Mutual localization in multi-robot systems using anonymous relative measurements

Antonio Franchi¹, Giuseppe Oriolo² and Paolo Stegagno²

Abstract

We propose a decentralized method to perform mutual localization in multi-robot systems using anonymous relative measurements, i.e. measurements that do not include the identity of the measured robot. This is a challenging and practically relevant operating scenario that has received little attention in the literature. Our mutual localization algorithm includes two main components: a probabilistic multiple registration stage, which provides all data associations that are consistent with the relative robot measurements and the current belief, and a dynamic filtering stage, which incorporates odometric data into the estimation process. The design of the proposed method proceeds from a detailed formal analysis of the implications of anonymity on the mutual localization problem. Experimental results on a team of differential-drive robots illustrate the effectiveness of the approach, and in particular its robustness against false positives and negatives that may affect the robot measurement process. We also provide an experimental comparison that shows how the proposed method outperforms more classical approaches that may be designed building on existing techniques. The source code of the proposed method is available within the MLAM ROS stack.

Keywords
Mobile robots, localization, multi-robot localization, unknown data-association, anonymous measurements, relative measurements, particle filters, EKF, RANSAC, range finders

1. Introduction

Mutual localization, defined as the estimation of relative configurations in a group of robots, is a prerequisite for many multi-robot applications such as formation control (Chen et al., 2010; Franchi et al., 2012), cooperative transportation (Fink et al., 2010), coverage/exploration (Howard et al., 2006; Schwager et al., 2009; Durham et al., 2012) and monitoring (Pasqualetti et al., 2012).

External localization tools such as motion capture systems or GPS may sometimes be used to perform mutual localization, at the cost of reducing the system autonomy and flexibility. These solutions are however unfeasible or unreliable in many real-world situations, for example when the robots are operating in unstructured indoor environments or under the surface of the water. A much preferable approach is to preserve the system autonomy by letting the robots perform the localization on their own. In this case, robot-to-robot measurements are collected and broadcast to the group to be used in a cooperative localization scheme. Clearly, this approach requires that each robot is equipped with a sensor that provides some form of relative measurement (e.g. relative distance or bearing) as well as a communication module for transmitting and receiving information.

Several solutions have been proposed in the literature for different versions of the mutual localization problem. Some authors have considered the localization of static agents, commonly referred to as network localization (Aspnes et al., 2006). Shames et al. (2009) address the problem of reconstructing relative positions from distance measurements, whereas the objective of Piovan et al. (2013) is to estimate relative orientations using bearing measurements. Dieudonne et al. (2008) have proven that the network localization problem is NP-hard in the deterministic (no noise) case.

Other works have dealt with moving robots, with the objective of estimating the configurations of the agents in

¹Max Plank Institute for Biological Cybernetics, Tübingen, Germany
²Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Corresponding author:
Antonio Franchi, Max Plank Institute for Biological Cybernetics, Spearmannstraße 38, 72012 Tübingen, Germany.
Email: antonio.franchi@tuebingen.mpg.de
a common fixed frame. This problem, first addressed by
Kurazume et al. (1994), is known as cooperative local-
ization, a term introduced by Rekleitis et al. (1998). The
use of teammates as landmarks for the localization prob-
lem has been proposed by Kurazume and Hirose (2000).
Early methods have been mostly proposed in conjunc-
tion with navigation and exploration problems (Sanderson,
1998; Grabowski et al., 2000; Rekletis et al., 2001). Other
works have explicitly addressed the localization problem
using particle filters (Fox et al., 2000), maximum likelihood
estimators (Howard et al., 2002), and Extended Kalman
filters (EKF) (Roumeliotis and Bekey, 2002; Martinelli
et al., 2005; Mourikis and Roumeliotis, 2006). All of these
works show that the agents’ ability of sensing each other
can improve the localization of the entire system.

A closely related problem, to which we refer as proper
mutual localization, is to consider different moving frames
attached to each robot and then attempt to estimate the
relative configurations among the moving frames (Howard
et al., 2003). Different estimation strategies have been pro-
posed for this problem, such as the use of hierarchically
distributed EKFs (Martinelli, 2007) or closed-form solvers
in conjunction with (nonlinear) iterative least squares, both
in the 2D (Zhou and Roumeliotis, 2008) and the 3D case
(Trawny et al., 2010). An interesting study on the role of the
communication in a decentralized localization algorithm
has been proposed by Leung et al. (2010).

To the best of the authors’ knowledge, in all previous
works on network, cooperative, or mutual localization it is
assumed that relative measurements include the identity
of the robots, i.e. that the sensor ‘knows’ to which robot each
measurement refers to. This ubiquitous assumption requires
in practice the identification of a distinctive feature for each
robot, an ability that is not embedded in simple sensors
(e.g. cameras, range finders, RGB-D scanners) and typi-
cally calls for sophisticated classification algorithms based
on some form of robot tagging.

However, individual identification via tagging has a num-
ber of potential drawbacks. First, the process may be prone
to failures and false positives whenever the conditions of the
surrounding environment are not ideal (e.g. visual tag-
ging may be dramatically affected by slight changes in the
lighting conditions). Second, it is difficult to discriminate
robots in large groups using features such as shape or color,
i.e. most kinds of tagging do not scale well with the number
of robots. In any case, assigning an identity via tagging is
a form of centralization, and hence decreases the autonomy
of the system as well as that of the individual robots.

A more radical approach is to accept the fact that the
robots’ identities in the relative measurements are
unknown, and then attempt to estimate these identities
together with the relative configurations by using a more
powerful localization strategy. The problem of mutual
localization with anonymous measurements was first
considered by Franchi et al. (2009a), where a two-phase
registration algorithm is used to build hypotheses on the
identities of the robot associated with the measurements,
and then a multi-hypothesis Extended Kalman filter (MH-
EKF) is used to incorporate odometric measurements and
reject associations that are inconsistent with the current
belief.

In a subsequent work (Franchi et al., 2010a), we have
proved that anonymity of relative position measurements
causes a combinatorial ambiguity in the geometrical reg-
istration when the spatial arrangement of the robots is (or
is close to being) rotational symmetric. In this case, the run
time of the algorithm proposed by Franchi et al. (2009a)
may become a factorial function of the number of agents.
In applications where multi-robot systems are required to
move in regular formations, this worst-case complexity may
indeed materialize, affecting the performance of localiza-
tion systems based on that algorithm. To overcome this
difficulty, we modified the previous localization system by
feeding back the current belief about the robot relative con-
figurations into the multiple registration step, so as to nar-
rrow its search space and ultimately keep under control the
total execution time even in the presence of ambiguities
(Franchi et al., 2010b). In addition, the use of particle filters
was advocated as a convenient alternative to the MH-EKF.

In this paper we develop to a mature stage the lat-
ter approach. In particular, the main contributions of the
present work are the following: (1) the introduction of an
algorithm which solves a novel, challenging, and practi-
cally relevant problem in the field of mutual localization;
(2) a theoretical analysis of the implication of anonymity
on the solutions of the localization problem; (3) an experi-
mental validation of the proposed method proving its feasibil-
ity and effectiveness; and (4) a quantitative comparison
of its performance against more classical approaches that
may be designed building on existing techniques, such as
FastSLAM.

The paper is organized as follows. Section 2 provides a
detailed formulation of the problem. The implications of the
measurements being anonymous are analyzed in Section
3. Section 4 gives an overview of the proposed approach,
whose two main stages are described in detail in Sections
5 and 6. Experimental results are presented in Section 7,
while a discussion and a quantitative comparison with a
possible alternative approach are given in Section 8. Sec-
tion 9 describes the source code release and Section 10
concludes the paper.

