

An Online Scalable Approach to Unified Multirobot Cooperative Localization and Object Tracking

Aamir Ahmad, Guilherme Lawless, and Pedro Lima, *Member, IEEE*

Abstract—In this paper, we present a unified approach for multi-robot cooperative simultaneous localization and object tracking based on particle filters. Our approach is scalable with respect to the number of robots in the team. We introduce a method that reduces, from an exponential to a linear growth, the space and computation time requirements with respect to the number of robots in order to maintain a given level of accuracy in the full-state estimation. Our method requires no increase in the number of particles with respect to the number of robots. However, in our method, each particle represents a full-state hypothesis, leading to the linear dependency on the number of robots of both space and time complexity. The derivation of the algorithm implementing our approach from a standard particle filter algorithm and its complexity analysis are presented. Through an extensive set of simulation experiments on a large number of randomized datasets, we demonstrate the correctness and efficacy of our approach. Through real robot experiments on a standardized open dataset of a team of four soccer-playing robots tracking a ball, we evaluate our method's estimation accuracy with respect to the ground truth values. Through comparisons with other methods based on 1) nonlinear least squares minimization and 2) joint extended Kalman filter, we further highlight our method's advantages. Finally, we also present a robustness test for our approach by evaluating it under scenarios of communication and vision failure in teammate robots.

Index Terms—Cooperative perception, distributed robot systems, localization, sensor fusion, visual tracking.

I. INTRODUCTION

ACHIEVING a complexity that scales well with the number of robots for the online cooperative simultaneous robot localization and object tracking (SLOT) problem in a multirobot scenario is the core topic of this paper. One of the

approaches to multirobot cooperative object tracking is to explicitly share information among the robots. This includes sharing self-localization estimates and environment observations to enrich other robot's observations. Moreover, for this purpose, one needs a well-calibrated self-localization confidence measuring technique to avoid reference frame inconsistencies. We addressed this situation previously in [1]. However, since the confidence measuring mechanisms are often based on heuristics, e.g., using the *number of effective particles* [2] as a measure of robot localization confidence, it might still result in false positives (e.g., a situation where the self-localization confidence of a robot is high in spite of it being wrongly localized). Similarly, cooperative robot localization can also be performed by explicitly sharing information regarding a mutually observed common object as showed in [3], but it would still remain highly dependent on the self-localization confidences of the teammate robots as well as their object observation confidence measures when using the teammate's observation or estimates. In a scenario where a team of robots needs to localize themselves as well as track an object, cooperative methods for doing both simultaneously can benefit from using a single unified approach because this would eliminate the need for separate self-localization/object-observation confidence measurements. Note that this does not imply first performing cooperative object tracking, and subsequently using the cooperatively tracked object's estimate to improve the self-localization of the robots (i.e., a two-step process, which some authors have previously explored [4]). In a unified method, both processes (cooperative localization of robots and cooperative estimation of the object) are performed in a single step to prevent the recursive propagation of estimation error from one process to the other, which is inevitable in a two-step process. In [5], we presented such a unified method, based on nonlinear least squares minimization, designed for offline implementation.

In this paper, we present a novel online¹ method for the multirobot unified cooperative localization and object tracking (UCLT) problem, based on a particle filter (PF). The goal is that a robot estimates the state composed of its own pose, the poses of its teammate robots, and the position of a mutually observed tracked object, at a given timestep. The input to the method is the state at the previous timestep and the most recent control and observation measurements made by the robots. The control measurements refer to those made by the proprioceptive sensors of

Manuscript received February 2, 2017; accepted May 6, 2017. Date of publication June 30, 2017; date of current version October 2, 2017. This paper was recommended for publication by Associate Editor C. Secchi and Editor C. Torras upon evaluation of the reviewers' comments. This work was supported in part by the Portuguese Government through FCT Project PCMMC: PTDC/EEA-CRO/100692/2008 and Fundação para a Ciência e a Tecnologia LARSyS strategic funding UID/EEA/50009/2013 and in part by the European Commission through the Marie Curie Intra-European Fellowship (Project 626050: TRaVERSE). (*Corresponding author: Aamir Ahmad.*)

A. Ahmad is with the Max Planck Institute for Intelligent Systems, Tübingen 72076, Germany, and also with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1049-001, Portugal (e-mail: aamir.ahmad@tecnico.ulisboa.pt).

G. Lawless and P. Lima are with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1049-001, Portugal (e-mail: guilherme.lawless@tecnico.ulisboa.pt; pedro.lima@tecnico.ulisboa.pt).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2017.2715342

¹An offline method/approach refers to a batch process which optimizes a full trajectory of states after all measurements are acquired, while an online approach recursively estimates the state only at a given timestep using the measurements at that timestep and the previous state.

the robots, e.g., odometry. The observations made by the robots include range and bearing measurements to the environmental landmarks/features and to the tracked object. We explicitly do not consider interrobot range and bearing measurements. Including such measurements would simply lead to another class of cooperative localization problem, which is not in the scope of this paper. The main motivation for our work in this paper is that through cooperatively tracking a mutually observed common object, a team of robots can implicitly perform cooperative localization. In comparison with noncooperative methods, this would not only improve the accuracy of the object position estimates but also improve the localization accuracy of the robots, especially in situations where the self-localization of the robots is prone to failure due to poor observation measurements of environmental landmarks/features. In our method (henceforth referred to as PF-UCLT algorithm), presented further in Section III, a particle's state hypothesis component (the other component being its weight) represents the pose of the robot running the algorithm, the poses of the teammate robots, and the position of the tracked object. The update step of our algorithm systematically overcomes the problem of error propagation from the object position estimate to the robots' pose estimates, which happens in a two-step process as previously mentioned. The core novel features of our approach are as follows.

- 1) A standard PF implementation for the UCLT problem will require an exponentially growing number of particles, hence exponentially growing space and computational time complexity, w.r.t. the number of robots in order to achieve a given level of estimation accuracy and to avoid the particle deprivation problem. Our PF-UCLT approach does not require any increase in the number of particles w.r.t. the number of robots. This is done by exploiting the properties of conditional and mutual independence of some measurements involved in the estimation process. However, in our method, each particle represents a full-state hypothesis, leading to the linear dependency on the number of robots of both space and time complexity. This makes our online method scalable w.r.t. the number of robots in the team. We show the method is scalable by applying it to several simulated datasets and a real robot dataset. To validate its superior performance we experimentally compare it on the same datasets with
 - a) an online joint extend Kalman filter (EKF)-based method and
 - b) an offline nonlinear least squares minimization-based method named MMG-O [5].
- 2) The method is also robust to communication and sensor measurement failures as it does not assume that the tracked object is visible to all robots at all times. Through a series of experiments we show how our PF-UCLT algorithm copes with such failures.

Note that in this paper we assume a known map of landmarks used by each robot to improve its self-localization estimate when a sufficient number of landmarks is visible. Hence, we do not address the simultaneous localization and mapping (SLAM) problem. Including the mapping aspect within our framework of cooperative localization and object tracking would not only increase the computational complexity by requiring additional

state components, but would also violate the properties of mutual and conditional independences in some measurements which we explore in this work. Consequently, the SLAM extension of our work deserves a separate treatment.

The rest of the paper is organized as follows. In Section II, we overview the state-of-the-art methods for cooperative localization and object tracking with brief comments on how our method overcomes some of the open challenges in this context. Section III presents the theoretical details of the PF-UCLT algorithm along with its complexity analysis. This is followed by the testbed and dataset description in Section IV. The simulation and real robot experimental results are presented in Sections V and VI, respectively. Section VII concludes the paper with comments on future directions.

II. STATE OF THE ART

The target tracking problem, manifested in several different forms, is now a very mature field of research [6]–[8]. Scenarios ranging from single to multiple robots tracking one or more objects/targets [9] to cases where tracking benefits from a network of static sensors in addition to moving robots [10] have been studied extensively. PFs are one of the widely used methods employed for target tracking [9], [11], [12] due to its multimodal and nonparametric form. PFs are suitable for scenarios where the target's motion model is unpredictable and/or switches to a different model over time. Using PFs, authors have efficiently addressed single-robot single-target problem, e.g., [12]–[14]. However, it is evident that occlusions and limitations on the range of sensors are some of the key reasons why multirobot/multisensor approach has gained more attraction in the recent years. In this context, a decentralized PF-based technique was developed in [9] and [15] to address inherent issues with multirobot cooperative object tracking, e.g., limited communication bandwidth (solved by sharing Gaussian mixture models instead of complete particle sets). Other explored techniques involve sharing target-related world-frame constraint relations among the robots, the solution of which provides the target's global position estimates [4], and sharing reduced sets of particles among the teammates [16]. While the case of multiple-static-sensors and single-target was addressed in [17], the problem of multiple moving targets using a single moving platform was handled in [18]. An important focus of our work in this paper is to address a more general situation, where a moving target is tracked by multiple moving robots which, at the same time, also need to localize themselves. Although the solution that we propose in this paper could be extended in a straightforward manner to multiple targets with known data association, the increase in computational complexity with the number of targets remains an open problem to address.

The cooperative multirobot localization problem has also attained a widespread attention [19]–[23]. It has been addressed through numerous viewpoints originating from different application areas, e.g., using interrobot measurements [19], [24] or communicating the relative positions among the teammates and treating them as observation measurements in a filtering algorithm where the states include the poses of all the robots [25]. Using interrobot measurements, the works of Kurazume *et al.*

[26]–[29] and of Rekleitis *et al.* [30] are among the first that introduced and formalized the idea of cooperative localization. Later, in [31], Roumeliotis and Bekey provide a principled KF-based framework to handle the interdependent and independent measurements, whereas in [32] Rekleitis *et al.* formulated the problem using a PF. More recently, in [33], Bailey *et al.* presented an efficient mechanism for cooperative robot localization, which is an alternative method to overcome the recursive propagation of errors. This is done by centralizing the cooperative estimation. In their method, each robot first estimates its own pose and then communicates it, along with an interrobot measurement, to a central data server. These data, received from all the robots in the team, are then fused to compute consistent relative localization of every robot. However, this implies that the robots need to have an additional sensor not only to measure the relative distances to the other robots in the team but also to detect their respective IDs. Besides being a practical issue, this leads to a different class of cooperative localization problem because 1) most measurements become strongly correlated and 2) a data association issue needs to be solved. Our work, in this paper, does not address this class of problems and we assume that direct interrobot measurements are not available. Other methods include communicating environment-specific static features and landmark information to robots that might not be able to observe those themselves due to occlusions/distance, e.g., [34], [35]. Another recent method to perform cooperative localization is to share information among the robots regarding a mobile object/target observable (and/or being tracked) in the world frame, at the same time, by all the robots in the team [3]. In such a case, if the commonly observed object is also being tracked by the robots, then both cooperative robot localization and cooperative object tracking can benefit from each other. This is one of the key concepts explored by the method we propose in this paper.

SLOT in the context of single-robot application has also been a subject of wide interest. For instance, in [36], a conditional PF localizes a mobile robot and tracks multiple people in the environment. Later in [37], the work was extended to address data-association with uncertainty using sample-based joint probabilistic data association. However, extending this to multirobot scenarios is a challenging step, which we address in this paper. The field of active cooperative perception, although slightly different in its goal from SLOT, provides significant insights to solve the SLOT problem in the context of multirobot scenarios. Jung and Sukhatme in [38] and Ahmad *et al.* [39] present active cooperative target tracking methods, where the goal is to control each of the robots in the team and move them to a location such that the cooperative target perception uncertainty is minimized. An approach by Zhou and Roumeliotis [40] addresses the issue of complexity that arises in active cooperative target tracking and reduces it from exponential to linear w.r.t. the number of robots. However, in their approach, robot localization is not addressed, and it only deals with the motion control of the robots (generating future trajectory of the robots) and cooperative tracking of the target. In contrast, our approach deals with both the cooperative localization of the robots and cooperative tracking of the object, in a

unified framework, and reduces the computational complexity from exponential to linear w.r.t. the number of robots. A similar idea was explored by Chang *et al.* in [41], where they presented an EKF formulation of the problem and apply their method to real humanoid robots. One of the experimental comparisons of our method, which is based on a PF, is with the EKF-based method. We show that our method significantly outperforms the EKF-based approach. Earlier, we had presented an offline solution to the UCLT problem in [5] based on nonlinear least squares minimization. In this paper, we will also compare our PF-based online solution with the offline approach presented in [5] and show that comparable accuracies can be achieved by our latest method in real time. Furthermore, it is a two-step approach in which the errors of target estimation are propagated to self-localization of the robots, although not recursively since target observations are in nonrobot-centric frames. The usage of nonrobot-centric frames can sometimes be difficult. For instance, in large outdoor scenarios with few observable landmarks at a given instant, robot-centric observations are necessary. In such a case, using a two-step approach, like in [4], would lead to recursive propagation of errors.

