Low-Cost Collaborative Localization for Large-Scale Multi-Robot Systems

Amanda Prorok, Alexander Bahr and Alcherio Martinoli

Abstract— **Large numbers of collaborating robots are advantageous for solving distributed problems. In order to efficiently solve the task at hand, the robots often need accurate localization. In this work, we address the localization problem by developing a solution that has low computational and sensing requirements, and that is easily deployed on large robot teams composed of cheap robots. We build upon a real-time, particle-filter based localization algorithm that is completely decentralized and scalable, and accommodates realistic robot assumptions including noisy sensors, and asynchronous and lossy communication. In order to further reduce this algorithm's overall complexity, we propose a low-cost particle clustering method, which is particularly well suited to the collaborative localization problem. Our approach is experimentally validated on a team of ten real robots.**

I. Introduction

A variety of tasks performed by multi-robot systems such as search and rescue and environmental monitoring often need accurate localization to succeed. Due to the intrinsic nature of such tasks, the individual agents are often confined to small size and weight, which sets hard limits on on-board resources. Simultaneously, a large portion of the robot's resources may be dedicated to the task at hand, leaving little room for solving the localization problem. These compounding problems pose the challenge of designing systems and algorithms that can flexibly accommodate given restrictions, without compromising performance.

In this paper, we consider the problem of absolute localization of a team of mobile robots for unknown initial robot positions. Our method builds upon previous work [15], where we proposed a collaborative localization algorithm targeting miniaturized, computationally limited platforms equipped with noisy, low-power sensing modalities. The collaboration strategy employs a robot detection model based on noisy, local, relative range and bearing observations. The work in this paper extends our previous approach by proposing a particle clustering method which reduces the complexity of the overall localization algorithm. We detail how the clustering method accommodates the constraints of the robot detection model, and show that, for modest range and bearing noise, it finds a near-optimal approximation. Finally, we test the scalability and robustness of our approach by performing experiments on a team of ten real robots.

A. Related Work

The problem of collaborative multi-robot localization was first addressed by Kurazume et al. [10], and was followed by a number of subsequent studies. In an early work, Roumeliotis et al. [18] enable the distribution of a Kalman estimation scheme by constructing communicating filters, which allows team-members to propagate their state and covariance estimates independently. Yet, as covariance matrix updates occur during each update step and require information exchange between all robots and a centralized processor, the method is particularly vulnerable to single-point failures. Particularly the requirement to update the information in all robots after a single observation of an individual robot assumes a communication infrastructure without any packet loss at all times which is impossible to achieve except for maybe a very small set of robots in a very confined environment. The method scales in $O(N^3)$ with respect to the number of robots, and thus limits its scalability due the high computational cost. In [11], Martinelli et al. propose an extension to [18], which relaxes the assumptions on relative observations, but without further improving the algorithm's scalability and cost. In a recent work, Nerurkar et al. [14] address the reduction of computational complexity and single-point failures by implementing a maximum a posteriori estimation method. Nevertheless, the $O(N^2)$ computational cost is significant. Also, the proposed method requires synchronous communication among the robots, and its feasibility still remains to be validated on real robots. Mourikis et al. [13] consider the problem of resource-constrained collaborative localization with the goal of deriving optimal sensing frequencies. Yet, as exteroceptive data is dealt with in a centralized way, the sensing frequencies inevitably decrease with an increasing number of robots, thus limiting the scalability of the approach. Finally, a decentralized localization algorithm presented in [3] tries to alleviate the problems described above. The approach is based on an extended information filter, whose implementation is distributed over the robot team members. However, its computational cost increases for each new observation made and it assumes bidirectional synchronous communication, the feasibility of which remains to be evaluated on real robots. Fox et al. first introduce a multi-robot Monte-Carlo localization algorithm in [4] which simultaneously relaxes noise assumptions as well as inter-robot dependencies. They propose a method with which robots mutually synchronize their position beliefs upon detection, and show successful global localization, yet provide no information about the scalability and processing requirements of their algorithm.

B. Problem Formulation

Our problem is described as follows. We have a multi-robot system of N robots $\mathcal{R}_1, \mathcal{R}_2, ...\mathcal{R}_N$, where the number N does not need to be known by the robots. The robots navigate in a bounded space. For a robot $\mathcal{R}_n$, at time t, the

All authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering at the Ecole Polytechnique Fédérale de Lausanne. The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 51NF40-111400. firstname.lastname@epfl.ch

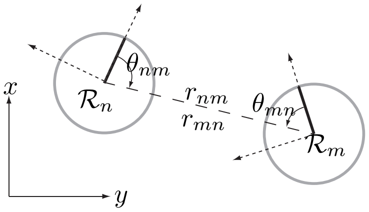


Fig. 1. System of two robots ($\mathcal{R}_n$ and $\mathcal{R}_m$) sharing a common localization frame. The figure illustrates the robots' relative range (r_{nm} and r_{mn}) and bearing (θ_{nm} and θ_{mn}) values.