2. Problem formulation

In this section, we describe the relevant geometry and pro-
vide a formulation of the mutual localization problem with
anonymous measurements. For the reader’s convenience,
the main symbols are collected in Table 1. Throughout the
section, refer to Figures 1 and 2 for an illustration.

Consider a multi-robot system \( \{R_1, R_2, \ldots, R_n\} \) moving
in a planar world. The number \( n \) of robots is not known and
may vary during the operation.
Fig. 1. Relevant frames and configuration variables for two robots \( R_i, R_j \).

<table>
<thead>
<tr>
<th>Table 1. Main symbols used in the paper.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_i )</td>
</tr>
<tr>
<td>( n )</td>
</tr>
<tr>
<td>( F_W )</td>
</tr>
<tr>
<td>( F_i )</td>
</tr>
<tr>
<td>( \mathcal{W}q_i )</td>
</tr>
<tr>
<td>( \mathcal{W}p_j )</td>
</tr>
<tr>
<td>( \mathcal{W}\theta_j )</td>
</tr>
<tr>
<td>( q_j )</td>
</tr>
<tr>
<td>( p_j )</td>
</tr>
<tr>
<td>( \theta_j )</td>
</tr>
<tr>
<td>( \oplus, \ominus )</td>
</tr>
<tr>
<td>( x^t )</td>
</tr>
<tr>
<td>( x^{1:t} )</td>
</tr>
<tr>
<td>( d_j^t )</td>
</tr>
<tr>
<td>( \delta_j^t )</td>
</tr>
<tr>
<td>( z )</td>
</tr>
<tr>
<td>( p(d_j^t</td>
</tr>
<tr>
<td>( D )</td>
</tr>
<tr>
<td>( O_{i,k} )</td>
</tr>
<tr>
<td>( O_{i,k}^t )</td>
</tr>
<tr>
<td>( p(p_j^t</td>
</tr>
<tr>
<td>( C )</td>
</tr>
<tr>
<td>( N_i^t )</td>
</tr>
<tr>
<td>( \text{bel}(x) )</td>
</tr>
<tr>
<td>( \text{bel}(x) )</td>
</tr>
<tr>
<td>(</td>
</tr>
<tr>
<td>( P )</td>
</tr>
<tr>
<td>( l(p) )</td>
</tr>
<tr>
<td>( L := {P, \mathcal{S}} )</td>
</tr>
<tr>
<td>( M )</td>
</tr>
<tr>
<td>( q_M, \mathcal{L}_M )</td>
</tr>
<tr>
<td>( q_{j'} )</td>
</tr>
<tr>
<td>( \beta^t )</td>
</tr>
</tbody>
</table>

As in the works by Howard et al. (2003) and Martinelli (2007), we look at the mutual localization problem from the subjective perspective of each robot. In the following, we describe the problem from the viewpoint of \( R_i \), whose objective is to localize every \( R_j \) with respect to itself. Clearly, \( R_i \) is completely generic.

With reference to Figure 1, denote by \( F_W = \{O_W, X_W, Y_W\} \) the world inertial frame and by \( F_j = \{O_j, X_j, Y_j\} \) the frame attached to robot \( R_j \). The absolute configuration of \( R_i \) is \( \mathcal{W}q_i = (\mathcal{W}p_i, \mathcal{W}\theta_i) \), with \( \mathcal{W}p_i \) and \( \mathcal{W}\theta_i \) respectively the position and orientation of \( F_j \) with respect to (w.r.t.) \( F_W \); whereas the relative configuration of \( R_j \) w.r.t. \( R_i \) is \( q_j = (p_j, \theta_j) \), with \( p_j \) and \( \theta_j \) respectively the position and orientation of \( F_j \) w.r.t. \( F_i \) (see Figure 1). Note that we have \( q_i = (\theta_i, 0, 0) \). Both absolute and relative configurations take values in \( SE(2) \).

Two robot configurations \( q' = (p', \theta') \) and \( q'' = (p'', \theta'') \) (absolute or relative) can be composed using the operators \( \oplus, \ominus : SE(2) \times SE(2) \rightarrow SE(2) \) introduced by Smith and Cheeseman (1986) and defined as

\[
q' \oplus q'' = (p' + R(\theta')p'', \theta' + \theta''),
\]

\[
q' \ominus q'' = (R(-\theta'')(p' - p''), \theta' - \theta''),
\]

where \( R(\phi) \in SO(2) \) is the planar rotation matrix associated with angle \( \phi \). For example, this allows us to express a relative configuration as

\[
q_j = \mathcal{W}q_j \ominus \mathcal{W}q_i.
\]

Note that in this context a configuration acts as a roto-translation. This dual interpretation will be used throughout the paper.

The above operators will also be used to compose a configuration with a position; in particular, \( q \oplus p' \) and \( p' \ominus q'' \) (note the order of the operands) return the position component of the right-hand side of (1) and (2), respectively. Accordingly, a relative position can be written as

\[
p_j = \mathcal{W}p_j \ominus \mathcal{W}q_i.
\]

We will also apply \( \ominus \) and \( \oplus \) to compose a configuration \( q \) with a set of positions \( P = \{p', p'', \ldots\} \), obtaining the new sets of points \( q \ominus P \) and \( P \oplus q \).

Let us now specify the sensor equipment of the robots. Clearly, we assume a discrete-time structure for the mutual localization system, in view of the use of sensor data with a certain refresh rate. To simplify the notation, time instants are denoted by 1, 2, \ldots, \( t \), but in fact the sampling interval can have any duration. We denote by a superscript \( t \) the value of a quantity at time \( t \), and by a superscript \( 1 : t \) the history of its values at times 1, 2, \ldots, \( t \); e.g. we write \( q_j^t \) and \( q_j^{1:t} \). Refer to Figure 2 for an illustration of the relevant measured quantities.

The generic robot \( R_i \) is equipped with an odometer and a robot detector. The odometer returns a measurement \( \delta_i^t \), the relative displacement of \( R_i \) between two consecutive time instants:

\[
\delta_i^t = q_{i'}^t \ominus q_i^{t-1}.
\]
To account for uncertainty, the odometer is represented with a probabilistic model \( p(d'|z) \), where \( z \) is used, throughout the paper, to indicate a generic measurement in its appropriate space; in this case, \( z = \delta_i^t \) is a measured roto-translation. Hence, \( p(d'|z) \) represents the probability that the true displacement of \( R_i \) is \( d_i^t \) given that the measured displacement is \( z \).

The robot detector of \( R_i \) measures the relative position \( p_i^t \) (not the orientation) of every \( R_j \) that falls in the detection region \( D \), a planar region attached to \( R_i \). No assumption is taken on the extension or shape of \( D \). Our experiments will be performed with limited detection regions that are not isotropic, and in particular have a blind zone; therefore, it may happen that \( R_i \) detects \( R_j \) but the reverse is not true. From a probabilistic point of view, the robot detector can be represented by a perception model \( p(p_i^t|z) \), where \( z \) represents in this case the measured relative position, whose ‘true’ value \( p_i^t \) is the position component of (3) at time \( t \).

In addition to the basic sensing uncertainty encoded in the ‘individual’ perception model, we assume that the actual operation of the detector is affected by the following complications, all stemming from the multi-robot nature of our problem (see Figure 2):

1. The detector may detect as robots certain deceiving objects which are similar to robots (false positives).
2. The detector may fail to detect robots in \( D \), e.g. due to line-of-sight occlusions (false negatives).
3. Most importantly, relative position measurements are anonymous, i.e. the detector cannot recognize the identities of the perceived robots.

