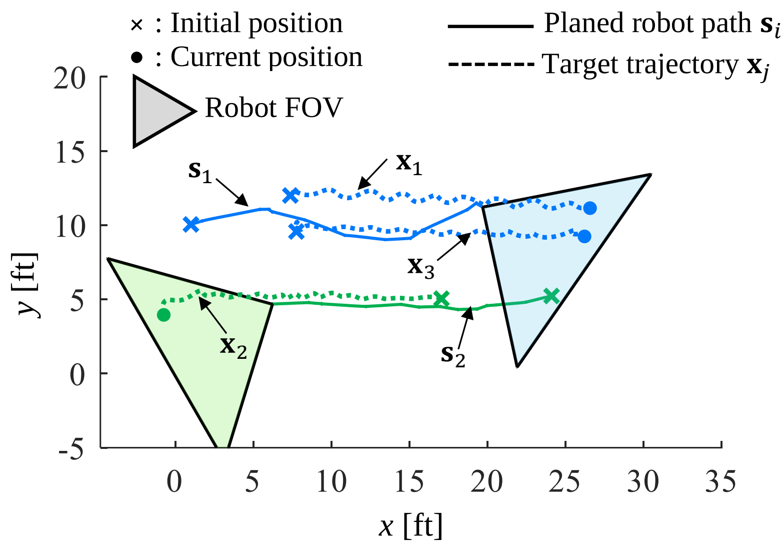

Figure 4.12: The trajectories of robots and targets for multi-robot multi-target tracking

This experiment incorporates online sensing, communication, and target tracking together, which is a complete process of multi-target tracking with MRN. It has been successfully repeated more than 20 times, which shows the feasibility of the proposed decentralized target tracking algorithm.

CHAPTER 5

FUTURE WORK

5.1 Target Assignment

In this work, the target assignment for robots is fixed and known a-*priori*. As targets move independently in the workspace, targets assigned to the same robot can be very far apart from each other after some time. It is very hard to cover targets that are far from each other with the field-of-view of the robot. To resolve this, one of the directions of future work is to develop a target assignment algorithm to dynamically assign targets to robots during tracking. The states of robots and targets are shared through the communication network. In particular, the target assignment can be completed based on the positions of targets and robots, which can assign targets to robots that are closer to those targets while avoiding conflicts. The target assignment is not fixed and can be changed to improve the tracking efficiency.

5.2 Mixed Human-Robot Team For Target Tracking

The directional and bounded field-of-view of the robot cannot cover the workspace, which means the target maybe not be observed for some time. The accuracy of state estimation of targets may be affected because of the lack of observation. Future work can improve this by building a mixed human-robot team for target tracking. Besides the MRN used in this work, a human operator can cooperate with the MRN to update the target states. The human operator can observe the workspace through a surveillance camera that covers the whole workspace. The human operator can communicate with the MRN to update the states of the target which are lost by the robots.

5.3 Target State Estimation

In this work, the state of each target is estimated by the robot to which is assigned, and then the estimated state is shared with other robots. That means each robot estimates the target states only based on its observation. However, the target may be observed by more than one robot for some time. The potential future work is to utilize the observation from other robots to improve the accuracy of the target estimation. Robots can communicate with each other to get observation from other robots and then fusing the measurements to improve the accuracy [35].

CHAPTER 6

CONCLUSION

This work proposes an approach to control multi-robot network for dynamic target tracking. Firstly, the online sensing algorithm is developed which implements robot localization, target detection, classification, and localization. Then a information-driven based tracking utility function is proposed. Each robot locally optimizes the utility function to get the optimal control to track targets. Then the proposed approach is tested through physical experiments. The experiments show that the proposed approach is feasible in real-time in physical world.

BIBLIOGRAPHY

[1] Wilbert G. Aguilar, Guillermo A. Rodríguez, Leandro Álvarez, Sebastián Sandoval, Fernando Quisaguano, and Alex Limaico. On-board visual slam on a ugv using a rgb-d camera. In YongAn Huang, Hao Wu, Honghai Liu, and Zhouping Yin, editors, *Intelligent Robotics and Applications*, pages 298–308, Cham, 2017. Springer International Publishing.

[2] Nima Amjady and Hadi Nasiri-Rad. Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm. *Expert Systems with Applications*, 37(7):5239–5245, 2010.

[3] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[4] Jacopo Banfi, Jérôme Guzzi, Francesco Amigoni, Eduardo Feo Flushing, Alessandro Giusti, Luca Gambardella, and Gianni A Di Caro. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Autonomous Robots*, 43(3):665–680, 2019.

[5] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 476–481 vol.1, 2000.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] Y.U. Cao, A.S. Fukunaga, A.B. Kahng, and F. Meng. Cooperative mobile robotics: antecedents and directions. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 226–234 vol.1, 1995.

[8] F. Capezio, A. Sgorbissa, and R. Zaccaria. Gps-based localization for a surveillance ugv in outdoor areas. In *Proceedings of the Fifth International Workshop on Robot Motion and Control, 2005. RoMoCo '05.*, pages 157–162, 2005.

[9] Jesús Capitan, Luis Merino, and Aníbal Ollero. Cooperative decision-making under uncertainties for multi-target surveillance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84(1):371–386, 2016.

[10] Micah Corah and Nathan Michael. Scalable distributed planning for multi-robot, multi-target tracking. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 437–444. IEEE, 2021.

[11] Tansel Dokeroglu, Ender Sevinc, Tayfun Kucukyilmaz, and Ahmet Cosar. A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering*, 137:106040, 2019.