pose $\mathbf{x}_{n,t}$ is given by the Cartesian coordinates $x_{n,t}, y_{n,t}$ and orientation $\phi_{n,t}$. Also, at time t, a robot $\mathcal{R}_m$ is in the set of neighbors $\mathcal{N}_{n,t}$ of robot $\mathcal{R}_n$ if robot $\mathcal{R}_m$ can determine a range $r_{mn,t}$ and bearing $\theta_{mn,t}$ to robot $\mathcal{R}_n$. Thus, at every moment in time, the neighborhood topology is defined by the physical constraints given by the relative observation sensors deployed on the robots. Also, if $\mathcal{R}_m \in \mathcal{N}_{n,t}$, we make the assumption that the robot $\mathcal{R}_m$ can communicate with the robot $\mathcal{R}_n$. Apart from a sensing modality which enables the robots to determine inter-robot range and bearing (including a unique robot identifier), they are also equipped with a dead-reckoning self-localization module (e.g., odometry), but do not make use of any exteroceptive sensors capable of feature recognition.

Every robot runs (locally) a collaborative particle filter with the goal of localizing itself, without any prior knowledge of the initial state or previous measurements. The belief of its pose is formulated as

$$\mathbf{Bel}(\mathbf{x}_{n,t}) \sim \{\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle | i = 1, ..., M\} = X_{n,t} \qquad (1)$$

where M is the number of particles, $\mathbf{x}_{n,t}^{[i]}$ is a sample of the random variable $\mathbf{x}_{n,t}$ (the pose), and $w_{n,t}^{[i]}$ is its weight. The symbol $X_{n,t}$ refers to the set of particles $\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle$ at time t belonging to robot $\mathcal{R}_n$. Given this context, the aim of our paper is to improve the default algorithm complexity of $O(|\mathcal{N}_{n,t}|M^2)$ [15], by devising a novel algorithm which respects an upper complexity limit of $O(|\mathcal{N}_{n,t}|MK)$ for a robot $\mathcal{R}_n$ at any time t, where $K \ll M$ is a user-defined value (K is detailed later, in Section III).

II. Collaborative Localization Algorithm

In this section, we briefly review the collaborative localization algorithm presented in [15], which, together with the Monte-Carlo Localization (MCL) method presented in [4], forms the baseline for our work. For convenience, the complete localization algorithm is shown in Algorithm 1.

A. Multi-Robot MCL

Let us from here on consider a robot $\mathcal{R}_n$, that is detected by robot $\mathcal{R}_m$, and simultaneously receives localization information from robot $\mathcal{R}_m$. At time t, after a sequence of motion control actions $u_{n,t}$, we formulate this event as

$$\begin{aligned}\mathbf{Bel}\left(\mathbf{x}_{n,t}\right) &= p\left(\mathbf{x}_{n,t}|u_{n,0..t}\right) \cdot \\ &\int p\left(\mathbf{x}_{n,t}|\mathbf{x}_{m,t}, r_{mn,t}, \theta_{mn,t}\right)\mathbf{Bel}\left(\mathbf{x}_{m,t}\right) \mathrm{d}\mathbf{x}_{m,t} \qquad (2)\end{aligned}$$

For such a collaboration to take place, robot $\mathcal{R}_m$ needs to communicate $r_{mn,t}, \theta_{mn,t}$ and $\mathbf{Bel}\left(\mathbf{x}_{m,t}\right)$ to robot $\mathcal{R}_n$. Thus a communication message is composed as $d_{mn,t} = \langle r_{mn,t}, \theta_{mn,t}, X_{m,t} \rangle$. If several robots in a neighborhood $\mathcal{N}_{n,t}$ communicate with robot $\mathcal{R}_n$, the received information is the set of all relative observations of robot $\mathcal{R}_n$ at time t, as well as the belief representations $X_{m,t}$ of all detecting robots $\mathcal{R}_m \in \mathcal{N}_{n,t}$. We denote this data set as $D_{n,t} = \{d_{mn,t} | \mathcal{R}_m \in \mathcal{N}_{n,t}\}$.

B. Range & Bearing Detection Model

The detection model $p_{mn}(\mathbf{x}_n|d_{mn})$ describes the probability that robot $\mathcal{R}_m$ detects robot $\mathcal{R}_n$ at pose $\mathbf{x}_n = [x_n, y_n, \phi_n]^\mathsf{T}$, given the detection data d_{mn}. This probability density function is applied to the ensemble of particles in the belief of robot $\mathcal{R}_n$, in order to adjust their weights to current relative observations. We define the transformation from Euclidean to polar coordinates $\mathbf{T}_e^p(\mathbf{x}_q, \mathbf{x}_p)$ as

$$\mathbf{T}_e^p(\mathbf{x}_q, \mathbf{x}_p) = \begin{bmatrix} r_{qp} \\ \theta_{qp} \end{bmatrix} \qquad (3)$$

where

$$\begin{aligned} r_{qp} &= \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \qquad (4) \\ \theta_{qp} &= \operatorname{atan2}((y_p - y_q), (x_p - x_q)) - \phi_q. \qquad (5) \end{aligned}$$