III. PF-BASED UNIFIED COOPERATIVE LOCALIZATION AND TRACKING

In this section, we first present the formulation of the online UCLT problem followed by the PF-UCLT algorithm, the core contribution of this paper, and its detailed description. The PF-UCLT algorithm is designed in a computationally decentralized manner, such that each robot runs its own instance of the algorithm where it receives the measurements made by its own sensors and by the sensors of its teammate robots, assumed to be transmitted to it through a wireless communication system.

A. Online UCLT Problem Formulation

We formulate the online UCLT problem using a recursive Bayesian filter. Let there be N robots r_1, \dots, r_N in a team tracking an object \mathbf{O} in an environment consisting of L static landmarks represented as a set \mathbf{L}_{map} . The IDs and the positions of the landmarks in the world frame are known. The state [two-dimensional (2-D) pose in the world frame] of the robot r_n is given by $\mathcal{L}_t^{r_n} = [x_t^{r_n} \ y_t^{r_n} \ \theta_t^{r_n}]^\top$ and the state (3-D position in the world frame) of the tracked moving object is given by $\mathbf{O}_t = [x_t^{\mathbf{O}} \ y_t^{\mathbf{O}} \ z_t^{\mathbf{O}}]^\top$ at the t th timestep.

The 2-D position of the l th known and static landmark is given $\mathbf{l}^l = [l_x^l \ l_y^l]^\top$. The landmarks are assumed to be fixed on the ground plane on which the robots move.

The odometry measurement made by the robot r_n at the t th timestep is given by the vector $\mathbf{u}_t^{r_n}$ and an associated noise with zero mean and covariance matrix $\mathbf{R}_t^{r_n}$. The static landmark observation measurement of the l th landmark made by the robot r_n in its local frame at the t th timestep is given by the vector $\mathbf{z}_t^{r_n, l}$ and an associated noise with zero mean and covariance matrix $\mathbf{Q}_t^{r_n, l}$. Similarly, the moving object \mathbf{O} 's observation measurement made by the robot r_n in its local frame at the t th timestep is given by the vector $\mathbf{z}_t^{r_n, \mathbf{O}}$ and an associated noise with zero mean and covariance matrix $\Sigma_t^{r_n, \mathbf{O}}$.

We now define \mathbf{x}_t as the full-state vector being estimated by stacking all individual states at the t th timestep as

$$\mathbf{x}_t = \left[\mathcal{L}_t^{r_1 \top} \quad \dots \quad \mathcal{L}_t^{r_N \top} \quad \mathbf{O}_t \top \right]^\top. \quad (1)$$

\mathbf{u}_t is obtained by stacking all control (robot odometry) measurements available at the t th timestep as

$$\mathbf{u}_t = \left[\mathbf{u}_t^{r_1 \top} \quad \dots \quad \mathbf{u}_t^{r_N \top} \right]^\top. \quad (2)$$

\mathbf{z}_t is obtained by stacking all the observation measurements available at the t th timestep as

$$\mathbf{z}_t = \left[\mathbf{z}_t^{r_1, 1 \top} \quad \dots \quad \mathbf{z}_t^{r_1, L \top} \quad \dots \quad \mathbf{z}_t^{r_N, 1 \top} \quad \dots \quad \mathbf{z}_t^{r_N, L \top} \quad \mathbf{z}_t^{r_1, \mathbf{o} \top} \quad \dots \quad \mathbf{z}_t^{r_N, \mathbf{o} \top} \right]^\top. \quad (3)$$

The online UCLT problem seeks to estimate $bel(\mathbf{x}_t)$, the posterior belief of the state \mathbf{x}_t at the t th timestep, given all the measurement data up to that timestep. This is given by a probability distribution over the state space $bel(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$, conditioned on the available measurement data. Using the recursive Bayesian filter equation, under the Markovian assumption of state's completeness [42], we obtain

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \quad (4)$$

where η is the proportionality constant.

B. PF-UCLT Algorithm: A Solution to the Online UCLT Problem

We first discuss a standard PF solution to the recursive Bayesian formulation of the online UCLT problem given by (4), its shortcomings, then describe the proposed PF-UCLT algorithm and how it addresses those shortcomings.

PF approximates $bel(\mathbf{x}_t)$ by a set of M particles $\mathcal{X}_t \triangleq \{\langle \mathbf{x}_t^{[m]}, w_t^{[m]} \rangle\}_{m=1}^M$, representing state hypotheses and their weights. A standard PF solution estimates \mathcal{X}_t from \mathcal{X}_{t-1} , \mathbf{u}_t , \mathbf{z}_t , and \mathbf{L}_{map} through the following steps.

Step 1: A temporary particle set $\bar{\mathcal{X}}_t \triangleq \{\langle \bar{\mathbf{x}}_t^{[m]}, \bar{w}_t^{[m]} \rangle\}_{m=1}^M$ is initialized with null values.

Step 2: Hypothesis prediction for all temporary particles

$$\bar{\mathbf{x}}_t^{[m]} \sim p\left(\mathbf{x}_t | \bar{\mathbf{x}}_{t-1}^{[m]}, \mathbf{u}_t\right). \quad (5)$$

Step 3: Weight update for all temporary particles

$$\bar{w}_t^{[m]} \propto p\left(\mathbf{z}_t | \bar{\mathbf{x}}_t^{[m]}, \mathbf{L}_{\text{map}}\right). \quad (6)$$

Step 4: $\bar{\mathcal{X}}_t$ is resampled [42] to obtain \mathcal{X}_t . Finally, the state estimate \mathbf{x}_t is obtained from \mathcal{X}_t using either a weighted average of the particles' state components or the particle with the maximum weight.

The performance of a PF is heavily dependent on the number of particles [43], [44]. In practice, to achieve a good approximation of the posterior belief $bel(\mathbf{x}_t)$ by a PF-based method and to

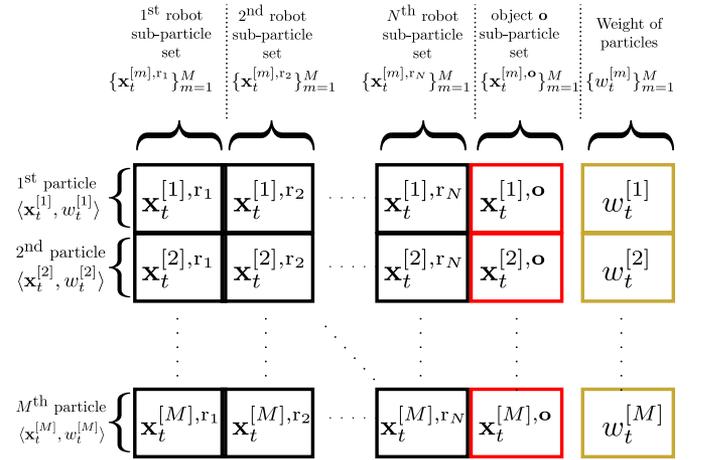


Fig. 1. Structure of particles, sub-particles, particle weights, and the associated notations defined in Section III and used in Algorithm 1 and subsequently.

not quickly fall into the particle deprivation problem [42], the number of particles depends exponentially [43], [45] on the dimension of the state space represented by a particle. The particle deprivation problem refers to a situation where none of the particles are in the vicinity of the correct state. This is more likely to happen as the dimension of the state space grows. In [43], Quang *et al.* formally prove that the PF error increases exponentially with the estimated state's dimension and therefore, to maintain a given accuracy, the number of particles used in a PF must increase exponentially with the state's dimension. We briefly discuss the role of state dimensionality in the context of the UCLT problem.

Let us assume that in case of a single robot localization, where the robot moves on a 2-D plane (such that the state-space dimension is 3 including its 2-D position and orientation), the required number of particles to achieve a given accuracy level in the localization estimates by a PF-based method is M (in practice, M is usually tractably small, i.e., M in the order of thousands results in an acceptable accuracy and computational speed). However, when the state space consists of poses of N robots tracking 3-D positions of O objects, the number of particles required must be $M^{(N+O)}$ to achieve the same accuracy level in all poses and position estimates similar to that obtained by M particles in case of a single-robot localization. This renders the usage of a standard PF implementation inefficient for the UCLT problem. Even with $N = 2$ and $O = 1$, which is the minimum requirement for a UCLT scenario, the required number of particles will be in the order of millions.

In our solution, the PF-UCLT algorithm (see Algorithm 1), we overcome the aforementioned issue by utilizing the properties of conditional and mutual independence of some of the involved variables and accordingly modifying the weight update step of the temporary particles. Consequently, the required number of particles remains constant w.r.t. N , whereas the space and time complexity grow only linearly w.r.t. N in order to maintain a given accuracy level in all the poses and position estimates. Furthermore, the time complexity in our solution, as described further, will also reduce from exponential to linear w.r.t. N .

PF-UCLT Algorithm Description: Before describing the PF-UCLT algorithm, it is important to introduce the concept of sub-particles used throughout the rest of the text (see Fig. 1). Since the state hypothesis component $\mathbf{x}_t^{[m]}$ of the m th particle $\langle \mathbf{x}_t^{[m]}, w_t^{[m]} \rangle$ is composed of the states of all robots r_1, \dots, r_N and the tracked object \mathbf{O} at the timestep t , we define a particle to be an $(N+2)$ -tuple as

$$\underbrace{\langle \mathbf{x}_t^{[m]}, w_t^{[m]} \rangle}_{m^{\text{th}} \text{particle}} \stackrel{\text{def.}}{=} \underbrace{\langle \mathbf{x}_t^{[m],r_1}, \dots, \mathbf{x}_t^{[m],r_N}, \mathbf{x}_t^{[m],\mathbf{O}}, w_t^{[m]} \rangle}_{(N+2)\text{-tuple}} \quad (7)$$

Algorithm 1: PF-UCLT($\mathcal{X}_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{L}_{\text{map}}, r_k$).