[12] Silvia Ferrari and Thomas A Wettergren. *Information-Driven Planning and Control*. MIT Press, 2021.

[13] Eric W. Frew and Jack Elston. Target assignment for integrated search and tracking by active robot networks. In *2008 IEEE International Conference on Robotics and Automation*, pages 2354–2359, 2008.

[14] Vandit Gajjar, Ayesha Gurnani, and Yash Khandhediya. Human detection and tracking for video surveillance: A cognitive science approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[15] Jake Gemerek, Silvia Ferrari, Brian H Wang, and Mark E Campbell. Video-guided camera control for target tracking and following. *IFAC-PapersOnLine*, 51(34):176–183, 2019.

[16] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.

[17] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3):16–25, 2006.

[18] Prabhat Hajela. Genetic search-an approach to the nonconvex optimization problem. *AIAA journal*, 28(7):1205–1210, 1990.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] M. Hussein, W. Abd-Almageed, Yang Ran, and L. Davis. Real-time human detection, tracking, and verification in uncontrolled camera motion

environments. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 41–41, 2006.

[21] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 and 2021): Developing Global Digital Economy after COVID-19.

[22] C. Kreucher, A.O. Hero, and K. Kastella. A comparison of task driven and information driven sensor management for target tracking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4004–4009, 2005.

[23] Chris Kreucher, Alfred O Hero, and Keith Kastella. A comparison of task driven and information driven sensor management for target tracking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4004–4009. IEEE, 2005.

[24] Jae Un Lee, Dong-Hyuk Park, and Jong-Hoon Won. A localization method with ukf/maf for a single antenna gnss/imu integration of autonomous unmanned ground vehicle. In *2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, 2022.

[25] Su-Jin Lee, Soon-Seo Park, and Han-Lim Choi. Potential game-based non-myopic sensor network planning for multi-target tracking. *IEEE Access*, 6:79245–79257, 2018.

[26] Keith A LeGrand, Pingping Zhu, and Silvia Ferrari. A random finite set sensor control approach for vision-based multi-object search-while-tracking. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021.

[27] Bin Liu, Wencang Zhao, and Qiaoqiao Sun. Study of object detection based on faster r-cnn. In *2017 Chinese Automation Congress (CAC)*, pages 6233–6236, 2017.

[28] Chang Liu, Zhihao Liao, and Silvia Ferrari. Rumor-robust decentralized gaussian process learning, fusion, and planning for modeling multiple moving targets. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3066–3071. IEEE, 2020.

[29] Juan Liu, James Reich, and Feng Zhao. Collaborative in-network process-

ing for target tracking. *EURASIP Journal on Advances in Signal Processing*, 2003:1–14, 2003.

[30] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5790–5799, 2017.

[31] Wenjie Lu, Guoxian Zhang, and Silvia Ferrari. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics*, 30(4):919–934, 2014.

[32] M. Marengoni, B.A. Draper, A. Hanson, and R. Sitaraman. A system to place observers on a polyhedral terrain in polynomial time. *Image and Vision Computing*, 18(10):773–780, 2000.

[33] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. 06 2009.

[34] Keisuke Okumura and Xavier Défago. Solving simultaneous target assignment and path planning efficiently with time-independent execution. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 270–278, 2022.

[35] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498, 2007.

[36] Lynne Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2, 01 2008.

[37] Ragesh Kumar Ramachandran, Nicole Fronda, and Gaurav S Sukhatme. Resilience in multirobot multitarget tracking with unknown number of targets through reconfiguration. *IEEE Transactions on Control of Network Systems*, 8(2):609–620, 2021.

[38] Stephen L. Smith and Francesco Bullo. Monotonic target assignment for robotic networks. *IEEE Transactions on Automatic Control*, 54(9):2042–2057, 2009.

[39] Mohib Ullah, Ahmed Kedir Mohammed, Faouzi Alaya Cheikh, and Zhaohui Wang. A hierarchical feature model for multi-target tracking. In *2017 IEEE international conference on image processing (ICIP)*, pages 2612–2616. IEEE, 2017.

[40] Peng Wang, Peter B. Luh, Shi-Chung Chang, and Kerry L. Marsh. Efficient optimization of building emergency evacuation considering social bond of evacuees. In *2009 IEEE International Conference on Automation Science and Engineering*, pages 250–255, 2009.

[41] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification across multiple resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8042–8051, 2018.

[42] Hongchuan Wei, Pingping Zhu, Miao Liu, Jonathan P How, and Silvia Ferrari. Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets. *IEEE Transactions on Automatic Control*, 64(1):159–173, 2018.

[43] Jingjin Yu, Soon-Jo Chung, and Petros G. Voulgaris. Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies. *IEEE Transactions on Automatic Control*, 60(2):327–341, 2015.

[44] Shun Zhang, Jinjun Wang, Zelun Wang, Yihong Gong, and Yuehu Liu. Multi-target tracking by learning local-to-global trajectory models. *Pattern Recognition*, 48(2):580–590, 2015.

[45] Yinggang Zhang, Yanduo Zhang, and Xun Li. Multi-robot exploration system in unknown environment based on submap. In *2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pages 256–260, 2021.

[46] Qin Zheng, JianXin Sha, and ChangLuan Fang. An effective genetic algorithm to vda with discontinuous "on-off" switches. *Science China Earth Sciences*, 55(8):1345–1357, 2012.

[47] Jianpeng Zhou and Jack Hoang. Real time robust human detection and tracking system. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 149–149, 2005.