Thus, assuming Gaussian noise and knowledge of the range and bearing standard deviations (σ_r and σ_θ, respectively), and the independence of range and bearing measurements, the detection model is

$$p_{mn}(\mathbf{x}_n|d_{mn}) = \eta \cdot \sum_{\left\langle \mathbf{x}_m^{[i]}, w_m^{[i]} \right\rangle \in X_m} \Phi\left(\mathbf{T}_e^p(\mathbf{x}_m^{[i]}, \mathbf{x}_n); \boldsymbol{\mu}, \Sigma\right) \cdot w_m^{[i]} \quad (6)$$

where $\Phi(\cdot; \boldsymbol{\mu}, \Sigma)$ is a multivariate normal probability density function with mean $\boldsymbol{\mu} = [r_{mn}, \theta_{mn}]^\mathsf{T}$ and where η is a normalization constant. The covariance matrix is $\Sigma = \operatorname{diag}([\sigma_r^2, \sigma_\theta^2])$ (the work in [5] provides experimental evidence for our platform showing that a range and bearing measurement behaves like two independent Gaussian variables). As seen in [15], the detection model can easily be augmented by an additional component in case robot $\mathcal{R}_n$ reciprocally detects robot $\mathcal{R}_m$.

C. Reciprocal Sampling

In addition to using a robot detection model for updating the belief representation $\mathbf{Bel}(\mathbf{x}_{n,t})$, our approach relies on a *reciprocal* sampling method. Let us refer to the iterative process described in Algorithm 1: instead of sampling from $\mathbf{Bel}(\mathbf{x}_{n,t-1})$ in line 11, the reciprocal MCL algorithm samples from the distribution $\mathbf{x}_{n,t}^{[i]} \sim p(D_{n,t}|\mathbf{x}_{n,t}^{[i]})$, according to a robot detection model. Thus, samples are drawn at poses which are probable given the reciprocal robot observations, and which are independent of the previous belief $\mathbf{Bel}(\mathbf{x}_{n,t-1})$. By defining a reciprocal proportion α, particles are sampled from the robot's own belief with a probability $1 - \alpha$, and with a probability of α from the probability density function proposed by the detection model (line 13).

Algorithm 1 MultiRob_Recip_MCL$(X_{n,t-1}, u_{n,t}, z_{n,t}, D_{n,t})$

```
$\bar{X}_{n,t} = X_{n,t} = \emptyset$
for $i = 1$ to $M$ do
    $\mathbf{x}_{n,t}^{[i]} \leftarrow$ Motion_Model$(u_{n,t}, \mathbf{x}_{n,t-1}^{[i]})$
    $w_{n,t}^{[i]} \leftarrow$ Measurement_Model$(\mathbf{x}_{n,t}^{[i]})$
    $w_{n,t}^{[i]} \leftarrow$ Detection_Model$(D_{n,t}, \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]})$
    $\bar{X}_{n,t} \leftarrow \bar{X}_{n,t} + \left\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \right\rangle$
end for
for $i = 1$ to $M$ do
    $r \sim \mathcal{U}(0, 1)$
    if $r \leq (1 - \alpha)$ then
        $\mathbf{x}_{n,t}^{[i]} \leftarrow$ Sampling$(\bar{X}_{n,t})$
    else
        $\mathbf{x}_{n,t}^{[i]} \leftarrow$ Reciprocal_Sampling$(D_{n,t}, \bar{X}_{n,t})$
    end if
    $X_{n,t} \leftarrow X_{n,t} + \left\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \right\rangle$
end for
return $X_{n,t}$
```

III. Particle Clustering

The algorithm complexity of the detection model $p_{mn}(\mathbf{x}_n|d_{mn})$ (Eq. 6) leads to $O(M^2)$ for Algorithm 1 (for better clarity in the following derivations, we will assume that $|\mathcal{N}_{n,t}| = 1$). This cost can be prohibitive for a large number of particles M (i.e., large with respect to available computational resources). Also, a multi-robot system may have communication constraints that make sending large particle sets infeasible. Hence, even though the method applied in this paper allows for very small particle sets [15], we resort to a clustering method to further reduce the computational and communication overhead.

Let us consider a case where robot $\mathcal{R}_m$ detects robot $\mathcal{R}_n$. The goal of the clustering method is to reduce the number of operations needed to compute the probability density function p_{mn}. Thus, for every detection that it makes, robot $\mathcal{R}_m$ resorts to a clustering method which summarizes its set X_m composed of M particles to a set $\hat{X}_m$ composed of K cluster abstractions (or centroids), reducing the overall computational cost to $O(MK)$ (this clustering routine is detailed later, in Algorithm 2 of Section III-A). The resulting partition of the particle set is denoted $\mathcal{C}_m$, with $|\mathcal{C}_m| = K$. An individual cluster $c_m^{[k]} \in \mathcal{C}_m$ is defined as the set of particles

$$c_m^{[k]} = \{\langle \mathbf{x}_m^{[i]}, w_m^{[i]} \rangle \mid f(\langle \mathbf{x}_m^{[i]}, w_m^{[i]} \rangle, \cdot) = k\}, \tag{7}$$

where f is a function mapping a particle to a cluster index. Also, we define $\bar{c}_m^{[k]}$ as the data abstraction of cluster $c_m^{[k]}$, representing all particles in its set by the tuple

