HOLISTIC SCENE PERCEPTION FOR COLLABORATIVE HUMAN-ROBOT TEAMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Junyi Dong

December 2022

HOLISTIC SCENE PERCEPTION FOR COLLABORATIVE HUMAN-ROBOT TEAMS

Junyi Dong, Ph.D.

Cornell University 2022

Holistic scene perception aims to provide comprehensive knowledge of the objects and environment in a scene, as well as the intrinsic relationships between them, which plays an important role in human cognition. Driven by the desire to build future cognitive robots that can form collaborative teams with humans, scene perception for autonomous robots has been at the frontier of the interdisciplinary research combining computer vision and robotics in the recent decades. Although humans are capable of perceiving variegated visual scenes effortlessly, such tasks remain difficult for autonomous robots. The primary challenge lies in the extraction of implicit and hidden context, such as the interaction between objects, and the integration of it with the explicit and task-relevant visual features to interpret the scene. This dissertation tackles the above challenge by providing a novel framework for holistic scene perception that integrates domain knowledge, image recognition, state estimation, inference of hidden variables, and anticipation of future actions. The proposed approach is tested in dynamic scenes that depict human team activities, such as the team sport of volleyball, with complex goals and variegated interactions. The approach relies on a novel dynamic Markov random field model to infer hidden variables in the scene, which are then combined with visual features and domain knowledge to perform action anticipation using a multi-layer perceptron.

In addition, recent advancements in robotics and processing capabilities point

to a future in which mobile robots equipped with onboard sensors will be able to perceive the environment as humans do. Therefore, the second part of this dissertation investigates scene perception from the aspect of employing a network of mobile robots to effectively recognize and track a larger number of dynamic targets. Of critical interest in this problem is the maximization of tracking quality by simultaneously determining the coordination and control of the robot network, which, however, is proven to be NP-hard. This dissertation presents a new decomposition-based framework to efficiently solve the NP-hard network optimization problem in two stages. Two novel decentralized coordination methods are proposed to find adaptive and conflict-free target assignments. Then, robots locally and concurrently determine their control to maximize a new tracking utility function in real time. Physical experiments with a network of ground robots tracking human targets validate the applicability of the proposed approach in real-world applications.

Finally, the long term vision for holistic scene perception is to have intelligent robots share common goals and perceive the targets and environments collaboratively with humans to gain improved efficiency and robustness. A collaborative human-robot team has the advantage of leveraging complementary skills such as human field experience and domain knowledge, and robot data processing and integrated sensor modalities. This dissertation develops a new collaborative control and communication framework applicable to human-robot teams engaged in visually detecting and tracking many targets in an obstacle-populated environment. In both numerical simulations and physical experiments, this new collaborative control and communication framework is shown to be capable of providing robust performance in the presence of uncertainties such as state estimation errors and intruders.

BIOGRAPHICAL SKETCH

Junyi is a PhD student in the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University. She received the B.S. degree in Automation from Beijing University of Chemical Engineering, China and the M.Sc. degree in Thermal Engineering from Tsinghua University, China. Her main research interests include probabilistic graphical model, computer vision, control, and optimization, focusing particularly on developing perception and control algorithms for autonomous robotic systems equipped with imaging sensors.

Dedicated to my loving and supportive family, especially Bei.

ACKNOWLEDGEMENTS

I am especially grateful to my advisor, Dr. Silvia Ferrari. Silvia is a great mentor who has provided invaluable help, support, and encouragement to me over the course of my graduate studies. I would also like to thank Prof. Douglas MacMartin and Prof. Peter Frazier for serving on my Ph.D. committee and for their useful advice on my research. In addition, I would like to give special mention to the grants that funded this research: the Office of Naval Research Grant N00014-17-1-2175 and the Lockheed Martin Grant S21-012.

I wish to thank everyone in the Laboratory for Intelligent Systems and Controls (LISC) who were always available to help. A special thank you to my great lab mates and fiends, Sushrut Surve and Qingze Huo, for their support and enlightening discussions along the way. I wish to dedicate this dissertation to my family: my better half Bei, my parents Xiaoling and Yanqi, and my little brother Junyu. Their unconditional love, understanding, and support made my accomplishments possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND MOTIVATION

Holistic scene perception is concerned with the problem of perceiving and understanding the content of a given scene that is composed of objects and surfaces arranged in a meaningful way for semantic interpretation. Holistic scene perception plays an important role in human cognition and in many future tasks envisioned for cognitive robots such as human-robot collaboration. Although humans are able to integrate sufficient information of a meaningful scene within milliseconds [64], scene perception from RGB images by a fully autonomous agent remains a challenge to be tackled. A key problem in scene perception is to create intelligent agents with the ability of anticipating human actions before they occur, which is crucial to a broad range of contexts and situations. For example, people tend to choose their greetings, such as "shaking hands" or "hugging", based on their anticipation of the most likely response by the recipient [78]. Drivers routinely predict future actions of pedestrians, cyclists, and other drivers, based on their appearance, trajectories, driving style, and inferred social role, in order to guarantee safe driving [26, 27]. Similarly, athletes make split-second decisions based on the behavior of their teammates and opponents, their knowledge of the game, as well as their anticipation of opponents' actions [135]. As such, the ability to anticipate human actions is essential for human social life and bears great potential for future development of intelligent systems. Team sports, in particular, provide an excellent benchmark problem for action anticipation because the rules and goals of the game are well defined, video data is broadly available from event broadcasting, and players' decisions depend on many factors ranging from team strategy to individual roles, from knowledge of the game to opponent behaviors [104, 106].

Our holistic approach for interpreting and predicting team behaviors is demonstrated on a new and challenging problem in Chapter 2, namely, anticipating fast actions executed by interacting members of a sport team. In a team sport, such as volleyball, individual players assume different roles during the game, contributing in different measure to game strategy and outcome, and influencing their teammates' behaviors in contrasting ways. The players' roles are, almost by definition, hidden or unobservable. Chapter 2 presents a novel dynamic Markov random field (DMRF) model that models the joint probability of players' roles based on the extracted players' feature vector, while also capturing players' interrelationships in a dynamic graph structure. The results from the DMRF inference stage are integrated with the visual cues and domain knowledge of the sport and of the team itself, in order to perform action anticipation using a multi-layer perceptron (MLP). Such integration of the inference and anticipation method provides a holistic approach for visual scene perception that allows to account for the implicit context, perceived through several inferred hidden variables, as well as for hybrid inputs comprised of spatio-temporal relationships, continuous variables, and categorical features that together describe the team players and their interactions.

In addition, recent advancements in robotics and processing capabilities point to a future in which mobile robots will be able to perceive the environment as humans do or even better. Many modern robotic platforms, such as unmanned ground vehicles (UGVs), are equipped with onboard sensors that allow robots to collect sensing data while traveling through an environment. Consequently, the control of multi-robot networks (MRNs) to perform scene perception tasks such as target tracking has received significant attention. More specifically, target tracking by MRNs deals with estimating the unknown kinematic states of moving targets, which is related to many promising applications. For instance, MRNs can patrol

high risky industrial workspace and track people working there for the purpose of safety monitoring. Similarly, robots can track rescuers in hazardous environments to assist searching and rescuing victims. In these applications, MRNs can operate in parallel to reduce the task completion time, communicate mission-relevant data to gain situational awareness, and create redundancy to improve fault tolerance, which renders them more promising than single robot systems.

Unlike a large volume of previous work that focused on the estimation aspect of the tracking problem using passive information received by static sensors [31, 102, 127, 148], research with MRNs aims to actively and cooperatively determine robot control in order to optimize the network tracking performance. Chapter 3 presents a new approach to the coordination and control of multi-robot networks (CCMRN) for the non-trivial tracking scenarios where targets outnumber the robots. The approach features decentralized optimization, in which the network goal is achieved by robots concurrently selecting the conflict-free target assignment through local communication and independently determining their control for target tracking. Two novel methods, the group-based algorithm and the bundle-based algorithm, are proposed to find target assignments at every time instant, with the latter achieving more effective coordination and guaranteeing $\frac{1}{2}$-approximation in the worst-case. In addition, the robot control is optimized to make the most informative future measurements and encourage the exploration of lesser tracked targets. The simulation results in Section 3.7 show that the performance of the proposed approaches is very close to that of the optimal solution and is higher than the other decentralized baseline methods.

Compared to MRNs consisting of homogeneous robotic agents, collaborative human-robot teams can potentially improve efficiency and robustness for scene

perception tasks by leveraging complementary skills of different team members. Therefore, many emerging robotic applications increasingly require autonomous robots to partner with humans to achieve shared goals. Multi-target tracking, in particular, provides an interesting testbed for studying human-robot collaboration because humans and robots can obtain complementary information about dynamic targets. For instance, although characterized by directional and bounded FOV, mobile robots can track dangerous targets at close distance to gain views with intricate features. In contrast, human operators possess better situational awareness and interpretation of complex mission objectives but have difficulty simultaneously observing many dynamic targets. Chapter 4 proposes a new approach to human-robot collaboration that enables the maximization of the cumulative tracking time when the targets outnumber the tracking agents. Collaboration entails a two-way message-exchange mechanism and distributed robot control that is a function of human actions. A new tracking utility function is proposed for the local estimation of the global tracking performance by the collaborative team, which accounts for the robot FOV geometry, kinematic constraints, target prediction, obstacle map, and human input. In both numerical simulations and physical experiments, this new collaborative control and communication framework is shown to be capable of providing robust performance in the presence of uncertainties such as state estimation errors and intruders. Moreover, the collaborative team can perform other high-level perception tasks such as human action and interaction recognition, as will be discussed in future work.

CHAPTER 2

A HOLISTIC APPROACH FOR ROLE INFERENCE AND ACTION ANTICIPATION IN HUMAN TEAMS

2.1 Introduction

As pointed out in the seminal work on mental cognition by Kenneth Craik in 1943 [34], animals utilize internal models of their external reality and of possible actions at their disposal in order to evaluate various alternatives and conclude which one to utilize to react to new situations. In the context of teams and collaborative groups, individuals use their ability to anticipate human actions in a broad range of situations in order to decide their own subsequent actions and behaviors. Often, action anticipation is based on inferred cues, such as social roles, intentions, and goals that are deduced from visual information interpreted in the context of domain knowledge and past experiences. For example, people tend to choose their greetings, such as "shaking hands" or "hugging", based on their anticipation of the most likely response by the recipient [78]. Drivers routinely predict future actions of pedestrians, cyclists, and other drivers, based on their appearance, trajectories, driving style, and inferred social role, in order to guarantee safe driving [26, 27]. Similarly, athletes make split-second decisions based on the behavior of their teammates and opponents, their knowledge of the game, as well as their anticipation of opponents' actions [135]. As such, the ability to anticipate human actions is essential for human social life and bears great potential for future development of intelligent systems and machines. Team sports, in particular, provide an excellent benchmark problem for action anticipation because the rules and goals of the game are well defined, video data is broadly available from event broadcasting, and play-

ers' decisions depend on many factors ranging from team strategy to individual roles, from knowledge of the game to opponent behaviors [104, 106].

In contrast to action recognition, which generates a semantic label from the video of an observed human behavior [41, 94, 116, 151], action anticipation aims at predicting one or more sequential human behaviors, several seconds into the future. Unlike traditional prediction algorithms, the approach presented in the chapter seeks to anticipate the semantic labels of a sequence of human actions before their onset, including sudden and radical behavioral changes such as switching from standing to hitting the ball. Existing methods for action anticipation can be categorized into feature-level, single-agent, and dual-agent anticipation. Feature-level anticipation predicts a convolutional feature representation of a future image for an ongoing action and, then, uses this representation to predict the action label classification [48, 108, 114, 120, 129]. These methods assume that a few initial frames of a human action is partially observed, based on which the remaining action sequences can be predicted. Moreover, feature-level anticipation relies primarily on prior data training and, therefore, fails in testing images that do not show globally similarity to the training data [130].

Single-agent anticipation predicts a semantic action label using appearance-based or motion-based features extracted from a sequence of frames preceding the onset of an action [24, 47, 109]. The input features can be enriched by incorporating information of the surrounding visual context, such as the presence of certain meaningful objects in the scene [81, 91]. A long short-term memory (LSTM) network was trained in [47] to predict an individual's cooking activity over the horizon of 0.25-2 s based on an observation time window of 1.75-3.5 s. The action anticipation performance of the cooking activity was quantitatively evaluated in [73] in

terms of the observation duration and prediction horizon, showing that an increase in prediction horizon is accompanied by deterioration in anticipation accuracy even with long observations of up to 30 s.

Dual-agent action anticipation methods rely on extracting action-reaction patterns from videos of two-person interactions such as "hugging" or "pushing", in order to leverage the causal relationship in social interactions [12, 63, 78, 81]. However, the resulting algorithms are limited in scope in that the interaction is known *a priori*, and the anticipation is from the perspective of the reactive agent by only anticipating the reactive actions based purely on visual cues. The approach presented in this chapter is applicable to diverse forms of interactions among two or more persons, including team strategies and individual roles that evolve over time, and is capable of predicting action sequences and timing. Previous work has shown that the temporal localization of future events can be performed by learning a probability distribution of the occurrence time conditioned on a sequence of observed features [95]. In particular, the former method quantizes the prediction horizon into discrete time intervals, one of which is predicted to contain the occurrence of the future event. One downside of such discrete-time model is the finite temporal resolution caused by quantization. As an improvement, a regression neural network was learned from data in [91, 95] to output a real positive value as a continuous approximate of the onset of the future action executed. In this work, the regression neural network is extended to the problem of predicting both the onset and duration of future actions in human teams.

Our holistic approach for interpreting and predicting team behaviors is demonstrated on a new and challenging problem, namely anticipating fast actions executed by interacting members of a sport team. In a team sport, such as volleyball,

not only the team strategy and circumstances of play are hidden and directly influence individual actions, but also are highly dynamic, in that they change significantly and rapidly over time. Additionally, individual players assume different roles during the game, contributing in different measure to game strategy and outcome, thus influencing their teammates' behaviors in contrasting ways. The team strategy and players' roles are, almost by definition, hidden or unobservable. In other words, they are not visually explicit in the scene, but they can be inferred from a combination of visual cues and domain knowledge of the sport and of the team itself, as will be demonstrated in this chapter.

Inferring team strategy bears similarities to the problem of group activity recognition, which seeks to identify an activity label for a group of participants [66, 67, 123, 142]. However, these methods require the user to pre-select a time window that centers around a group activity by manually clipping the video or choosing the initial and final image frame. As such, they can not be easily extended to dynamic settings where the team strategies evolve over time, gradually or suddenly at unknown instants. In contrast, this work infers the team strategy label in each frame, based on which the input video can be automatically partitioned into scene segments for action anticipation.

On the other hand, role inference derives motivation from the "Role Theory" in sociology [84, 103, 125], which is a key concept for understanding the organization of social life and social activity. Recently, [51] defined roles as "socially defined expectations that a person in a given status follows", showing that roles provide predictability of people's behaviors. The importance of individual social roles in human events, such as "listener", "speaker", "bride", and "groom", has also been recently recognized in the computer vision literature [44, 103]. These

methods, however, are not directly applicable to team action anticipation because they do not consider the rapid change in roles. Also, existing methods seek to label either the group activity or the individual role, whereas, in many events, such as sports, the individual role changes over time as a function of an evolving group activity/strategy. Furthermore, in many events, such as team sports, the interdependence between team strategies and players' roles cannot be necessarily categorized into a set of semantic classes identifiable *a priori*.

This work presents a novel dynamic Markov random field (DMRF) model that captures players' interrelationships using a dynamic graph structure, and learns individual player characteristics in the form of a feature vector based on a wealth of prior information, including domain knowledge, such as court dimensions and sport rules, and visual cues, such as homography transformations, and players' actions and jerseys. The DMRF unary and pairwise potentials can then be learned from data to represent the probability of individual feature realizations and the strengths of the corresponding players' interrelationships, respectively. Each new video frame is associated with a global hidden variable that describes the team strategy, within which each player is assigned a local hidden variable representing her/his role on the team. Then, given video frames of an ongoing game, the DMRF can be used to infer the players' roles using a Markov chain Monte Carlo (MCMC) sampling method, and to provide inputs to an MLP that anticipates the players' future actions.

The notion of key player is introduced to distinguish a small set of players who will perform dominant actions that directly influence the game progress. In the anticipation stage, an MLP is trained to predict future actions of key players based on visual features as well as the inference results. Action anticipation is

performed in each frame such that the anticipated results can be updated in a timely manner as the future unfolds. Inspired by recent work on predicting the temporal occurrence of future actions [91], the anticipation MLP is configured to simultaneously output the semantic label, onset and duration of the key players' future actions.

In comparison to the existing research on single-agent and dual-agent action anticipation, this work raises a distinctively new variant of visual forecasting problem that anticipates future action in human teams. By proposing a new problem formulation and solution for team action anticipation, the holistic approach presented in this work allows to account for the implicit context, perceived through several inferred hidden variables, as well as for hybrid inputs comprised of spatio-temporal relationships, continuous variables, and categorical features that together describe the team players and their interactions. The results obtained on testing database constructed from broadcasting videos of volleyball games demonstrate that this approach predicts the future actions of key players up to 46 frames into the future, with an accuracy of 80.50%. In addition, the approach achieves an average accuracy of 84.43% and 86.99% for inferring the team strategy and players' roles, respectively.

2.2 Background and Preliminaries

The role inference and action anticipation approach presented in this chapter is demonstrated on the team sport of volleyball, described here briefly for completeness. However, the approach can be similarly applied to other team sports and activities, as will also be shown in future work. A volleyball match consists of five

sets that are further broken into points. Each point starts with a player serving the ball to the opposite side. Each team must not let the ball be grounded within their own court by hitting the ball to the opponent after no more than three consecutive touches of the ball by three different players. The game continues until the ball is grounded, with the players moving around their own side of the court and assuming different roles over time, such as blocker, defense-libero, left-hitter, and so on (Fig. 2.1). This alternating pattern can be reflected by the transition of a finite class of team strategy labels (Fig.2.1(a)), whose semantic meaning describes the technical activity of the two teams. For instance, the team strategy label in Fig.2.1(b) indicates that the right team is setting the ball for the next-step attack and the left team is on defense, whereas Fig.2.1(b) shows that the the right is attacking and the left is blocking.

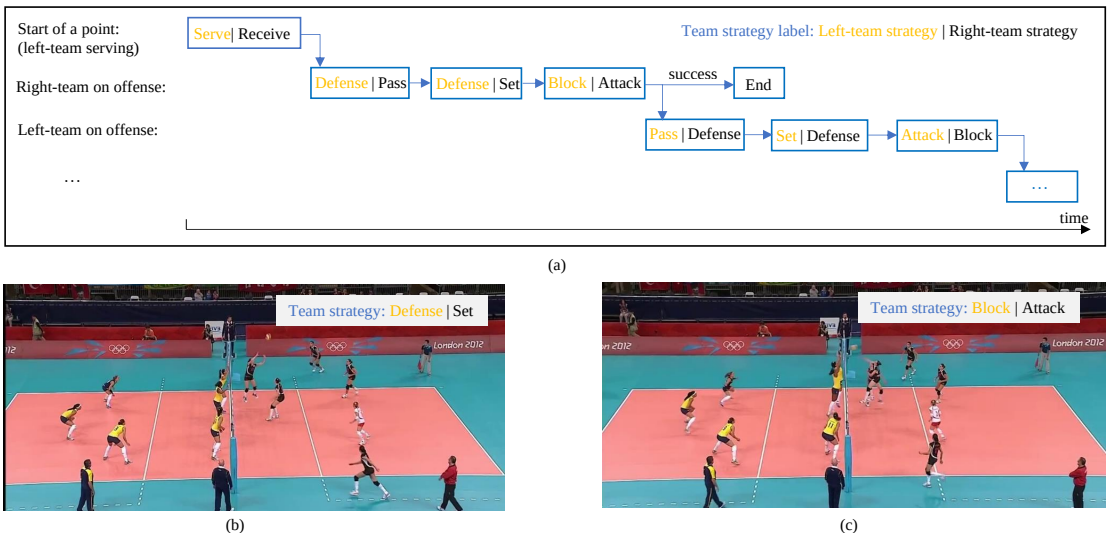


Figure 2.1: Example of temporal evolution of team strategies in a volleyball match (a) and corresponding visual scenes (b-c).

The two teams are divided by a net in the middle of the court, which simplifies the action anticipation problem compared to other team sports, such as football or hockey, which will be studied in future work. Like other sports, each team is

represented by a jersey color. But, in volleyball, some players within a team also wear a different jersey to indicate their "libero position" on the team. For effective coordination, players assume different roles in accordance to their expected duty in the team. Consequently, each player can be assigned a semantic role label that serves as an abstract representation of the player's intentions and possible actions. A complete description of the players' nine possible roles is shown in Fig.2.2. An important complexity is that the players roles change rapidly and unexpectedly over time, and some of the players can assume the same role at the same time.

Also volleyball actions can be categorized into nine well-defined classes: *spiking, blocking, setting, running, digging, standing, falling, waiting, and jumping*, extracted using computer vision algorithms [7, 8, 66, 67, 115]. However, actions are not unique to players' roles, nor there is any precise correspondence (e.g. one-to-one) between roles and actions. In this work, the action label *waiting* is replaced with *squatting* for a closer clarification on this defensive action that happens before a player digs the ball, as shown in Fig.2.3.

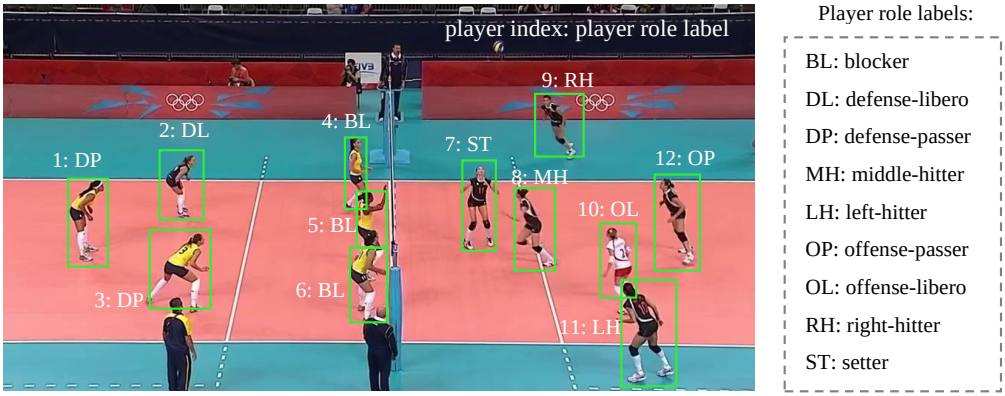


Figure 2.2: Volleyball players' roles.

During the volleyball match, players do not contribute equally. Rather, only a subset of players referred to as *key players* are actively engaged while the others

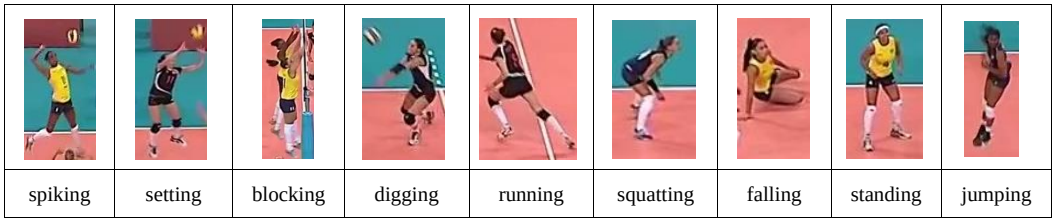


Figure 2.3: Examples of nine volleyball players' actions.

are waiting for their turns to enter into action. For instance, player 7 in Fig. 2.2 is a key player because her future action of *setting* will dominate the game.

2.3 Problem Formulation and Assumptions

The problem addressed in this chapter consists of anticipating future actions by multiple key players in the team sport of volleyball based on hidden information, such as players' roles and team strategy, domain knowledge, and visual features extracted from video using existing computer vision algorithms [59, 66, 67, 75, 142, 147]. The goal is to develop a general and systematic approach for interpreting visual scenes of human group activities with complex goals, dynamic behaviors, and variegated interactions. Although this work mainly considers video data, the proposed framework can be readily applied to data obtained from other sensing modalities, such as range finders, inertial navigation units, and wearable sensors [54]. The approach is holistic in that it integrates image *recognition*, namely the classification of visually explicit information, state *estimation*, *inference* of hidden variables, and *anticipation* of future actions and events. As schematized in Fig. 2.4, the approach consists of using the information extracted from domain knowledge (including prior videos) and streaming videos, using available image recognition and state estimation algorithms, to solve the problems of team/player inference and

action anticipation problem formulated in Sections 2.3.1 and 2.3.2, respectively.

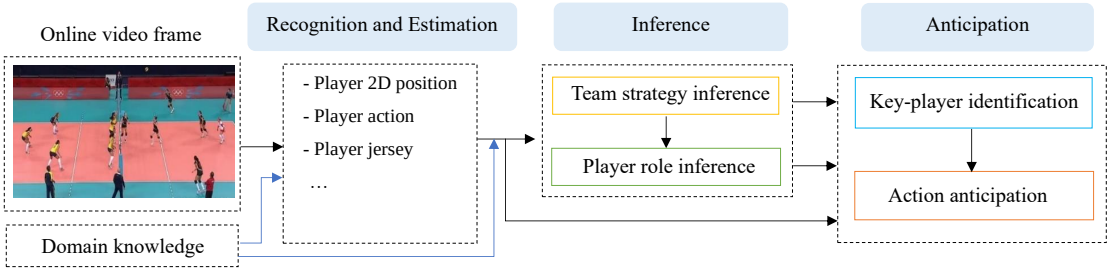


Figure 2.4: A holistic framework for action anticipation in team sports.

2.3.1 Inference Problem Formulation

Consider a video $\mathcal{V}$ comprised of $K \in \mathbb{N}^+$ consecutive frames obtained at discrete moments with a constant sampling interval Δt. Each frame $I(k) \in \mathbb{R}^{h \times w}, k = 1, \ldots, K$, corresponds to an image matrix of $h \times w$ pixel intensities, where $h, w \in \mathbb{N}^+$ are the frame size. Let $\mathcal{N} = \{1, \ldots, N\}$, $N \in \mathbb{N}^+$, denote the index set of players extracted from $I(k)$ using computer vision [59,67]. The frame index is omitted for $\mathcal{N}$ since the number of players is fixed in a volleyball video.

Each player in frame $I(k)$ can be associated with an index $i \in \mathcal{N}$ and a feature descriptor that contains a 2D position vector, an action label, and an appearance feature describing the player's jersey color. Other characteristics and state variables can be similarly included, depending on the application of interest. Let $\mathbf{p}'_i(k) = [x'_i(k)\ y'_i(k)]^T \in \mathbb{R}^{2\times 1}$ denote the $2D$ position of the i^{th} player with respect to the image frame, which can be approximated by the image coordinate at the bottom middle point of the player's bounding box. In order to gain immediate insight into players' spatial relationship, the position vector $\mathbf{p}'_i(k)$ is resolved into the inertial coordinate denoted by $\mathbf{p}_i(k) = [x_i(k)\ y_i(k)]^T \in \mathbb{R}^{2\times 1}$. Because the

volleyball court is planar, the image and inertial coordinate can be related via homograph transformation H, as shown in Fig.2.5,

$$\lambda \begin{bmatrix} x_i'(k) \\ y_i'(k) \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_i(k) \\ y_i(k) \\ 1 \end{bmatrix} \tag{2.1}$$

where $\lambda \neq 0$ is a scaling factor, and the homography matrix H can be estimated using domain knowledge of court dimensions and the geometry of the lines drawn on the volleyball court [43, 53, 126, 128].

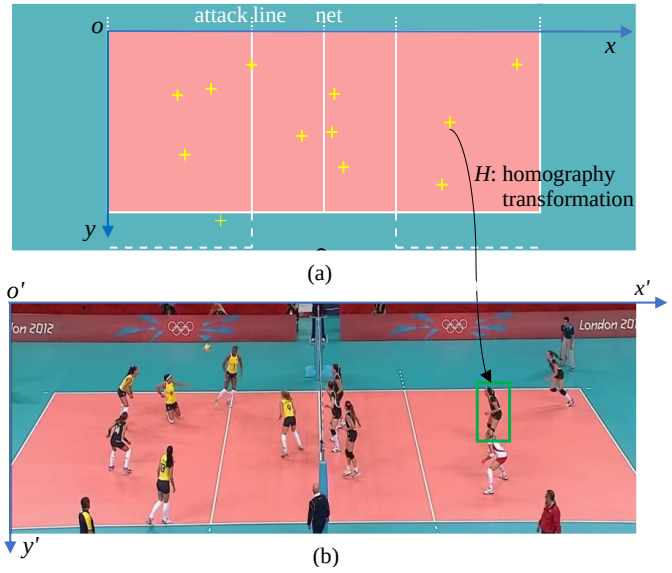


Figure 2.5: Projection between the inertial reference frame (a) and image reference frame (b).