Summarizing, at each time \( t \) the robot detector of \( R_i \) provides a set of measured positions (i.e. points) \( O_i^t = \{o_i^t,1, \ldots, o_i^t,m\} \), called observation in the following. The number \( m \) of points in each observation varies with \( t \); in addition, each element of \( O_i^t \) may represent an actual robot or a deceiving obstacle in \( D \). Finally, since position measurements are anonymous, the index \( k \) in \( o_i^t,k \) is only associated with the internal ordering of \( O_i^t \), and has no relationship with the identity of the robot it (possibly) represents. For example, if the robot detector is based on a laser range finder it is natural to use an angular ordering.

The last ingredient of our mutual localization system is communication. In particular, it is assumed that at each time \( t \) robot \( R_i \) broadcasts a message which can be received by any other robot \( R_j \) falling in its communication region \( C \), another planar region rigidly attached to \( R_i \). The message contains: (1) the integer \( i \), i.e. the identity of \( R_i \), (2) the measured displacement \( \delta_i^t \), (3) the observation \( O_i^t \). In turn, \( R_i \) receives messages from the robots in whose communication region it falls; for compactness, these robots will be called neighbors of \( R_i \) at \( t \) and collected in the set \( N_i^t \). False negatives may also affect the communication (messages lost between robots that should be able to communicate).

Based on the above description and notation, the problem addressed in this paper may be formulated as follows.

**Problem 1** (Mutual localization with anonymous position measurements, from the viewpoint of \( R_i \)). For \( t = 1, 2, \ldots \) and \( j \in \bigcup_{\tau=1}^{t-1} N_i^\tau \), compute an estimate of the relative configuration \( q_j^t \) of \( R_j \) in the form of a belief function:

\[
\text{bel}(q_j^t) := p(q_j^t|\delta_i^t, O_i^{\leq \tau}, \delta_j^t, O_j^t)_{\tau=1, \ldots, t; j \in N_i^t},
\]

that is, the posterior probability for \( q_j^t \) given:

1. the measurements of the odometer and of the robot detector of \( R_i \);
2. the measurements of the odometers and of the robot detectors of all robots that have been neighbors of \( R_i \) at some time instant.
3. The implications of anonymity

In order to analyze the implications of anonymity in Problem 1 and to justify the proposed solution approach, we consider first its deterministic-static version, often referred in the literature as network localization (Aspnes et al., 2006). Depending on whether the position measurements are anonymous or not, we have the two following problems.2

**Problem 2** (Network localization with anonymous position measurements, from the viewpoint of $R_i$). Modify Problem 1 by assuming that:

i) the robots do not move;

ii) relative position measurements are not affected by uncertainty (no noise, no false positives).

Compute a set of relative configurations $\{q_j, j \in N_i\}$ that are consistent with:

1. the observation of the robot detector of $R_i$;

2. the observations of the robot detectors of $R_j, \forall j \in N_i$.

**Problem 3** (Network localization with non-anonymous position measurements, from the viewpoint of $R_i$). Identical to Problem 2 with the following additional assumption:

iii) each observation contains the identity of the detected robots.

In both problems, localization only concerns robots that are neighbors of $R_i$, i.e., robots that communicate their observations to $R_i$. Note that Problem 3 is equivalent to the classical formulation of network localization.

Graph theory has been extensively used to analyze network localization problems with planar point (position-only) configurations and non-anonymous relative distance measurements (Aspnes et al., 2006; Jackson, 2007; Shames et al., 2009). In that context, it was shown the problem is uniquely solvable when the framework (i.e. the point formation) possesses a property known as rigidity. Proceeding from the work of Jackson (2007), we extend below the concept of rigid framework to our case, i.e., that of planar pose (position and orientation) configurations and relative position measurements, which may be anonymous or not.

A planar pose-configuration/position-measurements framework (simply framework in the following) is a pair $(G, Wq)$, where $G=(\mathcal{V}, \mathcal{E})$ is a directed graph with vertexes $\mathcal{V}$ and edges $\mathcal{E}$, and $Wq=(Wp, W\theta)$ is a map from $\mathcal{V}$ to $\text{SE}(2)$. The map $Wq$ associates every vertex $v \in \mathcal{V}$ to an absolute configuration $Wq(v) = (Wp(v), W\theta(v)) \in \text{SE}(2)$.

Consider two frameworks $(G, Wq_a)$ and $(G, Wq_b)$ that share the same graph but map the vertices to different configurations. We say that:

- $(G, Wq_a)$ and $(G, Wq_b)$ are equivalent if
  $$Wp_a(v) \oplus Wq_a(v) = Wp_b(v) \oplus Wq_b(v)$$
  for all $(v, v') \in \mathcal{E}$;

- $(G, Wq_a)$ and $(G, Wq_b)$ are congruent if
  $$Wp_a(v') \oplus Wq_a(v) = Wp_b(v') \oplus Wq_b(v)$$
  for all $v, v' \in \mathcal{V}$.

Using this terminology, we will say that $(G, Wq)$ is rigid if all of its equivalent frameworks are also congruent frameworks.

Assume now we are given an instance of Problem 2 or 3, and consider the associated framework $(G, Wq)$, defined by:

- $\mathcal{V} = \{v_k | k \in \{i \cup N_i\}\}$;

- $(v_k, v_h) \in \mathcal{E}$ if the measurement of the relative position $Wp(v_h) \oplus Wq(v_h)$ is available;

- $Wq(v_k) = Wq_k$.

The fact that our network localization problems are formulated from the viewpoint of $R_i$ entails that the associated framework only concerns its neighbors $N_i$. Note that robots that are detected by $R_i$ but do not communicate with it are not in $N_i$; on the other hand, $N_i$ may include robots that are not detected by $R_i$.

**Proposition 1.** An instance of Problem 3 is uniquely solvable if and only if its associated framework $(G, Wq)$ is rigid.

Proof. If the problem instance is uniquely solvable, the relative configurations of the robots in $N_i$ w.r.t. $R_i$ are uniquely determined. Now consider a pair of vertexes $v_k, v_h$ with $k, h \in \{i \cup N_i\}$. Since we have $Wq(v_k) \oplus Wq(v_h) = (Wq(v_k) \oplus Wq(v_h)) \oplus (Wq(v_h) \oplus Wq(v_h)) = Wq_k \oplus Wq_h$, also $Wp(v_k) \oplus Wq(v_k)$ is uniquely determined, which in turn implies that any framework which is equivalent to $(G, Wq)$ must also be congruent to it.

Now suppose that $(G, Wq)$ is rigid. Then, $Wp(v_k) \oplus Wq(v_k)$ is uniquely determined for any pair of vertexes $v_k, v_h$ with $k, h \in \{i \cup N_i\}$. Hence, $q_i, j \in N_i$, can be determined from simple geometry:

$$q_i = (p_{ji} - \pi) \cdot p_{ji} + \pi,$$

where $p_{ji} = Wp(v_i) \oplus Wq(v_i)$, $p_{ji} = Wp(v_j) \oplus Wq(v_j)$, and $\pi$ is the bearing angle associated with position vector $p$.

The following corollary, which descends from the same arguments used in the previous proof, identifies sufficient conditions for the associated framework to be rigid.

**Corollary 1.** An instance of Problem 3 is uniquely solvable if the graph $G=(\mathcal{V}, \mathcal{E})$ of the associated framework admits a directed spanning tree $T=(\mathcal{V}_T, \mathcal{E}_T)$ such that for any $(v_i, v_j) \in \mathcal{E}_T$ it is both $(v_j, v_i) \in \mathcal{E}$ and $(v_i, v_j) \in \mathcal{E}$. This is certainly true if $G$ is complete (i.e. all possible measurements are available).

The importance of rigidity for the unique solvability of Problem 3 is illustrated in Figure 3. In particular, rows 1–3 refer to the case of non-anonymous measurements (Problem 3). In row 1, the graph $G$ of the associated framework is

...
complete, and therefore the latter is rigid; a unique solution then exists. For the situation in row 2, \( G \) is not complete but the associated framework is still rigid; unique solvability is again guaranteed. The framework associated to the problem instance in row 3 is instead non-rigid, because the connection between \( R_i \) and \( R_j \) is weak, and therefore multiple (actually infinite) solutions exist.