$$\bar{c}_m^{[k]} = \langle \hat{\mathbf{x}}_m^{[k]}, \hat{w}_m^{[k]}, \hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]} \rangle, \tag{8}$$

where $\hat{\boldsymbol{\mu}}_m^{[k]}$ is a two dimensional vector and $\hat{\Sigma}_m^{[k]}$ is a covariance matrix. Thus, $\hat{X}_m = \{\bar{c}_m^{[k]} \mid c_m^{[k]} \in \mathcal{C}_m\}$ is the set of K cluster abstractions. Finally, we denote the clustered detection data as $\hat{d}_{mn} = \langle r_{mn}, \theta_{mn}, \hat{X}_m \rangle$. Formally, given the notation introduced above, finding an optimal particle clustering is equivalent to solving the following optimization problem

$$\min_{\hat{d}_{mn}} \mathbf{D}(p_{mn}(x_n|d_{mn}) \,||\, \hat{p}_{mn}(x_n|\hat{d}_{mn})), \tag{9}$$

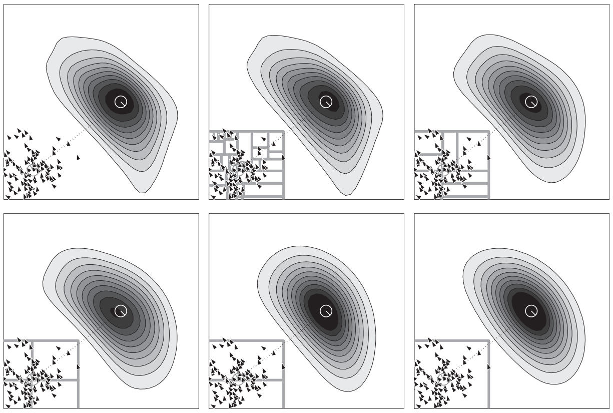


Fig. 2. The detection model (here with range and bearing noise $\sigma_r = 0.1$ and $\sigma_\theta = 0.2$) is projected on the detected robot (in white). Final cluster partitions are superimposed on the particles of the detecting robot. From left to right, top to bottom, the number of clusters K employed by the clustering algorithm is: 100, 32, 8, 4, 2, 1, for a total number of particles $M = 100$.

where $\hat{p}_{mn}$ is an approximated detection model, and $\mathbf{D}$ a distance measure between two probability density functions. Jain et al. [8] point out that in a typical clustering task, the actual grouping (or clustering) and cluster data abstraction (or cluster representation) are separate components of the task and are commonly treated sequentially. Hence, we deal with our problem by dividing it into the two following subproblems: *(i)* we consider the set of particles X_m and find an optimal way to create a partition $\mathcal{C}_m$, and *(ii)* we consider an arbitrary cluster $c_m^{[k]}$ in $\mathcal{C}_m$ and find an optimal way to determine its cluster abstraction $\bar{c}_m^{[k]}$. For a given set X_m, these two steps together ultimately lead to a set of cluster abstractions $\hat{X}_m$, which, instead of X_m, is included into the detection data tuple $\hat{d}_{mn}$ for every new detection made. The following paragraphs detail our low-cost clustering approach that aims to meet these specifications.

A. *Clustering Algorithm*

The optimal, combinatorial solution to the clustering problem of Equation 9 requires the evaluation of a very large number of partitions (the number of ways to partition a set of M data points into K non-empty clusters is given by Stirling number of the second kind). Even though efficient approaches have been proposed [9], combinatorial solutions still remain prohibitively expensive. Given the usefulness of clustering in a large range of disciplines, many non-combinatorial clustering approaches have been proposed [8]. Yet, since our goal is to reduce the final complexity of our algorithm, the complexity of the actual clustering algorithm must be at most equal to $O(MK)$. One of the most commonly used low-cost clustering methods is the k-means algorithm [12]. It starts off with a random initial cluster assignment and iteratively reassigns clusters until a convergence criterion is met or a maximum number of iterations L is attained. Although the algorithm has a low time complexity $O(MKL)$, its main disadvantage is that it is sensitive to the initial cluster assignment. The variant ISODATA algorithm [1] is also an iterative clustering algorithm with a time complexity of $O(MKL)$, with the additional capability to split and merge clusters according to predefined threshold values. It is therefore more flexible than the k-means and able to find the optimal partition, provided that the user is able to define correct threshold values. Non-iterative, incremental clustering algorithms have the advantage that they are even less time consuming than iterative algorithms. The leader algorithm [6] is the simplest of that kind. Data points are incrementally assigned to existing clusters based on a distance metric, with new clusters being created if all distance measures exceed a predefined criterion. Yet, given the algorithms incremental nature, the final clustering result is dependent on the order of the assignments made.