Next, let $A_i(k) \in \mathcal{A}$ represent the action label of player $i \in \mathcal{N}$ in an observed frame $I(k)$, where $\mathcal{A}$ is the discrete and finite range of the action classes shown in Fig.2.3. A player's jersey color is denoted by a discrete variable $C_i(k) \in \mathcal{C}$, which can be obtained using a color detector [28, 68, 124] or as prior knowledge. Together, the aforementioned features can be organized as a player feature vector $F_i(k) = [\mathbf{p}_i(k)^T \quad A_i(k) \quad C_i(k)]^T$.

Then, each frame $I(k) \in \mathcal{V}$ in a volleyball video can be assigned a semantic label describing the technical strategy of two teams, as illustrated in Fig.2.1(b-c). Inference of the team strategy requires the aggregation of features across players, which amounts to the concatenation of player feature vectors into a frame-wise team descriptor. In order to preserve the spatial relationship in a team, feature vectors of players on each side are sorted by the player's distance to the net. Then, the aggregated team feature descriptor can be constructed as

$$F(k) \triangleq [F_{l_1}^T(k) \quad \dots \quad F_{l_{\frac{N}{2}}}^T(k) \quad F_{r_1}^T(k) \quad \dots \quad F_{r_{\frac{N}{2}}}^T(k)]^T \tag{2.2}$$

with the range denoted by $\mathcal{F}$ and the indices of elements defined by the sorted index set

$$\hat{\mathcal{N}} = \{l_1, \ \dots, \ l_{\frac{N}{2}}, \ r_1, \ \dots, \ r_{\frac{N}{2}}\} \tag{2.3}$$

where $\{l_1, \dots, l_{\frac{N}{2}}\} \subset \hat{\mathcal{N}}$ represent the sorted indices of players on the left team and $\{r_1, \dots, r_{\frac{N}{2}}\} \subset \hat{\mathcal{N}}$ is the counterpart for the right team.

Let $S(k) \in \mathcal{S}$ be a global hidden variable representing the team strategy label in frame $I(k)$, where $\mathcal{S}$ is the finite range of the team strategy classes, as illustrated in Fig.2.1. In addition, let $X_i(k) \in \mathcal{R}, i \in \mathcal{N}$, be a local hidden variable representing the role of player i. $X_i(k)$ takes a realization from a set of role labels $\mathcal{R}$, which are illustrated in Fig.2.2. The labels of all players' roles can be denoted by a random vector $X(k) \triangleq [X_1(k) \quad \dots \quad X_N(k)]^T$ that has range $\mathcal{X} = \mathcal{R}^N$. Then, the inference problem can be formulated as follows:

Problem 1: Given the extracted features, $F(k)$, learn a multi-class classifier, $f_S : \mathcal{F} \to \mathcal{S}$, that maps $F(k) \in \mathcal{F}$ to a team strategy label $S(k) \in \mathcal{S}$. Subsequently, learn an inference model, $f_X : \mathcal{F} \times \mathcal{S} \to \mathcal{X}$, that maps the feature vector $F(k)$ and the inferred team strategy label $S(k)$ to the vector $X(k)$, representing role labels of all players.

2.3.2 Anticipation Problem Formulation

The goal of the action anticipation problem is to leverage the confluence of information including inferred team strategies, inferred players' roles and features, as well as domain knowledge, in order to predict which are the key players and what are their respective future action sequences. Given the inferred team strategy up to the current frame, κ, (obtained from *problem 1*), a scene change point is defined as a frame index τ such that

$$S(\tau) \neq S(\tau + 1), \quad \tau = 1, \ldots, \kappa - 1 \tag{2.4}$$

and is typically unknown *a priori*. Let $\boldsymbol{\tau} = [\tau_1 \ldots \tau_m]^T$ represent the scene change points up to the current time κ, where $\tau_1 = 1$ and $\tau_m \leq \kappa$. Video frames between every two consecutive scene change points have the same inferred team strategy and, therefore, can be automatically grouped as a scene segment, which eliminates the algorithm's dependence on pre-trimmed videos. Let $V_l, l = 1, \ldots, m$ denote the l^{th} scene segment with the frame-index set T_l defined as

$$T_l = \begin{cases} \{\tau_l, \ldots, \tau_{l+1} - 1\} & l = 1, \ldots, m - 1 \\ \{\tau_l, \ldots, \kappa\} & l = m \end{cases} \tag{2.5}$$

Consequently, V_l can be represented as

$$V_l = \{I(k) \mid k \in T_l\}, \quad l = 1, \ldots, m \tag{2.6}$$

The duration of V_l, denoted by d_l, equals the number of frames in T_l multiplied by the discrete-time sampling interval Δt

$$d_l = \begin{cases} (\tau_{l+1} - \tau_l)\Delta t & l = 1, \ldots, m - 1 \\ (\kappa - \tau_l + 1)\Delta t & l = m \end{cases} \tag{2.7}$$

After defining the scene segments, variables that are defined in each frame $I(k)$ can be upgraded to represent the whole segment, as shown in Table 2.1, where

Table 2.1: Notation of frame variables and segment variables

Frame variable	**Description**	**Segment variable**	**Description**
$S(k)$	Team strategy in frame $I(k)$	$S_l = \{S(k) \mid k \in T_l\}$	Team strategy in segment V_l
$A_i(k)$	Action of player i in frame $I(k)$	$A_{i,l} = \{A_i(k) \mid k \in T_l\}$	Action of player i in segment V_l
$X_i(k)$	Role of player i in frame $I(k)$	$X_{i,l} = \{X_i(k) \mid k \in T_l\}$	Role of player i in segment V_l
$\mathbf{p}_i(k)$	2D location of player i in frame $I(k)$	$P_{i,l} = \{\mathbf{p}_i(k) \mid k \in T_l\}$	2D location of player i in segment V_l

the argument in "()" represents the frame index, the subscript "i" represents the player index, and the subscript "l" represents the segment index.

In order to distinguish a small set of players who will perform dominant actions that influence the game progress, a binary indicator variable $\mu_i(\kappa) \in \{0, 1\}$ is introduced for a player i such that its value equals one if the corresponding player will become a key player, and equals zero otherwise. $\mu_i(\kappa)$ can be obtained by constructing a mapping, $f_\mu : \mathcal{S} \times \mathcal{R} \rightarrow \{0, 1\}$, that takes as input the inferred team strategy label $S(k)$ and role label $X_i(k)$ and outputs the binary indicator value

$$\mu_i(\kappa) = f_\mu(S(k), X_i(k)) \tag{2.8}$$

$f_\mu(\cdot)$ can be learned as a binary classifier based on a small amount of annotated data, or it can be derived using domain knowledge about the likelihood of a player being the key player given the corresponding role and team strategy. The complete set of predicted key players is

$$\mathcal{K} = \{i \mid \mu_i(\kappa) = 1,\ i \in \mathcal{N}\} \tag{2.9}$$

Action anticipation of a key player considers four types of information collected in the current scene segment V_m, i.e., the inferred team strategy S_m, the inferred role $X_{i,m}$, the ongoing action $A_{i,m}$ and the player's 2D spatial location $P_{i,m}$. Furthermore, the Markov assumption is adopted such that future action $A_{i,m+1}$, is independent from the past action $A_{i,m-1}$ with given $\{A_{i,m}, P_{i,m}, X_{i,m}, S_m\}, i \in \mathcal{K}$.

The Markov assumption is justifiable because the hybrid inputs encode information from multiple sources, hence enriching the model and reducing the dependence of future action on historical data. By virtue of such assumption, action anticipation only requires a short-term input with arbitrary starting scenes. Finally, the action anticipation problem can be summarized as follows:

Problem 2: Given the inferred team strategy label $S(\kappa)$ and role label $X(\kappa)$ of the current frame $I(\kappa) \in V_m$, predict the set of key players, $\mathcal{K} \subseteq \mathcal{N}$, using (2.8-2.9). Then, for each key player $i \in \mathcal{K}$, predict the semantic label, onset and duration of their future actions $A_{i,m+1}$ using aggregated input sequences $\{A_{i,m}, P_{i,m}, X_{i,m}, S_m\}$.

2.4 Inference Model

Inferring team strategy requires a multi-class classifier to map the feature vector $F(k)$ to a label $S(k)$ that represents the technical team activity in each frame. This work uses an MLP to perform the task while other classifiers such as random forests [132] are also applicable. The inferred team strategy label, $S(k)$, is appended to the feature vector of the i^{th} player to form an augmented feature vector, i.e., $Z_i(k) = [F_i(k)^T \quad S(k)]^T, i \in \mathcal{N}$, which can then be organized into an augmented feature matrix for all players

$$Z(k) = [Z_i(k) \quad \dots \quad Z_N(k)] \tag{2.10}$$

This section develops a novel dynamic Markov random field (DMRF) model with dynamical graph structures for inferring the joint probability of players' roles $X(k)$ from the augmented feature matrix $Z(k)$.

2.4.1 Dynamic Markov Random Field (DMRF) Model of Team Player Roles and Interactions

Classic MRFs are probabilistic models comprised of an undirected graph with a set of nodes that each represent correlated random variables, and a set of undirected arcs (i.e., graph structure) that represent a factorization of the joint MRF probability learned from data [45]. The advantages of MRFs over other probabilistic models are that they can model processes with both hidden and observable variables, as well as include both categorical and continuous variables by describing different types of relationships using unary and pairwise potentials. MRF was introduced into the image processing field in the 1980s [49] and was henceforth widely used in computer vision problems such as image segmentation [52, 96], image denoising [22] and image reconstruction [25, 93]. While in classic MRFs, the graph structure is fixed and decided *a priori*, this work presents an approach for constructing dynamic MRFs (or DMRFs) representations of the visual scene. The goal is to learn a temporally evolving graph structure from each frame for the inference of hidden role variables, where only the set of nodes remains unchanged, and the arcs appear or disappear from frame to frame based on the events in the scene.

In this approach, every hidden node, denoted by $X_i(k)$ ($i \in \mathcal{N}$), represents the hidden role of player i, and every observable node, denoted by $Z_i(k)$ ($i \in \mathcal{N}$), represents the feature vector of player i. The temporally evolving arc set, $\mathcal{E}(k)$, is then learned from the players' relative distance by minimizing an energy function such that the minimum value corresponds to the optimal arc configuration. In order to infer the players' roles from all available information, each node $X_i(k)$ is connected to the corresponding feature vector $Z_i(k)$. $X_i(k)$ is associated with

a unary potential $\phi(X_i(k), Z_i(k))$ that captures how probable feature $Z_i(k)$ is for different realizations of $X_i(k)$. Every arc is associated with a pairwise potential $\psi(X_i(k), X_j(k))$ that represents the strength of correlations between the two random variables $X_i(k)$ and $X_j(k)$ in a spatial neighborhood. Then, the joint probability distribution of the random variables can be factorized as the product of potential functions over the graph structure [76, 140]

$$P(X(k)|Z(k), \mathcal{E}(k)) = \frac{1}{C} \prod_{i \in \mathcal{N}} \phi(X_i(k), Z_i(k)) \prod_{i,j \in \mathcal{E}(k)} \psi(X_i(k), X_j(k)) \qquad (2.11)$$

where C is the partition function that guarantees $P(X(k)|Z(k))$ is a valid distribution and the scope of pairwise potentials is determined by the estimated graph structure $\mathcal{E}(k)$. An example of DMRF graph representation is illustrated in Fig.2.6 and the potential functions are learned as explained in the following subsections.

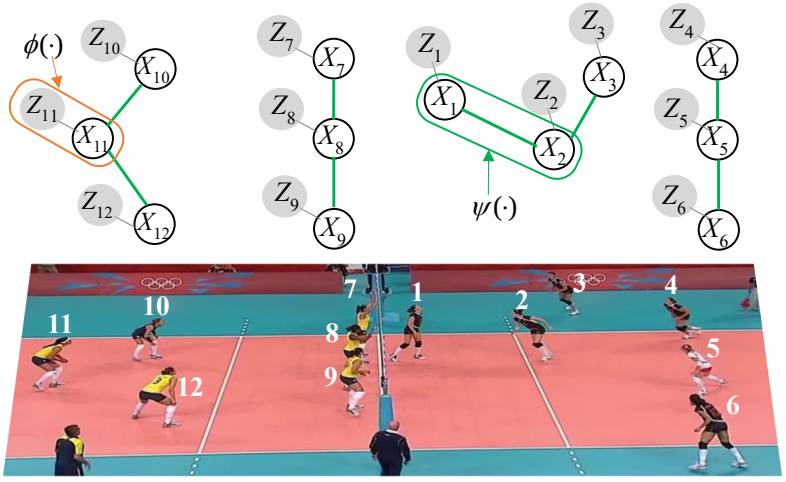


Figure 2.6: DMRF model for player role inference, where the time argument k is omitted for brevity.

DMRF Potential Functions

The unary potential $\phi(X_i(k), Z_i(k))$ expresses how probable the feature vector $Z_i(k)$ is for different realization of the role label $X_i(k)$, and can be modeled as a

likelihood function [11, 76, 112],

$$\phi_i(X_i(k), Z_i(k)) \triangleq P(Z_i(k)|X_i(k)) \tag{2.12}$$

Let $\mathcal{R} = \{1, 2, \ldots, R\}$ denote the set of role labels such that $X_i(k) = n$ ($n \in \mathcal{R}$) if player i assumes the n^{th} semantic role label. Let $\mathbf{1}_n \in \{0, 1\}^R$ be a R-dimensional one-hot vector where the n^{th} entry equals one and the rest entries equal zero. The likelihood function can be defined as

$$P(Z_i(k)|X_i(k) = n) = \frac{\exp\{\mathbf{1}_n^T \cdot [W_{u2} \cdot \sigma(W_{u1} \cdot Z_i(k))]\}}{\sum_{m=1}^{R} \exp\{\mathbf{1}_m^T \cdot [W_{u2} \cdot \sigma(W_{u1} \cdot Z_i(k))]\}} \tag{2.13}$$

where $\sigma(\cdot)$ is the sigmoid function, W_{u1} and W_{u2} are weights that will be learned from data and their dimensions are hyper-parameters selected to agree with the dot product.

Pairwise potential concerns the interrelationship between two node variables taking particular roles, with greater value indicating higher probability for the corresponding players to interact in a team. For instance, the pair "setter - hitter" has a higher chance to interact in a close proximity than "setter - blocker" pair since the latter only appears in two opposing teams. Let $W_p \in \mathbb{R}^{R \times R}$ denote the weight matrix that represents the correlation between a pair of roles. Then, the pairwise potential is defined as

$$\psi(X_i(k) = n, X_j(k) = m) \triangleq \mathbf{1}_n^T \cdot W_p \cdot \mathbf{1}_m \tag{2.14}$$

DMRF Graph Structure

The graph structure, $\mathcal{E}(k)$, determines the scope of pairwise potentials. Traditionally, the MRF graph structure is established a *priori* and remains fixed (e.g. [139,140]). In order to use MRF models for dynamic role inference, a new approach is developed here to learn and adapt the structure online based on streaming

video frames. In this approach, the structure can vary from an empty arc set to a fully connected configuration, as shown in Fig.2.7. An empty arc set (Fig.2.7(a)) indicates that all nodes (e.g. players' roles) are independent and there are no interactions between them. Conversely, a densely connected configuration (such as that in Fig.2.7(c)) captures many interrelationships, including redundant ones and, thus, may incur unnecessary computational burden. The approach developed in this work produces an efficient structure estimation algorithm (2.16-2.20) to dynamically estimate a sparse structure (Fig.2.7(b)) that captures only the most significant interactions in each video frame.

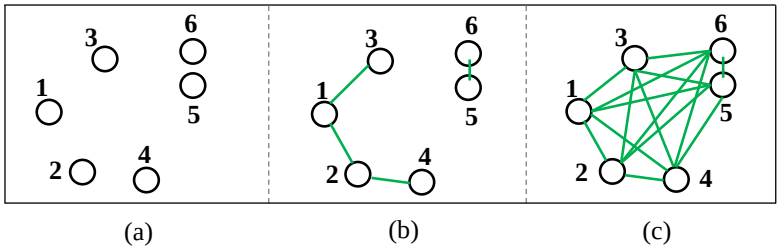


Figure 2.7: A graphical model of six nodes with an empty arc set (a), a sparse arc set (b), and a dense fully connected arc set (c).

Let $Y_{i,j}(k)$ denote a binary variable such that its value $y_{i,j}(k)$ equals one when an interaction arc exists between players labeled by i and j, and equals zero otherwise. Then the arc set can be denoted as $\mathcal{E}(k) = \{(i,j)|y_{i,j}(k) = 1, i,j \in \mathcal{N}\}$, and the structure estimation problem can be cast as a constrained optimization problem over the arc variables $Y_{i,j}(k)$. In many human team activities, such as sports, proximity is an indication of potential interactions and, therefore, in this work the DMRF graph structure is indicative of interrelationships between spatial neighbors. Other representations are also possible, depending on the application, and may be adopted in the proposed approach with small modifications. Then, the Euclidean distance $d_{i,j}(k) = \|\mathbf{p}_i(k) - \mathbf{p}_j(k)\|$ between every pair of players is used to construct

an energy function that is linear in the realizations of the arc variables $Y_{i,j}(k)$,

$$E(Z(k), \mathcal{E}(k)) \triangleq \sum_{(i,j)\in\mathcal{E}(k)} d_{i,j}(k)\ y_{i,j}(k) \tag{2.15}$$

such that the optimal arc configuration corresponds to the minimum of the energy function. Subsequently, minimizing the energy function can be approached by solving an Integer Linear Program

$$\min_{\mathcal{E}(k)} \quad \sum_{(i,j)\in\mathcal{E}(k)} d_{i,j}(k) y_{i,j}(k) \tag{2.16}$$

$$y_{i,j}(k) = y_{j,i}(k), \quad \forall (i,j) \in \mathcal{E}(k) \tag{2.17}$$

$$\text{sbj to} \quad \sum_{i\in\mathcal{N}} y_{i,j}(k) \geq 1, \quad \forall j \in \mathcal{N} \tag{2.18}$$

$$\sum_{i\in\mathcal{N}} y_{i,j}(k) \leq 2, \quad \forall j \in \mathcal{N} \tag{2.19}$$

$$y_{i,j}(k) \in \{0,1\}, \quad \forall (i,j) \in \mathcal{E}(k) \tag{2.20}$$

The constraint in (2.17) guarantees that interactions are symmetric, and (2.18) - (2.19) specify that a node has a minimum of one and maximum of two arcs connecting to its spatial neighbours, resulting in a sparse structure. Although only the proximity feature is considered, the proposed method is a generic algorithm that can incorporate other features to estimate social interactions. Details are referred to the previous work [39]. After $\mathcal{E}(k)$ is estimated, the joint probability distribution of the role variables in (2.11) is factorized as the product of potential functions over $\mathcal{E}(k)$.

2.4.2 Spatio-temporal MRF Model

In this subsection, an approach is presented for reconstructing the temporal evolution of random variables $X(k)$ across frames to recursively estimate the joint role

labeling using a sequence of feature vectors and the DMRF model of a single frame derived in (2.11). Let $\gamma(X_i(k-1), X_i(k))$ denote the temporal potential function that measures the compatibility of temporal transitions between $X_i(k-1)$ and $X_i(k)$. The temporal potential function can be modeled by a transition matrix $W_t \in \mathbb{R}^{R\times R}$ such that

$$\gamma(X_i(k-1)=n, X_i(k)=m) \triangleq \mathbf{1}_n^T \cdot W_t \cdot \mathbf{1}_m \tag{2.21}$$

The temporal potential function can be integrated with the pairwise potential function to construct a joint state transition function

$$P(X(k)|X(k-1)) \propto \prod_{i\in\mathcal{N}} \gamma(X_i(k-1), X_i(k)) \prod_{i,j\in\mathcal{E}(k)} \psi(X_i(k), X_j(k)) \tag{2.22}$$

On the other hand, the product of unary potentials can be treated as the joint likelihood function, assuming that individual features are conditionally independent given the realization of random variables

$$P(Z(k)|X(k)) = \prod_{i\in\mathcal{N}} P(Z_i(k)|X_i(k)) = \prod_{i\in\mathcal{N}} \phi(X_i(k), Z_i(k)) \tag{2.23}$$

Let $Z(1,k) = \{Z(l)|1 \le l \le k\}$ denote a sequence of extracted feature vectors obtained from an initial frame ($l = 1$) up to the k^{th} frame. Then, the joint probability of $X(k)$ can be recursively estimated from $Z(1,k)$ in a fashion similar to Bayesian filtering [37]

$$P(X(k)|Z(1,k)) = \frac{1}{\hat{C}} P(Z(k)|X(k)) \sum_{X(k-1)} P(X(k)|X(k-1)) P(X(k-1)|Z(1,k-1)) \tag{2.24}$$

where $\hat{C}$ is the partition function that guarantees $P(X(k)|Z(1,k))$ is a valid distribution. The proposed spatio-temporal MRF model is illustrated in Fig.2.8. The challenge arises because $P(X(k)|Z(1,k))$ is a multi-dimensional joint distribution that has significant computational ramifications. In order to keep the computation tractable, the joint distribution is achieved via the Markov chain Monte Carlo

(MCMC) sampling method [3,18,29] by constructing a set of random samples that constitute a Markov chain whose stationary distribution converges to the desired distribution.

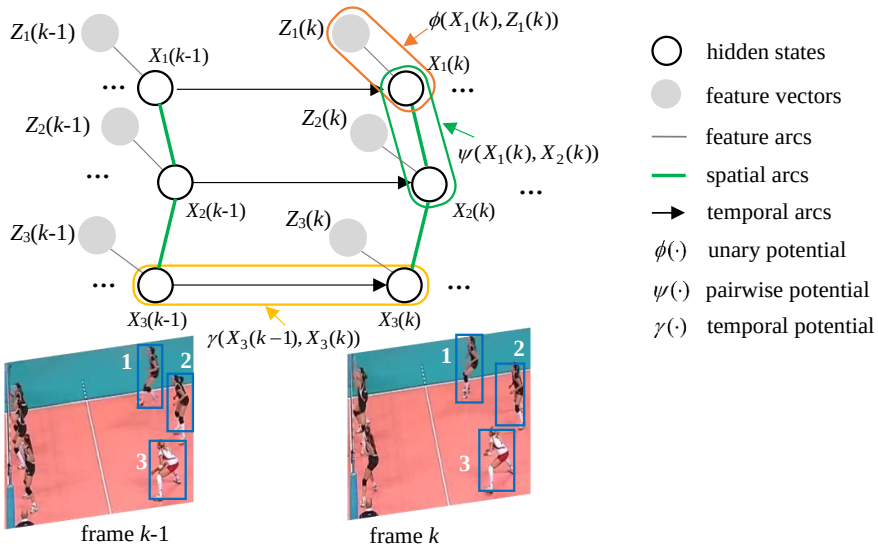


Figure 2.8: Spatio-temporal MRF model for modeling players' roles.

2.4.3 Learning of Potential Functions

The MRF model is trained in an incremental manner [4] in which the parameters of unary potentials are first trained and then fixed to learn the pairwise potentials. This incremental training allows the pairwise potentials to be built upon strong unary potentials, which makes the training more efficient because otherwise the pairwise potentials may not be able to capture the significant interactions from misleading unary potentials. In particular, the unary potential is trained by minimizing the cross entropy loss function, whereas the pairwise potential can be learned using the structural support vector machine framework [39, 71] or using domain knowledge about the relationship between different roles. This two-stage

learning is performed in a frame-wise manner by leaving out the temporal transition matrix, which is fine-tuned at last on the training database. This incremental training allows the model to learn specific information presented in each potential function [4] and reduces the computational burden that would otherwise be incurred if all potential functions are learned together.

2.4.4 MCMC Inference

Inferring a role labeling $X(k)$ from the joint distribution $P(X(k)|Z(1,k))$ suffers from an enormous combinatorial complexity. Naively searching through the set of all possible labeling is intractable because the set has a cardinality that is exponential in the number of states. This work adopts the MCMC method [3, 29] to address the computational ramifications, which generates a Markov chain over the space of the joint configuration $X(k)$, such that the chain has a stationary distribution converging to $P(X(k)|Z(1,k))$. Assume the posterior $P(X(k-1)|Z(1,k-1))$ at time $k-1$ is represented by a set of $N_s \in \mathbb{R}^+$ samples $\{X(k-1)^{(\ell)}\}_{l=1}^{N_s}$, and each sample corresponds to a joint role labeling of all players, i.e., $X(k-1)^{(\ell)} = [X_1(k-1)^{(\ell)} \quad \dots \quad X_N(k-1)^{(\ell)}]^T$. Then, the Monte Carlo approximation to the posterior distribution in (2.24) at time k is

$$P(X(k)|Z(1,k)) \approx \frac{1}{\hat{C}} P(Z(k)|X(k)) \sum_{\ell=1}^{N_s} P(X(k)|X(k-1)^{(\ell)}) \tag{2.25}$$

Substitute (2.22-2.23) into (2.25), which gives

$$P(X(k)|Z(1,k)) \approx \frac{1}{\hat{C}} \prod_{i \in \mathcal{N}} \phi(X_i(k), Z_i(k)) \prod_{i,j \in \mathcal{E}(k)} \psi(X_i(k), X_j(k)) \sum_{l}^{N_s} \prod_{i} \gamma(X_i(k-1)^{(\ell)}, X_i(k)) \tag{2.26}$$

resulting in a sample-based representation for the distribution $P(X(k)|Z(1,k)) \approx \{X(k)^{(\ell)}\}_{\ell=1}^{N_s}$. The Metropolis-Hastings (MH) algorithm with the symmetric ran-

dom walk proposal distribution [3, 29] is implemented for simulating the Markov chain.

2.5 Anticipation Model

The goal of action anticipation is to predict a set of key players and their future actions as time evolves. Existing methods can not be easily adapted to the action anticipation problem (*problem 2*) because they do not take into account the time varying team strategy and players' roles, which are core to team actions. The anticipation model presented in this work differs from the existing methods by the input information exploited, which aggregates inferred hidden variables (inferred team strategy and players' roles) with explicit visual features, forming a rich input representation. The prediction of key players, $\mathcal{K} \subset \mathcal{N}$, is first achieved via (2.8-2.9). Subsequently, for each predicted key player, $i \in \mathcal{K}$, the action anticipation model merges four types of information corresponding to the current scene segment, i.e., $\{S_m, X_{i,m}, A_{i,m}, P_{i,m}\}$, to anticipate the future action $A_{i,m+1}$. The representation of input segments directly affects the learning efficiency and computational cost of the model. Thus, it is worth exploring a compact representation of $\{S_m, X_{i,m}, A_{i,m}, P_{i,m}\}$. Based on the definition of the scene change point and scene segment in (2.4-2.6), the segment variable of team strategy, S_m (Table 2.1), takes a constant value within the scene segment V_m. Hence, S_m can be fully defined by its value at the current time, κ, and the duration of V_m up to κ, that is, $S_m \triangleq (S(\kappa), d_m)$. Although values of $X_{i,m}$, $A_{i,m}$, and $P_{i,m}$ can vary within a scene segment, it is observed that future actions are most closely related to their respective values at the current time κ. Furthermore, this work seeks a frame-wise representation of the anticipation input and output, such that they can be updated

instantaneously as time unfolds. As a result, only $A_i(\kappa)$, $X_i(\kappa)$, and $\mathbf{p}_i(\kappa)$ are preserved as inputs, as shown in Fig. 2.9, which, together with $(S(\kappa), d_m)$, constitute an input vector

$$\mathbf{u}_i(\kappa) = [S(\kappa) \quad X_i(\kappa) \quad A_i(\kappa) \quad \mathbf{p}_i(\kappa)^T \quad d_m]^T \tag{2.27}$$

where the time-varying characteristic of d_m represents the variable duration of the team strategy $S(\kappa)$. Likewise, the anticipation output, $A_{i,m+1}$, is designed to have an instantaneous representation of the future actions. Let t_s denote the *time to onset*, that is, the amount of time until the onset of $A_{i,m+1}$, and let d_{m+1} denote the duration of $A_{i,m+1}$. Then, $A_{i,m+1}$ can be defined as $A_{i,m+1} \triangleq (A_i(\kappa + t_s), d_{m+1})$, as shown in Fig. 2.9(b). Equivalently, $A_{i,m+1}$ can be specified by a vector representation comprising three unknown variables

$$\mathbf{y}_i(\kappa) = [A_i(\kappa + t_s) \quad t_s \quad d_{m+1}]^T \tag{2.28}$$

It follows from (2.27-2.28) that the goal of the action anticipation task is to predict $\mathbf{y}_i(\kappa)$ based on $\mathbf{u}_i(\kappa)$ as time evolves.