Let us now look at Problem 2, where anonymity of the measurements comes into play. To this end, we need a preliminary definition. We say that a set of planar points is rotational symmetric of order \( \lambda \) if there exist \( \lambda \) distinct rotations around its centroid that are associated with angles in \((0, 2\pi]\) and map the set into itself. Accordingly, a generic set of points is at least rotational symmetric of order 1 (a \( 2\pi \) rotation will always map the set into itself), whereas actual symmetries are associated to \( \lambda \geq 2 \).

**Proposition 2.** The number of solutions of an instance of Problem 2 is lower bounded by

\[
\begin{align*}
\text{if centroid}(P) \notin P, & \quad (\lambda - 1)!(|V|^{\frac{1}{\lambda^2}} - 1) \quad \text{(4)} \\
\text{if centroid}(P) \in P, & \quad (\lambda)!^{\frac{|W|}{\lambda}} \quad \text{(5)}
\end{align*}
\]

where \( \lambda \) is the order of rotational symmetry that characterizes the set \( P = \{p_k \mid k \in \{i\} \cup N_i\} \) of the robot cartesian positions.

**Proof.** Given an instance \( I_A \) of Problem 2, denote with \( I_N \) its non-anonymous version, and with \( (G, V, q) \) the framework associated to \( I_A \) and \( I_N \) (it is the same). Clearly, any solution of \( I_N \) is also a solution of \( I_A \); denote with \( \sigma \) the number of solutions of \( I_A \). Then consider another instance \( I'_N \) of Problem 3 such that the associated framework is \( (K, W, q) \), where \( K \) is the complete graph built from \( G \); let \( I'_A \) be the anonymous version of \( I'_N \) and \( \sigma' \) the number of solutions of \( I'_A \). Since \( G \) is a subgraph of \( K \), the input data of \( I'_A \) contain all the measurements that are present in the input of \( I_A \); therefore, it is \( \sigma' \leq \sigma \). The proof is concluded using Proposition 4 of (Franchi et al., 2010a), which states that \( \sigma' \) is given by (4)–(5).

Note that the above lower bounds on the number of solutions of Problem 2 hold regardless of the rigidity of the graph \( G \) in the associated framework, i.e. of the richness of the available measurements. In particular, both bounds reduce to 1 when \( \lambda = 1 \), i.e. in the absence of non-trivial rotational symmetries.

Since Problem 3 is a more informed version of Problem 2, any solution of the former is a solution of the latter (the reverse is not true). In view of this, we may combine Propositions 1 and 2 in the following statement.

The solution of a given instance of Problem 2 may be not unique for two independent reasons:

1. **Non-rigidity:** The framework associated with the problem is not rigid. In this case, clearly, an infinity of solution exist also for the problem with anonymous measurements.
2. **Ambiguity:** If the set of robot cartesian positions is rotational symmetric of order two or larger, there exists a
finite set of possible solutions whose number grows as a factorial function of the order of symmetry.

Clearly, these situations may occur simultaneously, and their effects cumulate.

We emphasize that ambiguity is a byproduct of the anonymity of the measurements. This is further illustrated in Figure 3, row 4. Even if the robots are in the exact same arrangement of row 1, and the graph $G$ of the associated framework is still complete, the set of robot cartesian positions is rotational symmetric of order $\lambda = 3$ and therefore, consistently with (4), two distinct solutions arise in the presence of anonymity.

It should be kept in mind that the relevance of the ambiguity phenomenon goes well beyond the deterministic case, in which rotational symmetric situations are a set of zero measure in the space of possible arrangements of the multi-robot system. In fact, when the localization problem is formulated in a stochastic setting to take into account measurement noise, ambiguity arises as soon as when the group is ‘close’ to being in a rotational symmetric arrangement, generating large uncertainties in the probability density of the solution.

Finally, note that rotational symmetry is not the only source of ambiguity for Problem 2. Indeed, if the framework associated with the problem is not complete (be it rigid or not), multiple solutions may exist even for non-symmetric arrangements of the system (Franchi et al., 2010a).

### 4. Overview of the Proposed Approach

As discussed in the introduction, the main distinctive aspects of Problem 1 with respect to standard versions of multi-robot localization problems found in the literature are the anonymity of the measurements and the presence of false positives/negatives. In principle, the anonymity issue could be addressed by using some form of data association technique. We shall delay the discussion of this classical approach to Section 8, where we will explicitly compare its results to those of our method.

The results of the previous section, and in particular those concerning Problem 2, provide useful guidelines for designing a solution method for Problem 1, which is the stochastic-dynamic version of the former. In particular:

- To consider all of the possible solutions, which may be multiple in the presence of ambiguity, first enumerate all possible arrangements of the observed robots (including their identities and orientations) that are consistent with the current observations, both endogenous (gathered by $R_i$) and hexogeneous (gathered and communicated by its neighbors). Incorporation of other robots’ observations will also allow $R_i$ to localize robots that it cannot directly detect, either due to the intrinsic limitations (e.g. anisotropy) of its robot detector or to line-of-sight occlusions (false negatives). This geometric step, called multiple registration, is also instrumental for rejecting false positives, for which there exists no matching observation.
- Since the number of possible solutions of a multiple registration can be factorial in case of ambiguity, use a probabilistic method such as random sampling consensus (RANSAC) (Fischler and Bolles, 1981) in order to keep the execution time under control.
- Make the multiple registration algorithm aware of the current belief (hence, of past history) to help discriminate among statically ambiguous situations.
- Using observations taken over a time interval may allow to extract a solution even when the multiple registration problem admits an infinite number of solutions due to non-rigidity of the associated framework. This leads to filtering as a way to incorporate successive measurements and odometric data. Since the measurements are uncertain, this is also instrumental in reducing the effect of noise.

This results in a two-phase approach to the problem solution, clearly reflected in the scheme of our mutual localization system shown in Figure 4.

In particular, robot $R_i$ runs a probabilistic multiple registration (PMR) algorithm to compute feasible hypotheses on the identities and relative configurations of the robots belonging to $N_i^t$, on the basis of the current observations $O_i^t$, $\{O_j^t\}_{j \in N_i^t}$, and of the current beliefs about $\{q_j^t\}_{j \in N_i^t}$. The
relative configurations thus obtained, together with the odometric measurements \( \delta'_i \) and \( \{ \delta'_j \}_{j \in N'_t} \), are used by \( |N'_t| \) particle filters to update the belief about the configuration of each robot in \( N'_t \). Note the feedback mechanism that brings the motion-updated belief of the particle filters in input to the multiple registration stage, allowing the latter to prune unreasonable solutions in ambiguous situations and thus reducing its computational load.

The main features of our two-stage approach are:

1. the mutual exclusive structure of the set of measurements is exploited in the registration phase;
2. the increased dimension of the measurements provided to the particle filters (the relative orientation is added by PMR) improves the convergence of the estimation process;
3. the simplicity of the probabilistic model of the robot detector used in the particle filter, which does not have to take into account the identity of the robots;
4. the extension of the perception capabilities of the system members beyond those of the individual robots, without any kind of centralized elaboration.

The multiple registration algorithm and the particle filters are described in detail in the next two sections.

5. Multiple registration stage

At each time instant \( t \), the generic robot \( R_i \) runs the PMR algorithm, that represents the part of the localization system which provides the input to the measurement update in the particle filters (see Figure 4).

In computer vision, registration between two images entails computing the roto-translation between the viewpoints from which the images were gathered. In our case, the ‘images’ to be registered are the observations gathered from which the images were gathered. In our case, entails computing the roto-translation between the viewpoints from which the images were gathered.