We take inspiration from the methods described above to develop a non-iterative, order-independent, non-parametric approach that produces a predefined number of K clusters. Our solution is inspired by the construction of multidimensional binary trees [2], and consists of a 2-dimensional sorting algorithm which repetitively separates the particle set along the mean of the dimension producing the highest variance, until the predefined maximum number of clusters K is attained. We note that splitting along the median instead of the mean incurs a higher complexity. A description of this algorithm is shown in Algorithm 2. The function on line 5 has a complexity $O(M)$, the function on line 6 has a complexity $O(|c_m^{[k_{max}]}|)$, and function on line 10 has a complexity $O(|c_m^{[k]}|)$. Hence, the total algorithm cost is $O(MK)$. Figure 2 shows examples of final cluster partitions for six different total number of clusters, performed on an identical set of 100 particles. We note that, even for maximal clustering ($K = 1$), the detection model is well approximated.

Algorithm 2 `Cluster`$(X_{m,t}, K)$

```
 1: $\hat{X}_m \leftarrow \emptyset$
 2: $c_m^{[1]} \leftarrow X_m$
 3: $\mathcal{C}_m \leftarrow c_m^{[1]}$
 4: for $k = 1$ to $K - 1$ do
 5:     $k_{max}, \mathrm{dim} \leftarrow \texttt{find\_highest\_variance\_cluster}(\mathcal{C}_m)$
 6:     $c_m^{[k_{max}]}, c_m^{[k+1]} \leftarrow \texttt{split\_cluster}(c_m^{[k_{max}]}, \mathrm{dim})$
 7:     $\mathcal{C}_m \leftarrow \mathcal{C}_m + c_m^{[k+1]}$
 8: end for
 9: for $k = 1$ to $K$ do
10:     $\bar{c}_m^{[k]} \leftarrow \texttt{assign\_data\_abstraction}(c_m^{[k]})$
11:     $\hat{X}_m \leftarrow \hat{X}_m + \bar{c}_m^{[k]}$
12: end for
13: return $\hat{X}_m$
```

B. Cluster Abstraction

For an arbitrary cluster $c_m^{[k]}$, we have the non-summarized detection data $d_{mn}^{[k]} = \langle r_{mn}, \theta_{mn}, c_m^{[k]} \rangle$. The problem of finding an optimal cluster abstraction $\bar{c}_m^{[k]}$ can, thus, be formalized as

$$\min_{\bar{c}_m^{[k]}} \mathbf{D_{KL}}(p_{mn}(x_n|d_{mn}^{[k]}) \,||\, \hat{p}_{mn}(x_n|\hat{d}_{mn}^{[k]})), \qquad (10)$$

where $\mathbf{D_{KL}}$ is the Kullback-Leibler divergence, and $\hat{d}_{mn}^{[k]} = \langle r_{mn}, \theta_{mn}, \bar{c}_m^{[k]} \rangle$ is the summarized detection data. Let us consider the following theorem.

Theorem 1: Given a point $\hat{\mathbf{x}}_\mathbf{m}^{[\mathbf{k}]} = [\hat{x}_m^{[k]}, \hat{y}_m^{[k]}, \hat{\theta}_m^{[k]}]^\mathsf{T}$, and the probability density function

$$\hat{p}_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]}) = \Phi\left(\mathbf{T}_e^p(\hat{\mathbf{x}}_m^{[k]}, \mathbf{x}_n); \hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]}\right), \qquad (11)$$

the Kullback-Leibler divergence between p_{mn} and $\hat{p}_{mn}$ is minimal if

$$\hat{\boldsymbol{\mu}}_m^{[k]} = \frac{1}{|c_m^{[k]}|} \sum_{\mathbf{x}_m^{[i]} \in c_m^{[k]}} \mathbf{v}_m^{[k,i]}, \qquad (12)$$

$$\hat{\Sigma}_m^{[k]} = \frac{1}{|c_m^{[k]}| - 1} \sum_{\mathbf{x}_m^{[i]} \in c_m^{[k]}} \left(\mathbf{v}_m^{[k,i]} - \hat{\boldsymbol{\mu}}_m^{[k]}\right)\left(\mathbf{v}_m^{[k,i]} - \hat{\boldsymbol{\mu}}_m^{[k]}\right)^\mathsf{T} \qquad (13)$$

are the mean and covariance of $\mathbf{v}_m^{[k,i]} = \mathbf{T}_e^p(\hat{\mathbf{x}}_m^{[k]}, \tilde{\mathbf{x}}_m^{[i]})$, with

$$\tilde{x}_m^{[i]} = x_m^{[i]} + r_{mn}\cos(\theta_{mn} + \phi_m^{[i]}) \qquad (14)$$

$$\tilde{y}_m^{[i]} = y_m^{[i]} + r_{mn}\sin(\theta_{mn} + \phi_m^{[i]}). \qquad (15)$$