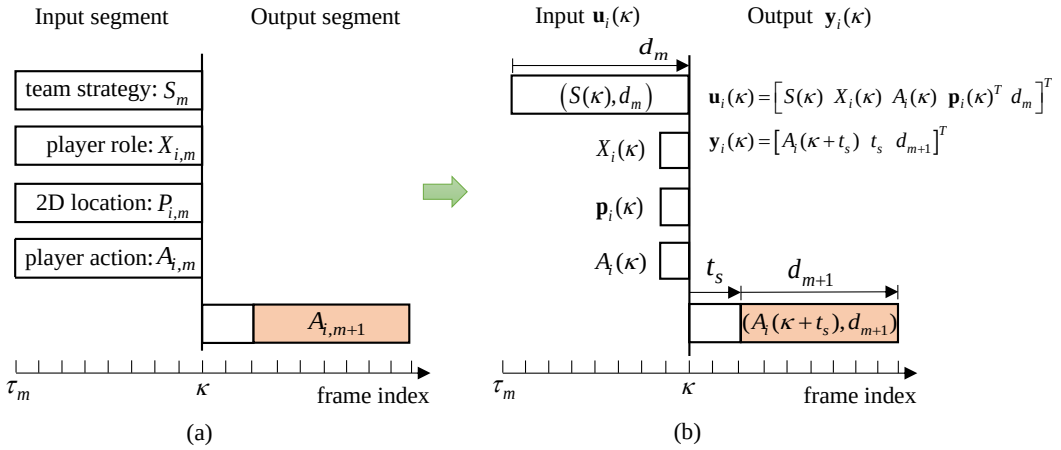


Figure 2.9: Input and output segment for action anticipation of the i^{th} key player (a) and the simplified instantaneous representation (b).

An MLP is designed to perform the anticipation task based on the proposed

input-output representation in (2.27-2.28). Categorical variables in $\mathbf{u}_i(\kappa)$ are converted to binary representations via one-hot encoding. The encoded $\mathbf{u}_i(\kappa)$ is passed through two branches, as shown in Fig.2.10, where the top branch is configured to output a probability distribution for the discrete variable $A_i(\kappa + t_s)$ and the bottom branch generates two positive scalar values for the continuous variables, t_s and d_{m+1}, respectively. In particular, the top branch first maps the input vector to a latent vector, $\mathbf{h}_1$, using a fully connected (FC) layer followed by the relu-activation function

$$\mathbf{h}_1 = relu(W_{h1}\mathbf{u}_i(\kappa)) \tag{2.29}$$

where W_{h1} is the weight matrix. Subsequently, $\mathbf{h}_1$ is fed to the output layer, composed of a FC layer and the softmax activation function, to generate the conditional probability distribution of $P(A_i(\kappa + t_s)|\mathbf{u}_i(\kappa))$. Let $\mathcal{A} = \{1, 2, \ldots, A\}$ denote the range of the action classes, where each integer, $a \in \mathcal{A}$, represents a semantic action label, and $W_{o1} = [\mathbf{w}_1 \quad \ldots \quad \mathbf{w}_A]^T$ denote the weight matrix of the output FC layer. Then, $P(A_i(\kappa + t_s) = a|\mathbf{u}_i(\kappa))$ is computed as

$$P(A_i(k + t_s) = a|\mathbf{u}_i(k)) = \frac{\exp(\mathbf{w}_a^T\mathbf{h}_1)}{\sum_{a'=1}^{A}\exp(\mathbf{w}_{a'}^T\mathbf{h}_1)}, \quad a \in \mathcal{A} \tag{2.30}$$

and the action class with the highest probability is chosen as the anticipated action. Although the bottom branch adopts the same structure as the top branch, the FC-layers can have different dimensions and the output activation function is designed to be a relu-activation function for guaranteeing real positive values of t_s and d_{m+1}. Let W_{h2} denote the weights of the hidden FC layer in the bottom branch, and $W_{o2} = [\mathbf{w}_\tau \quad \mathbf{w}_d]^T$ denote the weights of the corresponding output FC layer.

Then, t_s and d_{m+1} are obtained as follows:

$$\mathbf{h}_2 = relu\ (W_{h2}\mathbf{u}_i(\kappa)) \tag{2.31}$$

$$t_s = relu\ (\mathbf{w}_\tau^T \mathbf{h}_2)$$

$$d_{m+1} = relu\ (\mathbf{w}_d^T \mathbf{h}_2)$$

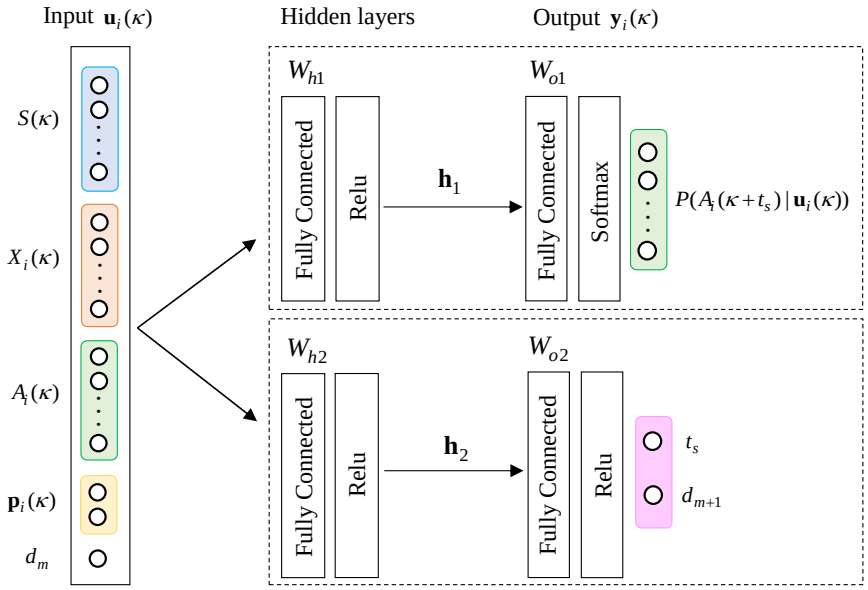


Figure 2.10: MLP for action anticipation.

The complete set of the MLP parameters, $\Theta_A = \{W_{h1}, W_{h2}, W_{o1}, W_{o2}\}$, is trained by minimizing an anticipation loss that is a function of the ground truth and the actual predicted output. In particular, the loss function is formulated as the summation of the cross-entropy loss of the discrete action variable, $A_i(\kappa + t_s)$, and the mean squared loss of the two timing variables, t_s and d_{m+1}.

In summary, the input-output representation in (2.27-2.28) allows the input to be updated in each frame and the anticipation output to progressively change as more observations stream in. Furthermore, the trained model is shared across all players, and, therefore, anticipation for multiple players can be performed simultaneously by constructing an input vector for each of them.

2.6 Experiments

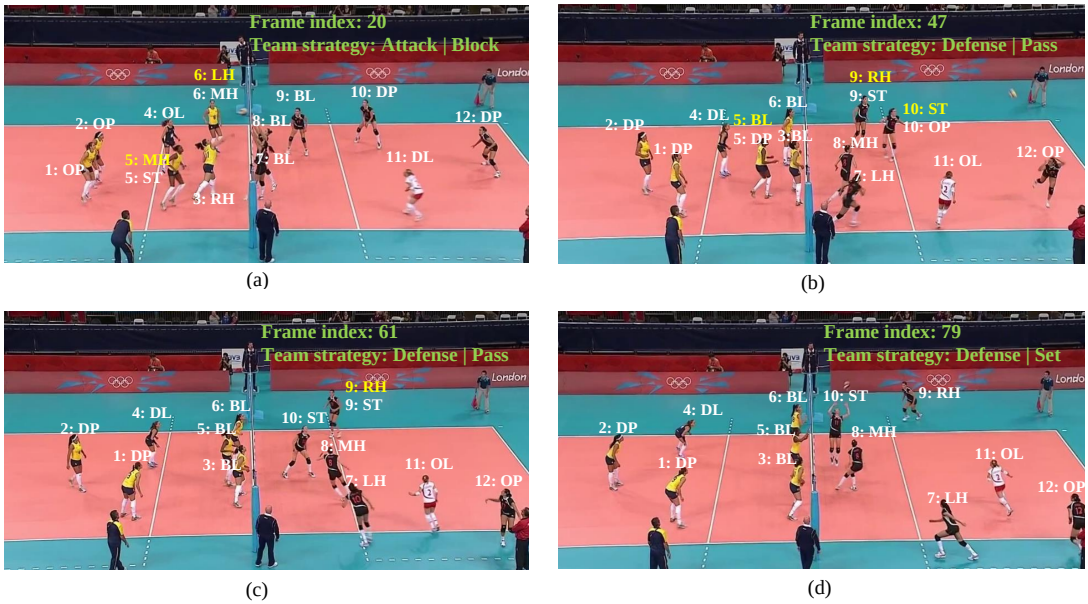
In this section, experiments are conducted in order to validate the accuracy of the proposed methods. Using the Volleyball Activity Dataset [67], a supervised training database for the proposed inference and anticipation algorithms was obtained by annotating team strategies, player's roles, player's actions and other necessary visual and positional information. Despite additional supervision required for learning the intermediate hidden variables, the overall labeling effort is less than that required by deep neural network models for action anticipation trained solely on images. The reason is that the proposed approach exploits the problem structure and incorporates domain knowledge before training the DMRF and MLP models. The inference and anticipation results are analyzed qualitatively and quantitatively on the testing data. Comparison with existing work on action anticipation was unfortunately not possible because existing algorithms are only applicable to single-agent or dual-agent activities [24, 47, 91, 109]. Therefore, the experiments in this work focused on evaluating the overall performance of the inference and anticipation model. Moreover, comparative studies (Section 2.6.2) that involve three types of experiments are carried out to determine the anticipation performance variability as a function of the hidden variables and corresponding inference accuracy.

2.6.1 Inference and Action Anticipation Results

The DMRF inference results are shown in Fig. 2.11 for a sample sequence of frames extracted from a testing video clip, where the inferred team strategies and players' roles evolve over time. Notice that a team strategy spans over several consecutive

frames, during which the action and spatial layout of players may be shifted, but not qualified to be inferred as a different category. The DMRF model presented in Section 2.4 correctly infers that the team strategy changes from "attack | block" (Fig.2.11 (a)) to "defense | pass" (Fig. 2.11 (b-c)) to "defense | set" (Fig.2.11 (d)), exemplifying the algorithm's robustness to the dynamically evolving scenes. Similarly, the players' roles change as the game unfolds. For example, the role of player 3 alters from "right-hitter" to "blocker", whereas player 7, originally a "blocker", becomes a "left-hitter". For comparison purposes, ground truth labels of the false inference results are shown in yellow above the (white) inferred roles in Fig. 2.11. It is seen that inference failures are likely to happen when players are shifting to new locations. For instance, the algorithm mistakenly infers the roles of player 9 and 10 in Fig.2.11 (b). However, as more observations are received, the updated inference results would be self-corrected and thus match the ground truth (Fig. 2.11 (c-d)). It is notable that such kind of error is inevitable, even for human experts who identify players' roles in a transitioning process without further information such as a player's name or jersey number, which is out of the scope of this work.

Action anticipation is performed using inferred team strategy and players roles, which is in accordance with Experiment 3 in Section 2.6.2. Anticipation results are shown in Fig. 2.12-2.14 for two testing video clips with a framerate of 25 fps. Fig. 2.12 (a) shows that the setter, marked by the black bounding box, is predicted as the key player who will dominate the game based on the inferred role and team strategy. The observed action, the ground truth future action, and the anticipated action are visualized in the bar chart of Fig.2.12 (b), and the red vertical line indicates where the current frame is temporally located in the testing sequence. More specifically, the first segment of the middle and bottom bar is of the same



BL: blocker, DP: defense-passer, DL: defense-libero, MH: middle-hitter, LH: left-hitter, OP: offense-passer, OL: offense-libero, RH: right-hitter, ST: setter

Figure 2.11: Evolution of the inferred team strategy from "attack | block" (a) to "defense | pass" (b-c) to "defense | set" (d) and the inferred players' roles in each frame.

color as the top bar, representing that the current action would keep until the onset of the future action with a different color. The anticipation MLP gives the credible prediction of the key player who will be setting the ball, in spite of the discrepancy of 7 frames (0.28s) between the predicted timing and ground truth, as shown in the length of the middle and bottom bars (Fig. 2.12 (b)). Moreover, as time evolves from Fig. 2.12 (b) to 2.12 (d), the difference in timing gradually reduces, indicating the update of anticipation result as the future unfolds.

On the other hand, more than one individuals can be predicted as key players, as shown in Fig.2.13, where the three key players are marked by the black bounding boxes. Based on a short observation sequence of 7 frames (0.28 s), the anticipation MLP predicts that both middle-hitter (player 8) and left-hitter (player 10) will launch a spiking, although the ground truth shows only the left-hitter eventually

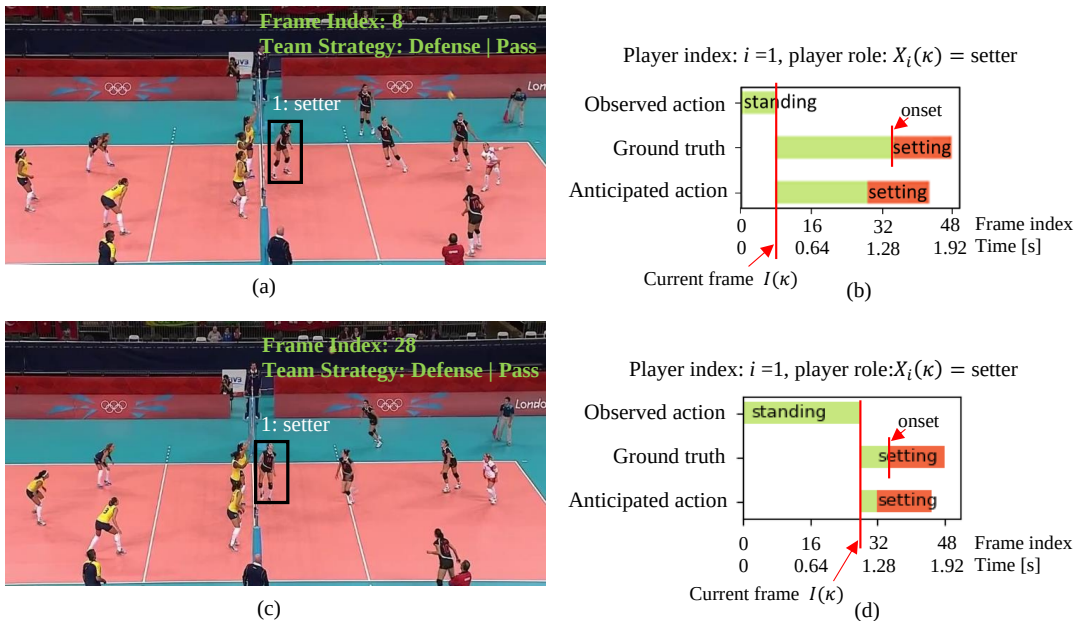


Figure 2.12: The anticipated key player and action in the 8^{th} frame (a-b) and the 28^{th} frame (c-d) in a testing video clip.

spikes the ball. Such mistake or conservatism is inevitable because it is yet uncertain in this moment who would launch the final attack as they both have great opportunity. This is also a general tactic when one of the hitters potentially makes a feint in order to distract blockers of the opposing team. As the game proceeds, the anticipated action of the middle-hitter evolves, finally reaching to the ground truth, as illustrated in Fig. 2.14(c). In addition, the onset and duration of the anticipated actions are indicated by the change of color and the length of the bars, respectively.

2.6.2 Performance Analysis and Results

The effectiveness of the inference and action anticipation algorithms presented in the previous sections is demonstrated using the metrics known as multi-class

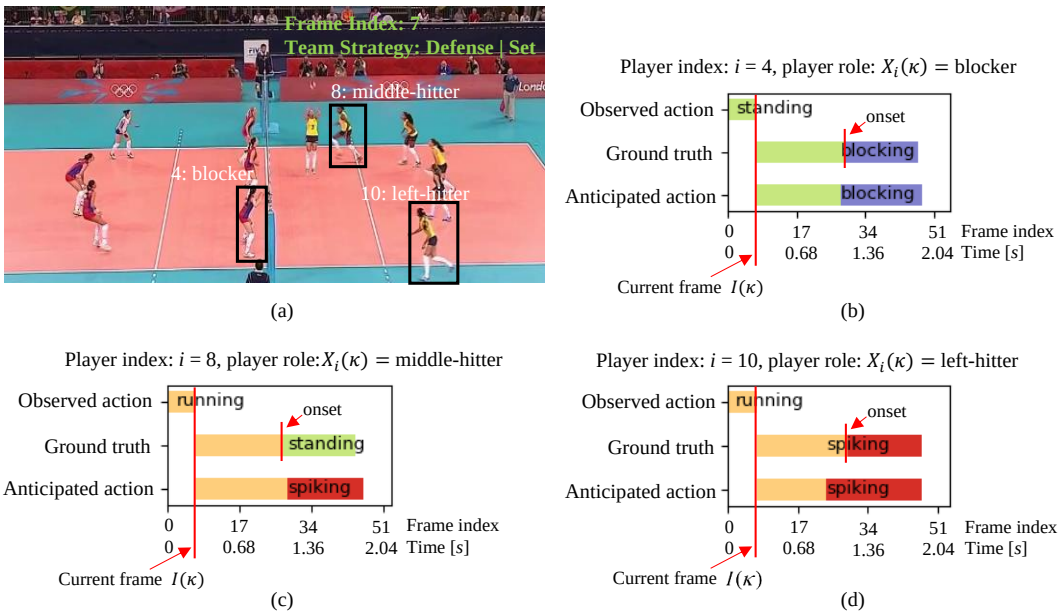


Figure 2.13: Three key players (a) and the anticipated actions (b-d) in the 7^{th} frame of a testing video.

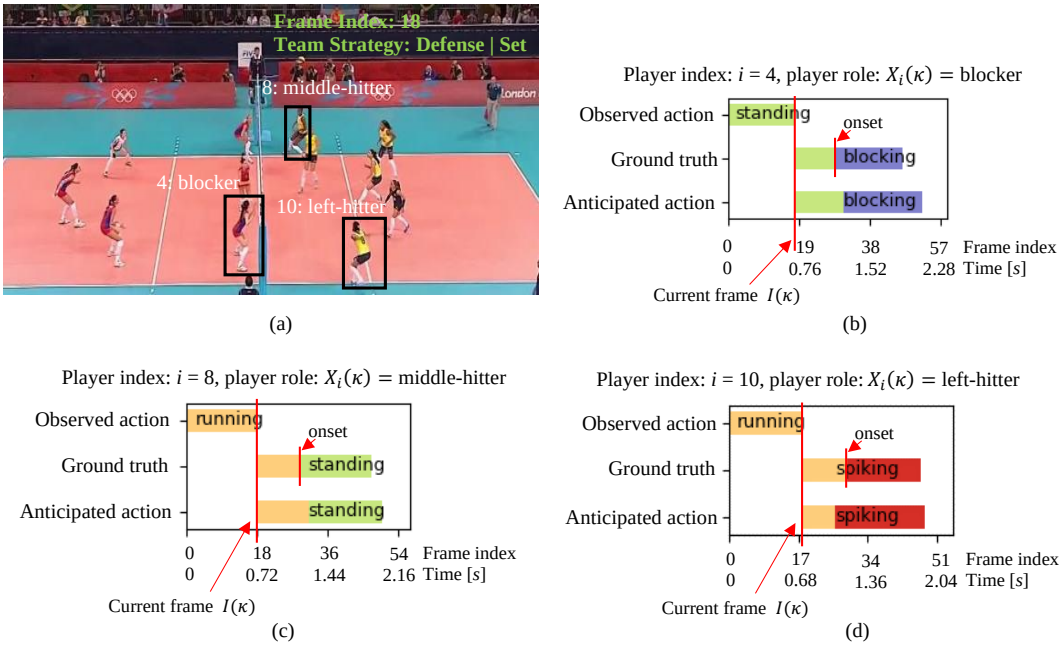


Figure 2.14: Three key players (a) and the anticipated actions (b-d) in the 18^{th} frame of a testing video.

average precision (APr), multi-class average recall (ARc), and multi-class average accuracy (Ac). The APr score concerns the proportion of inferred values, consisting of both true positive (TP) and false positive (FP), that is actually true (i.e., APr = TP/(TP + FP)). In contrast, the ARc score is the proportion of ground truth labels, including both true positive (TP) and false negative (FN), that is correctly inferred (i.e., ARc = TP/(TP + FN)). For both metrics, higher values correspond to better performance. Finally, Ac is defined as the harmonic mean of APr and ARc, which is also known as the F1-score

$$\mathrm{Ac} = 2\,\frac{\mathrm{APr} \times \mathrm{ARc}}{\mathrm{APr} + \mathrm{ARc}} \tag{2.32}$$

Two hidden variables, the team strategy ($S(\kappa)$) and the players' roles ($X(\kappa)$), are inferred in each frame with the overall results presented in Table 2.2. A comparative study is performed to assess the performance of the anticipation model as well as the robustness of the holistic framework, i.e., the dependence of the anticipating ability on the inferred hidden variables.

The comparative study involves three types of experiments aimed at determining the performance variability as a function of the hidden variables and corresponding inference accuracy:

- Experiment 1: perfect knowledge of team strategy ($S(\kappa)$) and player roles ($X(\kappa)$);
- Experiment 2: inferred team strategy ($S(\kappa)$) and perfect knowledge of player roles ($X(\kappa)$);
- Experiment 3: inferred team strategy ($S(\kappa)$) and player roles ($X(\kappa)$).

The purpose of the first experiment is to determine the performance of the action anticipation independently of the inference algorithm. The results in Table 2.2

Table 2.2: Inference and Action Anticipation Performance

Experiment	**Average Precision**	**Average Recall**	**Average Accuracy**
Team strategy inference	0.87	0.82	84.43%
Role inference	0.88	0.86	86.99%
Experiment 1	0.92	0.89	90.47%
Experiment 2	0.88	0.86	86.99%
Experiment 3	0.81	0.80	80.50%

show the important influence that the player role and team strategy have on the solution of the action anticipation problem (*problem 2*). As a result, the action anticipation performance degrades as errors are introduced in the inference stage, through Experiments 2 and 3. This is because, despite the excellent performance of the DMRF algorithm (Table 2.2), inferring the hidden variables from video introduces some errors (compared to perfect knowledge) that are, then, propagated to the action anticipation algorithm.

The advantage of this holistic approach is that action anticipation draws from the aggregation of both implicit hidden variables and explicit visual features. Therefore, errors from one source of information are potentially compensated by information obtained from other features. The performance results could be further improved by leveraging other variables and sensor modalities, which are easily incorporated in the proposed approach by augmenting the feature vectors. In addition, an ablation study is performed with a variant of the proposed model that excludes the inferred players' roles from the proposed holistic framework shown in Fig.2.4:

- Experiment 4: action anticipation without player roles ($X(\kappa)$) in the model input.

Table 2.3: Ablation Study Regarding the Hidden Role Variables

Model	**Average Precision**	**Average Recall**	**Average Accuracy**
Experiment 3	0.81	0.80	80.50%
Experiment 4	0.71	0.69	69.99%

Results of Experiment 4 are compared against results of the holistic approach (Experiment 3) in Table 2.3. Without the knowledge of players' roles, Experiment 4 sees a significant drop in the action anticipation accuracy, which, by contrast, shows the improvement brought by the inference of hidden role variables to the solution of the action anticipation problem (*problem 2*).

The ability to predict the onset and duration of a future action is also critical, as well as coupled with the problem of anticipating the action type, since many algorithms assume the starting time is known or even observed. Team sports offer an excellent benchmark problem, because players constantly adjust the timing and duration of their actions, speeding up or slowing down actions and behaviors for strategic purposes. These difficulties are exacerbated by varying contexts, for example, because the trajectory of the ball and the skills of the opponents differ greatly from one team to another, yielding different samples in the training and testing datasets. The performance of action timing prediction is evaluated by the time-relative error, which is defined as the ratio of the absolute prediction error to the corresponding prediction horizon. Then, the mean of the time-relative error (MTRE) of each testing instance is used as the metric to assess the performance on the test database. The proposed model achieves an MTRE of 14.57% and 15.67% for the prediction of the action onset and duration, respectively. When compared to the LSTM solution proposed in [47] for anticipating an individual's cooking

activity, the DMRF-MLP approach presented in this work achieves a comparable prediction horizon (0.48-1.84 s, versus 0.25-2 s) using a smaller observation time window (0.12-1.80 s, versus 1.75-3.50 s) and, thus, is applicable to fast actions and highly dynamic activities, such as sports.

2.7 Conclusion

This chapter presents a holistic approach that integrates image recognition, state estimation, and inference of hidden variables for the challenging problem of action anticipation in human teams. The approach is demonstrated on the team sport of volleyball, in which the team strategy and players' roles are unobservable and change significantly over time. The team strategy is first inferred by constructing a team feature descriptor that aggregates domain knowledge of volleyball games and features of individual players. Sequentially, the players' roles, modeled probabilistically as the DMRF graph, can be inferred using a Markov chain Monte Carlo sampling method. The dynamic graph structure that captures player interrelationships can be estimated by solving an integer linear program in each frame. By leveraging holistic information about the scene, including inferred team strategy, players' roles, as well as domain knowledge and instantaneous visual features, the action anticipation MLP is able to predict the semantic label and timing of the future actions by multiple interacting key players on the team. The numerical experiments show that this novel approach achieves an average accuracy of 84.43% for team strategy inference, 86.99% for role inference, and 80.50% for action anticipation. Additionally, the action onset and duration are predicted with a mean time-relative error of 14.57% and 15.67%, respectively.

CHAPTER 3

DECENTRALIZED COORDINATION AND CONTROL OF MULTI-ROBOT NETWORKS FOR ACTIVE TARGET TRACKING

3.1 Introduction

The coordination and control of multi-robot networks (MRNs) represent an important family of problems in robotics that has motivated numerous research directions in the past decade as more complex missions are envisioned [62, 85]. For example, mobile MRNs composed of unmanned aerial vehicles (UAVs) and/or unmanned ground vehicles (UGVs) are used to fulfill difficult tasks for urban search and rescue [9, 23], warehouses automation [40, 97], information gathering [17, 111], and surveillance [79, 134]. By intelligent coordination and autonomous control, MRNs can operate in parallel to reduce the task completion time, communicate mission-relevant data to gain situational awareness, and create redundancy to improve fault tolerance, which renders them more promising than single robot systems.

Recently, using MRNs to estimate the unknown kinematic states of moving targets, also known as target tracking, has drawn significant attention. Unlike most previous work that focused on the estimation aspect of the tracking problem using passive information received by static sensors [31, 102, 127, 148], research on MRNs aims to actively and cooperatively determine robot control in order to optimize the tracking performance. In earlier studies, algorithms were developed for MRNs to track a single target by minimizing the target uncertainty [58] or maximizing the target detection probability [144]. In addition, a competitive policy is investigated for the single target tracking [70], where only the winning robots in MRNs are activated to perform the tracking task. However, these algorithms

cannot be directly applied to multi-target tracking problems, which require effective network coordination to assign targets amongst the robots. In general, target assignment can be divided into two classes: i) single assignment that matches robots to an equal number of targets on a one-to-one basis [23, 118, 119, 145]; ii) multi-assignment that associates robots with a larger number of targets in a one-to-many relationship [10, 122]. Although there existed both centralized [72, 141] and decentralized solutions [14,98] to the single-assignment problem, the extension to multi-assignment is non-trivial because the latter essentially belongs to the class of combinatorial optimization problems which are NP-hard [110, 149].

This chapter focuses on the coordination and control of multi-robot networks (CCMRN) for the non-trivial tracking scenarios where targets outnumber the robots, which brings the challenge of simultaneously finding target assignment and determining robots' control in real time. Some existing methods [111,121,122] converted CCMRN to an integer linear program, which restricts the robot control to a few pre-specified actions, such that the decision variables are solely in the discrete space. On the other hand, CCMRN was considered as a graph-based path planning problem in [9, 10], where the goal is to ensure balanced observation of all targets by finding a walk on the graph that represents feasible movements of the robots. However, these approaches used centralized strategies and separated path planning from robot control by merely determining the robot waypoints without taking the robot kino-dynamic constraints into account. In addition, most of the above literatures assumed that robots are equipped with omnidirectional sensors and that multiple targets can be easily distinguished from each other, which overlook the difficulty of real-world target detection, classification, and state estimation using robot onboard sensors.