- 1: **Transmit** $\{z_t^{r_k,\mathbf{O}}, \Sigma_t^{r_k,\mathbf{O}}, u_t^{r_k}, \mathbf{R}_t^{r_k}, z_t^{r_k,l}$ and $\mathbf{Q}_t^{r_k,l}$ for all $l = 1, \dots, L\}$ to all robots $r_n; n = 1, \dots, N; n \neq k$
 - 2: **Receive** $\{z_t^{r_n,\mathbf{O}}, \Sigma_t^{r_n,\mathbf{O}}, u_t^{r_n}, \mathbf{R}_t^{r_n}, z_t^{r_n,l}$ and $\mathbf{Q}_t^{r_n,l}$ for all $l = 1, \dots, L\}$ such that $n \in \{1, \dots, N\}; n \neq k$
 - 3: $\bar{\mathcal{X}}_t \triangleq \{\langle \bar{\mathbf{x}}_t^{[m]}, \bar{w}_t^{[m]} \rangle\}_{m=1}^M \triangleq \{\langle \bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}}, \bar{w}_t^{[m]} \rangle\}_{m=1}^M = \emptyset$
 - 4: $\mathcal{W}_t \triangleq \{\langle w_t^{[m],r_1}, \dots, w_t^{[m],r_N}, w_t^{[m],\mathbf{O}} \rangle\}_{m=1}^M = \emptyset$
 - 5: **for** $m = 1$ to M **do**
 - 6: **for** $n = 1$ to N **do**
 - 7: $\bar{\mathbf{x}}_t^{[m],r_n} = \text{sample_robot_motion_model}(\mathbf{x}_{t-1}^{[m],r_n}, \mathbf{u}_t^{r_n})$
 - 8: **end for**
 - 9: $\bar{\mathbf{x}}_t^{[m],\mathbf{O}} = \text{sample_object_motion_model}(\mathbf{x}_{t-1}^{[m],\mathbf{O}})$
 - 10: **end for**
 - 11: **for** $m = 1$ to M **do**
 - 12: **for** $n = 1$ to N **do**
 - 13: $w_t^{[m],r_n} \propto \prod_{l=1}^L p(\mathbf{z}_t^{r_n,l} | \bar{\mathbf{x}}_t^{[m],r_n}, \mathbf{L}_{\text{map}})$
 - 14: **end for**
 - 15: **end for**
 - 16: **for** $n = 1$ to N **do**
 - 17: $\{\langle \bar{\mathbf{x}}_t^{[m],r_n}, w_t^{[m],r_n} \rangle\}_{m=1}^M \leftarrow$
 $\text{sort_descend}(\{\langle \bar{\mathbf{x}}_t^{[m],r_n}, w_t^{[m],r_n} \rangle\}_{m=1}^M)$
 w.r.t. $\{w_t^{[m],r_n}\}_{m=1}^M$
 - 18: **end for**
 - 19: **for** $m = 1$ to M **do**
 - 20: $m^* = \arg \max_{m' \in [m:M]} \prod_{n=1}^N p(\mathbf{z}_t^{r_n,\mathbf{O}} | \bar{\mathbf{x}}_t^{[m],r_n}, \bar{\mathbf{x}}_t^{[m'],\mathbf{O}})$
 - 21: **Swap** $\bar{\mathbf{x}}_t^{[m],\mathbf{O}}, \bar{\mathbf{x}}_t^{[m^*],\mathbf{O}}$
 - 22: $w_t^{[m],\mathbf{O}} \propto \prod_{n=1}^N p(\mathbf{z}_t^{r_n,\mathbf{O}} | \bar{\mathbf{x}}_t^{[m],r_n}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}})$
 - 23: $\bar{w}_t^{[m]} = w_t^{[m],\mathbf{O}} \prod_{n=1}^N w_t^{[m],r_n}$
 - 24: **end for**
 - 25: normalize $\{\bar{w}_t^{[m]}\}_{m=1}^M$
 - 26: $\mathcal{X}_t = \text{resample}(\bar{\mathcal{X}}_t)$
 - 27: **return** \mathcal{X}_t
-

where the first N elements of the $(N+2)$ -tuple, henceforth designated as the robot sub-particles of the m th particle, form an N -tuple $\langle \mathbf{x}_t^{[m],r_1}, \dots, \mathbf{x}_t^{[m],r_N} \rangle$ representing the state hypothesis of the robots r_1, \dots, r_N . The $(N+1)$ th element $\mathbf{x}_t^{[m],\mathbf{O}}$, henceforth designated as the object sub-particle of the m th particle, represents the state hypothesis of the tracked object \mathbf{O} . The last element $w_t^{[m]}$ represents the weight of the full m th particle. Similar sub-particle notations for temporary particles or particles at timestep $t-1$ will follow.

We now expand (5) and (6) to further facilitate Algorithm 1 description. Using the concept of sub-particles and the variable definitions in (1) and (2), we can expand the prediction step (5) as

$$\begin{aligned} \bar{\mathbf{x}}_t^{[m]} &\sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{[m]}, \mathbf{u}_t) \\ \Rightarrow \langle \bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}} \rangle &\sim p(\mathcal{L}_t^{r_1}, \dots, \mathcal{L}_t^{r_N}, \mathbf{O}_t | \\ &\langle \mathbf{x}_{t-1}^{[m],r_1}, \dots, \mathbf{x}_{t-1}^{[m],r_N}, \mathbf{x}_{t-1}^{[m],\mathbf{O}} \rangle, \\ &\mathbf{u}_t^{r_1}, \dots, \mathbf{u}_t^{r_N}). \end{aligned} \quad (8)$$

Property 1: Since all the control measurements are interoceptive, i.e., they concern only the internal measurements of each individual robot and are independent from each other, the prediction step of each sub-particle can be done separately.

Using *Property 1*, (8) can be written as the following set of equations:

$$\begin{aligned} \bar{\mathbf{x}}_t^{[m],r_1} &\sim p(\mathcal{L}_t^{r_1} | \mathbf{x}_{t-1}^{[m],r_1}, \mathbf{u}_t^{r_1}) \\ &\vdots \\ \bar{\mathbf{x}}_t^{[m],r_N} &\sim p(\mathcal{L}_t^{r_N} | \mathbf{x}_{t-1}^{[m],r_N}, \mathbf{u}_t^{r_N}) \\ \bar{\mathbf{x}}_t^{[m],\mathbf{O}} &\sim p(\mathbf{O}_t | \mathbf{x}_{t-1}^{[m],\mathbf{O}}). \end{aligned} \quad (9)$$

Using (1) and (3), we can expand (6) as

$$\begin{aligned} \bar{w}_t^{[m]} &\propto p(\mathbf{z}_t | \bar{\mathbf{x}}_t^{[m]}, \mathbf{L}_{\text{map}}) \\ &\propto p(\mathbf{z}_t^{r_1,1}, \dots, \mathbf{z}_t^{r_1,L}, \dots, \mathbf{z}_t^{r_N,1}, \dots, \mathbf{z}_t^{r_N,L}, \\ &\mathbf{z}_t^{r_1,\mathbf{O}}, \dots, \mathbf{z}_t^{r_N,\mathbf{O}} | \langle \bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}} \rangle, \mathbf{L}_{\text{map}}) \\ &\propto \prod_{n=1}^N \prod_{l=1}^L p(\mathbf{z}_t^{r_n,l} | \langle \bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}} \rangle, \mathbf{L}_{\text{map}}) \\ &\quad \prod_{n=1}^N p(\mathbf{z}_t^{r_n,\mathbf{O}} | \langle \bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}} \rangle, \mathbf{L}_{\text{map}}) \\ &\propto \prod_{n=1}^N \prod_{l=1}^L p(\mathbf{z}_t^{r_n,l} | \bar{\mathbf{x}}_t^{[m],r_n}, \mathbf{L}_{\text{map}}) \\ &\quad \prod_{n=1}^N p(\mathbf{z}_t^{r_n,\mathbf{O}} | \bar{\mathbf{x}}_t^{[m],r_n}, \bar{\mathbf{x}}_t^{[m],\mathbf{O}}). \end{aligned} \quad (10)$$

The third proportionality equation in (10) is obtained from the second using the following property of conditional independence.

Property 2: The observation measurements of all the static landmarks is independent of the observation measurements of the moving object, given the predicted poses of the observing robots and the predicted position of the tracked object.

The final proportionality equation in (10) is obtained from the third using the property of mutual independence for some (and not all) of the involved variables. These properties are as follows.

Property 3: The observation measurement for a given static landmark made by any robot r_n is dependent only on the predicted pose of the robot r_n and the fixed position of that static landmark. It is independent of the predicted poses of all the other robots, the fixed positions of all the other static landmarks and the predicted position of the tracked object.

Property 4: The observation measurement of the tracked object made by any robot r_n depends only on the predicted pose of the robot r_n and the predicted position of the tracked object. It is independent of the predicted poses of all the other robots and the fixed positions of all the known static landmarks.

Using the sub-particle concept, introduced notations and equations [(9) and (10)], we further describe Algorithm 1.

The algorithm itself is a recursive predict–update loop, an instance of which is required to run on each robot in the team. Furthermore, in the description, we assume that Algorithm 1 runs on the robot r_k . The algorithm’s first input is \mathcal{X}_{t-1} , the particle set returned from the immediate preceding iteration of the algorithm. The other inputs, as defined previously in this section, are \mathbf{u}_t , \mathbf{z}_t , \mathbf{L}_{map} , and r_k .

In lines 1, 2 of Algorithm 1, the robot r_k transmits its control and observation measurements to all the other robots in the team and receives the same information from whichever teammate possible. Note that our method is not decentralized from an information viewpoint; only from a computational viewpoint. This means that every robot expects to receive measurements from all its teammates and then locally performs the full-state estimation method. Even though computation is replicated at each robot, it should not be confused with a multicentralized approach. In a multicentralized approach, each robot or agent running the algorithm replicates the exact same computation and results in the exact same estimate, which could essentially be run on a central station and the results broadcasted back to the robots. In our computationally decentralized approach, it is not necessary that all robots have the exact same measurements at every timestep. This is quite possible in situations of communication delays and failures between one or more robots in the team but not among the other robots. Therefore, even though the computation is replicated, the measurements at each robot and the estimates can be different.

Lines 3, 4 define temporary sets $\bar{\mathcal{X}}_t$ and \mathcal{W}_t , both initialized as empty sets. Set $\bar{\mathcal{X}}_t$ is a temporary particle set, where a temporary particle is an $(N + 2)$ -tuple as described in (7). Set \mathcal{W}_t is defined as $\{\langle w_t^{[m],r_1}, \dots, w_t^{[m],r_N}, w_t^{[m],o} \rangle\}_{m=1}^M$ to hold temporary weight values of sub-particles corresponding to the robots and tracked object in the set $\bar{\mathcal{X}}_t$.

In lines 5–10 of Algorithm 1, $\bar{\mathcal{X}}_t$ stores, after computing, the predicted values of all particles using the input particle set \mathcal{X}_{t-1} . This is done by separately incorporating the odometry measurements of individual robots on the robot sub-particles $\mathbf{x}_{t-1}^{[m],r_1}, \dots, \mathbf{x}_{t-1}^{[m],r_N}$ and applying a user-defined motion model on the object sub-particle $\mathbf{x}_{t-1}^{[m],o}$. The separate prediction of the sub-particles is according to (9) and *Property 1*, as stated previously.

Lines 11–24 present the core of Algorithm 1. The fusion of all correlated observation measurements is performed here. Our weighting mechanism (lines 11–24) associates weights to the predicted particles without modifying its state hypothesis components as per (10). To this effect, an ingenious rearranging technique of sub-particles is used which eliminates the need of an exponentially high number of particles to maintain a given level of accuracy in the state estimates, thereby avoiding the particle deprivation problem. This is further described in detail.

Lines 11–15: for each robot r_n , its predicted sub-particle $\bar{\mathbf{x}}_t^{[m],r_n}$ is separately assigned a weight $w_t^{[m],r_n}$ proportional to $\prod_{l=1}^L p(z_t^{r_n,l} | \bar{\mathbf{x}}_t^{[m],r_n}, \mathbf{L}_{\text{map}})$, which corresponds to their observation measurements of the static landmarks. The separate weighting of the robot sub-particles at this stage is due to *Property 3*, as introduced previously.

In lines 16–18, each robot’s predicted sub-particle set $\{\bar{\mathbf{x}}_t^{[m],r_n}\}_{m=1}^M$ is sorted in descending order w.r.t. its temporary weight set $\{w_t^{[m],r_n}\}_{m=1}^M$, generated in lines 11–15. This means that the highest weighted sub-particle, separately for each robot, is assigned the particle index 1, the second-highest weighted sub-particle is assigned the particle index 2, and so on. It is crucial to understand why this is done. Given that a robot’s “good” predicted sub-particles approximate better the correct posterior of that robot’s state than its “bad” sub-particles, consider the following situation. If we were using the straightforward particle weighting mechanism, i.e., the first proportionality equation in (10), it would be very likely that a robot’s “good” sub-particle would get coupled with a “bad” sub-particle from another robot when computing the weight of a particle leading to a lower overall weight for that particle, even though a sub-particle in it was “good.” Consequently, that particle has a much higher chance to get eliminated in the resampling step of the PF. It is due to this loss of “good” sub-particles and eventually the loss of “good” particles that an exponentially high number of particles is required, so as not to fall into the particle deprivation problem when using the first proportionality equation of (10) directly. By sorting the individual robot’s sub-particles and then grouping the “best with the best” robots’ sub-particles, we make sure that the “good” sub-particles from every robot have a much higher chance to survive after the resampling step of the PF and the particle deprivation problem is solved partially [because up to this stage in the algorithm we still have not considered the object observation measurements in (10)]. Furthermore, recall that any robot r_n ’s sub-particle set $\{\bar{\mathbf{x}}_t^{[m],r_n}\}_{m=1}^M$ was predicted and weighted independently of all the other robots’ sub-particle sets. This implies that rearranging the robot sub-particle sets does not modify the distribution represented by the predicted particle set $\bar{\mathcal{X}}_t$.