Proof: Our proof is inspired by the general methodology shown in [7]. Let us use the polar coordinate frame ρ, ϑ with its origin at $\hat{\mathbf{x}}_m^{[k]}$ (cf. $\mathbf{T}_e^p(\hat{\mathbf{x}}_m^{[k]}, \cdot)$). The probability density function $\hat{p}_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]})$ is part of the exponential family of functions and can be written

$$\hat{p}_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]}) = h(\rho,\vartheta) g(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]}) e^{\eta(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]})\tau(\rho,\vartheta)} \qquad (16)$$

with $\tau(\rho,\vartheta) = [\rho, \rho^2, \vartheta, \vartheta^2, \rho\vartheta, 1]^\mathsf{T}$. The Kullback-Leibler divergence

$$\mathbf{D_{KL}}(p_{mn}||\hat{p}_{mn}) = \int_{-\infty}^{\infty} p_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]}) \log \frac{p_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]})}{\hat{p}_{mn}(\mathbf{x}_n|\hat{d}_{mn}^{[k]})} \mathrm{d}\mathbf{x}_n \qquad (17)$$

is minimized by taking its gradient with respect to $\hat{\boldsymbol{\mu}}_m^{[k]}$ and $\hat{\Sigma}_m^{[k]}$, and setting it to zero:

$$-\nabla \log g(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]}) = \nabla\eta(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]})^\mathsf{T} \mathbb{E}_{p_{mn}}\left[\tau(\rho,\vartheta)\right]. \qquad (18)$$

Note that the integral of $\hat{p}_{mn}$ is one, and thus we have

$$g(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]}) \int_{-\pi}^{\pi} \int_{0}^{\infty} h(\rho,\vartheta) e^{\eta(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]})\tau(\rho,\vartheta)} \mathrm{d}\rho \mathrm{d}\vartheta = 1. \qquad (19)$$

Computing the gradient on the left and right side of the above equation with respect to $\hat{\boldsymbol{\mu}}_m^{[k]}$ and $\hat{\Sigma}_m^{[k]}$ yields

$$-\nabla \log g(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]}) = \nabla\eta(\hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]})^\mathsf{T} \mathbb{E}_{\hat{p}_{mn}}\left[\tau(\rho,\vartheta)\right]. \qquad (20)$$

Finally, by combining Equations 18 and 20, we obtain

$$\mathbb{E}_{\hat{p}_{mn}}\left[\tau(\rho,\vartheta)\right] = \mathbb{E}_{p_{mn}}\left[\tau(\rho,\vartheta)\right] \qquad (21)$$

From this we conclude that mean and covariance, respectively, of p_{mn} and $\hat{p}_{mn}$ must be the same in order to minimize their Kullback-Leibler divergence, hence Equations 12 and 13. ■

We note that Theorem 1 does not take into account the uncertainty of the range and bearing observations. Thus, we propose a variant detection model $\hat{p}_{mn}$ (cf. Equation 6) that explicitly takes into account noise. We have

$$\hat{p}_{mn}(\mathbf{x}_n|\hat{d}_{mn}) = \eta \cdot \sum_{\bar{c}_m^{[k]} \in \hat{X}_m} \Phi\left(\mathbf{T}_e^p(\hat{\mathbf{x}}_m^{[k]}, \mathbf{x}_n); \hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]} + \Sigma\right) \cdot \hat{w}_m^{[k]} \qquad (22)$$

where $\hat{\boldsymbol{\mu}}_m^{[k]}$ and $\hat{\Sigma}_m^{[k]} + \Sigma$ approximate the true mean and covariance, respectively, in the presence of noise (we remind

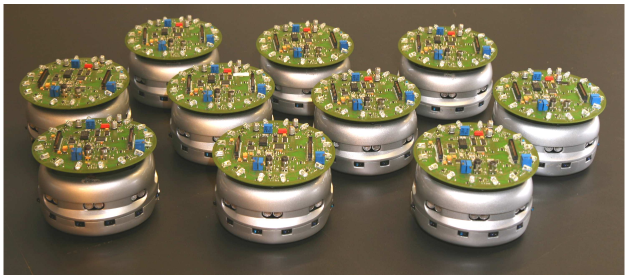


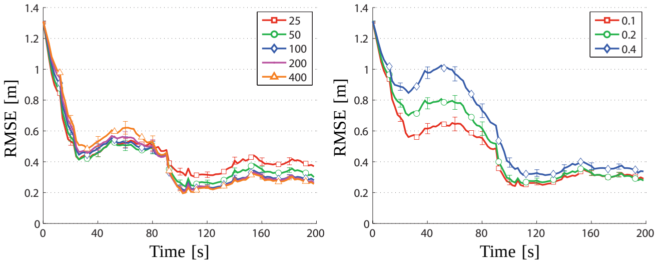
Fig. 3. Fleet of ten Khepera III robots. The robots are all equipped with an inter-robot, relative range and bearing module, which is composed of a ring of 16 infrared LEDs.

the reader that $\Sigma = \mathrm{diag}([\sigma_r^2, \sigma_\theta^2])$. Indeed, finding a closed form solution for the true values is intractable. However, if the set of particles $c_m^{[k]}$ is densely populated, our approximation is very good. Moreover, if the particle positions coincide, and if for a given cluster $c_m^{[k]}$ the point $\mathbf{x}_m^{[k]}$ is its center of mass, the solution is optimal. Hence, we complete the data abstraction $\bar{c}_m^{[k]} = \langle \hat{\mathbf{x}}_m^{[k]}, \hat{w}_m^{[k]}, \hat{\boldsymbol{\mu}}_m^{[k]}, \hat{\Sigma}_m^{[k]} \rangle$ (cf. Equation 8) with $\hat{\mathbf{x}}_m^{[k]}$ as the weighted center of mass, and $\hat{w}_m^{[k]}$ the cumulative weight

$$\hat{\mathbf{x}}_m^{[k]} = \frac{1}{\hat{w}_m^{[k]}} \cdot \sum_{\langle \mathbf{x}_m^{[i]}, w_m^{[i]} \rangle \in c_m^{[k]}} w_m^{[i]} \cdot [x_m^{[i]}, y_m^{[i]}, \phi_m^{[i]}]^\mathsf{T} \quad (23)$$