This chapter, easing the above limitations, proposes a new decentralized approach to the CCMRN problem that enables the maximization of the network tracking quality in real time regarding both the discrete target assignment and continuous control command. Two novel methods, the group-based algorithm and the bundle-based algorithm, are developed to find adaptive target assignments through multi-hop communication, with the latter achieving more effective coordination and guaranteeing $\frac{1}{2}$-approximation in the worst-case. A new tracking utility function is proposed for the local estimation of the global network tracking quality, which accounts for the sensor geometries, kinematic constraints, control bounds, and collision avoidance. The simulation results in Section 3.7 show that the performance of the proposed approaches is very close to that of the optimal solution and is higher than the other decentralized methods. Furthermore, the computational complexity analysis shows that the proposed methods have polynomial time complexity. Finally, the applicability of the proposed approaches is demonstrated in real-world physical experiments.

3.2 Problem Formulation and Assumptions

This chapter considers the problem of coordinating and controlling a network of mobile robots, such as UGVs, to track multiple moving targets for purposes such as surveillance and security. Let $\mathcal{N} = \{1, \ldots, N\}, N \in \mathbb{N}^+$ denote the index set of robots in a MRN, where N is the total number of robots and is known a-*priori*. The robots are assumed to operate in a closed and bounded two-dimensional (2D) workspace $\mathcal{W} = [0, L_x] \times [0, L_y] \subset \mathbb{R}^2$. Let $\mathcal{F}_\mathcal{W}$ denote the inertial frame, with origin $\mathcal{O}_\mathcal{W}$, embedded in $\mathcal{W}$ such that the $x_I y_I$-plane aligns with the ground plane. A moving Cartesian frame $\mathcal{F}_{\mathcal{A}_i}$ is embedded in robot $i, i \in \mathcal{N}$ by placing the

origin $\mathcal{O}_{\mathcal{A}_i}$ at the pinhole of the robot camera. The state vector of robot i can be represented by $\mathbf{s}_i = [x_i \quad y_i \quad \theta_i]^T$, where x_i, y_i, and θ_i are the 2D coordinates and orientation of $\mathcal{F}_{\mathcal{A}_i}$ with respect to $\mathcal{F}_{\mathcal{W}}$, as shown in Fig. 3.1. The robot state vector $\mathbf{s}_i$ can be estimated from onboard odometry sensors and communicated with other robots in the MRN. In addition, it is assumed that $\mathbf{s}_i$ obeys the unicycle motion model [45, 88],

$$\dot{\mathbf{s}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix} = \mathbf{f}(\mathbf{s}_i, \mathbf{u}_i), \quad \forall i \in \mathcal{N} \tag{3.1}$$

where the robot control vector, $\mathbf{u}_i = [v_i \quad w_i]^T \in \mathbb{R}^2$, consists of the linear velocity v_i and angular velocity w_i. Assuming a constant sampling interval $\Delta t \in \mathbb{R}^+$, the robot state and control vector at any time $k\Delta t$ can be represented by $\mathbf{s}_i(k) = \mathbf{s}_i(k\Delta t)$ and $\mathbf{u}_i(k) = \mathbf{u}_i(k\Delta t)$. Then, the state and control of the robot network can be written as $\mathbf{s}(k) = [\mathbf{s}_1^T(k) \quad \dots \quad \mathbf{s}_N^T(k)]^T \in \mathbb{R}^{3N}$ and $\mathbf{u}(k) = [\mathbf{u}_1^T(k) \quad \dots \quad \mathbf{u}_N^T(k)]^T \in \mathbb{R}^{2N}$, respectively.

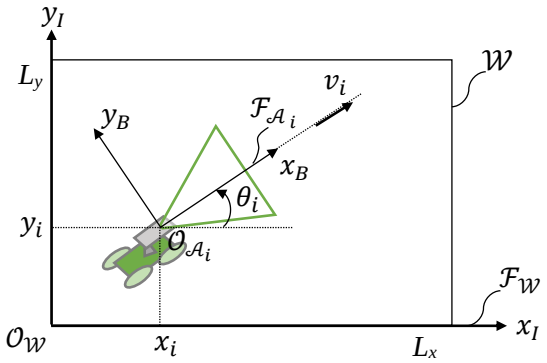


Figure 3.1: Definition of the state of a robot.

Consider that the MRN is interested in tracking a set of dynamic targets indexed by $\mathcal{M} = \{1, \dots, M\}, M \in \mathbb{N}^+$, where $j \in \mathcal{M}$ represents a unique target identity (ID). The state of target j is denoted by $\mathbf{x}_j(k) =$

$[x_j(k) \quad y_j(k) \quad v_{x,j}(k) \quad v_{y,j}(k)]^T \in \mathbb{R}^4$, which comprises the position and velocity with respect to $\mathcal{F}_\mathcal{W}$. Then, the states of all targets can be written as $\mathbf{x}(k) = [\mathbf{x}_1^T(k) \quad \dots \quad \mathbf{x}_M^T(k)]^T \in \mathbb{R}^{4M}$. Assuming constant target velocity and additive Gaussian process noise $\mathbf{w}(k)$, the target motion model at any discrete time k is given by

$$\mathbf{x}_j(k) = \mathbf{F}\mathbf{x}_j(k-1) + \mathbf{w}(k), \quad \mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{3.2}$$

where $\mathbf{F} \in \mathbb{R}^{4\times4}$ is the known state transition matrix [6], and $\mathbf{Q}$ is the covariance matrix of the process noise.

As targets travel in the workspace, robots actively observe the targets that are inside the robots' field-of-view (FOV), which is a compact subset of $\mathcal{W}$ denoted by $\mathcal{S}_i \subset \mathcal{W}, \forall i \in \mathcal{N}$. Existing methods that use vision-based sensors often measure target position in the camera frame or virtual image plane [50, 80, 136], leading to complex nonlinear measurement models. In contrast, this paper designs an online sensing pipeline (Section 3.3.2) that relies on RGB image, depth image, and robot localization to directly measure target positions in the inertial frame $\mathcal{F}_\mathcal{W}$ as follows:

$$\mathbf{z}_{i,j}(k) = \mathbf{H}\mathbf{x}_j(k) + \mathbf{v}(k) \quad \text{if} \quad \mathbf{x}_j(k) \in \mathcal{S}_i(k) \tag{3.3}$$

where $\mathbf{z}_{i,j}$ indicates the observation of the j^{th} target by the i^{th} robot, $\mathbf{H} = [\mathbf{I}_2 \quad \mathbf{0}_2]$ and $\mathbf{v}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is the zero-mean Gaussian noise. Let $\hat{\mathbf{x}}_{i,j}(k)$ represent the state estimate of target j by robot i, which should be distinguished from the true target state $\mathbf{x}_j(k)$. Given the target motion model and measurement model (3.2)-(3.3), $\hat{\mathbf{x}}_{i,j}(k)$ can be recursively estimated by Kalman filtering [137]:

$$\hat{\mathbf{x}}_{i,j}(k) = \begin{cases} \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) + \mathbf{K}(k)\mathbf{e}_{i,j}(k) & \text{if } \mathbf{x}_j(k) \in \mathcal{S}_i(k) \\ \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) & \text{if } \mathbf{x}_j(k) \notin \mathcal{S}_i(k) \end{cases} \tag{3.4}$$

where $\hat{\mathbf{x}}_{i,j}(k-1)$ is the state estimation obtained at time $k-1$, $\mathbf{e}_{i,j}(k) = \mathbf{z}_{i,j}(k) - \mathbf{HF}\hat{\mathbf{x}}_{i,j}(k-1)$ is the innovation term in Kalman filtering, and $\mathbf{K}(k)$ is the Kalman gain matrix in [137].

In addition, because this work considers the non-trivial tracking problems where targets outnumber the robots ($M > N$), not all targets can be consistently tracked at any given time. Therefore, network coordination and control plays a crucial role in determining valid target assignment and in obtaining the most informative measurements.

Definition 1 (Valid Target Assignment) : Given a set of targets indexed by $\mathcal{M} = \{1, \ldots, M\}$, a valid target assignment to a multi-robot network indexed by $\mathcal{N} = \{1, \ldots, N\}$ at any time k is defined as a collection of N subsets, $P(k) \triangleq \{P_1(k), \ldots, P_N(k)\}$, such that every element in $\mathcal{M}$ is included in one and only one subset in $P(k)$, i.e.,

$$P_i(k) \cap P_{i'}(k) = \emptyset, \text{ if } i \neq i', \text{ and } \bigcup_{i=1}^{N} P_i(k) = \mathcal{M} \tag{3.5}$$

The target assignment space, denoted by $\mathcal{P}$, is the family of all valid target assignments that satisfy (3.5). Because by definition $P_i(k) \cap P_{i'}(k) = \emptyset$, no conflicts may arise and, thus, a valid assignment is also called a conflict-free assignment. Moreover, $P(k)$ may vary over time as a function of the robot and target states, which forms a distinct contrast to the existing work that assumes static target assignment [17, 42].

In order to quantify the network tracking quality, a novel utility function, $U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k))$, is defined in Section 3.5, which measures the performance of tracking target j by robot i. Assuming the targets move independently of each other [79], the global tracking quality of an MRN is equivalent to the sum of

tracking utility over all targets:

$$U_g \triangleq \sum_{i \in \mathcal{N}} \left(\sum_{j \in P_i(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \right) \tag{3.6}$$

This work tackles a new network optimization problem which seeks to simultaneously find the optimal target assignment and network control that maximize U_g, i.e.,

$$\max_{P(k), \mathbf{u}(k)} \quad U_g \tag{3.7}$$

$$\text{s.t.} \quad \mathbf{s}_i(k+1) = \mathbf{f}(\mathbf{s}_i(k), \mathbf{u}_i(k)), \quad \forall i \in \mathcal{N} \tag{3.8}$$

$$\mathbf{a}_1 \leq \mathbf{s}_i(k) \leq \mathbf{a}_2, \quad \forall i \in \mathcal{N} \tag{3.9}$$

$$|\mathbf{u}_i(k)| \leq \mathbf{b}, \quad \forall i \in \mathcal{N} \tag{3.10}$$

$$P_i(k) \cap P_i'(k) = \emptyset, \ i \neq i', \ \bigcup_{i=1}^{N} P_i(k) = \mathcal{M} \tag{3.11}$$

where $\leq$ denotes elementwise inequalities, $\mathbf{a}_1 = [0 \quad 0 \quad 0]^T$, $\mathbf{a}_2 = [L_x \quad L_y \quad 2\pi]^T$, and $\mathbf{b}$ represents the physical bounds imposed on the control input. By organizing the robot control $\mathbf{u}_i(k)$ and the resulting one-step future state $\mathbf{s}_i(k+1)$ into a vector $\boldsymbol{\chi}_i(k) = [\mathbf{u}_i(k)^T \quad \mathbf{s}_i(k+1)^T]^T$, the constraints in (3.8)-(3.10) can be written compactly as

$$\mathcal{X}_i \triangleq \{\boldsymbol{\chi}_i(k) \in \mathbb{R}^5 | \ \mathbf{s}_i(k+1) = \mathbf{f}(\mathbf{s}_i(k), \mathbf{u}_i(k)), \\ \mathbf{D}\boldsymbol{\chi}_i(k) \leq \mathbf{d}\}, \ \ \forall i \in \mathcal{N} \tag{3.12}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & -\mathbf{I}_3 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{b} & \mathbf{b} & \mathbf{a}_2 & \mathbf{a}_1 \end{bmatrix}^T \tag{3.13}$$

Similarly, the network control and state can be expressed compactly as $\boldsymbol{\chi} = [\boldsymbol{\chi}_1^T(k) \ \ldots \ \boldsymbol{\chi}_N^T(k)]^T$. Then, the decision variables of the constrained optimization (CO) problem in (3.7)-(3.11) can be represented by the pair $(P(k), \boldsymbol{\chi}(k))$, whose search space is $\mathcal{P} \times \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$.

Directly solving the CO problem in (3.7)-(3.11) relies on accumulating and processing information from all robots in the network [5, 9], which is known to be highly demanding in both computation and communication resources and is prone to single-point failures [40, 113]. This next sections present novel decentralized approaches to solve the network optimization problem in real time, where cooperative robots locally determine their target assignment and control to accomplish the network tracking objective.

3.3 Decentralized Optimization Framework

In a decentralized framework, the network goal is achieved by robots concurrently optimizing their local utility function while satisfying physical constraints and guaranteeing conflict-free target assignments through communication. Therefore, each robot must solve the optimization problem,

$$\max_{P_i(k), \boldsymbol{\chi}_i(k)} \quad \sum_{j \in P_i(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \tag{3.14}$$

$$\text{s.t.} \quad \boldsymbol{\chi}_i(k) \in \mathcal{X}_i \tag{3.15}$$

$$\forall P_i(k) \subseteq P(k), P(k) \in \mathcal{P} \tag{3.16}$$

The computational complexity associated with the optimal solution to (3.14)-(3.16) is as follows.

Theorem 1: Determining the optimal solution to the decentralized CO problem in (3.14)-(3.16), is NP-hard.

Proof: The CO problem in (3.14)-(3.16) can be converted to a mixed integer nonlinear programming (MINLP) by introducing a set of binary variables $\Gamma_{i,j}(k), \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$, such that its value $\gamma_{i,j}(k)$ equals one if target j is assigned to robot i

at time step k, and equals zero otherwise, i.e., $\gamma_{i,j}(k) \in \{0,1\}$. Then, (3.14)-(3.16) can be equivalently written as

$$\max_{\boldsymbol{\chi}_i(k),\gamma_{i,j}(k)} \quad \sum_{j\in\mathcal{M}} U_{i,j}(\mathbf{s}_i(k),\hat{\mathbf{x}}_{i,j}(k)) \cdot \gamma_{i,j}(k) \tag{3.17}$$

$$\text{s.t.} \quad \boldsymbol{\chi}_i(k) \in \mathcal{X}_i \tag{3.18}$$

$$\sum_{i\in\mathcal{N}} \gamma_{i,j}(k) = 1, \quad \forall j \in \mathcal{M} \tag{3.19}$$

$$\gamma_{i,j}(k) \in \{0,1\} \tag{3.20}$$

where (3.18) imposes the kinematic and control constraints on $\boldsymbol{\chi}_i(k)$, and (3.19)-(3.20) guarantees a valid target assignment. It has been well established that MINLP is NP-hard [69, 82]. It concludes that the CO problem in (3.14)-(3.16), which can be equivalent written as an MINLP, is also NP-hard.

3.3.1 Decomposition-Based Approximation

In order to reduce the computational complexity of the above NP-hard problem, this work proposes a novel decentralized framework that decomposes the joint optimization of $P_i(k)$ and $\boldsymbol{\chi}_i(k)$ in (3.14) into a sequential optimization problem with two stages. The validity of the decomposition given independent constraints, such as (3.15)-(3.16), is discussed in [19]. After decomposition, the stage I problem can be summarized as follows.

Problem 1 (Decentralized Coordination): Given the online robot and target estimates, $\mathbf{s}_i(k)$ and $\hat{\mathbf{x}}_{i,j}(k)$, determine the target assignment $P_i(k)$ that maximizes the following tracking utility by keeping the robot state $\mathbf{s}_i(k)$ temporarily constant:

$$\max_{P_i(k)} \quad \sum_{j\in P_i(k)} U_{i,j}(\mathbf{s}_i(k),\hat{\mathbf{x}}_{i,j}(k)), \quad \forall i \in \mathcal{N} \tag{3.21}$$

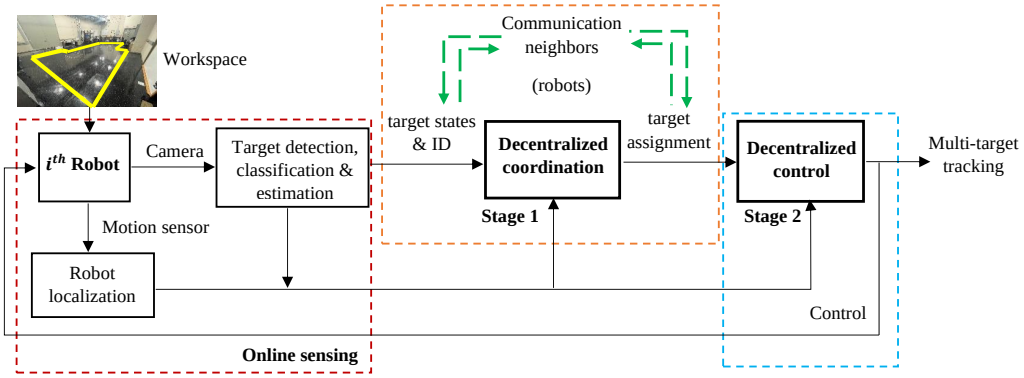


Figure 3.2: The decentralized coordination and control framework implemented on each individual robot.

Let $P_i^*(k)$ denote the optimal solution to (3.21), then the stage II problem can be formulated as follows.

Problem 2 (Decentralized Control): Given the optimal assignment $P_i^*(k)$, find the robot control that maximizes the tracking utility while satisfying the physical constraints, i.e.,

$$\begin{aligned} \max_{\boldsymbol{\chi}_i(k)} \quad & \sum_{j \in P_i^*(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \\ \text{s.t.} \quad & \boldsymbol{\chi}_i(k) \in \mathcal{X}_i, \quad \forall i \in \mathcal{N} \end{aligned} \tag{3.22}$$

Denoting the outcome to (3.22) by $\boldsymbol{\chi}_i^*(k)$, the combined $(P_i^*(k), \boldsymbol{\chi}_i^*(k))$ provides an approximate solution to the decentralized network optimization problem in (3.14)-(3.16). The two-stage decomposition reduces the complete search space in exchange for sub-optimal solutions that are, however, highly efficient and practical for real-time tasks. The decentralized optimization framework hinging on the two-stage decomposition is shown in Fig. 3.2, which elegantly integrates online sensing, communication, coordination, and control to run on every robot for target tracking.

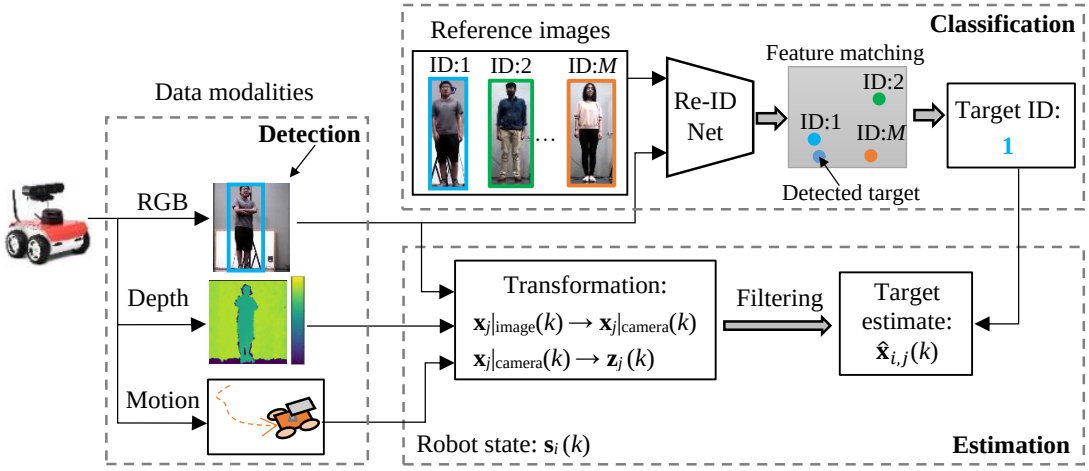


Figure 3.3: Online sensing pipeline for integrated target detection, classification, and state estimation.

3.3.2 Online Sensing

Online sensing plays a fundamental role in active perception of robots. In this work, online sensing is comprised of three sub-tasks: 1) robot state estimation that uses motion sensors for localizing the robot; 2) target detection and classification that relies on convolutional neural networks (CNN) to detect targets and recognize their unique IDs; 3) target state estimation that fuses RGB data, depth data, and robot localization to estimate the positions of dynamic targets while the robot is also in motion. Because robot state estimation can be readily achieved by existing techniques using either onboard odometry sensors or an external localization system, this section focuses on the integrated target detection, classification, and state estimation, as shown in Fig.3.3.

Due to rapid development in computer vision algorithms, human targets can be accurately detected by extracting bounding box approximations from robot RGB imagery using the well-known CNN-based detectors such as Yolo or Mask-RCNN [59,105]. Then, the goal of target classification is to associate the extracted

targets with the pre-specified targets-of-interest they most resemble. Considering that the targets-of-interest are each known to the MRN by a reference image annotated with a unique ID, as shown in Fig.3.3. Owing to the dynamic characteristics of the MRNs, target classification needs to be robust to viewpoint changes to prevent frequent ID-switching as robots and targets move over time. Consequently, a deep neural network trained for person re-identification, also known as the Re-ID Net, is adopted for the task due to its invariance to the translation and rotation of image features [87, 133]. The Re-ID Net implemented on each robot extracts convolutional features from the bounding box detection of each target, and compares those features to that of the reference images, so as to find the closest match as the recognized target. Further details on the implementation of Re-ID are stated in [87, 133].

After a target is detected and associated with a unique ID, the target state is measured using robot onboard sensors. Let $\mathbf{x}_j|_{\text{image}}(k) \in \mathbb{R}^2$ be the $2D$ position of the j^{th} target with respect to the image reference frame, which can be approximated by the image coordinate at the center of the target's bounding box. Given $\mathbf{x}_i|_{\text{image}}$, the target depth, $d_j(k)$, can be obtained by extracting the corresponding pixel value in the depth image. Then, the target position with respect to the camera frame, $\mathcal{F}_{\mathcal{A}_i}$, is given by

$$\mathbf{x}_j|_{\text{camera}}(k) = d_j(k)\mathbf{M}^{-1}[\mathbf{x}_j|_{\text{image}}(k) \quad 1]^T \tag{3.23}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix. The target measurement $\mathbf{z}_{i,j}(k)$ in the inertial frame $\mathcal{F}_{\mathcal{W}}$ is obtained by mapping $\mathbf{x}_j|_{\text{camera}}(k)$ from $\mathcal{F}_{\mathcal{A}_i}$ to $\mathcal{F}_{\mathcal{W}}$

$$\mathbf{z}_{i,j}(k) = \mathbf{R}_i(k)\mathbf{x}_j|^T_{\text{camera}}(k) + \mathbf{r}_i^T(k) \tag{3.24}$$

where $\mathbf{R}_i(k)$ and $\mathbf{r}_i(k)$ are camera extrinsic parameters estimated from the robot

state vector, $\mathbf{s}_i(k) = [x_i(k) \quad y_i(k) \quad \theta_i(k)]^T$, as follows:

$$\mathbf{R}_i(k) = \begin{bmatrix} \cos[\theta_i(k)] & -\sin[\theta_i(k)] & 0 \\ \sin[\theta_i(k)] & \cos[\theta_i(k)] & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{r}_i(k) = [x_i(k) \quad y_i(k) \quad 0]^T \tag{3.25}$$

Compared to the true target state $\mathbf{x}_j(k)$, the measurement $\mathbf{z}_{i,j}(k)$ does not contain the velocity terms and is assumed to be subjected to white, additive Gaussian noise $\mathbf{v}(k)$, which yields

$$\mathbf{z}_{i,j}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_j(k) + \mathbf{v}(k) \tag{3.26}$$

as specified in (3.3). This linear measurement model is used to recursively estimate the target state $\hat{\mathbf{x}}_{i,j}(k)$ in (3.4).

3.3.3 Local Communication

Network communication is assumed to be free of delays and established between robots within a limited communication range r_c measured by the inter-robot Euclidean distance [74, 89, 99, 122]. Since the robot network is dynamic, the network communication topology is time-varying and at any instant of time can be represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the node set $\mathcal{N}$ is defined by the robot index set such that each node corresponds to a robot, and the arc set $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ represents the collection of established communication links [61, 86, 131]. Owing to the limited communication range, robots that are spatially distant can be disconnected, forming local networks of various sizes.

The communication graphs are illustrated using a five-node (robot) network in Fig. 3.4. The five nodes form a single connected communication graph at time step k (Fig. 3.4(a)), which then becomes two locally connected graphs as the network configuration changes at time k' (Fig. 3.4(b)). Additionally, this work assumes multi-hop data transmission in the communication network, whereby a robot's messages can propagate through one or more intermediate neighbors to reach distant, non-adjacent neighbors. Therefore, multi-hop communication has the advantages of extended coverage and improved connectivity over the standard single-hop communication [9, 33, 117]. The following sections will present decentralized coordination and control approaches, which rely on information exchange by means of communication. For brevity, in the descriptions of the approaches, we assume N robots forming a single communication network to track M targets. When the communication graph is disconnected, the approaches hold without loss of generality for every local network.

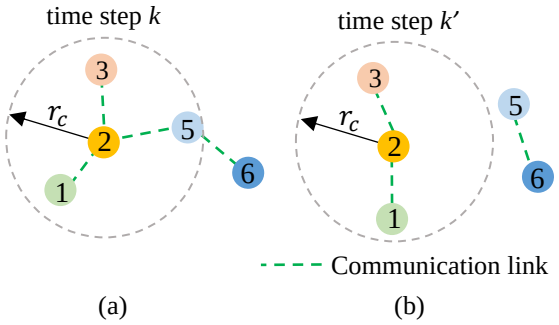


Figure 3.4: An illustrative example of (a) a single connected communication graph (a) and two locally connected communication graphs (b) formed by five nodes (robots) of different configurations.

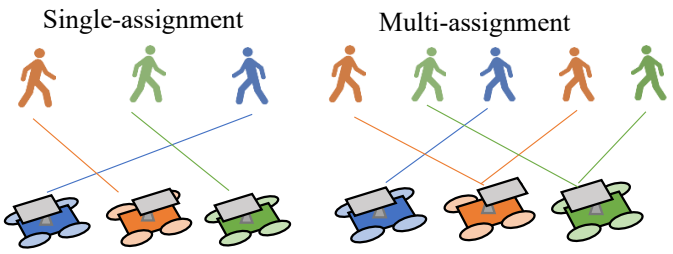


Figure 3.5: Illustration of the single-assignment and multi-assignment problems.

3.4 Decentralized Coordination

The goal of decentralized coordination is to find a valid target assignment, as defined in (3.5), by means of local robot estimation and communication. Mathematically, a tracking utility is associated with every robot-target pair, and the robots seek to find the assignments that maximize the utility of the network (3.6), which amounts to solving the stage I optimization problem (*Problem 1*). In general, target assignment can be divided into two categories, single-assignment and multi-assignment problems. In single-assignment problems, targets are matched to an equal number of robots on a one-to-one basis [118,119,145]. In multi-assignment problems, one or multiple targets are matched to each robot, leading to a more difficult problem where the target assignment space grows exponentially as the target number increases. These two types of assignment are illustrated in Fig.3.5.

Among existing methods for the single-assignment problem, auction-based algorithms put forward a distributed strategy that converges in polynomial time [146]. Auction was used by humans throughout history in market economies and was introduced to solve assignment problems in the late 1970s [15, 36]. In auction-based methods, robots are viewed as bidders that place bids for the most valued targets based on local criteria (e.g., utility functions). Robots can exchange information

with communication neighbors, whereby the targets' prices are updated based on the highest bids [113, 143]. Since robots can increase their bids following some pre-defined protocols to compete for the targets, the auction runs iteratively until the bids converge [98].

Inspired by the auction mechanism, this work proposes two novel decentralized algorithms for the multi-assignment problem, which are referred to as the group-based assignment and bundle-based assignment, respectively. Both algorithms perform an auction that iterates between robots bidding on the most valued targets and robots reaching a consensus on conflict-free assignments. Nevertheless, they differ in the ways of constructing assignment combinations and in the auction protocols being used. The notation shared by both algorithms is introduced here. During auction, robot i iteratively updates a winning bids list $\mathbf{y}_i(k)$ that records the highest bid on every target and a winning agent list $\mathbf{a}_i(k)$ that stores which agent owns each target. For brevity, the discrete-time index k is omitted in the rest of this section. Moreover, the tracking utility function $U_{i,j}(\cdot)$ defined in (3.48) is used as the local criterion for computing bids in auction, with larger values indicating more desired targets.