Lines 19–24 incorporate the observation measurements of the tracked object made by the robots in (10).

Following from *Property 1*, the prediction of the tracked object’s sub-particles $\{\bar{\mathbf{x}}_t^{[1],\circ}, \dots, \bar{\mathbf{x}}_t^{[M],\circ}\}$ was performed independently of the robots’ sub-particles prediction. This enables the freedom to rearrange the tracked object’s sub-particles before incorporating the object observation measurements in the particle’s weight. We perform this rearrangement of the object’s sub-particles to completely solve the problem of particle deprivation (recall that earlier it was partially solved by rearranging only the robot’s sub-particles). Note that even in this rearrangement step neither any new sub-particles will be created nor any of the existing ones will be destroyed, therefore ensuring that the distribution represented by the predicted particle set $\bar{\mathcal{X}}_t$ remains unaltered. This rearrangement is performed as follows. For the m th previously rearranged set of the robots’ predicted sub-particles $\{\bar{\mathbf{x}}_t^{[m],r_1}, \dots, \bar{\mathbf{x}}_t^{[m],r_N}\}$, obtained after the execution of lines 16–18, we find m^* in line 20. Here, m^* is the particle index ranging from m to M at which the weight contribution $\prod_{n=1}^N p(z_t^{r_n,\circ} | \bar{\mathbf{x}}_t^{[m],r_n}, \bar{\mathbf{x}}_t^{[m^*],\circ})$ by the object sub-particle $\bar{\mathbf{x}}_t^{[m^*],\circ}$ to the m th particle’s weight is maximum for all possible indices in $[m : M]$. Line 20 swaps $\bar{\mathbf{x}}_t^{[m],\circ}$ with $\bar{\mathbf{x}}_t^{[m^*],\circ}$ ensuring that the m^* th object sub-particle is grouped with the m th set of robot sub-particles, whereas the m th object sub-particle is saved at a later index. This way none of the object sub-particles get deleted or replicated. Lines 22, 23 then incorporate the weight contribution of the selected object sub-particle into the particle weight $\bar{w}_t^{[m]}$. Note that the computation of the object sub-particle’s contribution to the particle’s weight due the object observation measurement follows from *Property 4*. The particle weight computation in line 23 is in accordance with the final expression obtained in (10). The full rearrangement is achieved by executing lines 20–23 over all predicted sub-particle sets $[1 : M]$. The rearrangement of the tracked object’s sub-particles, in a way such that the “good” object sub-particle gets coupled with the “good” sub-particle set of the robots, ensures that the particle deprivation problem is completely addressed. This concludes the weight update step of Algorithm 1.

Lastly, line 25 performs the particle weight normalization followed by the resampling step in line 26. The resampling of the particle set $\bar{\mathcal{X}}_t$ is performed to obtain \mathcal{X}_t , which is eventually returned as the final output of this algorithm. This can be done using standard available methods, e.g., low variance resampling [42].

C. Space and Time Complexity Analysis of PF-UCLT

1) *Space Complexity*: Assuming that N is the total number of robots in the team, M is the number of particles required to obtain a given accuracy level by a PF-based method for a single robot localization and the number of tracked objects $O = 1$, the worst-case space complexity of a standard PF-based method for the online UCLT problem will be $\mathcal{O}((N + 1)M^{N+1})$. Algorithm 1 limits the worst-case space complexity to $\mathcal{O}((N + 1)M)$, since only M particles, with $(N + 1)$ sub-particles in each particle, are required in order to

maintain the same accuracy level in the state estimates. We also show this through extensive experimental results. This feature of the PF-UCLT algorithm makes it scalable to a large number of robots, even when the robots may have low memory capacity.

2) *Time Complexity*: The worst-case time complexity (WCTC) for a standard PF-based method would be $\mathcal{O}(M^{N+1})$, i.e., growing exponentially with the number of robots. In our solution, Algorithm 1, the WCTC of the weight update step due to the observation measurements of the fixed landmarks is $\mathcal{O}(NM)$. The WCTC of the sorting performed in lines 16–18 (assuming merge sort) will be $\mathcal{O}(NM \log M)$. For the weight update step due to the tracked object’s observation measurements, the WCTC will be $\mathcal{O}(NM^2)$. Summing the WCTC of all the individual steps and considering only the highest order term, while assuming that $M \gg (N + 1)$, the WCTC of the complete weight update process in Algorithm 1 will be $\mathcal{O}(NM^2)$. This is linear in terms of the number of robots N . Therefore, scaling Algorithm 1 to a larger number of robots is feasible.

IV. TESTBED, DATASETS, BENCHMARKS, AND EXPERIMENTS

A. Testbed and Implementation

We applied the PF-UCLT algorithm to the robot soccer scenario. Our testbed is the RoboCup Middle Sized League (MSL). In MSL, all sensors, actuators, and computation power must be on-board the robots. No remote control, sensing, or processing is allowed. However, interrobot communication is accepted. A FIFA standard size 5 ball of a prespecified color is used for the competition. In such a scenario, where all robots are fully autonomous, the self-localization of each robot and the knowledge of the soccer ball’s position is essential. Adding further to this challenge is the large field size, field symmetry, occlusions caused by the presence of other robots, fast motion of the ball, as well as the limited range of the sensors on each of the robots. Therefore, robotic soccer is an ideal testbed for the implementation and evaluation of the PF-UCLT algorithm presented in this paper. The implementation was done within the robot operating system (ROS) framework and the source code is available here.²

B. Datasets and Ground Truth

We present results on 1) an extensive set of simulated soccer-robots datasets and 2) a real soccer-robots dataset. Each dataset type is described in more detail at the beginning of the corresponding experimental results section. While the ground truth (GT) for simulated datasets is obvious, the real robot dataset also consists of GT 2-D positions of the robot and 3-D positions of the object (but not the robot orientations). This GT was obtained using an external stereo camera pair for durations when the robots or the ball were not occluded. Error metrics, as described below, make use of these GT values. Further details of the GT system, which has an average error of approx. 2 cm within a range of 9 m from the GT system’s stereo baseline, can be obtained from [46].

²https://github.com/guilhermelawless/pfuclt_omni_dataset

C. Metrics

The error, at each timestep, in the global estimated position of the ball is computed as the Euclidean distance between the ball's estimated 3-D position (by the proposed or other comparable methods) and the corresponding GT estimate for the same. The robot localization error, at each timestep, is computed as the Euclidean distance between the estimated 2-D position and its corresponding GT values. The GT system provides only the 2-D GT positions of the robots, and not the GT orientations of the robots. In the next sections, we describe the simulation and real robot experimental results using these error metrics.

V. SIMULATION EXPERIMENTS

We created a simulated dataset generator for the MSL soccer robot scenario.³ For a desired number of robots, size of playing field, number of landmarks, etc., it can generate randomized datasets that simulate 1) robot motions, including translation and rotations, on the 2-D playing field while avoiding collisions, and 2) arbitrary ball motion in 3-D space within the prespecified field dimensions. The generated datasets include the following.

- 1) Odometry measurements at 33 Hz with artificially added zero mean Gaussian noise [42] for each robot.
- 2) Landmark and ball measurements (x, y, z in each robot's local frame) at 16.5 Hz with added zero mean Gaussian noise (maximum $\sigma^2 = 0.64$ in X - and Y -directions and maximum $\sigma^2 = 0.25$ in the Z -direction in the robot's local frame).
- 3) GT poses of the robots and the ball. The observation measurements also keep a separate record of possible occlusions so that the dataset user could decide whether to use a measurement with or without considering occlusions.

All simulation experiments described further were performed using a field size of 9 m \times 12 m and 10 landmarks. The user-defined object motion model is as follows. The prediction step in line 9 of Algorithm 1 was made by simply adding a zero mean Gaussian acceleration to the object sub-particles. We explicitly avoided a separate computation of the object velocity in simulation, contrary to what we do in the case of real robot experiments. Since the simulation experiments focus on evaluating computational and space complexity of Algorithm 1 as well as the trend of estimation accuracies w.r.t. the number of robots, it was important to avoid any additional computational or space-expensive step that may bias the results. All simulation experiments were made on a laptop computer ASUS K550J with CPU Intel Core i7-4710HQ (2.50 GHz up to 3.50 GHz) and 8 GB RAM. 64-bit Ubuntu 14.04 LTS was used as the operating system.

A. Proof of Concept

The goal of this experiment is to show that performing cooperative localization through a mutually observed common object improves both the localization estimates of the robots and the tracked position of the object in comparison with the case where

each robot is localized independently from the other robots and object position estimation is accomplished by fusing the object position estimates from every robot. To this end, we performed the following experiment on a configuration of four robots and one tracked object (ball) in simulation. After generating an 8-min-long simulated dataset for this configuration, we applied a PF-based single-robot localization and object tracking method separately on each robot. This is also equivalent to a naïve version of the PF-UCLT Algorithm 1, i.e., with $N = 1$, $O = 1$ and without any sub-particle rearrangements as suggested in Algorithm 1. The 3-D position of the ball is then estimated using the following different fusion methods, where the left-superscript r_n denotes the robot running its separate PF.

- 1) Average of the ball position estimates from all robots as $\frac{1}{N} \sum_{n=1}^N r_n \mathbf{O}_t$, where $r_n \mathbf{O}_t$ denotes the ball position estimated by robot r_n .
- 2) Weighted average of the ball position estimates from all robots as $\frac{1}{\sum_{n=1}^N r_n \mathbf{W}_t} \sum_{n=1}^N r_n \mathbf{W}_t r_n \mathbf{O}_t$, where $r_n \mathbf{W}_t$ is the weight of the ball position estimate by robot r_n and is calculated as the sum of weights of the particles $r_n \mathbf{W}_t = \sum_{m=1}^M r_n w_t^{[m]}$.
- 3) Weighted average as above but the weight of the ball position estimate by robot r_n is calculated as the trace of that robot's most recent ball observation measurement information, i.e., $r_n \mathbf{W}_t = \text{Tr}(\Sigma_t^{r_n, \mathbf{o}^{-1}})$.
- 4) Estimate from the robot with maximum $r_n \mathbf{W}_t$ (two separate methods, each using one of the above methods to calculate $r_n \mathbf{W}_t$).

Subsequently, to perform cooperative localization and tracking, we ran the PF-UCLT Algorithm 1 with $N = 4$ and $O = 1$ on this dataset. The separate PFs (on each robot) and the PF-UCLT Algorithm 1 were run ten times each on the same dataset. In this dataset, the landmark observation range was limited to 2 m for each robot, whereas the ball observation range was unlimited (but within the field dimensions) and without considering occlusions. This was done to ensure that the robots often lose sight of the landmarks, thereby allowing us to clearly verify whether or not the mutually observed tracked object helps reduce the uncertainty in the robot pose estimates. The number of particles for both separate PFs and PF-UCLT was set to $M = 300$. However, recall that the PF-UCLT method requires sub-particles that grow linearly with the number of robots N (see Fig. 1 for particle structure).

The box plots in Fig. 2 summarize the results of this experiment over all the runs. It is evident that Robot 2 and Robot 3 had extremely poor localization when performing their separate PFs for robot localization and object tracking. This is due to observing very few or even no landmarks at times, and only for very short time intervals. However, the pose estimates of these robots substantially improve when using PF-UCLT. At the same time, there was no significant change in the localization estimates of Robot 1 and Robot 4, which were already well localized using their separate PFs. We also verify that compared to any of the previously mentioned methods to fuse the ball position estimates in the case of separate PFs, the PF-UCLT method results in higher accuracy.