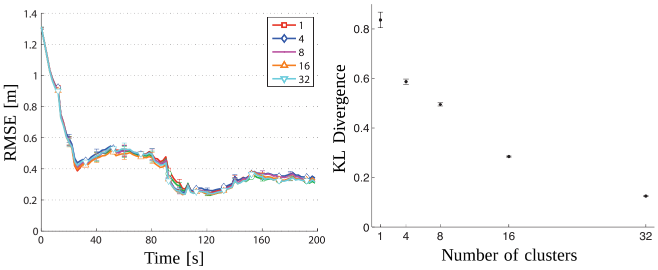
$$\hat{w}_m^{[k]} = \sum_{\langle \mathbf{x}_m^{[i]}, w_m^{[i]} \rangle \in c_m^{[k]}} w_m^{[i]}. \quad (24)$$

Finally, we note that the constraints given by our approximated detection model $\hat{p}_{mn}$ motivate the choice of a clustering algorithm which clusters densely located particles into common clusters (a condition which is satisfied by Algorithm 2).



(a) (b)

Fig. 4. Average localization error over 100 evaluations; the errorbars show 95% confidence intervals. (a) The algorithm is tested employing $\{25, 50, 100, 200, 400\}$ particles per robot. (b) The algorithm is tested with detection data messages corrupted by a $\{0.1, 0.2, 0.4\}$ failure rate.



(a) (b)

Fig. 5. (a) Average localization error over 100 evaluations. The localization algorithm is tested, employing the clustering method using $\{1, 4, 8, 16, 32\}$ clusters. (b) The Kullback-Leibler divergence between the full and the approximated detection models, as a function of the number of clusters employed by the clustering method.

IV. Experimental Evaluation

We validate our proposed approach by performing experiments on a team of ten Khepera III robots[1] [16] (see Figure 3). Our real experimental setup consists of a 3m large empty square arena. In order to measure the ground truth, we installed an overhead camera system, which allows us to monitor the robot positions in real-time with a maximum error below 3cm (as reported in our previous work [16]). The robots move straight at a speed of one robot-size per second (12cm/s), and perform standard Braitenberg obstacle avoidance. They are equipped with wheel encoders and use odometry for self-localization (the dead-reckoning error of the Khepera III robot has been reported to reach 0.1m after traveling for 18 meters [16]). We note that our measurement model routine (line 4 in Algorithm 1) simply reduces the particles' weights as they leave the bounded space, and does not take into account any exteroceptive sensor readings. The robots use a relative range and bearing module [17], which provides the measures used by the detection model. Figure 3 shows this platform mounted on every robot. The noise values were experimentally determined on our actual hardware setup ($\sigma_r = 0.15 \cdot r_{mn}$, and $\sigma_\theta = 0.15$ rad), with a maximum detection range of 3m. We tested our algorithm on a data set gathered over experiments of 3.5 minutes duration, where one of the ten robots was initially localized (and all other robots initially lost) . The data is comprised of inter-robot detection data messages (including range and bearing measurements), and odometry readings (recorded at 5Hz). Reciprocal robot detections are made asynchronously, and in average once every 5 seconds. If not stated otherwise in the following evaluations, the default algorithm configuration employs 100 particles per robot, no clustering ($K = M$), and a reciprocal sampling proportion $\alpha = 0.06$ (empirically determined, see [15]). We discuss the localization performance in terms of the mean positioning error of all particles in a given robot's belief (RMSE), a metric which implicitly includes the spread of the particle positions, and thus represents the uncertainty of the position estimate. Figure 4 discusses the sensitivity of our algorithm with respect to the number of particles M, as well as its robustness with respect to communication failures. Figure 4(a) shows the localization performance for a variable number of particles. Larger particle sets contribute to an improved localization accuracy. Yet, an 8-fold increase in the number of particles produces a reduction of only 25% of the localization error. This result coincides with the conclusions made in [15], where it was shown that by increasing the number of particles, the performance converges to that of an ideal localization filter with an infinity of particles. Figure 4(b) shows the localization performance for variable message failure rates. Increasing failure rates induce a graceful degradation of the localization performance. This result confirms the algorithm's robustness with respect to communication

[1] http://www.k-team.com/

failures, which ultimately reinforces the underlying asynchronous nature of our collaborative paradigm.