3.4.1 Group-Based Assignment

The primary idea of group-based assignment is to divide the set of targets ($\mathcal{M}$) into a number of groups equal to the robots' number (N) and, then, assign every target group to one and only one robot by auction. Assuming targets are grouped using the Euclidean distance, the tracking utility gained by robot i from selecting the j^{th} target group is denoted by $U_{i,j}(\cdot)$, which can be obtained by replacing the individual target estimate in (3.48) with the group centroid. However, robot i also

pays a bid $y_{i,\jmath}$ to claim the $\jmath^{th}$ target group. As a result, the net utility associated with a target group is defined as $U_{i,\jmath} - y_{i,\jmath}$, which is to be maximized by each robot $i \in \mathcal{N}$ while avoiding conflicts.

The rest of this subsection presents one auction iteration which consists of a single run of bidding and consensus. Assume the winning bids list and the winning agent list carried by robot i up to the l^{th} iteration are written as $\mathbf{y}_i^{(l)} = \{y_{i,\jmath}^{(l)} \mid \jmath \in \{1, \ldots, N\}\}$ and $\mathbf{a}_i^{(l)} = \{a_{i,\jmath}^{(l)} \mid \jmath \in \{1, \ldots, N\}\}$, respectively. Entering the $(l+1)^{th}$ iteration, robot i seeks to claim the target group $\jmath_i$ that maximizes its net utility

$$\jmath_i = \arg\max_{\jmath} \; U_{i,\jmath} - y_{i,\jmath}^{(l)}, \quad \jmath \in \{1, \ldots, N\} \tag{3.27}$$

However, if the $\jmath_i^{th}$ target group was won by another robot $i', i' \neq i$ in the l^{th} iteration, robot i will need to increase its bid to compete for the target group in the $(l+1)^{th}$ iteration

$$y_{i,\jmath_i}^{(l+1)} = y_{i,\jmath_i}^{(l)} + \delta \tag{3.28}$$

where δ is the largest increment by which the bid can be increased, with the $\jmath_i^{th}$ target group still being the best option for robot i [14, 16, 98]. Therefore, δ is the net utility difference between the current best and the second best target group

$$\delta = (U_{i,\jmath_i} - y_{i,\jmath_i}^{(l)}) - \max_{\jmath' \neq \jmath_i} (U_{i,\jmath'} - y_{i,\jmath'}^{(l)}) \tag{3.29}$$

In practice, a minimum bidding increment of $\epsilon \in \mathbb{R}^+$ is added to δ, so as to the guarantee convergence in auction [16]. By increasing the bid according to (3.29), robot i is recorded as the new winning agent for the $\jmath_i^{th}$ target group, i.e., $a_{i,\jmath_i}^{(l+1)} = i$. In order to reach a consensus, the winning bids list $\mathbf{y}_i^{(l+1)}$ of each robot is broadcast to their neighbours in the network. Then, robot i, after receiving its neighbours'

winning bids list, performs the following updates:

$$y_{i,\jmath}^{(l+1)} = \max_{\imath \in \mathcal{N}} \; y_{i,\jmath}^{(l+1)}, \quad \jmath \in \{1, \ldots, N\} \tag{3.30}$$

$$a_{i,\jmath}^{(l+1)} = \arg\max_{\imath \in \mathcal{N}} \; y_{i,\jmath}^{(l+1)}, \quad \jmath \in \{1, \ldots, N\} \tag{3.31}$$

The above auction and consensus in (3.27-3.31) repeat until the winning agent lists $\mathbf{a}_i, i \in \mathcal{N}$ converge.

The group-based method imposes the assumption that distinct targets in the same group are best represented by the group centroid and can be viewed as a single item in auction. This abstraction leads to the loss of valuable information regarding the contribution of each target to the tracking utility that is to be optimized. This issue will be addressed by the bundle-based assignment in the next subsection.

3.4.2 Bundle-Based Assignment

In the bundle-based assignment, each robot constructs a bundle by sequentially including targets based on the tracking utility of individual targets. This dynamic assignment combination encodes a decreasing order of target significance and is called a *bundle* to be distinguished from a *group* that is a static target combination formed before auction. The proposed bundle-based assignment draws inspiration from the consensus-based bundle algorithm (CBBA) in [30], which allocates a set of tasks that would happen at fixed locations and within limited time windows to several agents. However, task allocation in CBBA cannot be dynamically adjusted. In contrast, the proposed bundle-based assignment can instantaneously adapt to the movements of robots and targets.

The rest of this subsection presents one auction iteration in the bundle-based

method, to highlight the primary differences from the group-based method. Apart from the winning bids list $\mathbf{y}_i$ and winning agent list $\mathbf{a}_i$, robots carry a bundle list $\mathbf{b}_i$ to record the assignment combination. The length of $\mathbf{b}_i$ is an empirically chosen parameter, representing the maximum allowable targets assigned to a robot. Assume that $\mathbf{y}_i^{(l)}$, $\mathbf{a}_i^{(l)}$ and $\mathbf{b}_i^{(l)}$ have been obtained by robot $i, i \in \mathcal{N}$ in the l^{th} iteration. Letting n_i denote the index of the first empty element in $\mathbf{b}_i^{(l)}$, robot i selects the best target that is not yet in the bundle according to the following protocol for the $(l+1)^{th}$ iteration:

$$j_i = \arg\max_j \ \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \setminus \mathbf{b}_i^{(l)} \tag{3.32}$$

where $0 < \beta \leq 1$, and β^{n_i-1} is a factor that hinders a target from being selected by the robot who has already won many targets (a large n_i). Hence, the protocol in (3.32) prevents cases in which a large number of targets are assigned to a small subset of robots. After selecting the j_i^{th} target, $\mathbf{y}_i^{(l+1)}$ is updated by $y_{i,j_i}^{(l+1)} = \max_j \ \beta^{n_i-1} U_{i,j}$, $j \in \mathcal{M} \setminus \mathbf{b}_i^{(l)}$, which is different from the bid update (3.28) in the group-based method. Then, the n^{th} element in the bundle list $\mathbf{b}_i^{(l+1)}$ is determined as follows:

$$b_{i,n_i}^{(l+1)} = \begin{cases} j_i & \text{if} \quad y_{i,j_i}^{(l+1)} > y_{i,j_i}^{(l)} \\ \emptyset & \text{otherwise} \end{cases} \tag{3.33}$$

where $\emptyset$ means no target will be added to $\mathbf{b}_i^{(l+1)}$ because robot i fails to bid higher than the existing winning bid.

Next, each robot broadcasts $\mathbf{y}_i^{(l+1)}$ through multi-hop communication such that the up-to-date bids for all targets are known to the entire network N. The consensus is reached by the i^{th} robot replacing $\mathbf{y}_i^{(l+1)}$ with the highest bid offered by itself or one of its neighbours [65] and releasing the selected targets that are outbid from $\mathbf{b}_i^{(l+1)}$. Finally, the winning agent list $\mathbf{a}_i^{(l+1)}$ is determined based on the updated

$\mathbf{b}_i^{(l+1)}$:

$$a_{i,j}^{(l+1)} = \begin{cases} i & \text{if} \quad j \in \mathbf{b}_i^{(l+1)} \\ a_{i,j}^{(l)} & \text{otherwise} \end{cases}, \ \forall i \in \mathcal{N}, \ \forall j \in \mathcal{M} \tag{3.34}$$

In summary, despite different ways of bidding on the targets, both the group-based and bundle-based methods iteratively perform an auction until the winning agents list $\mathbf{a}_i$ converge. Also, it will be shown in Section 3.6 that both algorithms are polynomial time algorithms. Finally, given the converged $\mathbf{a}_i$, the target assignment solution to the decentralized coordination (*Problem 1*) is obtained as

$$P_i = \{j \ \in \mathcal{M} \mid \mathbf{1}(a_{i,j} = i)\}, \quad \forall i \in \mathcal{N} \tag{3.35}$$

where $\mathbf{1}(\cdot)$ is an indicator function that equals one when the enclosed statement holds true, and zero otherwise.

3.4.3 Performance Analysis of Decentralized Coordination

The group-based method has practical advantages of providing an efficient polynomial time solution to the NP-hard multi-assignment problem, as discussed further in Section 3.6. However, the grouping criterion (Euclidean distance) is inconsistent with the objective function of the multi-assignment problem, which introduces a degree of approximation that is difficult to bound. Therefore, this section focuses on the performance analysis of the bundle-based method that directly solves the multi-assignment problem. Let OPT and ALG be the objective function value of the optimal solution and an approximate solution to the same maximization problem, respectively. The definition of an *α-approximation algorithm* [138] is introduced as follows.

Definition 2 (α-approximation algorithm) : An α-approximation algorithm for a maximization problem is a polynomial time algorithm that produces a solution whose value is within a factor of α of OPT, that is:

$$\alpha \cdot \text{OPT} \leq \text{ALG} \tag{3.36}$$

It follows that $\alpha < 1$, and α is also called the performance guarantee of the approximation algorithm.

Proposition 1 (Bundle-based Assignment Performance Guarantee): Bundle-based assignment is a $\frac{1}{2}$-approximation algorithm.

Proof: The key step of the proof is to show the objective function of the bundle-based assignment satisfies the condition of diminishing marginal gain (DMG). Then, the *Lemma 1* in [30] is adopted to show that the bundle-based assignment is a $\frac{1}{2}$-approximation algorithm with the DMG objective function.

For the optimization problem in (3.22), diminishing marginal gain [13] refers to the phenomenon that each additionally assigned target leads to an ever-smaller increase in the value of the objective function $\sum_{j \in P_i} U_{i,j},\ \forall i \in \mathcal{N}$. Clearly, the incremental value gained by adding target j to the assignment of robot i is $U_{i,j}$. In the proposed bundle-based method, let n_i and m_i denote the n_i^{th} and m_i^{th} entry in the i^{th} robot's bundle list ($\mathbf{b}_i$). In order to show that the objective function satisfies the DMG condition, it suffices to prove the following relation:

$$U_{i,n_i} \geq U_{i,m_i}, \quad \text{if} \quad n_i \leq m_i \tag{3.37}$$

For brevity in notation, the index of the auction iteration (l) is omitted in this proof. According to (3.32)-(3.33), the n_i^{th} element in $\mathbf{b}_i$ represents the index of the target selected by robot i at the n_i^{th} place, which is given by

$$b_{i,n_i} = \arg\max_j \ \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \setminus \{b_{i,1}, \ldots, b_{i,n_i-1}\} \tag{3.38}$$

Assume that target c is added to the m_i^{th} ($n_i < m_i$) entry of $\mathbf{b}_i$, which provides an incremental value of U_{i,m_i} to the objective function. It follows that c was not in the bundle when b_{i,n_i} was selected ($c \in \mathcal{M} \setminus \{b_{i,1}, \ldots, b_{i,n_i-1}\}$) and $c \neq b_{i,n_i}$ since all elements in the bundle are unique. If $U_{i,n_i} < U_{i,m_i}$ while selecting the n_i^{th} element, then

$$c = \arg\max_j \ \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \setminus \{b_{i,1}, \ldots, b_{i,n_i-1}\} \tag{3.39}$$

Comparing (3.38) to (3.39) gives $c = b_{i,n_i}$, which contradicts $c \neq b_{i,n_i}$. Thus, it proves by contradiction that the relation in (3.37) is true, i.e., the objective function of bundle-based assignment satisfies the DMG condition. Then, the following optimality analysis is taken from [20, 30]. It was proven in [30] that the single assignment problem with DMG objective functions can achieve $\frac{1}{2}$-approximation. The multi-assignment problem in (3.22) can be treated as a single assignment by assuming a total number of $N \cdot \sum_{i=1}^{M} M!/i!$ expanded robots such that each robot can only select up to one target from the search space $\mathcal{P}$ [20]. It follows that the proposed bundle-based assignment with a DMG objective function guarantees $\frac{1}{2}$-approximation. However, this performance guarantee is for the worst-case scenarios [30]. The numerical results in Section 3.7.1 will show that the proposed bundle-based assignment provides much better performance in general.

3.5 Decentralized Control

Once the target assignment is obtained by solving the network coordination problem, robots individually determine their control in a decentralized fashion to maximize the network tracking performance (*problem 2*). Recent works have shown that the information gain of the (future) target measurements can be used to select sensor actions which reduce the most uncertainty in the target states [32, 79, 80].

Drawing inspiration from these works, this paper defines the information gain as the one-step expected entropy reduction (EER) in the target states.

3.5.1 Information Gain

Since robots have a bounded FOV, the one-step future measurement is modeled as Bernoulli random finite set (RFS) $Z_{i,j}(k+1)$, which can either contain a single target measurement ($\mathbf{z}_{i,j}(k+1)$) or be an empty set ($\emptyset$) depending on whether a target is detected or not. Therefore, the probability density function of $Z_{i,j}(k+1)$ can be described by

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

where $g(\cdot)$ is a Gaussian distribution derived from (3.3) and p_D describes the target detection probability and is characterized by the Bernoulli probability distribution

$$p_D = \begin{cases} 1 & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \in \mathcal{S}_i(k+1) \\ 0 & \text{if} \quad \hat{\mathbf{x}}_{i,j}(k+1) \notin \mathcal{S}_i(k+1) \end{cases} \tag{3.40}$$

that assumes no missed detections when the future target state is inside the planned robot FOV $\mathcal{S}_i(k+1)$.

Let $\mathbf{\Sigma}_{i,j}(k+1|k)$ and $\mathbf{\Sigma}_{i,j}(k+1|k+1)$ represent the prior and posterior covariance matrix of the target estimate before and after a future measurement $Z_{i,j}(k+1)$ is

made, respectively. The two covariance matrices can be recursively updated by

$$\boldsymbol{\Sigma}_{i,j}(k+1|k) = \mathbf{F}\boldsymbol{\Sigma}_{i,j}(k|k)\mathbf{F}^T + \mathbf{Q} \tag{3.41}$$

$$\boldsymbol{\Sigma}_{i,j}(k+1|k+1)$$
$$= \begin{cases} (\mathbf{I} - \mathbf{K}(k)\mathbf{H})\boldsymbol{\Sigma}_{i,j}(k+1|k) & \text{if } Z_{i,j}(k+1) = \{\mathbf{z}_{i,j}(k+1)\} \\ \boldsymbol{\Sigma}_{i,j}(k+1|k) & \text{if } Z_{i,j}(k+1) = \emptyset \end{cases} \tag{3.42}$$

where $\boldsymbol{\Sigma}_{i,j}(k|k)$ is the posterior covariance matrix at time k. The entropy reduction in the state estimate that would be gained by obtaining $Z_{i,j}(k+1)$ is [79, 85]

$$R_{i,j}(Z_{i,j}(k+1)) = \frac{1}{2}\log \frac{|\boldsymbol{\Sigma}_{i,j}(k|k)|}{|\boldsymbol{\Sigma}_{i,j}(k+1|k+1)|} \tag{3.43}$$

where $|\cdot|$ represents matrix determinant. The information gain or EER can be obtained by taking expectation over the unknown future measurement $Z_{i,j}(k+1)$ [80]:

$$I_{i,j} = \mathbb{E}_{Z_{i,j}(k+1)}[R_{i,j}(Z_{i,j}(k+1))] \tag{3.44}$$

The detailed derivation of $I_{i,j}$ is shown in Appendix A.

3.5.2 Tracking Utility Function

Since this paper considers the non-trivial tracking scenarios where the targets have a more dominant number, some targets cannot be consistently tracked at every time instant. Therefore, a new navigation reward that takes into account the bounded and directional robot FOV geometry is introduced to encourage the exploration of lesser tracked targets. Let $t_j(k)$ denote the cumulative tracking time of target j up to time step k, i.e.,

$$t_j(k) = \sum_{\kappa=1}^{k} \left(\sum_{i=1}^{N} \mathbf{1}(\mathbf{x}_j(\kappa) \in \mathcal{S}_i(\kappa)) \cdot \mathbf{1}(j \in P_i(\kappa)) \right) \tag{3.45}$$

The missed-tracking time can be defined accordingly as $\tau_j(k) = k - t_j(k)$, which serves as a priority indicator with larger values indicating that the target is tracked for lesser time. Letting $\mathbf{p}_{x,y} \in \mathcal{W}$ be the 2D coordinate of an arbitrary point in the workspace, the navigation reward designed to "push" robot i toward the lesser tracked target j is given by

$$J_{i,j} = -\,\tau_j(k) \cdot \oint_{\mathcal{S}_i(k+1)} \|\mathbf{p}_{x,y} - \hat{\mathbf{x}}_{i,j}(k+1)\|\; dxdy \tag{3.46}$$

where the integration over $\mathcal{S}_i(k+1)$ accounts for the geometry of the robot's bounded and directional FOV, and the negative sign ensures consistency with the maximization framework such that higher values of $J_{i,j}$ incentivize exploration more.

Next, letting $\mathcal{B} \subset \mathcal{W}$ denote the obstacles in the workspace, collision avoidance is guaranteed by incurring a cost $\gamma \in \mathbb{R}^+$ on the planned robot states that will collide with either the moving targets or the obstacles:

$$C_{i,j} = \gamma\Big(\mathbf{1}\big(\|\mathbf{s}_i(k+1) - \hat{\mathbf{x}}_{i,j}(k+1)\| \le \epsilon\big) + \mathbf{1}\big(\mathbf{s}_i(k+1) \in \mathcal{B}\big)\Big) \tag{3.47}$$

where $\epsilon \in \mathbb{R}^+$ is a threshold on the Euclidean distance within which collision is considered to occur. Then, the utility function of i^{th} robot tracking the j^{th} target is defined as

$$U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) = I_{i,j} + J_{i,j} - C_{i,j} \tag{3.48}$$

that consists of the EER, navigation reward, and collision penalty. Although the robot control $\mathbf{u}_i(k)$ does not directly appear in (3.48), it determines the planned robot state through the robot motion model (3.1), thus affecting the tracking utility.

Substituting (3.48) to (3.22) gives the objective function of the decentralized control optimization, i.e., $\sum_{j \in P_i^*(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k))$, which is discontinuous, non-convex, and multimodal. The discontinuity is introduced by the collision penalty

($C_{i,j}$) and by the bounded FOV model used in deriving the EER ($I_{i,j}$). The non-convexity is owing to the sum of tracking utility over multiple assigned targets ($|P_i^*(k)| \geq 1$), which also leads to a multimodal function. Characteristics of the objective function is further discussed in Appendix B using a representative example. The above characteristics prevent the use of classical gradient-based methods and sub-gradient methods to solve the decentralized control optimization problem in (3.22). Instead, modern metaheuristic algorithms provide viable solutions because they do not require any special characteristics of the objective functions. In particular, this work employs the Genetic Algorithm (GA) due to its popularity and effectiveness in solving the non-convex and discontinuous optimization problems [2, 21, 38, 55, 150].

In summary, the integration of the two assignment methods (stage I optimization) proposed in Section 3.4 and the control optimization (stage II optimization) in this section gives two novel decentralized approaches for network coordination and control, which are referred to as Group-based Assignment and Control (GBAC) and Bundle-based Assignment and Control (BBAC) hereon in the chapter. An important implementation consideration is that the EER term, $I_{i,j}$, does not contribute to the objective function in the stage I optimization because robot states are assumed temporarily constant when optimizing $P_i(k)$ (*Problem 1*). Also, when computing the objective function for the group-based assignment, the state estimate $\hat{\mathbf{x}}_{i,j}(k)$ in (3.48) is replaced by the group centroid. GBAC and BBAC differ in the coordination strategy used to find the online adaptive target assignments. Nevertheless, they are both fully decentralized approaches running concurrently on every robot to optimize the network tracking performance.

3.6 Computational Complexity Analysis

Since it has been proven in Section. 3.3 that obtaining the optimal solution of network coordination and control problem is NP-hard, this section presents the computational complexity analysis for the proposed algorithms, namely, the GBAC and BBAC. For both methods, the analysis is presented for a network of N robots forming a single communication network to track M targets. When the communication graph is disconnected, the analysis holds without loss of generality for every connected component in this graph. Decoupling the network optimization problem defined in (3.14) into decentralized coordination and control enables the complexity of each stage to be analyzed separately and then, combined to give the overall complexity of the proposed methods.

3.6.1 Complexity Analysis of Group-based Assignment

The complexity of group-based assignment is derived from three primary steps executed sequentially in its implementation. The first step requires $\mathcal{O}(NM)$ time to form a communication network and for each robot in this network to propagate the locally estimated target states to their respective neighbors. This communication ensures that the information used by the robots for further steps is consistent within the network. The second contribution to the algorithm's complexity comes from target grouping, which divides M targets into a number of groups equal to the number of robots (N). This work implements the k-means clustering [57, 83] for target grouping, whose run time is $\mathcal{O}(\kappa_1 NM)$ and κ_1 is the iterations required for convergence in clustering. The last step obtains a valid assignment through iterative auction and incurs the complexity of $\mathcal{O}(N^3 \max_{i,j}(U_{i,j}/\epsilon))$ for the

worst-case scenario, where ϵ is the minimum bid increment introduced in Section 3.4.1. Inspired by [146], the worst-case scenario is constructed to satisfy three conditions: 1) robots form a network of chain structure such that it takes $N-1$ communication rounds to propagate information over the entire network; 2) there exists a conflict on the assignment of every target group for all robots; 3) all robots persistently place minimum bid increments of ϵ to compete for a target group [146] until it is no longer attractive, thus delaying the auction. Because it takes no more than $N \max_{i,j}(U_{i,j}/\epsilon)$ iterations to resolve conflicts on a single assignment among all robots [14], it follows that the worst-case scenario requires no more than $\mathcal{O}(N^3 \max_{i,j}(U_{i,j}/\epsilon))$ time to terminate. Combining the above three terms, the final computational complexity of the group-based assignment can be obtained as

$$\mathcal{O}(NM + \kappa_1 NM + \kappa_2 N^3) \tag{3.49}$$

where κ_2 denotes $\max_{i,j}(U_{i,j}/\epsilon)$ for brevity.

3.6.2 Complexity Analysis of Bundle-based Assignment

Compared to the group-based method, the time complexity of the bundle-based assignment consists of one less term, because the targets need not be grouped before assignment. The first contributing term comes from the construction of the communication network, which takes the same $\mathcal{O}(NM)$ time as in (3.49). Although the running time required by the iterative auction in bundle-based algorithms differs from that in the group-based method due to different bidding schemes (Section 3.4), it is computed by constructing a similar worst-case scenario as described in the above subsection. The only difference is that robots bid on M targets separately without dividing them into N groups ($N \leq M$). Because conflicting

assignments are resolved by each robot releasing the targets that are outbid, it leads to a complexity of N^2M to reach consensus on all targets in the worst-case scenario. Therefore, the overall computational complexity is

$$\mathcal{O}(NM + N^2M) \tag{3.50}$$

Apparently, the run time of the algorithm is dominated by $\mathcal{O}(N^2M)$, which is a polynomial time algorithm in terms of both the network size N and the target population (M). However, it should be noted that, in general, the algorithm converges much earlier than N^2M iterations, because some robots are likely to form smaller local networks and to have a more efficient communication topology than the chain graph assumed in the worst-case scenario.

3.6.3 Complexity Analysis of Control Optimization

The complexity of the control optimization is dominated by the calculation of the utility function in (3.48) and the number of iterations (denoted by N_G) the GA algorithm takes to converge. The time required by each GA iteration is considered negligible and is beyond the scope of this discussion. Within each GA iteration, three sub-objectives are evaluated by each robot, which includes the EER ($I_{i,j}$), the navigation reward ($J_{i,j}$), and the collision penalty ($C_{i,j}$). Computing $I_{i,j}$ takes $\mathcal{O}(M)$ time due to the simplified expression in (3.44). Finding the navigation reward $J_{i,j}$ in has a complexity of $\mathcal{O}(LM)$, which assumes the robot FOV can be discretized into L grids in order to approximate the integration in (3.46) by a finite sum. Next, $C_{i,j}$ in (3.47) penalizes potential collisions between a robot with any targets and obstacles in the environment, thus causing $\mathcal{O}(M + |\mathcal{B}|)$ time complexity. Finally, the time required by the control optimization can be written

as

$$\mathcal{O}(N_G(M + LM + |\mathcal{B}|)) \tag{3.51}$$

By considering a sparse obstacle populated environment, i.e., $|\mathcal{B}| < M$, (3.51) can be simplified as $\mathcal{O}(N_G(L+1)M)$, which is linear in terms of the target population M in the network.

GBAC is composed of the group-based assignment and control optimization in a sequential manner, which leads to an overall complexity of $\mathcal{O}((\kappa_1+1)NM+\kappa_2 N^3+ N_G(L+1)M)$ by omitting the non-dominant term in (3.51). Likewise, BBAC that involves the bundle-based assignment and control optimization generates a total run time of $\mathcal{O}((N + N^2)M + N_G(L + 1)M)$. It can be concluded that the two decentralized network coordination and control methods developed in this chapter achieve polynomial time complexity and, thus, are scalable to large-size networks. Experimental results on the computation complexity of the proposed algorithms will be presented in Section. 3.7.

3.7 Decentralized Network Optimization Experiments And Results

The effectiveness of the decentralized coordination and control approaches presented in the chapter is verified for a variety of tracking scenarios characterized by randomized initial network configurations, different target trajectories, and varying communication ranges for inter-robot communication. In Section 3.7.1, the impact of network coordination is first investigated, followed by a comparative study that involves six types of methods for analyzing the tracking performance in Section 3.7.2. The performance variation as a function of the communication range is also

explored in 3.7.3. Then, physical experiments with a network of UGVs tracking multiple human targets are presented in Section 3.7.4, which demonstrates that the proposed approach can be implemented in real-time for robot network applications.

3.7.1 Simulation Results

The simulation environment includes four mobile robots and six moving targets that are randomly initialized in a bounded 100 m $\times$ 50 m workspace, as shown in Fig.3.6. It is assumed that the initial target states are known to the robot network. Also, targets are denoted by the color of the robot they are assigned to throughout the simulation.

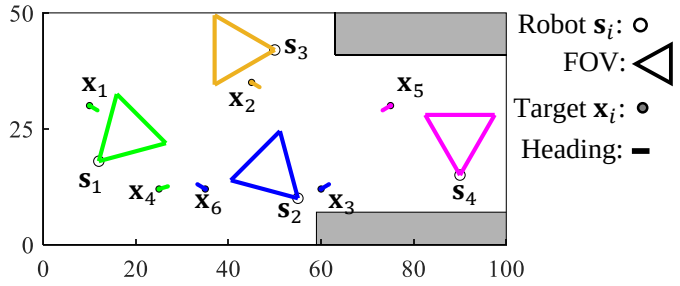


Figure 3.6: An example of the initial network configuration with robots and the assigned targets visualized in the same color.

The impact of network coordination is demonstrated qualitatively by comparing tracking with and without adaptive assignment. Tracking without adaptive assignment is realized by fixing the initial target assignment throughout the simulation and only optimizing the proposed EER-based tracking utility for network control. Therefore, this baseline is referred to as EER control here on in the chapter. On the other hand, the two novel network coordination and control methods, GBAC and BBAC, represent tracking with adaptive target assignment. For a fair

comparison, the three methods are tested using the same initial configuration in Fig.3.6 and the same communication range of 30 m.

The tracking results of the three methods corresponding to the same time instant are highlighted in Fig. 3.7. Attention is given to robot $\mathbf{s}_1$ and $\mathbf{s}_2$, who are initially assigned two targets moving in opposite directions (Fig.3.6), creating a challenging tracking scenario. Due to lack of coordination in EER control, targets $\mathbf{x}_4$ and $\mathbf{x}_6$ are tracked by robot $\mathbf{s}_1$ and $\mathbf{s}_2$ according to the initial assignment, respectively, although the two targets are very close and can be easily tracked by a single robot. Moreover, the tracking of target $\mathbf{x}_3$ is adversely affected since robot $\mathbf{s}_2$ would have to travel a large distance to track it after tracking target $\mathbf{x}_6$. Consequently, the uncertainty in the estimates of target $\mathbf{x}_3$ may increase to the level that the tracking eventually fails. In comparison, robots in GBAC (Fig.3.7(b)) and BBAC (Fig.3.7(c)) form a local network to coordinate the target assignment because they are currently within the communication range of 30m. As described in Section 3.4, coordination in GBAC is achieved by robots selecting target groups formed using the distance criterion. The group of targets $\mathbf{x}_1$, $\mathbf{x}_4$ and $\mathbf{x}_6$ is selected by robot $\mathbf{s}_1$, leading to targets $\mathbf{x}_4$ and $\mathbf{x}_6$ being tracked and freeing robot $\mathbf{s}_2$ to track target $\mathbf{x}_3$. Even more effective coordination is achieved by BBAC in Fig.3.7(c), where the change in assignments of targets $\mathbf{x}_3$ and $\mathbf{x}_4$ is adaptive to the target and robot movements. As a result, the targets are either tracked instantaneously or very close to the assigned robots such that they (targets $\mathbf{x}_3$ and $\mathbf{x}_5$) can be easily tracked in the next time steps.