³https://github.com/guilhermelawless/randgen_omni_dataset/

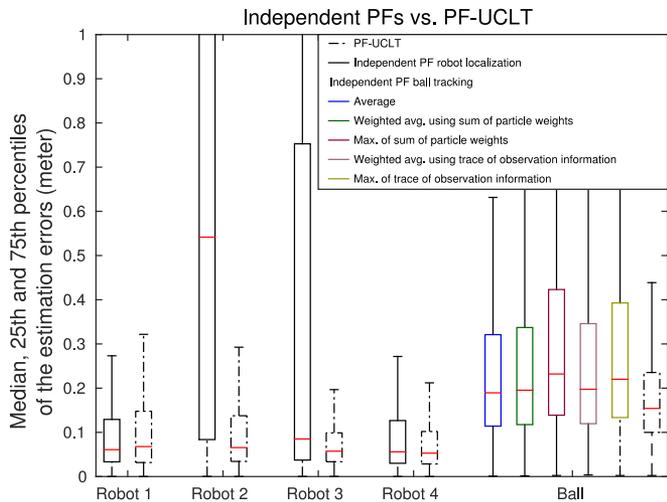


Fig. 2. Comparison of independent PFs for single-robot localization and object tracking with PF-UCLT. Solid box plots in black represent errors in robot localization estimates obtained through the independent PFs, solid box plots in color represent errors in ball position estimates obtained through ad hoc fusion techniques on the independent PFs, and dashed-line box plots represent the errors in the estimates obtained through the PF-UCLT method.

B. Scalability

The goal of this set of experiments is twofold. First, we will show that as the number of robots N grows in the PF-UCLT Algorithm 1, the growth of computational time is only linear. Second, we will also show that as N grows, we do not require an increase in the number of particles, but only a linear increase in the number of sub-particles in order to maintain an approximately constant accuracy level in the state estimates. For each of the two goals, we performed separate experiments as described in the sections below.

1) *Computation Time Versus Number of Robots*: For this goal we performed the following subexperiments.

- Subexperiment A*: Two separate runs on three different simulated datasets for each of the following robot configurations: $N = 2, N = 3, \dots, N = 10$. For all datasets $O = 1$. In this subexperiment, we generated datasets with unbounded observation range for the landmarks and the ball from the observing robots without considering occlusions (by ignoring occlusion information). The range in this and the next subexperiment is unbounded due to the goal of analyzing the trend of computation time of the algorithm. The higher the range, the more measurements are processed in every iteration of the algorithm, thus leading to worst-case computation time situations.
- Subexperiment B*: Same as subexperiment A but with occlusions. Meaning, if a landmark or the tracked object is occluded from the robot due to the presence of another physical entity in the line of sight, its simulated measurement is discarded.

Plots in Fig. 3 confirm that the average computation time for one iteration of the PF-UCLT Algorithm 1 grows linearly w.r.t. the number of robots for both subexperiments. The linear growth in computation time is in fact due to the linear growth in the number of landmark and ball observations as the number of

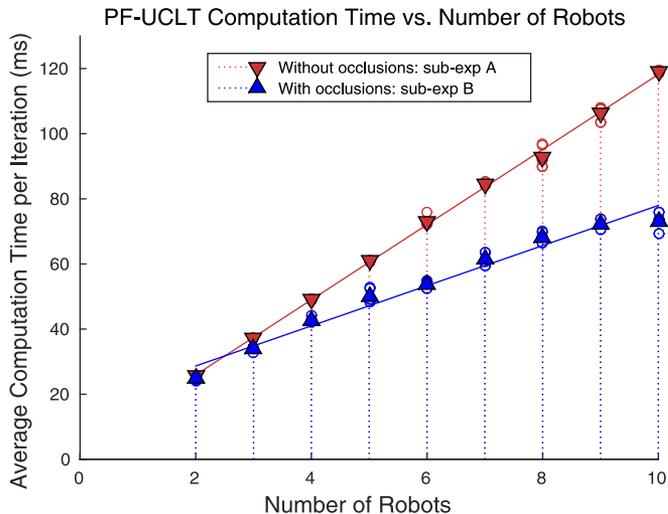


Fig. 3. Average computation time growth w.r.t. the increase in the number of robots. Multiple points on the Y-axis denote multiple runs (2) and multiple datasets (3) for a given number of robots.

robots increases. However, occlusions cause these observations to drop by a near-constant factor per robot which explains the smaller slope for subexperiment B. Moreover, a further increase in the number of robots could itself cause the occlusion rate to increase. Hence, we see that for the ten robot configuration, the computation times tend to grow sublinearly in subexperiment B. Note that for both subexperiments, we kept the number of particles constant at 250. However, since a particle consists of $N + O$ subparticles (see Fig. 1 for particle structure), for every additional robot an additional subparticle set is used which causes the memory requirement to increase linearly with N (see Section III-C for the space complexity analysis).

2) *State Estimate Accuracy Versus Number of Robots*: For this goal, we performed the following experiment: Two separate runs on ten different simulated datasets for each of the following robot configurations: $N = 2, N = 3, \dots, N = 10$. For all datasets $O = 1$. In each configuration, the maximum observation range for the landmarks and the ball was set to 3.0 m and occlusions were considered. The lower observation range is not only to closely mimic the real robot scenario, which is described in the next section, but also to ensure that the estimation accuracy is due to the cooperative nature of the algorithm and not simply due to having multiple landmark observations. The plots in Fig. 4 show the trend of accuracy in the state estimation w.r.t. the number of robots as well as the computation time growth for the sake of completeness. The plots in Fig. 4(a) show the robot localization errors for all the robots, the two runs and the ten datasets for a given configuration of robots, combined into a single box plot. For a given configuration, we also combine the errors in the global position of the ball over all the runs and datasets into a single box plot to facilitate comparison between different configurations. Note that the errors in the global position of the ball are excluded when it is out of the field of view of all the robots leading to a nonconvergent object subparticle set. This happens less as the number of robots increases. An additional visibility fraction plot in Fig. 4(a) denotes the ratio

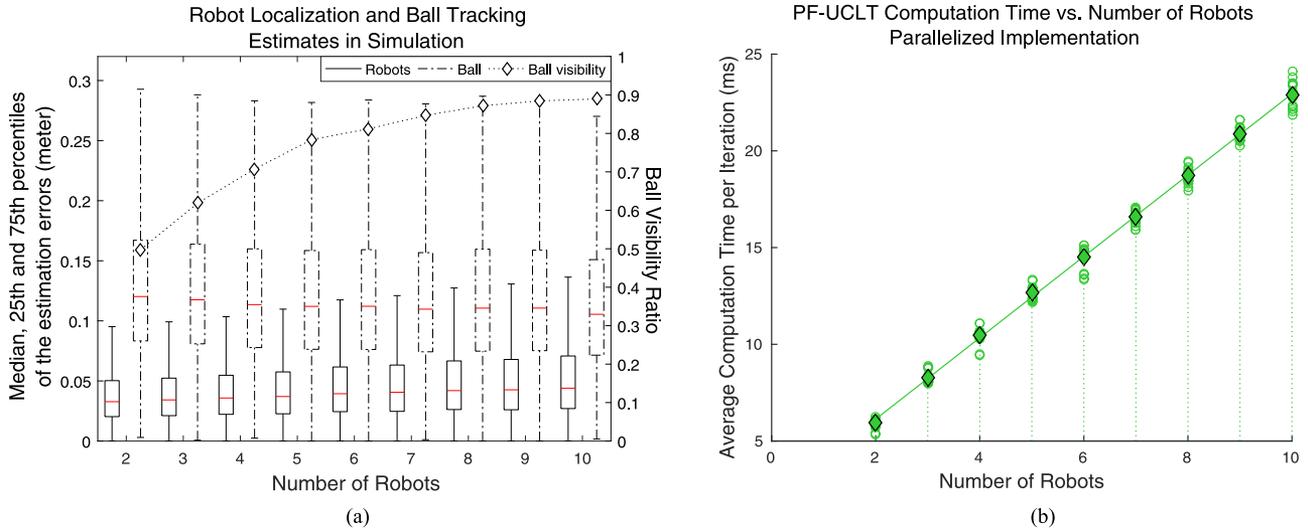


Fig. 4. Trend of estimation accuracy and computation times in the experiment described in Section V-B1. Note that here the computation times are in general much lower than the experiments described in Section V-B1. This is not only due to a highly restricted observation range (3 m) causing fewer observations to be processed in the update step of our algorithm, but also due to a parallelized implementation of our method over six CPU threads on the computer described at the beginning of Section V. (a) State estimation accuracy w.r.t. the number of robots. (b) Computation time w.r.t. the number of robots.

between the time the ball could be tracked and the total dataset duration.

For all configurations in this experiment also we kept the number of particles constant at 250. However, since a particle consists of $N + O$ subparticles (see Fig. 1 for particle structure), for every additional robot an additional subparticle set is used. This causes the memory requirement to increase linearly with the number of robots N (see Section III-C for the space complexity analysis). The trend in Fig. 4(a) confirms that for all configurations, the PF-UCLT algorithm was able to maintain the robot localization estimate errors at a median of approx. 3–4 cm and the errors in global ball position estimate at a median of approx. 10–12 cm (this implies even lower errors in the ball position estimate in a robot’s local reference frame). The distribution of errors is well within the 2-sigma error bounds of the actual measurement errors. We also experimentally established that the PF-UCLT Algorithm 1 does not require an increase in the number of particles, and needs only a linear increase in the number of subparticles w.r.t. the number of robots (therefore, only a linear increase in the memory requirements) in order to maintain an approximately constant accuracy level in all the state estimates (robots and object).

There is, however, a slight but noticeable increase in the robot localization errors as the number of robot grows and an initial slight decrease in the ball position errors which then saturates. This is due to our dataset implementation strategy. The field size is constant over all robot configurations. Therefore, the higher the number of robots, the more is the field cluttered with robots. This leads to extended periods of time when some robots observe neither a landmark nor the ball due to occlusions. During that period, those robots are deemed to degrade their localization estimate. Without any observation, their corresponding subparticles play no role in the update steps of Algorithm 1. On the other hand, with an increasing number of robots there is an

increasing chance that the ball is in the field of view of more robots until a certain number, after which the additional robots are simply beyond the observation range of the ball and/or the ball is occluded from them. Therefore, the ball position estimate slightly improves before saturating w.r.t. the number of robots. Note that the aforementioned degradation in the pose estimates of the robots does not affect the ball position estimates. In fact, in the time periods when the robots lose their localization they are also not observing the ball (and landmarks), and thus the ball position estimation is unaffected.

VI. REAL ROBOT EXPERIMENTS

For the real robot experiments, we used a prerecorded dataset from a team of four MSL robots (named as OMNI1–OMNI4). These robots are three-wheeled omnidirectional robot soccer platforms (see [47] and [48] for a detailed description of these robots as well as the datasets). The main sensor on the robot is a dioptic vision system consisting of a camera providing omnidirectional vision through a fish-eye lens. This vision system, pointing downward to the ground, enables the robots to detect objects of interest including the ball and landmarks up to ~ 3.5 m. In these experiments, the software was run on a computer with CPU Quad Core Intel(R) Core(TM) i5 CPU 750 @ 2.67 GHz and 8 GB RAM.

In the experiments described further in this section, the PF-UCLT algorithm is implemented on a part of this dataset. We used the observation measurements of six landmarks (out of ten), the orange ball observation measurements and the odometry measurements of all four robots for a period corresponding to approximately 6 min of real-time data acquisition. This is done in order to be consistent with the part of the dataset used by other methods in our previous work [5] which we compare against the PF-UCLT algorithm. Note that the ball measurements from each robot are quite noisy. A separate characterization of the sensor

noise, using the GT positions of the robots and the ball, shows that the mean of the noise in the range component of these measurements for all the robots is approx. -15 cm and the standard deviation ranges from 30 to 70 cm. Since the GT orientation of the robot is not available, this noise characterization is done only for the range component of the ball measurements. The noise is due to multiple sources of error including inaccuracies in color segmentation for the ball, delays in the system (between image acquisition time and processing, etc.) as well as partial occlusions to the ball. One can also use this noise characterization to assist in designing a filter for the ball position estimation. However, we did not do so to be fairly comparable with the other methods in [5]. More details can also be obtained from the wiki^{4,5} page of the dataset.