To generalize this concept, define a set of labeled points as a pair \( \mathcal{L} = (P, I) \) where \( P \) is a set of points and \( I : P \rightarrow \{0, 1, \ldots, n\} \) is an injective labeling map. The label \( l(p) \) is the index of the robot to which point \( p \) is associated, with \( l(p) = 0 \) indicating that \( p \) is not yet associated with any robot. We may then say that the previous processing transforms an observation into a particular set of labeled points, in which all points have label zero but one.

Consider now two sets of labeled points \( \mathcal{L}' = (P', I') \) and \( \mathcal{L}'' = (P'', I'') \), representing two ‘views’ of the same scene by different robots. In particular, each view may be either an ‘original’ observation (only one point has a non-zero label), or an ‘augmented’ observation incorporating the identities of some robots as the result of previous computations (multiple points have non-zero labels). Loosely speaking, our objective is to determine the roto-translation between the two robots that produces the best match between the two views.

Denote by \( q \) a generic roto-translation acting on \( \mathcal{L}'' \) (more precisely, on \( P'' \)). Under \( q \), a point \( p' \) of \( \mathcal{L}' \) is associated with a point \( p'' \) of \( \mathcal{L}'' \) if

\[
\|p' - q \oplus p''\| \leq \eta,
\]

where \( \eta > 0 \) is the distance threshold under which the two points are identified as the same. Denote by \( B_q \subset P' \times P'' \)
between Liu (2005) and Cunningham et al. (2012). In any case, the
directly or with little adaptation, e.g. those by Thrun and

points) resulting from the application of rule (6).

Definition 1 (Matching between sets of labeled points). Given two sets of labeled points \( L' = (P', l') \) and \( L'' = (P'', l'') \), a roto-translation \( q \) acting on \( L'' \) is a matching between \( L' \) and \( L'' \) if the corresponding relation \( B_q \) is such that:

1. each point in \( P' \) is associated to at most one point in \( P'' \), and vice versa;
2. points that have different non-zero labels are never associated.

The second condition guarantees that two points that have been assigned to different robots are not associated in \( B_q \).

An example of matching is shown in Figure 5.

The quality of a matching can be measured by the number of points it associates. We assume an algorithm is available that can compute all matchings above a certain quality.

Definition 2 (Matching algorithm). A matching algorithm is any procedure that, given two sets of labeled points \( L' \) and \( L'' \), returns all of the matchings \( q \) for which the cardinality of \( B_q \) (i.e. the number of pairs of associated points in \( L' \) and \( L'' \)) is larger or equal to a specified threshold \( w \).

A matching algorithm will return each matching as a roto-translation with the corresponding binary relation. The number of returned matchings clearly depends on the cardinality threshold \( w \); a lower threshold will typically produce more matchings. Even when the threshold increases, however, multiple matchings may be returned, e.g., in the presence of rotational symmetry (see Section 3).

With the above definitions, the basic structure of our binary registration algorithm is given in Algorithm 1. A more detailed discussion of each step follows.

Matching (line 1): For this step, we use the technique introduced by Franchi et al. (2009b), which is based on RANSAC (Fischler and Bolles, 1981). Since our matching problem is closely related\(^3\) to point-pattern matching, several other methods available in the literature can be used directly or with little adaptation, e.g. those by Thrun and Liu (2005) and Cunningham et al. (2012). In any case, the choice of the matching algorithm is essentially an implementation issue that does not affect the general working principle of the PMR algorithm to be described.

Correction (line 3): This consists in replacing each roto-translation \( q \) with

\[
q_M = \arg\min_{q \in \text{SE}(2)} \sum_{(p', p'') \in B_q} \|p' - q \oplus p''\|^2. \tag{7}
\]

This means that, independently of how \( q \) was computed, it is corrected by choosing the roto-translation that minimizes the squared error over all of the pairs associated by \( B_q \). The minimization problem \( (7) \) can be solved by standard methods; in particular, in our implementation we have used the technique proposed by Umeyama (1991).

Merging (line 4): A merged set of labeled points \( L_M = (P_M, l_M) \) is generated as follows:

1. For each pair \((p' \in P', p'' \in P'')\) contained in \( B_q \), compute the ‘averaged’ point

\[
p = \frac{\kappa}{\kappa + 1} p' + \frac{1}{\kappa + 1} (q_M \oplus p''), \tag{8}
\]

and put it in \( P_M \). In this formula, \( \kappa \) is the generation counter of \( p' \); i.e. \( \kappa = 1 \) if \( p' \) is an observed point, \( \kappa = 2 \) if \( p' \) is the result of a previous binary registration between two observed points, and so on. The structure of Equation (8) guarantees that all of the points that have contributed to a certain average, either in the current or in previous binary registrations, are given the same weight.

2. Put in \( P_M \) each point \( p' \in P' \) that has not been associated. For each point \( p'' \in P'' \) that has not been associated, put in \( P_M \) the point \( q_M \oplus p'' \).

3. For averaged points, define the labeling map \( l_M \) by letting

\[
l_M(p) = \max\{l(p'), l(p'')\}. \tag{9}
\]

For the other points in \( P_M \), which come from \( P' \) or \( P'' \), the label is simply inherited from \( l' \) or \( l'' \).

Note that Equation (9) implies that an averaged point will be unlabeled if it comes from two unlabeled points, and will get the non-zero label if one of the two points is actually labeled. Remember that the two points \( p' \) and \( p'' \) cannot have different non-zero labels due to the definition of matching.

Wrapping up, a binary registration produces multiple pairs of the form \( M = (q_M, L_M) \), which actually consist of a (corrected) roto-translation \( q_M \) and the corresponding (merged) set of labeled points \( L_M \). Generalizing the concept introduced in Definition 1, in the following we call matching any pair \( M \) that comes out of a binary registration.

---

**Algorithm 1: Binary registration.**

**input:** sets of labeled points \( L' \) and \( L'' \)

**output:** multiple matchings, each matching \( M \) consisting of a corrected roto-translation \( q_M \) and a merged set of labeled points \( L_M \)

1. run a matching algorithm on \( L' \) and \( L'' \);
2. foreach returned roto-translation \( q \) do
   3. correct the roto-translation \( q \) to \( q_M \);
   4. merge \( L' \) and \( L'' \) into a new set of labeled points \( L_M \);

---

1. each point in \( P' \) is associated to at most one point in \( P'' \), and vice versa;
2. points that have different non-zero labels are never associated.

The second condition guarantees that two points that have been assigned to different robots are not associated in \( B_q \).

An example of matching is shown in Figure 5.

The quality of a matching can be measured by the number of points it associates. We assume an algorithm is available that can compute all matchings above a certain quality.

**Definition 2 (Matching algorithm).** A matching algorithm is any procedure that, given two sets of labeled points \( L' \) and \( L'' \), returns all of the matchings \( q \) for which the cardinality of \( B_q \) (i.e. the number of pairs of associated points in \( L' \) and \( L'' \)) is larger or equal to a specified threshold \( w \).

A matching algorithm will return each matching as a roto-translation with the corresponding binary relation. The number of returned matchings clearly depends on the cardinality threshold \( w \); a lower threshold will typically produce more matchings. Even when the threshold increases, however, multiple matchings may be returned, e.g., in the presence of rotational symmetry (see Section 3).

With the above definitions, the basic structure of our binary registration algorithm is given in Algorithm 1. A more detailed discussion of each step follows.