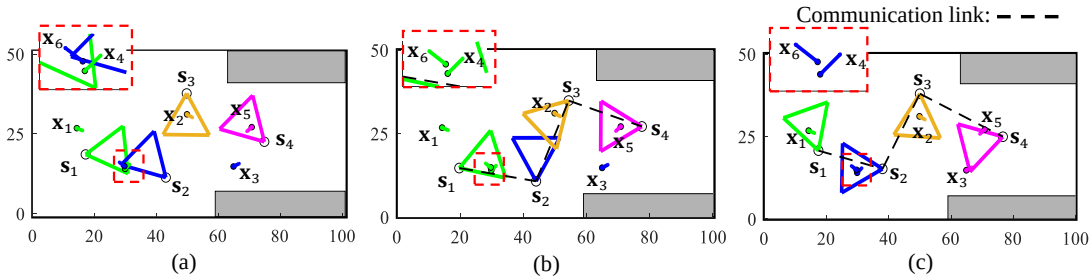


Figure 3.7: Demonstration of tracking without coordination (EER control) (a) and tracking with coordination by GBAC (b) and BBAC (c), and the robots and targets are denoted by their states $\mathbf{s}_i$ and $\mathbf{x}_j$, respectively.

3.7.2 Quantitative Analysis

The quantitative analysis involves the comparison of the proposed GBAC and BBAC methods, to the optimal solution obtained offline, and to three decentralized approximate approaches featuring different network coordination and control strategies:

- Optimal solution: Solving the network optimization problem defined in (3.7)-(3.11) without the decomposition presented in Section 3.3;
- Winner takes all (WTA) approach [122]: The state-of-art decentralized approach to simultaneous control and target assignment for multi-target tracking;
- EER control: The proposed decentralized network optimization with, however, a-*priori* target assignments;
- PD control: Decentralized control by minimizing the Euclidean distance between the initially assigned targets and the center of the robot FOV.

Notice that the comparative study focuses on the non-trivial tracking scenarios

where the targets outnumber the robots and, thus, only a fraction of the total target population can be consistently tracked. Also, the robot network is assumed to operate on a constant communication range of 30m for all of these methods, while the impact of varying communication ranges will be further studied in this section.

The effectiveness of the above methods is demonstrated using two metrics: 1) the average target tracking rate (ATTR) which is the ratio of the average target tracking time to the total simulation time; 2) the average robot travelling distance (ARTD) which is the average distance travelled by all robots in the network. For all methods, 15 tests of varying initial network configurations were performed in simulation. The ATTR and ATRD metrics recorded for each of these tests are presented in Fig.3.8-3.9, whereas the average statistics are summarized in Table. 3.1. The ARTD metric solely does not represent the tracking performance but can be used in combination with ATTR to analyze the tracking performance with respect to the energy consumed in achieving it. Thus, the tracking efficiency defined as the ratio of ATTR to ATRD is presented in Table. 3.1, which evaluates the average target tracking rate per unit distance travelled by the robot.

It can be seen from Table. 3.1 that the proposed GBAC and BBAC methods outperform all the baselines, except the optimal solution, which is obtained offline and, as expected, acts as an upper bound on the performance of the decentralized approaches (Fig. 3.8). The comparison to the WTA method shows the advantages of the full decentralized coordination and control framework proposed in this chapter. The advantages can be attributed to the effective auction-based assignment and to the information driven tracking utility, as opposed to the deterministic utility in WTA which does not account for the uncertainty in target state estimates.

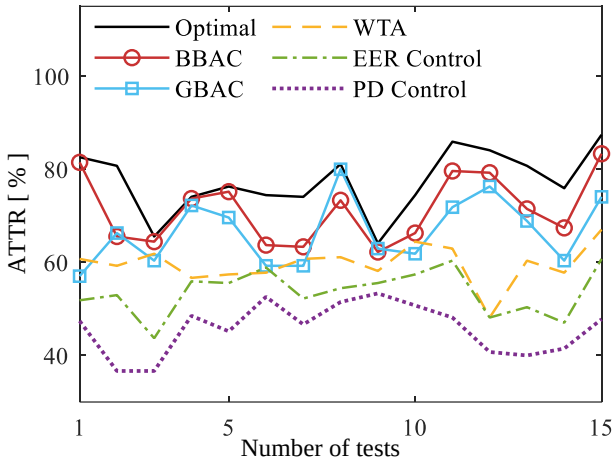


Figure 3.8: The ATTR metric obtained by the six network coordination and control algorithms when the communication range is 30 m.

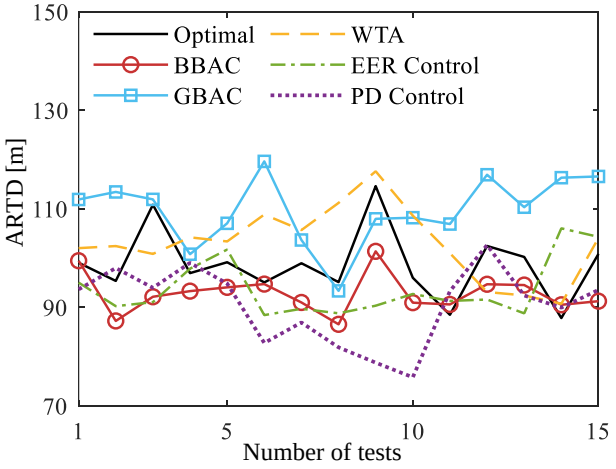


Figure 3.9: The ATRD metric obtained by the six network coordination and control algorithms when the communication range is 30 m.

Furthermore, when compared to EER control and PD control, the superior performance of GBAC and BBAC is owing to the adaptive target assignments achieved by network coordination. Amongst the two proposed methods, BBAC demonstrates a higher ATTR than GBAC on average while maintaining a lower ARTD, which is also reflected in the tracking efficiency metric (Table. 3.1). GBAC falls short because it uses an abstract group representation for distinct targets when determining the target assignments. In comparison, BBAC which achieves more ef-

fective coordination by considering individual targets' contribution to the network tracking utility.

The computational complexity derived in Section. 3.6 and the computation times observed experimentally in simulations are summarized in Table. 3.2. The results are obtained on a Alienware Aurora R13 with 3.19 GHz Intel(R) Core(TM) i9-12900KF and 128 GB installed RAM. The two proposed methods, GBAC and BBAC, demand drastically lesser time in comparison to the optimal solution. When compared to the other baselines, the computation complexity of GBAC and BBAC is higher, but they can still be implemented in real-time while achieving much better performance.

Table 3.1: Comparison of Tracking Performance

Methods	**Assignment**	**Average ATTR**	**Average ARTD (m)**	**Average efficiency**
Optimal	Adaptive	77.43%	98.71	0.78
BBAC	Adaptive	71.36%	92.80	0.77
GBAC	Adaptive	66.69%	109.65	0.61
WTA	Adaptive	59.63%	103.06	0.58
EER control	Pre-defined	53.68%	92.22	0.58
PD control	Pre-defined	45.83%	90.48	0.51

3.7.3 Influence of Communication Range

The tracking performance of the two proposed methods as a function of the communication range r_c is investigated by conducting 15 tests at each of the nine consecutive communication ranges varying from 0m to 80 m with an increment of

Table 3.2: Comparison of Computation Complexity

Methods	**Theoretical complexity**	**Experimental complexity (sec)**
Optimal	NP hard	119.03
BBAC	$\mathcal{O}((N+N^2)M + N_G(L+1)M)$	0.88
GBAC	$\mathcal{O}((\kappa_1+1)NM + \kappa_2 N^3 + N_G(L+1)M)$	0.74
WTA	$\mathcal{O}(NM + NP + M)$	0.44
EER control	$\mathcal{O}(N_G(L+1)M)$	0.56
PD control	$\mathcal{O}(M)$	0.18

10 m. The average ATTR across these tests is plotted against the communication range in Fig.3.10. As the communication range increases, more robots join the same local network to attain consensus on assignments that are beneficial for the network as a whole, improving the overall network tracking performance for both GBAC and BBAC. Interestingly, GBAC and BBAC witness a significant improvement in the average ATTR as the communication range increases to 30m, after which the improvement is less noticeable. This can be attributed to the use of multi-hop communication adopted in this work, which enables the propagation of information to distant non-adjacent neighbors and, thus, improve the network connectivity and reduces the impact of larger communication ranges.

On the other hand, the communication range of 0m simulates an interesting scenario where robots independently complete the tracking task based on their initial knowledge and local estimation, which is also representative of the network tracking performance in event of loss of communication. Therefore, the performance drop from a certain communication range (e.g., $r_c = 30$ m) to the range of

0m demonstrates the adverse impact of communication failures in these systems. However, due to the decentralized nature of the proposed approaches, an average ATTR of approximately 50% is obtained at $r_c = 0$ m, showing the robustness to scenarios of communication failures.

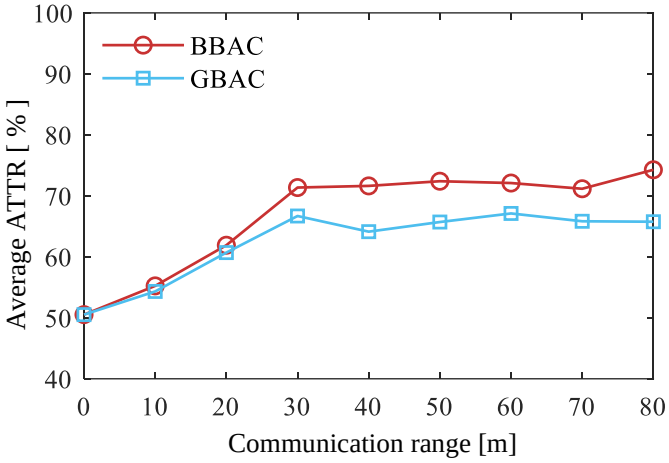


Figure 3.10: The average ATTR against varying communication ranges.

3.7.4 Experimental Results

The primary purpose of conducting experiments with a physical robot network is to ensure that the proposed control and coordination algorithms can run reliably on multiple robots in a decentralized manner, while achieving the network tracking objective in real-time. Two types of experiments are conducted in an indoor lab workspace (Fig. 3.11), which include the testing of the vision-based target detection, classification, and state estimation pipeline using a single target, and the testing of the full decentralized network optimization framework for multi-target tracking. The robots used in the experiments are Husarion ROSBots equipped with a Orbecc Astra RGBD cameras, odometry sensors, and a wifi Antenna, as shown in Fig. 3.12. Although onboard odometry sensors can be easily used for robot

localization, due to the accumulation of localization errors, they are less accurate than the external motion capture system that estimates the robot state independently at every time step. In addition, inter-robot communication is achieved by robots exchanging data over a shared WiFi network. For demonstration purposes, the experiments were recorded by a surveillance camera installed in a pre-selected location that covers the majority of the workspace.

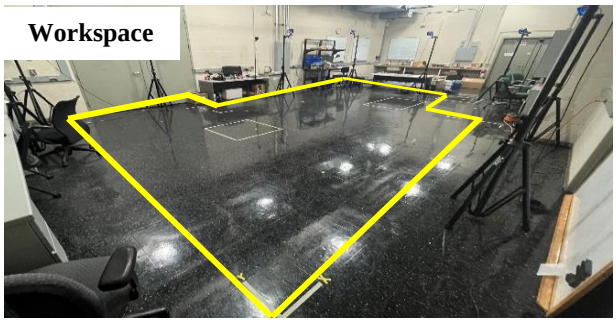


Figure 3.11: The indoor workspace.

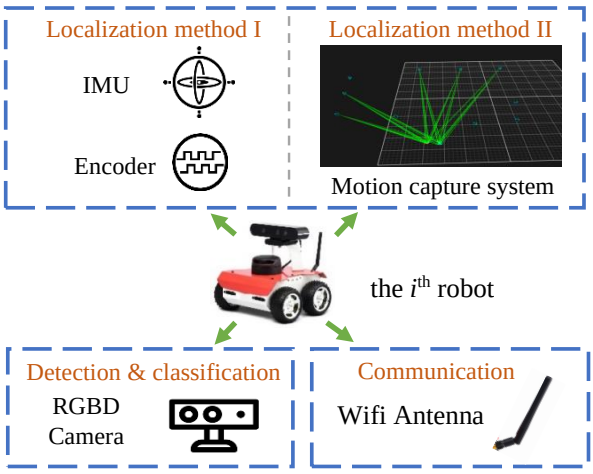


Figure 3.12: The UGV with various sensing capabilities in physical experiments.

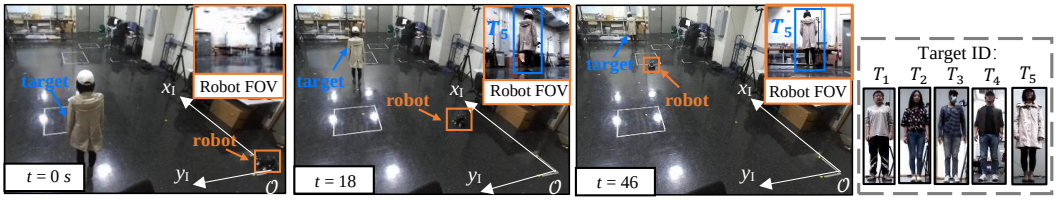


Figure 3.13: Demonstration of vision-based tracking with the robot view superimposed with the recording camera view and the reference images of the targets-of-interest.

Experiment on the Vision-based Online Sensing

The goal of this experiment is to validate the vision-based target detection, classification, and state estimation pipeline (Fig. 3.12) independent of the network coordination and communication. In the experiment, the human target is detected using Mask R-CNN implemented on the robot, which outputs bound box approximations of the target in the image frame, as shown in the robot view in Fig. 3.13. Then, the ReID Net that is also implemented on the robot recognizes the target ID based on the bounding box approximations. A sequence of frames in one of the experimental trials are shown in Fig. 3.13, where the initial target position and the reference images of the targets-of-interest were provided to the robot a-*priori*. Although the target is not inside the robot FOV at $t = 0$ s, the robot successfully tracks the target based on the predicted target dynamics at $t = 18$ s and correctly recognizes the target ID. Throughout the run, the robot continuously measures the target states by fusing the RGB data, depth data and robot localization according to (3.23)-(3.25). The robot path is planned by optimizing the tracking utility so as to maintain the target inside its FOV, which can be seen in the snapshot at $t = 46$ s. The target trajectory and the planned robot path are shown in Fig.3.14.

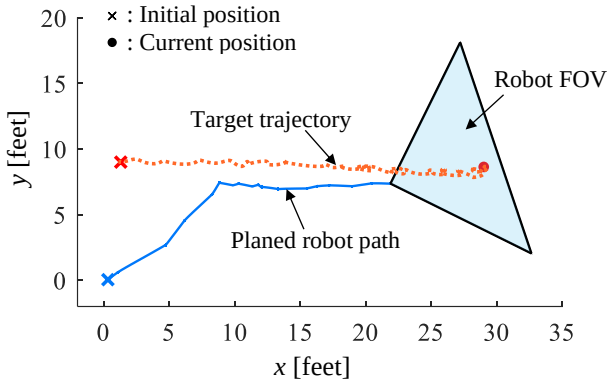


Figure 3.14: Planned robot path for target tracking corresponding to Fig. 3.13.

Experiment on the Network Coordination and Control

The second type of experiments aim to test the holistic framework of network coordination and control, with online sensing and communication as two fundamental components included in the tests. Similar to the above experiment, the initial target positions and the reference images of the targets-of-interest were provided to the robot network a-*priori*. Thus, the experiments simulate a real-life tracking application in which some preliminary location and visual cues of the targets-of-interest are provided to the robot network tasked with tracking them. The ability of the network to perform real-time tracking was successfully validated in 20 trials. One of the experiments is demonstrated as a case study, which features three targets walking in opposite directions. This case study creates a challenging scenario since spatially approximate targets that pass each other may easily confuse the robots in terms of target classification and assignment, and may adversely affect the tracking performance. Both the EER control and the proposed methods are tested on this scenario, with the EER control demonstrating the real-time tracking ability independent of adaptive target assignment and, in comparison, the proposed methods showing the influence of network coordination in real-world experiments.

The tracking results of the EER control are demonstrated by a sequence of images from the view of the recording camera, as shown in Fig.3.15, where targets are denoted by the color of the robot they are assigned to. It can be seen that both robots successfully and consistently track the initially assigned targets ($t = 0s$) throughout the experiment ($t = 23s$ and $t = 48s$), without being confused by the other targets in the network. Because the targets are assumed to not exit the workspace, they stop when reaching the boundary of the workspace ($t = 48s$). The trajectory of the targets and the optimized robot paths are shown in Fig.3.16.

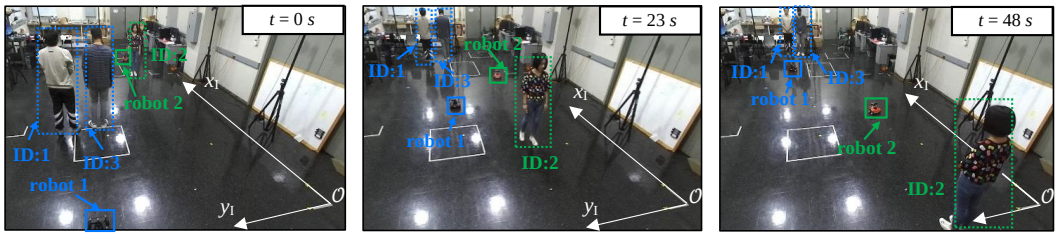


Figure 3.15: Demonstration of a robot network tracking three moving targets with the view of the recording camera.

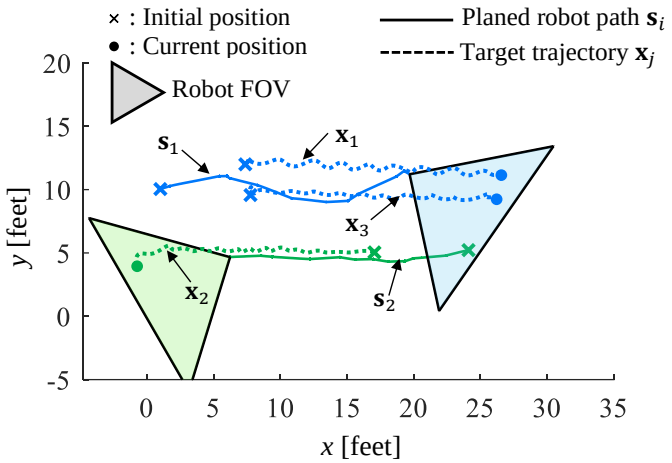


Figure 3.16: The optimized robot path ($\mathbf{s}_i$) for tracking the initially assigned targets ($\mathbf{x}_j$) without adaptive target assignment.

In contrast, experiments using the proposed methods witness changes in the target assignment given the same initialization in Fig. 3.15 ($t = 0$ s), demonstrating

that the assignment is able to dynamically adapt to robot and target movements by network coordination. One of the tracking results obtained by implementing GBAC is shown in Fig. 3.17-3.18, where the change of target assignment can be observed by comparing the optimized robot paths (Fig. 3.18) to the results in EER control (Fig. 3.16). Although targets 1 and 3 are initially assigned to robot 1, they move toward the direction that approaches robot 2 and, therefore, are automatically re-assigned to robot 2 when the change of assignment brings improvement on the network tracking utility. For the same reason, target 2 is re-assigned to robot 1 despite its initial assignment to robot 2. Moreover, the change of color in the target trajectories in Fig. 3.18 indicates the timing when the swapping of target assignment occurs. It can also be seen that the robots' control are optimized to track the newly selected targets, leading to a sudden change in the planned path of both robots. Additionally, no conflict exists in the target assignment throughout the experiment owing to the inter-robot communication that allows the network to reach consensus on a valid target assignment. When implementing the BBAC method in this case study, the experiment results show similar adaptive assignment to the above results obtained by GBAC and, thus, are omitted for brevity.

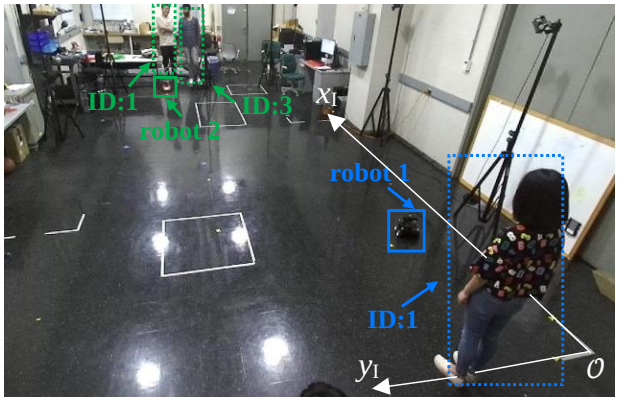


Figure 3.17: Final network configuration for tracking with adaptive assignment when the initialization is the same as that depicted in Fig. 3.15 at $t = 0s$.

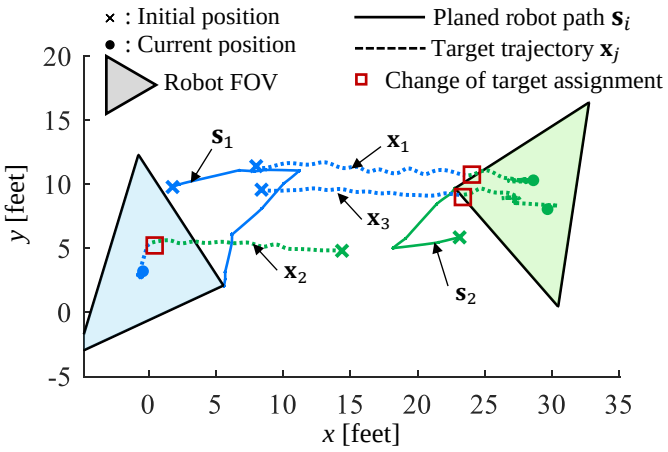


Figure 3.18: The optimized robot path ($\mathbf{s}_i$) for tracking the initially assigned targets ($\mathbf{x}_j$) with adaptive target assignment.

3.8 Conclusion

This chapter presents a novel decentralized optimization framework that integrates online sensing, communication, coordination, and control of multi-robot networks in applications that require tracking of multiple dynamic targets, such as surveillance and security. Because the optimization problem is NP-hard, two approximate approaches, referred to as GBAC and BBAC, are proposed to decompose the optimization into two stages, where target assignment and robot control are determined in a sequential manner. Both GBAC and BBAC rely on the auction mechanism which leverages local communication to achieve conflict-free assignments with, however, different strategies to construct assignment combinations during the auction. For both methods, a novel utility function is locally optimized by each robot to determine the best control in real-time to track the assigned targets. Simulation results show that the performance of the proposed approaches is very close to that of the optimal solution and is better than the other decentralized methods. Moreover, demonstrations of a physical multi-robot network tracking human targets show the applicability in real-world applications.

CHAPTER 4

MIXED HUMAN-ROBOT TEAMS FOR COLLABORATIVE MULTI-TARGET TRACKING

4.1 Introduction

Building on the theory developed in Chapter 3 for robot network optimization, this chapter introduces human partners to form a mixed team with the robotic agents for collaborative perception. Many emerging robotics applications, ranging from nursing homes to urban search and rescue [62, 77, 101], require robots to partner with humans to achieve shared goals. Compared to homogeneous teams of robots, mixed human-robot teams (MHRT) can potentially improve efficiency and robustness by leveraging complementary skills such as human field experience and domain knowledge [35], and robot data processing and integrated sensor modalities. Multi-target tracking, in particular, provides an interesting testbed for studying human-robot cooperation because humans and robots can obtain complementary information about dynamic targets. For instance, although characterized by directional and bounded FOV, mobile robots can track dangerous targets at close distance to gain views with intricate features. In contrast, human operators possess better situational awareness and interpretation of complex mission objectives but have difficulty simultaneously observing many dynamic targets. MHRT cooperation can both harness the strengths and mitigate the deficiencies of different members [100], while maintaining humans in the loop.

To date, many computer vision-based algorithms have been developed to extract visual features for tracking targets using stationary cameras [60, 90, 92]. Other studies have investigated multi-target tracking by controlling the pan and

tilt angle of a single camera, in order to minimize the uncertainty in target estimation [80, 136]. Recently, multi-robot multi-target tracking was explored using robotic platforms, such as unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), that have various sensing capabilities. Range and bearing measurements were used in [46] to track multiple moving targets under the assumption that the number of targets does not exceed the number of robots. Simulated radar-like measurements of quadrotors were used in [79] to develop a game-theoretic approach for planning quadrotor motions to track a group of ground targets from a pre-specified height. Due to the increased availability of high-resolution cameras, vision-based tracking is gaining popularity in a number of robotic applications. In [62], a team of camera-equipped UAVs were deployed by first detecting targets based on color histograms, and, then, planning their paths to track the detected targets. In [134], a group of mobile cameras selects their tracking targets according to a criterion that accounts for the pre-specified target priority, viewing quality, and energy consumption.

This chapter proposes a new approach to human-robot collaboration that enables the maximization of the cumulative tracking time in real time when the targets outnumber the tracking agents. Cooperation entails a two-way message-exchange mechanism and distributed robot control that is a function of human actions. A new tracking utility function is proposed for the local estimation of the MHRT global tracking performance, which accounts for the robot FOV geometry, kinematic constraints, target prediction, obstacle map, and human input. The effectiveness of the proposed approach is demonstrated in simulation involving a human-robot team with one human operator collaborating with four robot agents to track six dynamic targets. Additionally, physical experiments were conducted to validate the applicability of the proposed approach in real-world applications.

4.2 Problem Formulation and Assumptions

This chapter considers the problem of tracking multiple moving targets by means of human-robot collaboration. The problem is relevant to many security and surveillance applications involving UGVs and UAVs teaming with human operators to actively observe a set of targets in a large region of interest. Let $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$ represent a team of mobile robots operating in a closed and bounded two-dimensional (2D) workspace $\mathcal{W} = [0, L_x] \times [0, L_y]$, where $L_x, L_y \in \mathbb{R}^+$ represent the length and width of the workspace and $\mathcal{N} = \{1, \ldots, N\}$ is the index set of the mobile robots. Let $\mathcal{F}_\mathcal{W}$ denote the inertial frame, with origin $\mathcal{O}_\mathcal{W}$, embedded in $\mathcal{W}$ such that the xy-plane aligns with the ground plane. In addition, a moving Cartesian frame, $\mathcal{F}_{\mathcal{A}_n}$, is embedded in each robot, where the origin $\mathcal{O}_{\mathcal{A}_n}$ is located at the principal point of the robot's camera, such that $\mathcal{F}_{\mathcal{A}_n}$ aligns with the robot camera frame [56]. Without loss of generality, the state vector of the n^{th} robot $\mathcal{R}_n$ is represented by $\mathbf{s}_n = [\mathbf{p}_n^T \quad \theta_n]^T$, where $\mathbf{p}_n = [x_n \quad y_n]^T$ and θ_n are the 2D coordinates and orientation with respect to $\mathcal{F}_\mathcal{W}$ [88], as shown in Fig. 4.1. The state vector $\mathbf{s}_n$ satisfies the robot dynamics equation which, in this chapter, is given by the unicycle model [45, 88]

$$\dot{\mathbf{s}}_n = \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{\theta}_n \end{bmatrix} = \begin{bmatrix} v_n \cos\theta_n \\ v_n \sin\theta_n \\ \omega_n \end{bmatrix} = \mathbf{f}(\mathbf{s}_n, \mathbf{u}_n), \quad \forall n \in \mathcal{N} \tag{4.1}$$

where the robot control vector, denoted by $\mathbf{u}_n = [v_n \quad w_n]^T \in \mathbb{R}^2$, consists of linear velocity v_n and angular velocity w_n. Assuming a constant sampling interval Δt, the robot state and control vector at any discrete time k can be written as $\mathbf{s}_n(k)$ and $\mathbf{u}_n(k)$, respectively.