In our previous work [5] with EKF and MMG-O (MMG-O performs nonlinear least squares minimization), object velocity was considered as a component of the estimated state. Doing so in a PF-based method would substantially increase the computational burden. In the PF-UCLT algorithm we assumed that the velocity of the tracked object is neither directly measurable by any exteroceptive sensor nor considered as a component of the estimated state. Note that even though we do not explicitly estimate the object velocity in the proposed approach, our comparison for the rest of the estimated state components with the aforementioned approaches is fair. This is because all the methods use exactly the same raw sensor measurements. As an implementation strategy, what we did for PF-UCLT algorithm here is as follows. We computed the ball velocity at every timestep by performing a linear regression over the tracked ball positions during a fixed number of previous timesteps. Subsequently, this velocity, along with a zero mean Gaussian acceleration noise, is used for predicting the object subparticles in line 9 of Algorithm 1. Thus, the user-defined object motion model in the real robot case is constant velocity with zero-mean Gaussian acceleration. As an extra note, the success of all our experiments also shows that the PF-UCLT algorithm is able to accommodate different kinds of object motion models.

A. Comparison With Other Methods

The goal of this experiment is to compare PF-UCLT with other aforementioned approaches in terms of estimation accuracy. The robots are henceforth mentioned as OMNI1–OMNI4. Algorithm 1 is implemented on OMNI1. In addition to its own measurement data, OMNI1 receives odometry, landmarks, and object observation measurements from OMNI2, OMNI3, and OMNI4 (lines 1, 2 of Algorithm 1). In order to emulate communication between the two robots, at every iteration of the algorithm on OMNI1, we select the closest timestamped measurements from the data logs of other robots.

Fig. 5 presents the results of this experiment. The estimation errors are calculated as described in Section IV-C. In comparison with the MMG-O approach, we observe that the PF-UCLT method results in slightly less accurate localization of all robots.

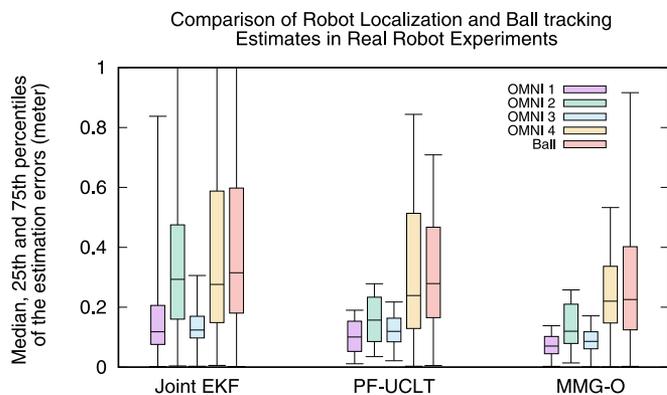


Fig. 5. Real Robot Experiment: Estimation error comparison between online PF-UCLT, offline MMG-O, and online EKF-based method.

The reason lies in the fact that PF-UCLT is an online approach where the measurement data at each timestep is incorporated only once. In contrast, MMG-O is an offline batch-processing method that employs an iterative local linearization technique for the least squares error minimization over the whole dataset, and in every such iteration it improves the estimate of the complete trajectory of all the robots and the ball. This enables the MMG-O method to mitigate the effect of noisy measurements, which are abundant in the dataset used for these experiments. In case of the PF-UCLT method, if noisy measurements exist for a significant period of time, estimates start to become poor. However, they recover to better estimates when less noisy measurements arrive. Unlike the offline approach of MMG-O, PF-UCLT is online and therefore not designed to update the previous trajectory at any timestep, hence the mean error of localization tends to be higher in its case when compared to the MMG-O method. Nevertheless, compared to the online EKF-based approach, the PF-UCLT method substantially improves the localization estimates, e.g., $\sim 65\%$ reduction in the mean localization error for OMNI1, $\sim 46\%$ for OMNI2, $\sim 24\%$ for OMNI3, and $\sim 31\%$ for OMNI4.

On the other hand, the PF-UCLT method shows a slightly higher precision in tracking the ball when compared to both the other methods. This is supported by a lower variance of the errors in the ball's global position estimates (0.05 m² for PF-UCLT, 0.334 m² for EKF, and 0.327 m² for MMG-O). Note that a lower mean/median error occurs when estimates are closer to the GT, indicating higher accuracy in tracking, whereas a lower variance of errors indicates higher precision in tracking (also true for the robot localization estimates). The higher precision in tracking the ball, achieved by the PF-UCLT method, is also reflected by its smoother trajectory in the video⁶ accompanying this paper, compared to that in the MMG-O video⁷ accompanying [5]. PF-UCLT's overall superior results over the EKF-based approach is primarily due to a PF's ability to effectively handle multiple modalities in the observations as well as arbitrary changes in

⁶Video link of the PF-UCLT experiments. http://users.isr.ist.utl.pt/~aahmad/TRO2017_video/TRO_17_PFUCLT.mp4

⁷This video can be downloaded from the multimedia tab of the [5]'s weblink <http://ieeexplore.ieee.org/xpl/abstractMultimedia.jsp?arnumber=6631396>

⁴https://github.com/aamirahmad/read_omni_dataset/wiki

⁵https://github.com/aamirahmad/read_omni_dataset/tree/master/docs

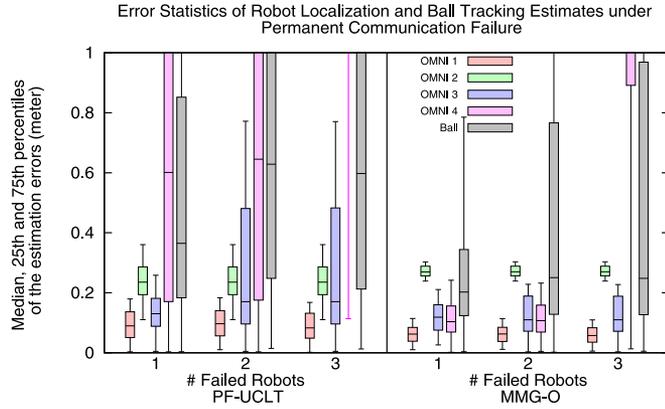


Fig. 6. Real Robot Experiment: Estimation error comparison between online PF-UCLT and offline MMG-O methods in the situation of permanent communication failure.

the object’s motion direction. On the other hand, PF-UCLT’s higher precision than MMG-O is due to the following reason. Least squares are inherently sensitive to outliers unless they are robustified, which was not the case in [5]. The real robot dataset is contaminated with intermittent outliers in the ball observation measurements. The low-pass filtering property of the PF-UCLT method efficiently handles such outliers as it is quite unlikely that a sudden wrong measurement (outlier) would affect a correctly converged object subparticle set.

The average iteration time taken by the PF-UCLT algorithm on the real robot dataset was 0.050 s. As reported in [5], the EKF required 138 s for the full dataset (i.e., 20–30 ms per iteration, showing real-time performance), whereas MMG-O took 20 s for the whole dataset. Note that the computing hardware in [5] was inferior to the one used here for PF-UCLT, as described previously. Even though MMG-O is fast, the design of MMG-O is such that it can only run on the full trajectory of states, thus being a batch process and running offline by design.

B. Robustness to Failure Scenarios

The goal of this set of experiments is to perform robustness analysis of the PF-UCLT algorithm and compare it with the robustness of MMG-O in the presence of communication or vision failures. To this end, a set of 12 experiments (3 situations in 4 different scenarios) were designed and in each experiment both methods were tested on the same portion of real-robot dataset as in the previous experiment. Results of PF-UCLT algorithm implementation on OMNI1 are analyzed. The scenarios include permanent and temporary communication failures as well as permanent and temporary vision failures. Each of the four scenarios consists of three different situations that correspond to an increasing number of failed robots in that scenario. To emulate the communication failure odometry, landmark and object observation measurements corresponding to specific timestamps were excluded from the algorithm’s execution process. To emulate vision loss, only the landmark and object observation measurements were excluded. We further present the results of these experiments grouped according to the failure scenarios.

The plots in Figs. 6–9 help visualize the evolution of estimation errors with the increasing number of failed robots in every

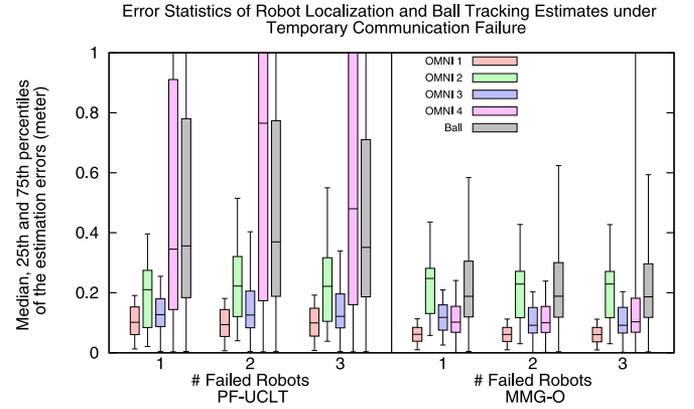


Fig. 7. Real Robot Experiment: Estimation error comparison between online PF-UCLT and offline MMG-O methods in the situation of temporary communication failure.

scenario. Note that the errors are calculated as explained in Section IV-C. For a particular robot, the error is calculated only for the duration which does not correspond to the communication failure time period (during that time period the teammate robot’s localization is irrecoverable). For the ball, as for any of the previous experiments, the errors are calculated only if the ball is present in the field of view of at least one of the robots. Also, the errors are calculated only if the corresponding GT is present and not unavailable due to occlusions from the overhead stereo GT system. Prior to the analysis of results, it is also worth mentioning that in the real robot dataset, the individual robot’s odometry noise and observation measurement noise were much higher for OMNI4 compared to the other robots. High slippage in OMNI4’s motors lead to noisier odometry measurements. The noisy observation measurements are caused by poor color segmentation calibration for its vision system.

1) *Permanent Communication Failure*: The following three situations were experimented under the scenario of permanent communication failure.

One Failed Robot: Communication loss between OMNI2 and OMNI1 [80 s–end of dataset (EOD)].

Two Failed Robots: Communication loss between OMNI2 and OMNI1 (80 s–EOD), OMNI3 and OMNI1 (85 s–EOD).

Three Failed Robots: Communication loss between OMNI2 and OMNI1 (80 s–EOD), OMNI3 and OMNI1 (85 s–EOD), OMNI4 and OMNI1 (92 s–EOD).

The EOD in all the experiments here onward is at ~ 360 s. Plots in Fig. 6 present the results of this scenario.

Primarily, we observe that for all the three situations, the errors in position estimates for the robots and the ball are lower for the MMG-O method in comparison to the PF-UCLT algorithm for reasons explained in the previous section. Although it gains on computational speed and allows real-time execution, PF-UCLT compromises on accuracy to some extent when compared to MMG-O.

For both PF-UCLT and MMG-O, we observe from the plots in Fig. 6 that the position estimation accuracy of the “disconnected” robots (robots which lose communication with OMNI1) degrades, while the position accuracy of the “connected” robots (robots whose communication with OMNI1 has not failed) is maintained. It is noticeable that OMNI1, on which

the algorithm runs in the case of PF-UCLT or whose self-measurement/observation data is never lost in case of the MMG-O approach, is able to maintain its position estimate accuracy when one or more teammates fail to communicate. This particularly highlights the robustness of both the approaches in the case of permanent communication loss. It is also interesting to observe that when OMNI4 is “disconnected” (loses communication), OMNI1 slightly improves its own position estimate accuracy (mean error reduces from 0.100 to 0.093 m in the case of PF-UCLT and from 0.64 to 0.62 m in the case of MMG-O approach) while OMNI4’s position accuracy reduces drastically (mean error increases from 1.40 to 2.76 m in the case of PF-UCLT and from 0.187 to 2.153 m in the case of MMG-O approach). The main reason behind it is that OMNI4’s measurements are extremely noisy. Such a phenomenon highlights a beneficial feature of both approaches. The position estimates of the robots with highly noisy measurements improve quite significantly in the presence of robots with less noisy measurements, while at the same time the position accuracy of the latter robots is only slightly compromised.