Figure 5 discusses our proposed clustering method. Figure 5(a) shows the localization performance when employing the clustering method for a variable number of clusters K. We note that the difference of performance between maximal clustering ($K = 1$) and modest clustering ($K = 32$) is very small. Figure 5(b) shows the Kullback-Leibler divergence between the full and the approximated detection models p_{mn} and $\hat{p}_{mn}$. The more clusters we employ in the clustering method, the smaller the divergence to the true probability density function. This shows that our clustering method produces a valid representation of the original probability density functions, which is simultaneously confirmed by the results shown in Figure 5(a). We note that the work in this paper considers a baseline experimental setup, where the belief of a robot's position is well represented by a single particle cluster (confirmed by results in 5(a)). More complex scenarios (including obstacles and multi-modal sensor models) should exhibit a larger spread of performance when clustering. In such cases, a trade-off between number of clusters K and accuracy must be determined (this exceeds the scope of this paper).

Finally, to illustrate the localization process, Figure 6 shows eight snapshots based on real data from an experiment performed over an interval of 126s during which one robot (in red) is initially localized. Each robot employed 100 particles with a reciprocal proportion $\alpha = 0.06$, and used the clustering routine with maximal clustering ($K = 1$).

V. Conclusion

In this work, we presented a fully scalable, probabilistic, multi-robot localization algorithm based on the Monte Carlo method. Its maximal overall complexity is $O(|\mathcal{N}|MK)$, where $|\mathcal{N}|$ is the number of neighboring robots (at a given time, for a given robot in the system), M the number of particles, and K an adjustable number of clusters produced by the clustering algorithm. This clustering method has shown to produce increasingly accurate probability density function representations for large K, and when employed in practice, has shown to perform well even for very small K. Furthermore, given the asynchronous paradigm of our collaboration strategy, the algorithm's update rate is much higher than the inter-robot message communication rate. Thus, the number of detected neighbors $|\mathcal{N}|$ is in practice no higher than 1, and the complete routine complexity is reduced to $O(MK)$. Thus, the algorithm is fully scalable with respect to the number of robots in the system. In addition, the algorithm poses no communication constraints and shows a graceful performance degradation in case of message failures. Our approach was experimentally validated on a team of ten real robots.

References

[1] G. H. Ball and D. J. Hall. ISODATA, a novel method of data analysis and classification. Technical report, 1965.

[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[3] A. Cristofaro, A. Renzaglia, and A. Martinelli. Distributed Information Filters for MAV Cooperative Localization. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2010.

[4] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots*, 8:325–344, 2000.

[5] S. Gowal and A. Martinoli. Bayesian rendezvous for distributed robotic systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2765–2771, 2011.

[6] J. A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, Inc., New York, NY, 1975.

[7] R. Herbrich. Minimising the Kullback-Leibler Divergence. Technical report, 2005. URL `http://research.microsoft.com/pubs/74555/KL.pdf`.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):265–323, 1999.

[9] L. G. W. Koontz, P. Narendra, and K. Fukunaga. A Branch and Bound Clustering Algorithm. *IEEE Transactions on Computers*, C-24(9):908–915, 1975.

[10] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1250–1257, 1994.

[11] A. Martinelli, F. Pont, and R. Siegwart. Multi-Robot Localization Using Relative Observations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2797–2802, 2005.

[12] J. McQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[13] A. Mourikis and S. I. Roumeliotis. Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5):917–931, 2006.

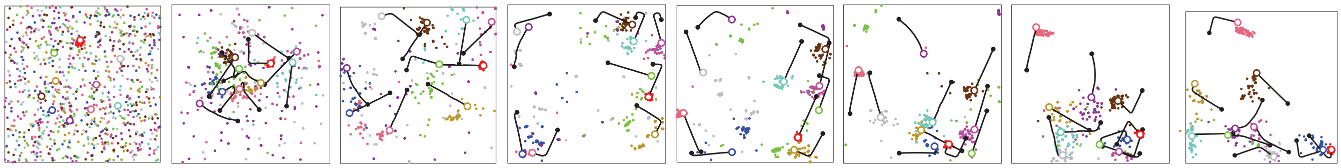
[14] E. Nerurkar, S. I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1402–1409, 2009.

[15] A. Prorok and A. Martinoli. A Reciprocal Sampling Algorithm for Lightweight Distributed Multi-Robot Localization. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3241–3247, 2011.

[16] A. Prorok, A. Arfire, A. Bahr, J. R. Farserotu, and A. Martinoli. Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010. doi: 10.1109/IPIN.2010.5647880.

[17] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli. A Fast On-Board Relative Positioning Module for Multi-Robot Systems. *IEEE Transactions on Mechatronics*, 14(2):151–162, 2009.

[18] S. I. Roumeliotis and G. A. Bekey. Distributed Multirobot Localization. *IEEE Transactions on Robotics and Automation*, 14(5):781–795, 2002.



(a) t = 0 (b) t = 18s (c) t = 36s (d) t = 54s (e) t = 72s (f) t = 90s (g) t = 108s (h) t = 126s

Fig. 6. The figure shows eight snapshots with 18s intervals of an experimental run on the team of 10 Khepera III robots. Each robot employed 100 particles with a reciprocal proportion $\alpha = 0.06$, and used the clustering routine with $K = 1$. The black lines show the trajectories completed in the time intervals between snapshots, with the filled black dots representing the robot positions at the end of the previous snapshots. The red robot was initially localized.