Consider that the MHRT is interested in tracking a set of human targets in-

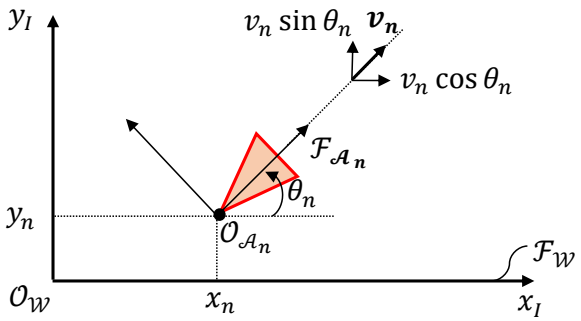


Figure 4.1: Definition of the state of a robot.

dexed by $\mathcal{M} = \{1, \ldots, M\}$. This work tackles the challenging tracking problem where the number of targets is no less than the size of the robot team, i.e., $M \geq N$. The state of the i^{th} target with respect to the inertial frame is represented by $\mathbf{x}_i = [x_i \quad y_i \quad v_{x,i} \quad v_{y,i}]^T \in \mathbb{R}^{4\times 1}, i \in \mathcal{M}$. Then, the state vector of all targets can be written as $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_M^T]^T \in \mathbb{R}^{4M\times 1}$. Assume that target velocity is constant and is subjected to process noise $\mathbf{w}(k)$ which is assumed to be additive, Gaussian, and white. The target motion model, also known as the target prediction model, can be described as

$$\mathbf{x}_i(k) = \mathbf{F}\mathbf{x}_i(k-1) + \mathbf{w}(k), \quad \mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{4.2}$$

where $\mathbf{F} \in \mathbb{R}^{4\times 4}$ is the state transition matrix for the constant-velocity model [6] and $\mathbf{Q}$ is the prediction covariance matrix. Let $P = \{P_1, \ldots, P_N\}$ be the target assignment to the robot team, such that $P_n \cap P_{n'} = \emptyset,\ n \neq n'$ and $\bigcup_{n=1}^{N} P_n = P$. For simplicity, P is assumed to be known a *priori*, which will not change throughout the mission. However, future work will investigate adaptive target assignment which incorporates robot and target dynamics. Let $\mathbf{z}_{n,i}(k)$ represent the measurements of the i^{th} target by the n^{th} robot, which can be obtained once the target is inside the robot's FOV denoted by $\mathcal{S}_n$. Unlike conventional sensors that measure the range and bearing to targets [46, 79], a vision-based measurement model relying on images streamed by the robot camera is derived in Section 4.3.1, which directly

measures target positions in the inertial frame $\mathcal{F}_{\mathcal{W}}$ as follows:

$$\mathbf{z}_{n,i}(k) = \begin{cases} H\mathbf{x}_i(k) + \mathbf{v}(k) & \text{if} \quad \mathbf{x}_i(k) \in \mathcal{S}_v(k) \\ \emptyset & \text{if} \quad \mathbf{x}_i(k) \notin \mathcal{S}_n(k) \end{cases} \tag{4.3}$$

where $\mathbf{H} = [\mathbf{I}_2 \quad \mathbf{0}_2]$ and $\mathbf{v}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is Gaussian noise.

Consider that robots cooperate with human operators in a tracking mission with the goal of maximizing the cumulative tracking time of all targets. Cooperation in the mixed team is realized by establishing a two-way message exchanging mechanism to share mission-relevant data. Both the robot message and human message, represented by $\mathbf{q}_n, n \in \mathcal{N}$ and $\mathbf{m}$ respectively, will be introduced in Section 4.3.2. The tracking time of the i^{th} target ($i \in \mathcal{P}_n$) up to time k is defined as

$$t_i(k) = \sum_{\tau=1}^{k} \mathbf{1}(\mathbf{x}_i(\tau) \in \mathcal{S}_n(\tau)) \tag{4.4}$$

Then, the objective of the MHRT can be written as a global utility function U_g

$$U_g = \sum_{i=1}^{M} t_i(T_f) \tag{4.5}$$

that is to be maximized with respect to the robot control for achieving optimal tracking performance by task end time T_f.

Optimizing a global utility in a multi-robot team demands accumulating and processing information from all agents in the team [5,9], which leads to high communication and computational load. This work presents a distributed approach where the global objective is achieved by individual robots concurrently and efficiently optimizing a local utility function. The local utility function, denoted by $U_{n,i}(\cdot)$, is a local measure of tracking performance that aligns with U_g [5] and is defined as a function of the robot's state $\mathbf{s}_n(k)$, the target measurement $\mathbf{z}_{n,i}(k)$, and the human message $\mathbf{m}$ in Section 4.3.3. Also, let $\mathcal{B}$ denote an obstacle map of

the workspace, and let $C(\mathcal{B})$ be the penalty function for obstacle collision. Then, the distributed cooperative tracking problem can be summarized as follows:

Problem 1 (Distributed Cooperative Tracking): Given the models for robot dynamics, target prediction and measurement (3.1-4.3), the obstacle map $\mathcal{B}$, and the target assignment P, estimate the robot states $\mathbf{s}_n, n \in \mathcal{R}$ and the assigned target states $\mathbf{z}_{n,i}, i \in P_n$ online, and determine the robot control $\mathbf{u}_n$ collaboratively with the human operator by solving the distributed optimization problem in (4.6) on each robot

$$\begin{aligned} \max_{\mathbf{u}_n(k)} \quad & \sum_{i \in P_n} U_{n,i}(\mathbf{z}_{n,i}(k), \mathbf{s}_n(k), \mathbf{m}) - C(\mathcal{B}) \\ \text{s.t.} \quad & \mathbf{s}_n(k+1) = \mathbf{f}(\mathbf{s}_n(k), \mathbf{u}_n(k)), \\ & \mathbf{a}_1 \leq \mathbf{s}_n(k) \leq \mathbf{a}_2, \\ & \mathbf{b}_1 \leq \mathbf{u}_n(k) \leq \mathbf{b}_2, \\ & \forall n \in \mathcal{N} \end{aligned} \tag{4.6}$$

where $\leq$ denotes elementwise inequalities, $\mathbf{a}_1 = [0 \quad 0 \quad 0]^T$, $\mathbf{a}_2 = [L_x \quad L_y \quad 2\pi]^T$, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are physical bounds on control input.

4.3 Cooperative Target Tracking

This section proposes a novel approach for multi-target tracking that achieves human-robot cooperation, online sensing, and distributed control optimization in a single framework, as shown in Fig.4.2. Because the robot states $\mathbf{s}_n$ can be easily estimated using onboard or external motion sensors, this section focuses on introducing the vision-based target state estimation, the cooperation mechanism in the MHRT, and the tracking utility function for distributed optimization.

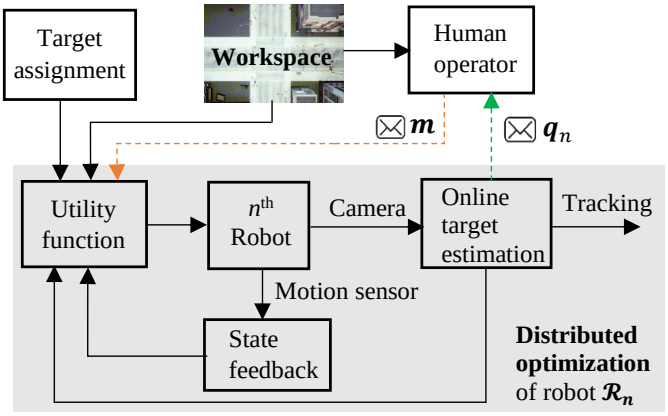


Figure 4.2: Cooperative tracking framework.

4.3.1 Online Target State Estimation

Existing works on vision-based target estimation often use a monocular camera to obtain measurements in the image frame or virtual image plane [80, 136], resulting in complex non-linear sensor measurement models. In contrast, this work integrates the CNN-based target detection with the ray-tracing method (Fig.4.3) to directly estimate the target position in the inertial frame $\mathcal{F}_{\mathcal{W}}$ using RGBD images streamed from the onboard robot camera, as shown in Fig.4.4.

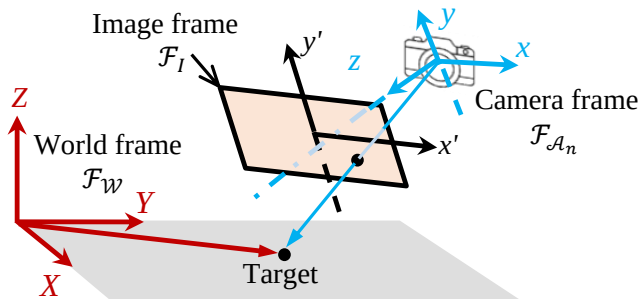


Figure 4.3: Illustration of the ray tracing method.

For brevity, the discrete-time index k is omitted in this subsection. Let $\mathcal{F}_{\mathcal{I}}$ denote the image reference frame, and let $\mathbf{x}_i|_{\text{image}}$ denote the $2D$ position of the i^{th} target with respect to $\mathcal{F}_{\mathcal{I}}$, which can be approximated by the image coordinate at

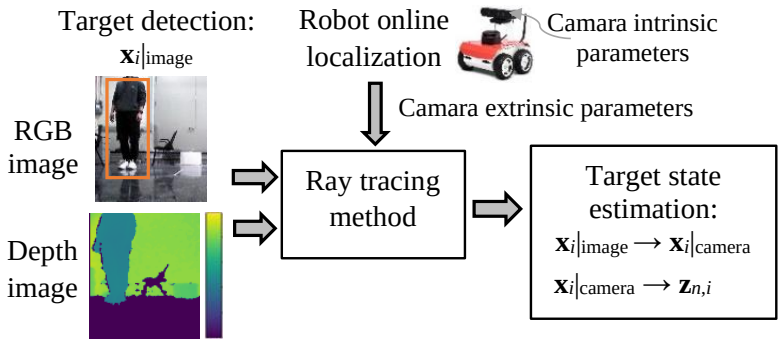


Figure 4.4: Vision-based online target estimation.

the center of the target's bounding box obtained from detection algorithms such as MASK-RCNN [59]. Given $\mathbf{x}_i|_{\text{image}}$, the target depth, d_i, can be estimated by the corresponding pixel value in the depth image. Then, the target position with respect to the camera frame $\mathcal{F}_{\mathcal{A}_n}$ is given by

$$\mathbf{x}_i|_{\text{camera}} = d_i \mathbf{K}^{-1} [\mathbf{x}_i|_{\text{image}} \quad 1]^T \tag{4.7}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix. The target measurement $\mathbf{z}_{n,i}$ in the inertial frame $\mathcal{F}_{\mathcal{W}}$ is obtained by transforming $\mathbf{x}_i|_{\text{camera}}$ from $\mathcal{F}_{\mathcal{A}_n}$ to $\mathcal{F}_{\mathcal{W}}$

$$\mathbf{z}_{n,i} = \mathbf{R}_n \mathbf{x}_i|_{\text{camera}}^T + \mathbf{r}_n^T \tag{4.8}$$

where $\mathbf{R}_n$ and $\mathbf{r}_n$ are camera extrinsic parameters that can be estimated from the robot state vector as follows:

$$\mathbf{R}_n = \begin{bmatrix} cos\theta_n & -sin\theta_n & 0 \\ sin\theta_n & cos\theta_n & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{r}_n = [x_n \quad y_n \quad 0]^T \tag{4.9}$$

Compared to the true target state $\mathbf{x}_i$, the measurement $\mathbf{z}_{n,i}$ does not contain the velocity terms and is assumed to be subjected to white, additive Gaussian noise $\mathbf{v}$, which gives

$$\mathbf{z}_{n,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_i + \mathbf{v} \tag{4.10}$$

and this completes the measurement model stated in (4.3).

4.3.2 Human Robot Cooperation Strategy

In order to cooperate in MHRT, two-way communication [1] is established for message exchanging through a shared network between the human operator and the robots, which is assumed to have negligible delays. Because of the uncertainty in target dynamics, the longer a target is not tracked, the more inaccurate is a robot's prediction and, thus, the less likely will the robot recover that target. Therefore, robot $\mathcal{R}_n$ actively broadcasts *query* messages to the human operator, requesting information of its least tracked target, whose index can be expressed as

$$\imath_n = \arg\max_{i \in P_n} \ (k - t_i) \tag{4.11}$$

The missed-tracking time of the least tracked target is $\tau_n = \max_{i \in P_n} \ (k - t_i)$, which is a priority indicator with larger values representing higher priorities for the operator to provide expert information about the target. The *query* message sent by $\mathcal{R}_n$ is defined as $\mathbf{q}_n \triangleq [\imath_n \quad \tau_n \quad n]^T$, in which the robot index n serves as a message identifier and the priority indicator τ_n guides the operator to make more informed decisions that improves the robot team's tracking time.

In response to robot queries, human operators send *update* messages to selectively update the states of the least tracked target among all queried targets. The *update* messages can be sent asynchronously, letting the operator decide when to update the robot team without adversely affecting the tracking task. Moreover, humans excel in discovering unexpected changes in the environment, such as the appearance of new targets (also called intruders). Once an intruder is recognized, human operators can designate robots to track it through the *intruder* message.

These two types of messages are distinguished by a binary variable $l \in \{0, 1\}$, whose value is specified by the operator such that $l = 0$ represents an *update* message and $l = 1$ means an *intruder* message. Then, the human message is compactly encoded as $\mathbf{m} \triangleq \begin{bmatrix} l & \hat{n} & \jmath & \boldsymbol{\chi}_\jmath^T \end{bmatrix}^T$, where $\hat{n}$ is the index of the robot that will receive the message, $\jmath$ is the index of the human selected target, and $\boldsymbol{\chi}_\jmath \in \mathbb{R}^2$ is the human observed target position.

The two-way message-exchange scheme is illustrated in Fig. 4.5. A user interface is developed for operators to communicate messages to the robot specified by the operator (refer Section 3.7.4). Then, upon receiving an *update* message, the robot updates its online estimation of the target states, $\hat{\mathbf{x}}_{n,i}(k)$, as follows:

$$\hat{\mathbf{x}}_{n,i}(k) = \begin{cases} \boldsymbol{\chi}_\jmath & \text{if } \hat{n} = n,\ l = 0,\ \jmath = i \\ \mathbf{z}_{n,i}(k) & \text{if } \hat{n} \neq n,\ \mathbf{x}_{n,i} \in \mathcal{S}_n(k) \\ \mathbf{F}\mathbf{x}_{n,i}(k-1) & \text{otherwise} \end{cases} \tag{4.12}$$

In comparison, robots that receive an *intruder* message will append the new target index to its original assignment, namely, $P_n = P_n \cup \{\jmath\}$, with the following initialization:

$$\hat{\mathbf{x}}_{n,|P_n|}(k) = \boldsymbol{\chi}_\jmath, \quad \text{if} \quad \hat{n} = n,\ l = 1 \tag{4.13}$$

The intruder will then be actively tracked by robot $\mathcal{R}_n$.

4.3.3 Tracking Utility Function

The distributed cooperative tracking task in (4.6) is achieved by robots concurrently maximizing a local utility function. The design of local utility should be aligned with the global utility (4.5), such that distributed optimization by all

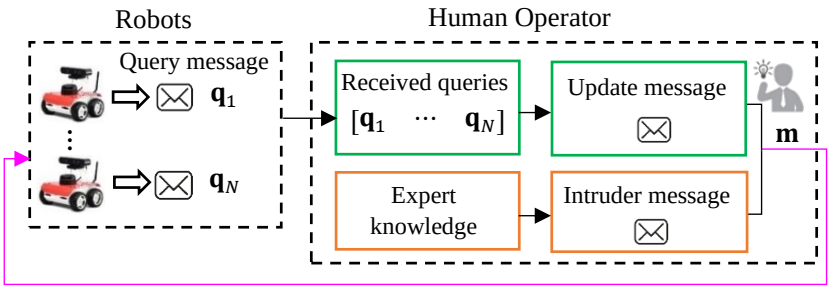


Figure 4.5: Message exchange within the human-robot team.

robots amounts to improving the MHRT goal [5], which, in this work, is to maximize the cumulative tracking time of all targets. Also, because this work considers the challenging tracking scenarios where the targets outnumber the robots, all targets cannot be simultaneously and consistently tracked. Thus, a novel local utility function is designed as a combination of a tracking reward and a navigation reward to encapsulate the trade-off between the instantaneous improvement in tracking time and the exploration of missed targets.

The tracking reward is a function of the planned robot FOV $\mathcal{S}_n(k+1)$ and the predicted target states $\hat{\mathbf{x}}_{n,i}(k+1)$ obtained by applying the prediction model (4.2) on the target state estimates $\hat{\mathbf{x}}_{n,i}(k)$. The tracking reward is defined as

$$D_{n,i} = \gamma \cdot \mathbf{1}(\hat{\mathbf{x}}_{n,i}(k+1) \in \mathcal{S}_n(k+1)) \tag{4.14}$$

where $\gamma \in \mathbb{R}^+$ is an empirically-chosen reward generated if the planned robot FOV tracks a target at the next future time step. Therefore, the tracking reward encourages an instantaneous improvement in tracking time.

Next, assume the workspace $\mathcal{W}$ is tessellated into equally sized and disjoint 2D cells [74] represented by $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_q\}$. Letting $\mathbf{y}_\ell \in \mathbb{R}^2$ denote the 2D centroid coordinate of the ℓ^{th} cell, an attractive potential for the i^{th} target is defined in terms of the Euclidean distance between the centroid of each cell and the target's

predicted position

$$M_i(\ell) = \eta(t_i) \cdot \|\mathbf{y}_\ell - \hat{\mathbf{x}}_{n,i}(k+1)\|, \quad \forall \ell \tag{4.15}$$

where $\eta(t_i)$ is a factor that is inversely proportional to the tracking time t_i, hence incentivizing exploration of the lesser tracked targets. Then, the navigation reward for robot $\mathcal{R}_n$ is defined as the cumulative negative attractive potential of the cells covered by the planned FOV $\mathcal{S}_n(k+1)$

$$M_{n,i} = -\sum_{\ell=1}^{q} M_i(\ell) \cdot \mathbf{1}(\mathcal{C}_\ell \in \mathcal{S}_n(k+1)) \tag{4.16}$$

where the negative sign ensures consistency with the maximization framework such that larger rewards pull the robot closer to the targets. Then, the local utility is defined as

$$U_{n,i}(\mathbf{z}_{n,i}, \mathbf{s}_n, \mathbf{m}) = M_{n,i} + D_{n,i} \tag{4.17}$$

Notice that the target observation $\mathbf{z}_{n,i}$, robot state $\mathbf{s}_n$, and human message $\mathbf{m}$ are implicitly integrated to the utility function through $\hat{\mathbf{x}}_{n,i}(k)$ and $\mathcal{S}_n(k+1)$ in (4.14-4.16).

Finally, collision with static obstacles $\mathcal{B} \in \mathcal{W}$ is avoided by incurring a penalty $\zeta \in \mathbb{R}^+$ on the planned robot state $\mathbf{s}_n(k+1)$ that will collide with the obstacles

$$C(\mathcal{B}) = \zeta \cdot \mathbf{1}(\mathbf{s}_n(k+1) \in \mathcal{B}) \tag{4.18}$$

Substituting (4.17-4.18) into (4.6) gives the objective function for distributed optimization that is solved locally and concurrently by each robot.

4.4 Cooperative Tracking Results

The proposed cooperative tracking approach is tested both in simulations and in real-world experiments. For comparison purposes, this work also implements a

weak-cooperative tracking algorithm and a non-cooperative tracking algorithm in simulation. Subsequently, physical experiments are conducted with a real human-robot team to demonstrate the approach's effectiveness in real-world applications.

4.4.1 Simulation Results

The simulated MHRT includes four mobile robots randomly initialized in a $100m \times 50m$ workspace and a human operator that observes the workspace through a surveillance camera, with the goal of tracking six targets that move freely in the same workspace. One of the testing instances is demonstrated in Fig.4.6-4.7, where targets are denoted by the color of the robot they are assigned to and the human view is shown by the pink polygon. Although the initialization in Fig.4.6 creates a challenging tracking scenario with no target being inside the robots' FOV, the mixed team is able to cooperatively track all targets at time step $k = 17$ (Fig.4.7).

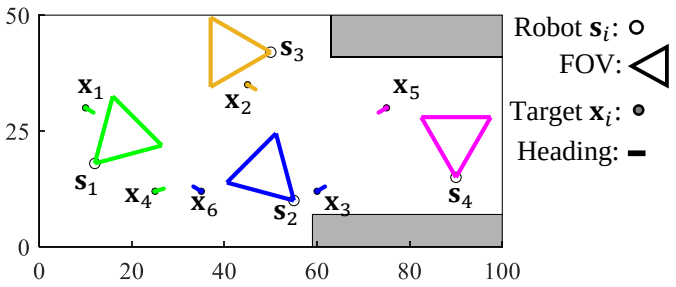


Figure 4.6: The initial configuration of a testing instance.

Another study is performed to compare the proposed *Cooperative tracking* approach against the following two baselines: 1) *Non-cooperative tracking*, comprised of robots that independently optimize the local tracking utility function without any interaction with the human operator; 2) *Weak cooperative tracking*, comprised of robots that send query messages but without the priority indicator (defined in

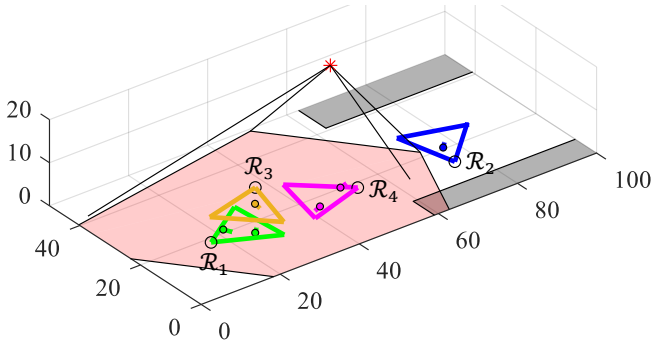


Figure 4.7: Illustration of the MHRT tracking all targets at time step $k = 17$.

Section 4.3.2) to inform human operators of the least tracked target. The comparative study involves 20 testing instances for the above three strategies, with the results quantitatively evaluated using two temporal metrics: the average target tracking rate (ATTR) which is the ratio of the average target tracking time to the total simulation time, and the minimum target tracking rate (MTTR) defined as the ratio of the tracking time for the least tracked target to the total simulation time. For both metrics, higher values correspond to better performance.

The results summarized in Table.4.1 show the performance variation across different tracking strategies. The small value of MTTR in non-cooperative tracking indicates that the homogeneous robot team is likely to lose at least one target for the majority of the time. This happens because the robots, with limited and directional FOV, can not guarantee to simultaneously and consistently track a larger number of freely moving targets. Moreover, once a target is lost, the growing uncertainty in target estimates may prevent the robots from recovering the lost target. The weak-cooperative tracking achieves limited improvement over the non-cooperative tracking by establishing communication in the mixed human-robot team. Nevertheless, robots fail to share data about the lesser tracked targets, leading to human operators being myopic about target priorities in choosing which

targets to update. Finally, the cooperative tracking consistently performs the best due to the proposed two-way cooperation. On one hand, robots actively send messages with priority indicators about the missed targets to support human decision-making, while on the other, the human operator selectively updates the target with the highest priority in accordance with robot queries, which guides the robots to recover the lost targets and improve the total tracking time.

Table 4.1: Tracking Performance Comparison

Methods	**ATTR**	**MTTR**
Non-cooperative tracking	44.2%	5.6%
Weak-cooperative tracking	50.6%	11.8%
Cooperative tracking	57.0%	26.5%

4.4.2 Experimental Results

The cooperative tracking framework is tested in physical experiments with a real human operator and Husarion ROSbot 2.0 ground robots that are each equipped with an ASTRA RGBD camera and a BNO055 IMU sensor. The human operator observes the workspace through a surveillance camera (Orbecc Astra camera) and communicates with the robot team over a shared WiFi network. A user-friendly interface is designed to allow the human operator to pass messages by simply clicking on the targets in the view of the surveillance camera. Then, the corresponding target position is automatically extracted with respect to the inertial frame and encoded for message passing. The ability to perform cooperative tracking in real time was successfully validated in 20 trials, two of which are demonstrated as case studies.

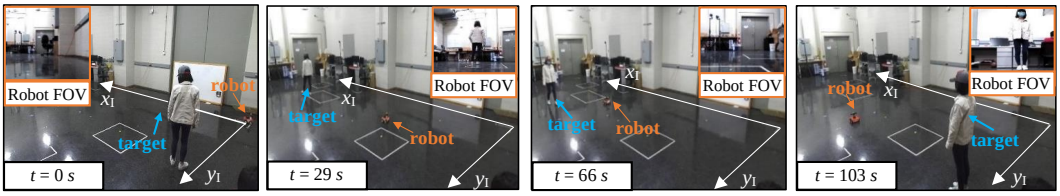


Figure 4.8: Demonstration of resilience to unpredictable target dynamics using *update* message with the robot view superimposed with the operator view.

The first case study demonstrates the resilience of the MHRT to unpredictable target dynamics through human-robot cooperation. As shown in Fig. 4.8, although the target is not inside the robot FOV at $t = 0s$, the robot successfully tracks the target at $t = 29s$ by optimizing the tracking utility. The target then makes a sudden change in the heading direction, which cannot be accounted for by the robot's prediction of the target dynamics and causes the robot to lose the target at $t = 66s$. Observing this unexpected change in target motion, the human operator sends an *update* message of the current target position to the robot. Owing to this message, the robot re-plans its path to successfully track the target again at $t = 103s$. The target trajectory, the timing of the *update* message, and the robot planned path before and after receiving the message are shown in Fig.4.9.

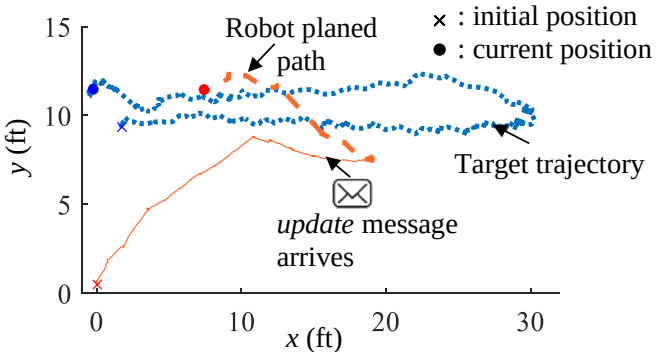


Figure 4.9: Target trajectory (dotted line) and planned robot path before (solid line) and after (dashed line) the *update* message corresponding to Fig.4.8.

The second case study demonstrates the flexibility of the network to incorporate a new target assignment through the *intruder* message sent by the human operator at any arbitrary time in the experiment. The robot initially tracks an assigned target while a new unknown target enters the workspace (Fig. 4.10). The human operator classifies the new target as an intruder and sends the new target's states through the *intruder* message. Having received the human message, the robot reconfigures its path by incorporating the intruder's information into its optimization framework and eventually tracks the intruder (Fig. 4.11). The target trajectory, the timing of the *intruder* message, and the robot planned path before and after receiving the message is shown in Fig.4.12.

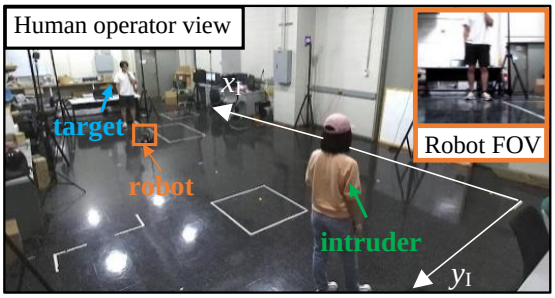


Figure 4.10: Demonstration of the human operator detecting an intruder while the robot is tracking its initially assigned target.

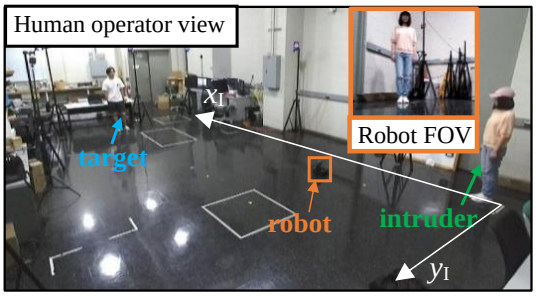


Figure 4.11: Demonstration of the intruder being tracked by robot replanning its path after receiving an *intruder* message from the human operator.

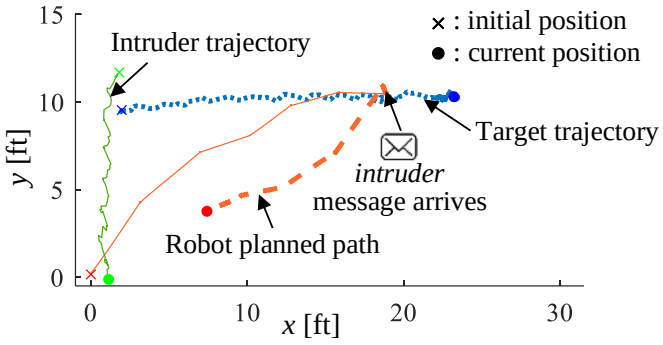


Figure 4.12: Target trajectory (dotted line) and planned robot path before (solid line) and after (dashed line) *intruder* message corresponding to Fig.4.10-4.11.