The accuracy of the ball’s position estimate degrades with the number of robots losing communication. This is simply due to the fact that with less communications, there are less object observation measurements. The ball is not only completely unobserved by OMNI1 for extended periods of time but also, when visible, far fewer teammates are able to provide the object observation measurements to OMNI1. Therefore, the information gain on the ball’s position estimate is stunted. This eventually leads to an increase in the overall mean error of the ball position estimation w.r.t. an increase in the number of failed robots.

2) *Temporary Communication Failure:* The following three situations were experimented under the scenario of temporary communication failure.

One Failed Robot: Communication loss between OMNI2 and OMNI1 (80–120 s).

Two Failed Robots: Communication loss between OMNI2 and OMNI1 (80–112 s), OMNI3 and OMNI1 (85–120 s).

Three Failed Robots: Communication loss between OMNI2 and OMNI1 (80–112 s), OMNI3 and OMNI1 (85–120 s), OMNI4 and OMNI1 (92–128 s).

Plots in Fig. 7 present the results of this scenario.

Similar to the previous scenario, the mean errors in the position estimates of all the robots and the ball are comparatively lower in the case of MMG-O method than the PF-UCLT, although both methods were able to effectively localize the robots and track the ball. An important observation in this scenario is the stability of errors. With the increase in the number of failed robots, the mean and median errors of all the robots and the ball position estimates change only slightly (0%–10%). This means that both the unified methods, MMG-O and PF-UCLT, were robust to temporary communication failures and were able to recover the temporarily “disconnected” robot’s localization estimates, once they got “reconnected” (regained communication).

3) *Vision Failure Scenarios:* The rest of the two scenarios involve permanent and temporary vision failure with the same situations as previously described in the communication-related scenarios. The only difference in the vision-related failure scenario is that the odometry measurements from the team-

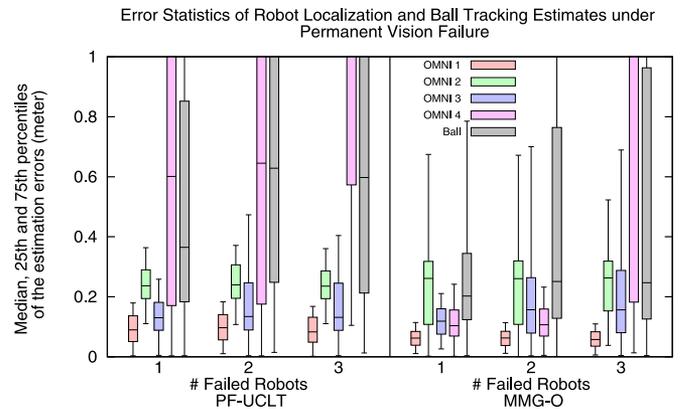


Fig. 8. Real Robot Experiment: Estimation error comparison between online PF-UCLT and offline MMG-O methods in the situation of permanent vision failure.

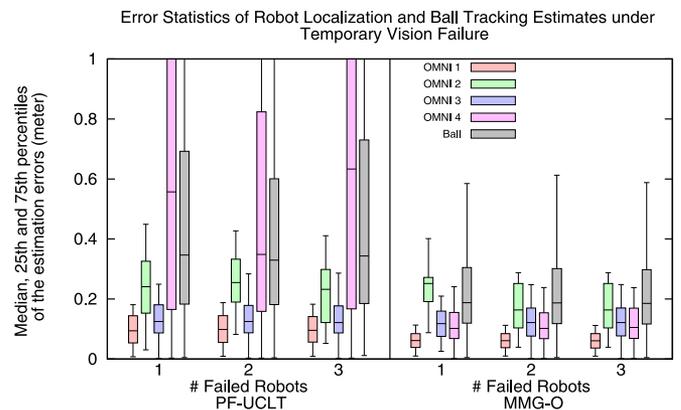


Fig. 9. Real Robot Experiment: Estimation error comparison between online PF-UCLT and offline MMG-O methods in the situation of temporary vision failure.

mate robots were always available. Therefore, contrary to the communication-failure scenarios, the “faulty” robots (i.e., the robots which lose camera vision) in the vision-failure scenarios were still attempted for localization using only odometry and their large estimation errors were considered in the statistics. In the communication-failure scenarios, the disconnected time periods were not considered in the error statistics. Through Figs. 8 and 9, we find that the results in vision failure scenarios are very similar to those of communication failure scenarios with slightly higher position estimation errors in the localization due to the aforementioned issue of having different time periods of the error statistics. Fig. 9 also confirms that our method recovers the “faulty” robot localizations and object tracking estimates after their vision system is restored.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach for unified cooperative robot localization and object tracking (UCLT) in a multi-robot scenario. The approach is based on a PF and designed for real-time applications. After formulating the online UCLT problem, we presented a recursive Bayesian filter solution and then described a standard PF-based method for it. Subsequently,

we introduced our approach, the PF-UCLT method, describing in detail its complete algorithm. In it, a new concept of subparticles and a rearranging technique for them was introduced. Through rigorous experiments on a large number of simulated datasets we showed that the PF-UCLT method

- 1) implicitly performs cooperative localization through a mutually observed common object and significantly improves the localization of the robot and its teammates, which would have been much poorer if using only a map or landmark-based localization for the robots;
- 2) simultaneously, the cooperatively estimated object position is also significantly improved in comparison to a weighted or simple average of individually tracked object positions by each robot;
- 3) reduces the exponentially growing computational and space requirements of a standard PF-based method to linear w.r.t. the number of robots in the team. Consequently, in order to maintain a given level of accuracy in the state estimates it does not require an increase in the number of particles, and needs only a linearly growing number of subparticles w.r.t. the number of robots. This was achieved through a novel subparticle rearranging technique, which in turn was possible by exploiting the underlying properties of conditional and mutual independence of the measurements. This also demonstrates how complexity can be reduced in other PF-based estimation algorithms that tend to be computationally expensive.

Through extensive experiments on a real-robot dataset, we showed that the PF-UCLT method

- 1) performs not only significantly better than an online EKF-based method designed to solve the same problem, but also very similar to an offline nonlinear least squares minimization-based method designed as a full-state trajectory optimizer. Since PF-UCLT is designed as an online estimator for real-time applications, this result is quite significant;
- 2) is robust to communication and sensor failures to the extent that the state estimates of the robots that do not lose their communication or vision remain unaffected in the presence of defective teammate robots. Moreover, the method recovers the state estimates of the defective robots in case their communication or vision gets restored.

Thus, we conclude that our unified framework for cooperative estimation provides unique advantages over ad hoc multiple estimation methods. Through careful exploitation of conditional and mutual independences in measurements we could take advantage of a PF-based estimator to construct such a unified framework. One limitation of our framework is that it currently assumes a known landmark-based map of the environment. However, in scenarios, such as indoor office spaces and factory floors, where building a landmark-based map is easily possible, our unified framework is applicable in a straightforward manner. Additionally, the mutually observed tracked objects in those scenarios must be ascertained beforehand and their models must be available to all robots. Within this context, an interesting extension of our framework would be to dynamically include or exclude mutually observed tracked objects from the estimated state space depending on their visibility periods. Furthermore,

based on their experience, robots could learn which tracked objects are more useful to use within our unified framework. On the other hand, in scenarios, e.g., search and rescue, where it is hard to predefine models for mutually observed tracked objects, our framework could leverage the possibility of using one of the robot teammates as a mutually observed object. Finally, in scenarios of feature-based maps, our method will require careful adaptation. One approach would be to consider either individual features or semantically segmented sets of features as static landmarks. However, the increased state-space dimension would remain a major challenge in such a map representation.

The novelties of our PF-UCLT method also give rise to several new threads of possible future developments. This includes the following:

- 1) considering inter-robot measurements;
- 2) tracking of multiple objects;
- 3) performing cooperative SLAM and object tracking, to name but a few.

Although the PF-UCLT method can be used to track multiple objects, it is not yet scalable w.r.t. the number of objects. The complexity would grow exponentially in such case. Similar complexity issues can be foreseen by adding interrobot measurements to this problem.

APPENDIX

PROOF OF SUBPARTICLE REARRANGEMENT WITH MAXIMUM LIKELIHOOD IN THE PF-UCLT ALGORITHM

In a PF, the empirical density

$$\hat{p}_t^M(\mathbf{x}_t) = \sum_{m=1}^M w_t^{[m]} \delta_{\mathbf{x}_t^{[m]}}(\mathbf{x}_t) \quad (11)$$

is a discrete estimate of the posterior density $p_t(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$, where $\delta_{\mathbf{x}_t^{[m]}}(\mathbf{x}_t)$ denotes the Dirac delta mass located at $\mathbf{x}_t^{[m]}$. M denotes the total number of particles.

In the update step derivation of PF-UCLT, using the properties of conditional and mutual independence and the concept of subparticles, the weight of a particle was obtained as the following product:

$$w_t^{[m]} \propto \prod_{n=1}^N \prod_{l=1}^L p(\mathbf{z}_t^{r_n, l} | \bar{\mathbf{x}}_t^{[m], r_n}, \mathbf{L}_{\text{map}}) \prod_{n=1}^N p(\mathbf{z}_t^{r_n, \circ} | \bar{\mathbf{x}}_t^{[m], r_n}, \bar{\mathbf{x}}_t^{[m], \circ}) \quad (12)$$

where N is the number of robots, L is the number of static and known landmarks, and \circ represents the tracked object. r_n denotes the n th robot.

Let

$$w_t^{[m], r_n} = \prod_{l=1}^L p(\mathbf{z}_t^{r_n, l} | \bar{\mathbf{x}}_t^{[m], r_n}, \mathbf{L}_{\text{map}}) \quad (13)$$

and

$$w_t^{[m], \circ} = \prod_{n=1}^N p(\mathbf{z}_t^{r_n, \circ} | \bar{\mathbf{x}}_t^{[m], r_n}, \bar{\mathbf{x}}_t^{[m], \circ}). \quad (14)$$

Using (12)–(14) we can rewrite the empirical posterior density (11) as

$$\hat{p}_t^M(\mathbf{x}_t) = \sum_{m=1}^M \left(\left(\prod_{n=1}^N w_t^{[m],\tau_n} \delta_{\mathbf{x}_t^{[m],\tau_n}}(\mathbf{x}_t^{\tau_n}) \right) \left(w_t^{[m],\circ} \delta_{\mathbf{x}_t^{[m],\circ}}(\mathbf{x}_t^{\circ}) \right) \right). \quad (15)$$

In Algorithm 1, we introduced a rearranging method for the subparticles which was possible not only due to the mutual and conditional independences in the involved variables but also because the rearrangement did not change the distribution of the predicted particle set. The intuition behind the rearrangement is that grouping the best subparticles provides a higher possibility for the good subparticles and eventually for the better particles to be retained after the resampling step. Furthermore, we also show that for all possible rearrangements of the subparticles, the proposed rearrangement will result in the maximum of the empirical posterior density given by 15. As this density is essentially a sum of products (SOP), where each product is a result of grouped subparticle weights, the assertion that the rearrangement results in the maximum posterior density is true if the following proposition is proved.

Proposition:

Consider a set $\mathbf{A} = \{a_1, a_2, \dots, a_M\}$ and a set $\mathbf{B} = \{b_1, b_2, \dots, b_M\}$, such that

- 1) $a_i, b_i \in \mathbb{R}^+$, where $1 \leq i \leq M$; $i, M \in \mathbb{N}^+$;
- 2) $a_1 \geq a_2 \geq a_3 \geq \dots \geq a_M$;
- 3) $b_1 \geq b_2 \geq b_3 \geq \dots \geq b_M$.

The SOP combination $(a_1 b_1 + a_2 b_2 + \dots + a_M b_M)$ is the maximum for all other possible SOP combinations. The summands of an SOP are obtained by multiplying an element of the set A by any element of the set B , such that each element from both sets is used once and only once.