**Matching (line 1):** For this step, we use the technique introduced by Franchi et al. (2009b), which is based on RANSAC (Fischler and Bolles, 1981). Since our matching problem is closely related\(^3\) to point-pattern matching, several other methods available in the literature can be used directly or with little adaptation, e.g. those by Thrun and Liu (2005) and Cunningham et al. (2012). In any case, the choice of the matching algorithm is essentially an implementation issue that does not affect the general working principle of the PMR algorithm to be described.

**Correction (line 3):** This consists in replacing each roto-translation \( q \) with

\[
q_M = \arg\min_{q \in \text{SE}(2)} \sum_{(p', p'') \in B_q} \|p' - q \oplus p''\|^2. \tag{7}
\]

This means that, independently of how \( q \) was computed, it is corrected by choosing the roto-translation that minimizes the squared error over all of the pairs associated by \( B_q \). The minimization problem \( (7) \) can be solved by standard methods; in particular, in our implementation we have used the technique proposed by Umeyama (1991).

**Merging (line 4):** A merged set of labeled points \( L_M = (P_M, l_M) \) is generated as follows:

1. For each pair \((p' \in P', p'' \in P'')\) contained in \( B_q \), compute the ‘averaged’ point

\[
p = \frac{\kappa}{\kappa + 1} p' + \frac{1}{\kappa + 1} (q_M \oplus p''), \tag{8}
\]

and put it in \( P_M \). In this formula, \( \kappa \) is the generation counter of \( p' \); i.e. \( \kappa = 1 \) if \( p' \) is an observed point, \( \kappa = 2 \) if \( p' \) is the result of a previous binary registration between two observed points, and so on. The structure of Equation (8) guarantees that all of the points that have contributed to a certain average, either in the current or in previous binary registrations, are given the same weight.

2. Put in \( P_M \) each point \( p' \in P' \) that has not been associated. For each point \( p'' \in P'' \) that has not been associated, put in \( P_M \) the point \( q_M \oplus p'' \).

3. For averaged points, define the labeling map \( l_M \) by letting

\[
l_M(p) = \max\{l(p'), l(p'')\}. \tag{9}
\]

For the other points in \( P_M \), which come from \( P' \) or \( P'' \), the label is simply inherited from \( l' \) or \( l'' \).

Note that Equation (9) implies that an averaged point will be unlabeled if it comes from two unlabeled points, and will get the non-zero label if one of the two points is actually labeled. Remember that the two points \( p' \) and \( p'' \) cannot have different non-zero labels due to the definition of matching.

Wrapping up, a binary registration produces multiple pairs of the form \( M = (q_M, L_M) \), which actually consist of a (corrected) roto-translation \( q_M \) and the corresponding (merged) set of labeled points \( L_M \). Generalizing the concept introduced in Definition 1, in the following we call matching any pair \( M \) that comes out of a binary registration.
Consider now another set of labeled points \( \mathcal{R}_j \), whereas \( \mathcal{R}_1 \) and \( \mathcal{R}_3 \) are irreconcilable because the same point is labeled once as \( \mathcal{R}_1 \) and once as \( \mathcal{R}_3 \).

### 5.2. Irreconcilable matchings

Our multiple registration algorithm, to be discussed in Section 5.3, builds solutions (matchings) by performing recursive binary registrations on partial matchings. To reduce its running time, it is important to avoid the propagation of partial matchings that will eventually generate to the same solution. To this end, we introduce the following definition.

**Definition 3** (Irreconcilability). Consider two matchings \( M_1 \) (between \( \mathcal{L}' \) and \( \mathcal{L}'' \)) and \( M_2 \) (between \( \mathcal{L}' \) and \( \mathcal{L}''' \)), where it may be \( \mathcal{L}''' = \mathcal{L}'' \), and their merged sets of labeled points \( \mathcal{L}_{M_1} \) and \( \mathcal{L}_{M_2} \). \( M_1 \) and \( M_2 \) are said to be irreconcilable if one of the following holds for at least one pair of points \((p_1, p_2) \in M_1 \times M_2\):

1. \( p_1, p_2 \) are the same point (in the sense of (6)) but have different labels;
2. \( p_1, p_2 \) are different points but have the same non-zero label.

Figure 6 illustrates the two sources of irreconcilability. Irreconcilable matchings will always generate different solutions. In fact, assume we have two irreconcilable matchings \( M_1 \) (between \( \mathcal{L}' \) and \( \mathcal{L}''' \)) and \( M_2 \) (between \( \mathcal{L}' \) and \( \mathcal{L}''' \)), and their merged sets of labeled points \( \mathcal{L}_{M_1} \) and \( \mathcal{L}_{M_2} \). Consider now another set of labeled points \( \mathcal{L}''' \). Then, all of the matchings produced by binary registration of \( \mathcal{L}_{M_i} \) with \( \mathcal{L}''' \) will be different from the matchings produced by binary registration of \( \mathcal{L}_{M_2} \) with \( \mathcal{L}''' \).

### 5.3. Probabilistic multiple registration

We may now present PMR, our multiple registration algorithm. The inputs to PMR are the observations available to \( \mathcal{R}_t \) at time \( t \), i.e. \( O_t' \), \( \{O'_j\}_{j \in N_t'} \), and the motion-updated beliefs on \( \{q'_j\}_{j \in N_t'} \) coming from the filtering stage (see Figure 4). The outputs are estimates of the relative configurations \( q_t' \), \( j \in N_t' \).

A pseudocode description of PMR is shown in Algorithm 2. A detailed step-by-step illustration is given below.

![Figure 6](image-url)
The solution(s) computed by PMR can be further refined by performing a final batch least-squares (BLS) minimization. In fact, each solution entails a particular data association, and one may then consider the problem of adjusting the robot configuration estimates so as to simultaneously take into account all measurements that are included in that data association. The solution of this nonlinear least-squares problem, that can be found using numerical methods, would be more accurate that the solution produced by successive pairwise minimizations such as (7).

Note the following:

- Irreconcilability-based pruning (line 5) is aimed at reducing the computational load of PMR by pruning branches that will sooner or later lead to the same solution. No solution is lost by doing this.
- Belief-based pruning (line 6) is particularly useful to discriminate statically ambiguous situations (where the robots are in a rotational symmetric arrangement) by discarding labelings (i.e. solutions) that are inconsistent with the current belief.
- PMR will find and provide partial solutions in situations where a complete solution cannot be found (line 14 with M empty); in particular, if the framework underlying the given instance of the problem is non-rigid PMR will only return configuration estimates for robots belonging to a rigid component.
- The solution(s) computed by PMR can be further refined by performing a final batch least-squares (BLS) minimization. In fact, each solution entails a particular data association, and one may then consider the problem of adjusting the robot configuration estimates so as to simultaneously take into account all measurements that are included in that data association. The solution of this nonlinear least-squares problem, that can be found using numerical methods, would be more accurate that the solution produced by successive pairwise minimizations such as (7).

Fig. 7. Execution of PMR in a non-ambiguous situation with three robots and a deceiving obstacle: (a) spatial arrangement and measurements; (b) initial sets of labeled points corresponding to the observations; (c) matchings produced by binary registration of \(L'_1\) and \(L'_2\), respectively, with \(L'_1\); (d) matching retained after irreconcilability-based pruning; (e) matching retained after belief-based pruning; (f) matching produced by binary registration of \(L'_k\) with \(L\); (g) final solution.

5.4. Illustrative examples

To clarify how PMR works, we work out two simple examples in the following.

Figure 7 describes the execution of PMR by \(\mathcal{R}_i\) in a non-ambiguous situation involving three robots (\(\mathcal{R}_i\), \(\mathcal{R}_j\), \(\mathcal{R}_k\)) and an obstacle which is erroneously detected as a robot (Figure 7(a)). The objective of the algorithm is to perform a registration of the sets \(L'_j\) and \(L'_k\), obtained from the communicated observations \(O'_j\) and \(O'_k\), and stored in \(L\) during the initialization phase, with the set \(L'_i\) obtained from the observation \(O'_i\) (Figure 7(b)).