4.5 Conclusion

This chapter presents a novel human-robot teaming framework that integrates online sensing, distributed control optimization, and human-robot communication applicable to tracking multiple targets in an obstacle-populated environment. The dynamic target states are estimated using RGBD images streamed by the robot camera and extrinsic camera parameters such as camera orientation and translation, which are estimated online. A new tracking utility function is locally optimized by each robot to determine the control input that maximizes the cumulative target tracking time. Human-robot cooperation is realized by robots actively querying target information, and by human operators selectively responding to robot queries and providing additional reasoning-based information such as the discovery of intruders. Numerical simulations show that the mixed team achieves superior performance compared to the homogeneous robot teams. Moreover, testing the proposed approach in physical experiments shows its applicability for real-world implementations.

CHAPTER 5

CONCLUSION

With rapid development in computer vision and sensor design, holistic scene perception has been at the frontier of robotics research in recent decades. This dissertation covers novel approaches to three challenging problems for holistic scene perception, which are relevant to a wide range of applications, including but not limited to sport analytics, surveillance, autonomous driving, environmental monitoring, search and rescue, etc. First, to elevate robots intelligence to understand complex scenes, we developed algorithms for inferring social roles and predicting future actions in human team activities. Then, we study multi-target tracking using robot networks due to their advantages over single robot systems in terms of execution time, efficiency, and robustness. Finally, human intelligence is incorporated into the tracking framework by teaming robots with human partners to enlarge the complexity of situations that can be handled by the robotic agents alone.

Chapter 2 presents a holistic approach that integrates image recognition, state estimation, and inference of hidden variables is proposed for the challenging problem of action anticipation in human teams. The approach is demonstrated on the team sport of volleyball, in which the team strategy and players' roles are unobservable and change significantly over time. The team strategy is first inferred by constructing a team feature descriptor that aggregates domain knowledge of volleyball games and features of individual players. Sequentially, the players' roles, modeled probabilistically as the DMRF graph, can be inferred using a Markov chain Monte Carlo sampling method. By leveraging holistic information about the scene, including inferred team strategy, players' roles, as well as domain knowledge

and instantaneous visual features, the action anticipation MLP is able to predict the semantic label and timing of the future actions by multiple interacting key players on the team.

In addition to action anticipation in human team activities characterized by dynamic team interactions, the perception task of multi-target tracking is studied in Chapter 3. In particular, Chapter 3 presents a novel decentralized optimization framework that integrates online sensing, coordination, and control of multi-robot networks for applications that require tracking of multiple dynamic targets, such as surveillance and security. Two approximate approaches are proposed to decompose the optimization into two stages, where target assignment and robot control are determined in a sequential manner. Both approaches rely on the auction mechanism which leverages local communication to achieve conflict-free assignments, and rely on the local estimation to determine the best robot control in real-time to track the assigned targets. Simulation results show that the performance of the proposed approaches is very close to that of the optimal solution and is better than the other decentralized methods. Moreover, demonstrations of a physical multi-robot network tracking human targets show the applicability of the proposed approaches in real-world applications. The integration of the robot network control and the holistic scene understanding is further discussed as future work in the next Section.

To further enhance the tracking performance of the homogeneous robot network used in Chapter 3, a mixed human-robot team is constructed in Chapter 4, which leverages complementary skills of different team members for cooperative tracking. A new approach to human-robot collaboration is proposed, which entails the bi-directional message-exchange mechanism and distributed robot control that is a function of human actions. Human-robot collaboration is realized by robots

actively querying target information, and by human operators selectively responding to robot queries and providing additional reasoning-based information such as the discovery of intruders. The incorporation of human intelligence in the mixed team enlarges the complexity of situations that can be handled by the robotic agents alone. Moreover, human input can be sent to the robots asynchronously, letting the human operator decide when to interact with the robot team without adversely affecting the robot performance. Comparative studies in simulation show that the proposed cooperative tracking outperforms other baselines including the weak-cooperative tracking and non-cooperative tracking. Physical experiments demonstrate that this new MHRT control and communication framework is able to provide robust performance in the presence of uncertainties such as state estimation errors and intruders.

CHAPTER 6

FUTURE WORK

Beyond the novel active perception methods proposed in this dissertation, there exist three main possible directions for the future work. Firstly, the proposed robot network coordination and control framework can be extended to 3D scene reconstruction, which is a different aspect of holistic scene perception and has promising applications ranging from robot exploration in unknown indoor environment to 3D content creation for virtual reality. Because robots can work in parallel to cooperatively reconstruct the environment, the network utility can be formulated as minimizing the task completion time, or conversely, maximizing the efficiency. In order to leverage the network coordination algorithms developed in Chapter 3, the cooperative reconstruction can be converted to a dynamic task assignment problem. In particular, at each time instant, a set of viewpoints corresponding to the unknown or uncertain region of the environment can be extracted based on the scene map which is partially reconstructed up to that time instant. Then, robots determine amongst themselves the assignment of the viewpoints and locally plan their control to reach the assigned viewpoints where they can take images to reconstruct the scene.

Secondly, a further extension to the above 3D scene reconstruction includes the high-level scene interpretation, such as understanding the danger levels or identifying the salient regions in a scene, by human-robot collaboration. A unique characteristic of human intelligence is interpreting complex scenarios that would require sophisticated computer vision systems to perform the same task. Consider the search and rescue task as an example, for which a team of robots are dispatched to explore a hazardous environment whereas a human operator remotely monitor

the images sent by the robots. By looking at these images, humans can effectively estimate the dangerous level of the environment and recognize the salient regions that are more likely to have victims as compared to regions that are not worth exploring. Such information can be sent back to the robots, forming a bidirectional communication loop that facilitates the mission execution. In order to reduce latency in data transmission, the images transmitted by the robots must be compressed into a compact form to ease the communication burden. Moreover, human inputs should be incorporated into the robot control optimization, which can be solved by encoding human messages as a numerical representation compatible to the robot utility function following some pre-specified rules.

Thirdly, the robot control optimization in Chapter 3 can be integrated with the holistic scene understanding approach developed in Chapter 2 for high-level perception tasks such as human activity (action and interaction) recognition and crowd analysis. Potential applications include but not limited to anomaly detection in healthcare center, patrolling robots in public venues such as airports and sport events. In these applications, the workspace often greatly exceeds the size of the robot camera's FOV and, thus, managing the robot's viewpoint is crucial to the performance of the perception tasks. Human operators in the collaborative team can provide preliminary information to guide the robots to the vicinity of the targets-of-interest such that robots can perform online viewpoint estimation more accurately. Another difficulty in the above applications is to avoid collisions from people moving around in the environments. Therefore, the objective function should include the metric of activity recognition accuracy as a function of the robot viewpoint, and also include the collision penalty. The fusion of data from multiple sensing modalities such as vision and lidar sensors can be explored to enable the robots to achieve high recognition accuracy and fast collision avoidance in complex

dynamic environments populated with humans. In addition, collaborative robots that work closely with humans can anticipate the future actions of the human partners to make better plans and interactions.

APPENDIX A

EXPECTED ENTROPY REDUCTION

The EER term in (3.44) is derived by taking expectation with respect to the Bernoulli RFS $Z_{i,j}(k+1)$. For brevity, $Z_{i,j}(k+1)$ is denoted by $Z_{i,j}$ in the following derivation:

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

Readers are redirected to [107] for the derivation of the set integration. Substitute (3.40)-(3.41) in (A.1) gives the value of $I_{i,j}$.

APPENDIX B

DISCONTINUOUS, NON-CONVEX AND MULTI-MODAL OBJECTIVE FUNCTION FOR CONTROL OPTIMIZATION

The discontinuity in the objective function is introduced by the collision penalty ($C_{i,j}$) and by the bounded FOV model used in deriving the EER ($I_{i,j}$). The non-convexity is owing to the sum of tracking utility over multiple assigned targets ($|P_i^*(k)| \geq 1$), which also leads to a multimodal function. It suffices to show that there exists a case where the objective function is discontinuous, non-convex and multimodal. Without loss of generality, assume that the robot's FOV is omni-directional, which eliminates the effect of the robot orientation on the tracking utility. Consider a representative example, in which robot i is assigned two targets located at $[18 \quad 20]m$ and $[38 \quad 20]m$, respectively. The contribution of the EER term, the navigation reward, and the collision penalty to the objective function are visualized separately in Fig.B.1. The combined objective function is plotted in Fig. B.2, in which the discontinuity and non-convexity is clearly evident. With the introduction of a directional FOV into the problem, the above characteristics of the objective function are further justified. For the same reason that objective function being discontinuous, non-convex and multimodal, the optimality guarantees of the control optimization remains an open question.

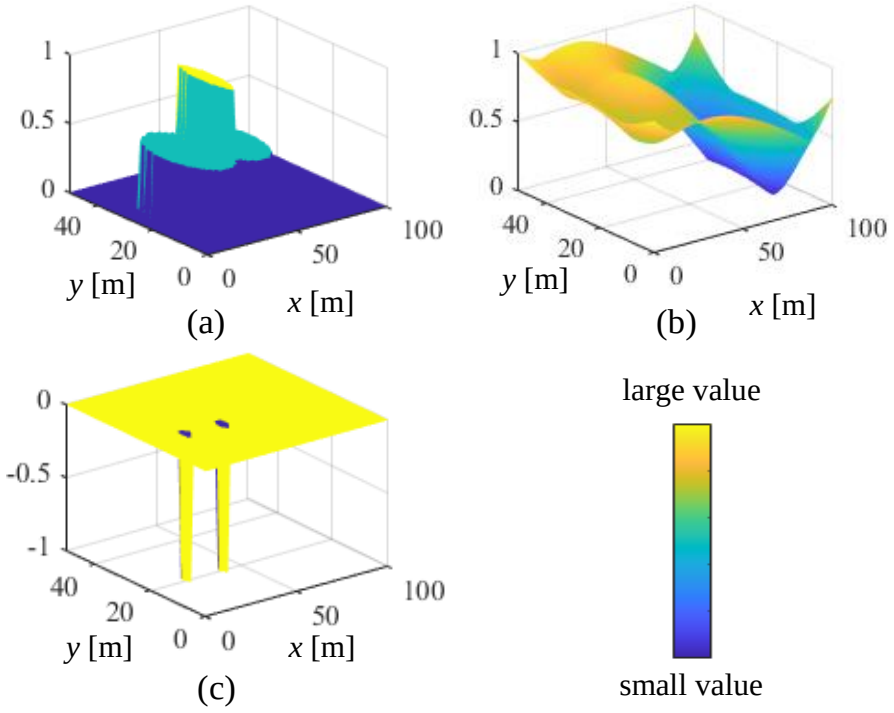


Figure B.1: Visulization of the EER (a), navigation reward (b), and collision penalty (c) for control optimization when two assigned targets are at $[18 \quad 20]m$ and $[38 \quad 20]m$.

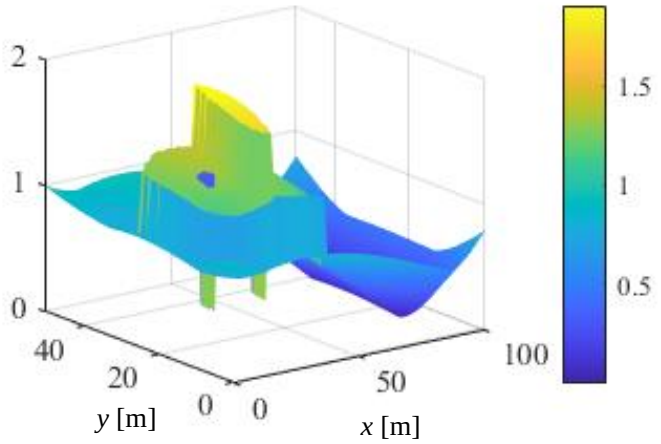


Figure B.2: A representative example of the objective function for robot control optimization.

BIBLIOGRAPHY

[1] Marco Aggravi, Giuseppe Sirignano, Paolo Robuffo Giordano, and Claudio Pacchierotti. Decentralized control of a heterogeneous human-robot team for exploration and patrolling. *IEEE Transactions on Automation Science and Engineering*, 2021.

[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] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

[4] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Måns Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip HS Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35(1):37–52, 2018.

[5] Gürdal Arslan, Jason R Marden, and Jeff S Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. 2007.

[6] 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.

[7] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, Long Beach, 2019.

[8] Timur Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4315–4324, Honolulu, 2017.

[9] 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.

[10] Jacopo Banfi, Jérôme Guzzi, Alessandro Giusti, Luca Gambardella, and Gianni A Di Caro. Fair multi-target tracking in cooperative multi-robot systems. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5411–5418. IEEE, 2015.

[11] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, Salt Lake City, 2018.

[12] Murchana Baruah and Bonny Banerjee. A multimodal predictive agent model for human interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1022–1023, 2020.

[13] Elliot T Berkman, Lauren E Kahn, and Jordan L Livingston. Valuation as a mechanism of self-control and ego depletion. In *Self-regulation and ego control*, pages 255–279. Elsevier, 2016.

[14] Dimitri Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66, 1992.

[15] Dimitri P Bertsekas. A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT*, 1979.

[16] Dimitri P Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, 1990.

[17] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019.

[18] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[20] Luc Brunet, Han-Lim Choi, and Jonathan How. Consensus-based auction approaches for decentralized task assignment. In *AIAA guidance, navigation and control conference and exhibit*, page 6839, 2008.

[21] P Cage, I Kroo, and R Braun. Interplanetary trajectory optimization using a genetic algorithm. In *Astrodynamics Conference*, page 3773, 1994.

[22] Yang Cao, Yupin Luo, and Shiyuan Yang. Image denoising based on hierarchical markov random field. *Pattern recognition letters*, 32(2):368–374, 2011.

[23] 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.

[24] Anirban Chakraborty and Amit K Roy-Chowdhury. Context-aware activity forecasting. In *Asian Conference on Computer Vision*, pages 21–36, Singapore, 2014. Springer.

[25] Michael Chan, Gabor T Herman, and Emanuel Levitan. Bayesian image reconstruction using a high-order interacting mrf model. In *International Conference on Image Analysis and Processing*, pages 608–614. Springer, 1995.

[26] Chao Chen, Shuhai Jiao, Shu Zhang, Weichen Liu, Liang Feng, and Yasha Wang. Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data. *IEEE Transactions on Intelligent Transportation Systems*, 19(10):3292–3304, 2018.

[27] Chao Chen, Qiang Liu, Xingchen Wang, Chengwu Liao, and Daqing Zhang. semi-traj2graph: Identifying fine-grained driving style with gps trajectory data via multi-task learning. *IEEE Transactions on Big Data*, pages 1–1, 2021.

[28] Evan Cheshire, Cibele Halasz, and Jose Krause Perin. Player tracking and analysis of basketball plays. In *European Conference of Computer Vision*, 2013.

[29] Siddhartha Chib and Srikanth Ramamurthy. Tailored randomized block mcmc methods with application to dsge models. *Journal of Econometrics*, 155(1):19–38, 2010.

[30] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decen-

tralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.

[31] Daniel E Clark. Multi-sensor network information for linear-gaussian multi-target tracking systems. *IEEE Transactions on Signal Processing*, 69:4312–4325, 2021.

[32] 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.

[33] Jeferson Rodrigues Cotrim and João Henrique Kleinschmidt. Lorawan mesh networks: A review and classification of multihop communication. *Sensors*, 20(15):4273, 2020.

[34] Kenneth James Williams Craik. *The nature of explanation*, volume 445. CUP Archive, 1952.

[35] Mary L Cummings, Jonathan P How, Andrew Whitten, and Olivier Toupet. The impact of human–automation collaboration in decentralized multiple unmanned vehicle control. *Proceedings of the IEEE*, 100(3):660–671, 2011.

[36] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[37] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

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

[39] Junyi Dong, Pingping Zhu, and Silvia Ferrari. Oriented pedestrian social interaction modeling and inference. In *2020 American Control Conference (ACC)*, pages 1373–1370, Virtual, 2020. IEEE.

[40] Ivica Draganjac, Damjan Miklić, Zdenko Kovačić, Goran Vasiljević, and Stjepan Bogdan. Decentralized control of multi-agv systems in autonomous warehousing applications. *IEEE Transactions on Automation Science and Engineering*, 13(4):1433–1447, 2016.

[41] Nour Eldin Elmadany, Yifeng He, and Ling Guan. Improving action recognition via temporal and complementary learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(3):1–24, 2021.

[42] Abdul Qadir Faridi, Sanjeev Sharma, Anupam Shukla, Ritu Tiwari, and Joydip Dhar. Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment. *Intelligent Service Robotics*, 11(2):171–186, 2018.

[43] Dirk Farin, Susanne Krabbe, Wolfgang Effelsberg, et al. Robust camera calibration for sport videos using court models. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 80–91. International Society for Optics and Photonics, 2003.

[44] Alircza Fathi, Jessica K Hodgins, and James M Rehg. Social interactions: A first-person perspective. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1226–1233, Providence, 2012. IEEE.

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

[46] 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. IEEE, 2008.

[47] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6252–6261, Long Beach, 2019.

[48] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5562–5571, Long Beach, 2019.

[49] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-6(6):721–741, 1984.

[50] 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.

[51] Anthony Giddens and Philip W Sutton. *Essential concepts in sociology*. John Wiley & Sons, 2021.

[52] Josep M Gonfaus, Xavier Boix, Joost Van de Weijer, Andrew D Bagdanov, Joan Serrat, and Jordi Gonzalez. Harmony potentials for joint classification and segmentation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3280–3287, San Francisco, 2010. IEEE.

[53] Ankur Gupta, James J Little, and Robert J Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *2011 Canadian Conference on Computer and Robot Vision*, pages 32–39, St Johns, 2011. IEEE.

[54] Fasih Haider, Fahim A Salim, Dees BW Postma, Robby Van Delden, Dennis Reidsma, Bert-Jan van Beijnum, and Saturnino Luz. A super-bagging method for volleyball action recognition using wearable sensors. *Multimodal Technologies and Interaction*, 4(2):33, 2020.

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

[56] Christian G Harris, Zachary I Bell, Runhan Sun, Emily A Doucette, J Willard Curtis, and Warren E Dixon. Target tracking in the presence of intermittent measurements by a network of mobile cameras. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5962–5967. IEEE, 2020.

[57] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[58] Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13):1660–1677, 2015.

[59] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, Venice, 2017.

[60] Li He, Guoliang Liu, Guohui Tian, Jianhua Zhang, and Ze Ji. Efficient multi-view multi-target tracking using a distributed camera network. *IEEE Sensors Journal*, 20(4):2056–2063, 2019.

[61] Shiwen He, Shaowen Xiong, Yeyu Ou, Jian Zhang, Jiaheng Wang, Yongming Huang, and Yaoxue Zhang. An overview on the application of graph neural networks in wireless networks. *IEEE Open Journal of the Communications Society*, 2021.

[62] Jonathan P How, Cameron Fraser, Karl C Kulling, Luca F Bertuccelli, Olivier Toupet, Luc Brunet, Abe Bachrach, and Nicholas Roy. Increasing autonomy of uavs. *IEEE Robotics & Automation Magazine*, 16(2):43–51, 2009.

[63] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *European Conference on Computer Vision*, pages 489–504, Zurich, 2014. Springer.

[64] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[65] Simon Hunt, Qinggang Meng, Chris Hinde, and Tingwen Huang. A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements. *Cognitive computation*, 6(3):338–350, 2014.

[66] Mostafa S Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–736, Munich, 2018.

[67] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1980, Las Vegas, 2016.

[68] Luca Iocchi. Robust color segmentation through adaptive color distribution transformation. In *Robot Soccer World Cup*, pages 287–295. Springer, 2006.

[69] Asuncion Jimenez-Cordero, Juan Miguel Morales, and Salvador Pineda. Warm-starting constraint generation for mixed-integer optimization: A machine learning approach. *Knowledge-Based Systems*, 253:109570, 2022.

[70] Long Jin, Shuai Li, Hung Manh La, Xin Zhang, and Bin Hu. Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach. *Automatica*, 100:75–81, 2019.

[71] Thorsten Joachims. Structured output prediction with support vector machines. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 1–7, Hong Kong, 2006. Springer.

[72] Roy Jonker and Ton Volgenant. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, 1986.

[73] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, Long Beach, 2019.

[74] Asif Khan, Evsen Yanmaz, and Bernhard Rinner. Information exchange and decision making in micro aerial vehicle networks for cooperative search. *IEEE Transactions on Control of Network Systems*, 2(4):335–347, 2015.

[75] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10(7):1755–1758, 2009.

[76] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[77] Hema S Koppula, Ashesh Jain, and Ashutosh Saxena. Anticipatory planning for human-robot teams. In *Experimental robotics*, pages 453–470. Springer, 2016.

[78] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704, Zurich, 2014. Springer.

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

[80] 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.

[81] Kang Li and Yun Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1644–1657, 2014.

[82] Leo Liberti. Undecidability and hardness in mixed-integer nonlinear programming. *RAIRO-Operations Research*, 53(1):81–109, 2019.

[83] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[84] Ralph Linton. *The study of man: An introduction*. D. Appleton-Century company, incorporated, 1936.

[85] 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.

[86] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4106–4115, 2020.

[87] 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.

[88] 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.

[89] Sean Luke, Keith Sullivan, Liviu Panait, and Gabriel Balan. Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 911–917, 2005.

[90] Nima Mahmoudi, Seyed Mohammad Ahadi, and Mohammad Rahmati. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, 78(6):7077–7096, 2019.

[91] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5773–5782, Venice, 2017.

[92] Anton Milan, S Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad

Schindler. Online multi-target tracking using recurrent neural networks. In *Thirty-First AAAI conference on artificial intelligence*, 2017.

[93] Partha Pratim Mondal, Giuseppe Vicidomini, and Alberto Diaspro. Markov random field aided bayesian approach for image reconstruction in confocal microscopy. *Journal of applied Physics*, 102(4):044701, 2007.

[94] M Naveenkumar and S Domnic. Deep ensemble network using distance maps and body part features for skeleton based action recognition. *Pattern Recognition*, 100:107125, 2020.

[95] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Future event prediction: If and when. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, Long Beach, 2019.

[96] Sebastian Nowozin, Christoph H Lampert, et al. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

[97] 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.

[98] Rahul Pal, KP Sarawadekar, and KV Srinivas. A decentralized beam selection for mmwave beamspace multi-user mimo system. *AEU-International Journal of Electronics and Communications*, 111:152884, 2019.

[99] Lynne E Parker. Decision making as optimization in multi-robot teams. In *International Conference on Distributed Computing and Internet Technology*, pages 35–49. Springer, 2012.

[100] Jeffrey R Peters, Amit Surana, and Francesco Bullo. Robust scheduling and routing for collaborative human/unmanned aerial vehicle surveillance missions. *Journal of Aerospace Information Systems*, 15(10):585–603, 2018.

[101] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8:191617–191643, 2020.

[102] 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.

[103] Vignesh Ramanathan, Bangpeng Yao, and Li Fei-Fei. Social role discovery in human events. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2475–2482, Portland, 2013.

[104] João Ramos, Rui J Lopes, and Duarte Araújo. What's next in complex networks? capturing the concept of attacking play in invasive team sports. *Sports medicine*, 48(1):17–28, 2018.

[105] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[106] João Ribeiro, Keith Davids, Duarte Araújo, Pedro Silva, João Ramos, Rui Lopes, and Júlio Garganta. The role of hypernetworks as a multilevel methodology for modelling and understanding dynamics of team sports performance. *Sports Medicine*, 49(9):1337–1344, 2019.

[107] Branko Ristic, Ba-Tuong Vo, Ba-Ngu Vo, and Alfonso Farina. A tutorial on bernoulli filters: theory, implementation and applications. *IEEE Transactions on Signal Processing*, 61(13):3406–3430, 2013.

[108] Cristian Rodriguez, Basura Fernando, and Hongdong Li. Action anticipation by predicting future dynamic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–10, Munich, 2018.

[109] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 280–289, Venice, 2017.

[110] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence*, 135(1-2):1–54, 2002.

[111] Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032, 2018.

[112] Paul Schnitzspan, Mario Fritz, Stefan Roth, and Bernt Schiele. Discriminative structure learning of hierarchical representations for object detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2238–2245, Miami, 2009. IEEE.

[113] Weihua Sheng, Qingyan Yang, Jindong Tan, and Ning Xi. Distributed multi-robot coordination in area exploration. *Robotics and autonomous systems*, 54(12):945–955, 2006.

[114] Yuge Shi, Basura Fernando, and Richard Hartley. Action anticipation with rbf kernelized feature mapping rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 301–317, Munich, 2018.

[115] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. Cern: confidence-energy recurrent network for group activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5523–5531, Honolulu, 2017.

[116] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network. *Pattern Recognition*, 107:107511, 2020.

[117] Ajit Kumar Singh, Sandeep Rajoriya, Subham Nikhil, and Tapan Kumar Jain. Design constraint in single-hop and multi-hop wireless sensor network using different network model architecture. In *International Conference on Computing, Communication & Automation*, pages 436–441. IEEE, 2015.

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

[119] Stephen L Smith and Francesco Bullo. Target assignment for robotic networks: Asymptotic performance under limited communication. In *2007 American Control Conference*, pages 1155–1160. IEEE, 2007.

[120] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Online localization and prediction of actions and interactions. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):459–472, 2018.

[121] Yoonchang Sung, Ashish Budhiraja, Ryan K Williams, and Pratap Tokekar. Distributed assignment with limited communication for multi-robot multi-target tracking. *Autonomous Robots*, 44(1):57–73, 2020.

[122] Yoonchang Sung, Ashish Kumar Budhiraja, Ryan K Williams, and Pratap Tokekar. Distributed simultaneous action and target assignment for multi-robot multi-target tracking. In *2018 IEEE International conference on robotics and automation (ICRA)*, pages 3724–3729. IEEE, 2018.

[123] Yansong Tang, Jiwen Lu, Zian Wang, Ming Yang, and Jie Zhou. Learning semantics-preserving attention and contextual interaction for group activity recognition. *IEEE Transactions on Image Processing*, 28(10):4997–5012, 2019.

[124] Rajkumar Theagarajan, Federico Pala, Xiu Zhang, and Bir Bhanu. Soccer: Who has the ball? generating visual analytics and player statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1749–1757, Salt Lake City, 2018.

[125] Henry L Tischler. *Cengage advantage books: Introduction to sociology*. Cengage Learning, 2013.

[126] Xiaofeng Tong, Jia Liu, Tao Wang, and Yimin Zhang. Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(2):1–32, 2011.

[127] 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.

[128] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[129] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, Las Vegas, 2016.

[130] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[131] Lei Wang, Wei-jie Feng, Michael ZQ Chen, and Qing-guo Wang. Global bounded consensus in heterogeneous multi-agent systems with directed communication graph. *IET Control Theory & Applications*, 9(1):147–152, 2015.

[132] Suhang Wang, Charu Aggarwal, and Huan Liu. Random-forest-inspired neural networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(6):1–25, 2018.

[133] 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.

[134] Yiming Wang and Andrea Cavallaro. Prioritized target tracking with active collaborative cameras. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 131–137. IEEE, 2016.

[135] Yingying Wang, Qingchun Ji, and Chenglin Zhou. Effect of prior cues on action anticipation in soccer goalkeepers. *Psychology of Sport and Exercise*, 43:137–143, 2019.

[136] 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.

[137] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[138] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[139] Qiong Wu and Pierre Boulanger. Enhanced reweighted mrfs for efficient fashion image parsing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(3):1–16, 2016.

[140] Zhirong Wu, Dahua Lin, and Xiaoou Tang. Deep markov random field for image modeling. In *European Conference on Computer Vision*, pages 295–312, Amsterdam, 2016. Springer.

[141] Jiang Yan, Wang Daobo, Bai Tingting, and Yan Zongyuan. Multi-uav objective assignment using hungarian fusion genetic algorithm. *IEEE Access*, 10:43013–43021, 2022.

[142] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li, and Qi Tian. Participation-

contributed temporal dynamic model for group activity recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1292–1300, Seoul, 2018.

[143] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.

[144] Huili Yu, Kevin Meier, Matthew Argyle, and Randal W Beard. Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles. *IEEE/ASME transactions on mechatronics*, 20(2):541–552, 2014.

[145] 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, 2014.

[146] Michael M Zavlanos, Leonid Spesivtsev, and George J Pappas. A distributed auction algorithm for the assignment problem. In *2008 47th IEEE Conference on Decision and Control*, pages 1212–1217. IEEE, 2008.

[147] Shengping Zhang, Hongxun Yao, Xin Sun, and Shaohui Liu. Robust visual tracking using an effective appearance model based on sparse coding. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–18, 2012.

[148] 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.

[149] Wanqing Zhao, Qinggang Meng, and Paul WH Chung. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE transactions on cybernetics*, 46(4):902–915, 2015.

[150] 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.

[151] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing multiple features for depth-based action recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2):1–20, 2015.