Proof: Assume the proposition does not hold, and assume $|A| = |B| > 1$ (in the case $|A| = |B| = 1$ there is no choice to be made). Now choose two pairs out of the optimal assignment: (a_i, b_j) and (a_k, b_h) for which $a_i \geq a_k$, but $b_j \leq b_h$. If the proposition does not hold, then such an assignment must exist. The optimal SOP would then be $a_i b_j + a_k b_h + c$, for a certain constant c given by the remaining part of the SOP. Now consider the alternative SOP $a_i b_h + a_k b_j + c$. If the proposition indeed does not hold, the following must be true: $a_i b_j + a_k b_h + c \geq a_i b_h + a_k b_j + c$. However, if we rearrange the inequality, we find $(a_i - a_k)(b_j - b_h) \geq 0$, which is true if and only if $a_i = a_k$ or $b_j = b_h$ (since $a_i - a_k \geq 0$ and $b_j - b_h \leq 0$). Therefore, if the successions are not ordered, we can always find an assignment switch that will increase the value of the SOP. Hence, we conclude that the successions must be ordered in order to maximize the SOP. ■

ACKNOWLEDGMENT

The authors would like to acknowledge all reviewers and editors who gave extensive and constructive criticism, which improved this work, with special mention to one anonymous

reviewer for providing a more elegant proof of the Appendix, now in use.

REFERENCES

- [1] A. Ahmad and P. Lima, "Multi-robot cooperative spherical-object tracking in 3D space based on particle filters," *Robot. Auton. Syst.*, vol. 61, no. 10, pp. 1084–1093, 2013.
- [2] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statist. Comput.*, vol. 6, pp. 113–119, Jun. 1996. [Online]. Available: <http://dx.doi.org/10.1007/BF00162521>
- [3] P. U. Lima, P. Santos, R. Oliveira, A. Ahmad, and J. Santos, "Cooperative localization based on visually shared objects," in *RoboCup 2010: Robot Soccer World Cup XIV*. Berlin, Germany: Springer, 2011, pp. 350–361.
- [4] D. Gohring and H.-D. Burkhard, "Multi robot object tracking and self localization using visual percept relations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 31–36.
- [5] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least square minimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5696–5701.
- [6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006, Art. no. 13.
- [7] J. Santos and P. U. Lima, "Multi-robot cooperative object localization: Decentralized bayesian approach," in *RoboCup 2009: Robot Soccer World Cup XIII*. Berlin, Germany: Springer, 2010, pp. 332–343.
- [8] L. Merino, F. Caballero, J. Martnez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *J. Field Robot.*, vol. 23, nos. 3/4, pp. 165–184, 2006.
- [9] L.-L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte, "A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 4539–4544.
- [10] A. Sanfeliu *et al.*, "Decentralized sensor fusion for ubiquitous networking robotics in urban areas," *Sensors*, vol. 10, no. 3, pp. 2274–2314, 2010.
- [11] M. Taiana, J. Santos, J. Gaspar, J. Nascimento, A. Bernardino, and P. Lima, "Tracking objects with generic calibrated sensors: An algorithm based on color and 3D shape features," *Robot. Auton. Syst.*, vol. 58, no. 6, pp. 784–795, 2010.
- [12] J. Inoue, A. Ishino, and A. Shinohara, "Ball tracking with velocity based on monte-carlo localization," in *Intelligent Autonomous Systems*, T. Arai, R. Pfeifer, T. R. Balch, and H. Yokoi, Eds. Amsterdam, The Netherlands: IOS Press, 2006, pp. 686–693.
- [13] C. Kwok and D. Fox, "Map-based multiple model tracking of a moving object," in *RoboCup 2004: Robot Soccer World Cup VIII*. Berlin, Germany: Springer, 2005, pp. 18–33.
- [14] P. A. V. Raúl, A. Lastra, and J. R. del Solar, "Integrated self-localization and ball tracking in the four-legged robot soccer league," in *Proc. 1st IEEE Latin Amer. Robot. Symp.*, Mexico City, Mexico, 2004, pp. 54–59.
- [15] L.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent methods for decentralised data fusion using particle filters," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, Sep. 2006, pp. 85–91.
- [16] W. Nisticò, M. Hebbel, T. Kerkhof, and C. Zarges, "Cooperative visual tracking in a team of autonomous mobile robots," in *RoboCup 2006: Robot Soccer World Cup X*. New York, NY, USA: Springer, 2007, pp. 146–157.
- [17] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2005, pp. 181–188.
- [18] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.
- [19] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Auton. Robots*, vol. 8, pp. 325–344, 2000.
- [20] A. Howard, M. Matarić, and G. Sukhatme, "Localization for mobile robot teams: A distributed mle approach," in *Experimental Robotics VIII* (Springer Tracts in Advanced Robotics), vol. 5, B. Siciliano and P. Dario, Eds. Berlin, Germany: Springer, 2003, pp. 146–155.
- [21] K. Leung, T. Barfoot, and H. Liu, "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 62–77, Feb. 2010.

- [22] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed heterogeneous outdoor multi-robot localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 374–381.
- [23] H. Mu, T. Bailey, P. Thompson, and H. Durrant-Whyte, "Decentralised solutions to the cooperative multi-platform navigation problem," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1433–1449, Apr. 2011.
- [24] A. Franchi, G. Oriolo, and P. Stegagno, "Mutual localization in multi-robot systems using anonymous relative measurements," *Int. J. Robot. Res.*, vol. 32, pp. 1303–1322, 2013.
- [25] A. C. Sanderson, "A distributed algorithm for cooperative navigation among multiple mobile robots," *Adv. Robot.*, vol. 12, no. 4, pp. 335–349, 1997. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1163/156855398X00235>
- [26] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proc. 1994 IEEE Int. Conf. Robot. Autom.*, May 1994, vol. 2, pp. 1250–1257.
- [27] R. Kurazume, S. Hirose, S. Nagata, and N. Sashida, "Study on cooperative positioning system (basic principle and measurement experiment)," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, vol. 2, pp. 1421–1426.
- [28] R. Kurazume and S. Hirose, "Study on cooperative positioning system: Optimum moving strategies for cps-iii," in *Proc. 1998 IEEE Int. Conf. Robot. Autom.*, May 1998, vol. 4, pp. 2896–2903.
- [29] R. Kurazume and S. Hirose, "An experimental study of a cooperative positioning system," *Auton. Robots*, vol. 8, no. 1, pp. 43–52, 2000. [Online]. Available: <http://dx.doi.org/10.1023/A:1008988801987>
- [30] I. M. Rekleitis, G. Dudek, and E. E. Milios, "On multiagent exploration," in *Proc. Vis. Interface*, 1998, pp. 455–461.
- [31] S. I. Roumeliotis and G. A. Bekey, "Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Autom. Symposia Proc.*, 2000, vol. 3, pp. 2958–2965.
- [32] I. Rekleitis, G. Dudek, and E. E. Milios, "Probabilistic cooperative localization and mapping in practice," in *Proc. 2003 IEEE Int. Conf. Robot. Autom.*, Sep. 2003, vol. 2, pp. 1907–1912.
- [33] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2859–2865.
- [34] J. W. Fenwick, P. M. Newman, and J. J. Leonard, "Cooperative concurrent mapping and localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 1810–1817.
- [35] C. Jennings, D. Murray, and J. Little, "Cooperative robot localization with vision-based mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, vol. 4, pp. 2659–2665.
- [36] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 1, pp. 695–701.
- [37] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robot. Res.*, vol. 22, no. 2, pp. 99–116, 2003.
- [38] B. Jung and G. Sukhatme, "Cooperative multi-robot target tracking," *Distrib. Auton. Robot. Syst.* 7, pp. 81–90, 2006.
- [39] A. Ahmad, T. Nascimento, A. G. S. Conceição, A. P. Moreira, and P. Lima, "Perception-driven multi-robot formation control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1851–1856.
- [40] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 678–695, Aug. 2011.
- [41] C.-H. Chang, S.-C. Wang, and C.-C. Wang, "Vision-based cooperative simultaneous localization and tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5191–5197.
- [42] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents). Cambridge, MA, USA: MIT Press, 2005.
- [43] P. Quang, C. Musso, and F. Le Gland, "An insight into the issue of dimensionality in particle filtering," in *Proc. 13th Conf. Inf. Fusion*, Jul. 2010, pp. 1–8.
- [44] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proc. IEEE Aerosp. Conf.*, 8–15, 2003, vol. 4, pp. 4_1979–4_1993.
- [45] S. Thrun, "Particle filters in robotics," in *Proc. 18th Conf. Uncertainty Artif. Intell.*, 2002, pp. 511–518.
- [46] A. Ahmad, J. Xavier, J. Santos-Victor, and P. Lima, "3D to 2D bijection for spherical objects under equidistant fisheye projection," *Comput. Vis. Image Understanding*, vol. 125, pp. 172–183, 2014.
- [47] A. Ahmad, "An integrated Bayesian approach to multi-robot cooperative perception," Ph.D. thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisbon, Portugal, 2013.
- [48] A. Ahmad and P. U. Lima, "Robot soccer data-set," Lisboa, Portugal, May 2012. [Online]. Available: http://datasets.isr.ist.utl.pt/lrmdataset/4_Robots_DataSet/



Aamir Ahmad received the B.Tech. (Hons.) degree in civil engineering from Indian Institute of Technology (IIT), Kharagpur, India, in July 2008 and the Ph.D. degree (with merit and European Doctorate) in electrical and computer engineering from Instituto Superior Técnico (IST), University of Lisbon, Lisbon, Portugal, in April 2013.

He is currently a Research Scientist at the Max Planck Institute for Intelligent Systems, Tübingen, Germany, where he is leading the multi-UAV aerial outdoor motion capture project. Until August 2016

he was a Marie-Curie Postdoctoral Fellow with Max Planck Institute for Biological Cybernetics, Tübingen. From April 2013 to August 2014, he was a Postdoctoral Researcher with Institute for Systems and Robotics (ISR), Instituto Superior Técnico, Lisbon. His main research focuses on the fields of Bayesian sensor fusion, multirobot systems, robot localization, object tracking, cooperative methods for tracking and localization, and 3-D object detection.

Dr. Ahmad was a member of the ISR scientific board and a program committee member of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2014, 2016, and the European Conference on Mobile Robotics (ECMR) 2013. He has served as a Reviewer of various prestigious conferences and journals, e.g., International Conference on Robotics and Automation, IEEE/RSJ International Conference on Intelligent Robots and Systems, ECMR, *Robotics and Autonomous Systems Journal*, *Autonomous Robots*, and *International Journal of Social Robotics*.



Guilherme Lawless received the Master's degree in electrical and computers engineering from the Faculty of Engineering, University of Porto, Porto, Portugal, in 2016. He is currently working toward the Ph.D. degree with Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal. He is being advised by Prof. Dr. Pedro U. Lima.

His research focus is in multirobot systems, active perception, and cooperative active SLAM. He is currently involved in the Aerial Outdoor Motion Capture (AirCap) project at the Max Planck Institute, Tübingen, Germany, where he is working with Dr. Aamir Ahmad.



Pedro U. Lima received the Licenciatura (5 years) and M.Sc. degrees in electrical and computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal, in 1984 and 1989, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1994.

He is a Professor at IST, Universidade de Lisboa, Lisbon, and a Researcher with Institute for Systems and Robotics, where he is the Coordinator of the Intelligent Robots and Systems group. He is the co-author of two books. His research interests include the areas

of discrete event models of robot tasks and planning under uncertainty, with applications to networked robot systems.

Dr. Lima is a Trustee of the RoboCup Federation (2003–2012, 2016–present), and was the General Chair of RoboCup2004, held in Lisbon. He was the President and a founding member of the Portuguese Robotics Society, was a National Delegate to EU and ESA Space Robotics programs and was awarded a six-month Chair of Excellence at the Universidad Carlos III de Madrid, Spain, in 2010. He was also the Coordinator of the FP7 Coordination Action RoCKIn and a member of the Advisory Board of the Mohamed Bin Zayed International Robotics Challenge. He is an Associate Editor of *Elsevier's Journal of Robotics and Autonomous Systems*.