During the first iteration, PMR performs \(|N'_i| = 2\) binary registrations of \(L'_j\) and \(L'_k\), respectively, with \(L'_i\) obtaining two matchings that are not irreconcilable (Figure 7(c)).
pruning, only one of them (the matching between \( L'_i \) and \( L'_j \)) is retained. This matching also survives belief-based pruning (Figure 7(d)). At this point, \( L'_i \) is deleted from \( L \) and \( L \) is accordingly updated (Figure 7(e)).

In the second iteration, registration with \( L'_i \) produces a better matching than before (Figure 7(f)), with three associated pairs instead of the previous two. Having passed belief-based pruning (Figure 7(g)), this matching represents the final solution. In particular, the algorithm returns the estimated configurations stored in \( Q \), i.e. \( \hat{q}_i \) and \( \hat{q}_j \).

This example well illustrates the incremental nature of PMR, and in particular how the iterative registration mechanism allows to localize robots that are not directly detected by \( R_i \). Note also that the deceiving obstacle is automatically ‘rejected’ because no communicated observations fits its location. However, it is useful because it acts as a landmark improving the quality of the matching and, ultimately, the localization accuracy.

To appreciate how PMR handles ambiguous situations, consider instead the example of Figure 8, in which the spatial arrangement of the robots is rotational symmetric (Figure 8(a)). As before, the objective of the algorithm is to perform a registration of the sets \( L'_i \) and \( L'_j \), obtained from the communicated observations \( O'_i \) and \( O'_j \) and stored in \( L \) during the initialization phase, with the set \( L'_i \) obtained from the observation \( O'_i \) (Figure 8(b)).

During the first iteration, PMR performs \(|N'_i| = 2| \) binary registrations of \( L'_i, L'_k \), respectively, with \( L'_i \), obtaining four matchings (Figure 7(c)). A maximal subset of two irreconcilable matchings (Figure 8(d)) is then retained. These are then rated on the basis of their fitness w.r.t. the current belief; in this case, the current belief indicates that only one of the two matchings can be accepted (Figure 7(e)). At this point, \( L'_j \) is deleted from \( L \) and \( L \) is accordingly updated.

In the second iteration, registration of \( L'_j \) with \( L \) produces (as in the previous case) a better matching than with \( L'_j \) only (Figure 7(f)), with three associated pairs instead of the previous two. After belief-based pruning (Figure 7(g)), this matching survives as the final solution and the outputs of PMR are the estimated configurations stored in \( Q \), i.e. \( \hat{q}_j \) and \( \hat{q}_k \).

5.5. Analysis

The run time of PMR, which accounts for most of the cycle time of our mutual localization system, depends on the number \(|N'_i| \leq n \) of observations that \( R_i \) receives (\( n \) is the unknown number of robots).

In normal operation (no ambiguity, or ambiguities that can be resolved by belief-based pruning), PMR expands a single branch, which requires \((|N'_i| - 1)(|N'_i| - 2)/2\) binary registrations to produce a solution: in turn, each binary registration requires constant time if performed with the technique described by Franchi et al. (2009b). This leads to a worst-case complexity \( O(n^2) \), while the average-case complexity can be significantly lower if the number of neighbors (robots communicating with \( R_i \)) is smaller than \( n \).

In the presence of ambiguities that cannot be resolved, the algorithm expands a branch for each solution, with the complexity of each branch being that of normal operation. In principle, however, the number of solutions can grow quite large. In particular, it is easy to prove that both the lower bounds of Proposition 2 have a maximum value of \( N_i! \) (set \( \lambda = N_i + 1 \) if the centroid is occupied by a robot, \( \lambda = N_i \) otherwise).

One way to keep the complexity polynomial in spite of the presence of ambiguities would be to modify PMR so as to perform at each \( t \) a random partial (rather than complete)
exploration of the space of possible solutions. In any case, especially in the early stages of localization, it is advisable to break rotational symmetry whenever possible, e.g., by using an anti-symmetry control law (Franchi et al., 2010a).

6. Filtering stage

In the proposed mutual localization system, the multiple registration stage represented by PMR is followed by a filtering stage that takes into account the odometry measurements to reduce the noise, prune ambiguous solutions based on the current certainty and finally compute a belief on \( q'_j \), for \( j \in N^i_t \). The multi-hypothesis nature of the problem suggests the use of particle filters, since they are intrinsically multi-modal.

In principle, \( R_i \) should maintain a single particle filter to compute a joint belief \( \text{bel}(\{q'_j\}_{j \in N^i_t}) \). However, this is not feasible from a computational point of view, since the dimension of the filter would grow linearly with the number of robots, and the number of particles required to represent its state would grow exponentially. To overcome this problem, \( R_i \) maintains one particle filter for each \( R_j \), computing a separate belief \( \text{bel}(q'_j), j \in N^i_t \), instead of a single joint belief. This choice, already implicit in the formulation of Problem 1, relies on the independence assumption

\[
p(q'_j \mid j \in N^i_t) = \prod_{j \in N^i_t} p(q'_j),
\]

which is acceptable in a pure localization scenario. A drawback of this choice is the loss of the explicit mutual exclusion guaranteed by the multiple registration procedure (i.e., the fact that a measurement is associated to a single robot). However, the mutual exclusion implicitly stands, since it is already included in the measurements feeding the filter, which are the outputs of PMR.

At each time instant \( t \), the \( j \)-th filter (\( j \in N^i_t \)) receives as inputs the motion displacement \( \delta'_j \) of \( R_i \) plus, only for those \( j \in N^i_t ;
\)

1. the motion displacements \( \delta^{(\tau+1):t} \) (sent by \( R_i \)), where \( \tau_j \) is the last time instant before \( t \) in which \( R_j \) communicated with \( R_i \);
2. the set of all the relative configuration estimates \( \{q'_j\} \) computed by PMR for \( R_j \).

The filter equations are very similar to those of Howard et al. (2003). In particular, the update rule that accounts for the motion of \( R_i \) is

\[
p(q'_j | \delta'_j) = v_i \int p(d | \delta'_j) \text{bel}(q'_{j-1})|_{q'_{j-1} = q'_{j-1} \oplus d} \, dd,
\]

where \( v_i \) is a normalization factor and \( p(d | \delta'_j) \) is the model of the odometer. Equation (13) leads to the following update for the single particle of value \( q'_j \):

\[
\tilde{q'_j} = q'_j \oplus (\delta'_j + \nu \delta),
\]

where \( \nu \delta \) is a sample taken from \( p(d | \delta'_j) \). Similarly, the update rule that accounts for the motion of \( R_j \) is

\[
p(q'_j | \delta'_j) = v_j \int p(d | \delta'_j) \text{bel}(q'_{j-1})|_{q'_{j-1} = q'_{j-1} \oplus d} \, dd,
\]

where \( v_j \) is a normalization factor, and the update equation for the single particle is

\[
\tilde{q'_j} = q'_j \oplus (\delta'_j + \nu \delta).
\]

The consecutive application of Equations (13) and (15) allows the computation of the prediction

\[
\text{bel}(q'_j) = p(q'_j | \delta'_j, \delta'_j).
\]

After the update, the motion of \( R_i \) and \( R_j \) causes a translation of \( p(q'_j) \), while the additive noise introduces a blur, i.e., a decrease in sharpness of the probability density by dispersing the particles.

In normal operation, \( \tau_j = t - 1 \) and \( R_j \) uses \( \delta'_j \) as motion measurement for the motion update of the robot \( R_j \). However, if \( R_i \) and \( R_j \) have not communicated over a non-zero time interval \([\tau_j + 1, \ldots, t - 1]\), the motion update of \( R_j \) has not been performed since \( \tau_j \). When, at \( t \), communication is resumed, \( R_j \) uses the whole set \( \delta^{(\tau+1):t} \) as motion measurement to perform a cumulative motion update.

As for the measurement update, assume first that the set of relative configuration estimates \( \{q'_j\} \) produced by PMR is composed by a single element. Then, using Bayes law, the measurement update is given by

\[
\text{bel}(q'_j) = v' \int p(q'_j \mid q'_j) \text{bel}(q'_j),
\]

where \( v'_j \) is another normalization factor, and the probabilistic model \( p(q'_j \mid q'_j) \) is the same appearing in Equation (12). When the set \( \{q'_j\} \) consists of more than one element, each of them is equally probable, so that using the theorem of total probability we obtain the update equation:

\[
\text{bel}(q'_j) = v' \sum_{q'_j \in \{q'_j\}} p(q'_j \mid q'_j) \text{bel}(q'_j),
\]

with \( v'_j \) different from that in (18).

A number of standard practical techniques have been used to improve the performance of the filter. For example, the initial prior distribution for \( R_i \) is generated using the first configuration estimate(s) \( \{q'_1\} \) computed by PMR. Moreover, at each step a small percentage of particles are re-initialized using directly the new measurements; for example, this enables the localization system to handle the kidnapped robot situation (see the last experiment in the next section).

7. Experimental validation

Before presenting experimental results, we provide a description of our implementation setup.
7.1. Hardware/software setup

The proposed mutual localization algorithm has been tested on a team of four differential-drive Khepera III robots moving in a 3.6 m × 1.9 m wide arena. In all of the experiments, mutual localization is performed from the perspective of robot $R_1$ ($i = 1$). To obtain a ground truth against which to evaluate the accuracy of the proposed localization algorithm, we have used an external three-camera system which is able to track the robots with errors on the position and orientation that are less than 0.01 m and $5^\circ$, respectively.

Motion control, sensor data management, and localization have been implemented using MIP, an in-house developed software platform specifically aimed at multi-robot systems. The source code of the localization algorithm is also available in the Mutual Localization with Anonymous Measurements (MLAM) ROS stack.

Each Khepera III robot has been equipped with a Hokuyo URG-04LX laser sensor that provides bearing-plus-range (i.e. position) measurements at 10 Hz within a 240$^\circ$ field of view (thus leaving a 120$^\circ$ blind zone behind the robot). The robot detector is then implemented as a memoryless feature extraction algorithm that inspects the laser scan, looking for the indentations caused by a white cardboard box mounted on top of each robot. All cardboard boxes have the same shape and size, making impossible to retrieve the identity of the detected robot on the basis of the profile of the indentation. Each robot measures its own displacement by odometric integration over a 0.1 s time interval. Simultaneous calibration of odometric and sensor parameters was performed using the algorithm by Censi et al. (2013). Communication introduces no significant delay.

The binary registration used in PMR is a RANSAC-based algorithm (Franchi et al., 2009b) with a maximum of 100 trials and a tolerance $\eta = 0.06$ m for point association. Each particle filter is run with 300 particles. The whole mutual localization algorithm by robot $R_1$ (PMR + particle filters) runs at 10 Hz on an entry-level GNU-Linux computer, proving that it can be easily executed in real time in a practical applicative scenario.
7.2. Experimental results

Figure 9 summarizes the results of the first experiment. The top part of the figure shows four snapshots of the scene at successive instants of time with the corresponding relative configuration estimates computed by $R_1$ in the form of clouds of particles. The orientation of each robot, which is actually included in the estimates, is not shown by the particle. The bottom part of the figure shows the norm of the position error (left) and the orientation error (right), defined respectively as $|p_j^t -  \bar{p}_j^t|$ and $\theta_j^t - \bar{\theta}_j^t$, for $j = 2, 3, 4$, with $q_j^t = (p_j^t, \theta_j^t)$ the ground truth and $\bar{q}_j^t = (\bar{p}_j^t, \bar{\theta}_j^t)$ the weighted average of the particle distribution for the robot $R_j$.

The four robots start in a square spatial arrangement, which is rotational symmetric and hence ambiguous (first snapshot). Only one robot ($R_{4}$) moves over a closed path (second and third snapshots) to conclude the experiment in the starting configuration (fourth snapshot). At the beginning, PMR finds $3! = 6$ solutions (as predicted by Equation (4) with $\lambda = 4$ and $N = 3$) which it is not able to prune due to the absence of a priori discriminating knowledge. The result is that the particle filters are initialized at three different configurations for each robot, so that the initial clouds of particles are multi-modal and overlapping. Accordingly, the errors are quite large in the first part of the experiment. When the symmetry is broken ($t \approx 10$ s) PMR is able to extract a single solution, allowing in turn the filters to disambiguate the estimates; as a consequence, the clouds of particles separate and the errors quickly decrease. In the rest of the experiment the localization system uses new measurements to maintain the accuracy of the estimates. The particle distributions clearly indicate the progressive reduction of uncertainty. When the experiment ends, the spatial arrangement is again rotational symmetric, but PMR is now able to extract a single solution thanks to the use of the current belief.

The objective of the second experiment, shown in Figure 10, is to test the robustness of the system with respect to false positives and negatives of the detection process. To this end, we introduced two obstacles that are seen as robots by the robot detector, and moved all of the robots so as to generate temporary occlusions. At the beginning of the experiment, robots and obstacles are arranged in a regular lattice (first snapshot) and there is an ambiguity in the configuration of $R_3$ and $R_4$. In particular, according to PMR, $R_3$ may be placed at the four corners of the rectangle, whereas $R_2$ may be located either at its true position or at the specular corner. This particular situation is due to the fact that while $R_4$ detects the obstacle in the opposite corner, $R_3$ does not due to a momentary failure of the detection process (note that the theoretical analysis of Section 3 does not directly apply due to the presence of false positives). The robots start moving simultaneously (second snapshot), under the action of a decentralized pseudo-random control law with integrated obstacle avoidance. After the symmetry is broken ($t \approx 8$ s), PMR is once again able to extract a single solution, the clouds of particles separate, and the errors suddenly decrease. Then, the localization system continues to improve the estimates by the use of the measurements (third snapshot).

Note that $R_1$ does not detect all of the other robots through the whole experiment, due to the limited field of view of the range finder and reciprocal occlusions. For example, the second snapshot shows that the line of sight between $R_1$ and $R_3$ is almost occluded by $R_4$, so that a false negative is generated around the corresponding time instant. However, the estimate is not affected by this temporary lack of measurements, because PMR is still able to place $R_3$ by incorporating the observations of other robots that see it. Even when PMR does not produce valid matchings, the particle filters are still updated by incorporating odometric measurements; for example, this happens in the situation of the third snapshot, in which $R_1$ is facing the outer wall and all of the other robots are in the blind zone of its detector.

As discussed at the end of Section 5.3, one may perform a batch least-squares optimization with known correspondences to further refine each solution found by PMR before feeding it to the filtering stage. Figure 11 shows the effect of this refinement on a sample solution computed by PMR during the second experiment. The BLS problem was solved by the MATLAB function lsqnonlin in 0.204 s. The entity of the correction on the robot configuration estimates is limited: the mean square difference is 0.72 cm for the position and 1.66° for the orientation. Therefore, to avoid the added computational load, we chose not to integrate the BLS refinement step in our localization system.

In the third experiment (Figure 12), the robots start in a non-symmetric spatial arrangement and are driven by the same decentralized pseudo-random control as before. As expected, the particle distributions are well separated and single-modal from the very beginning. At time $t \approx 20$ s, $R_3$ (green) is kidnapped (first snapshot) and released in a different position (second snapshot). The localization system quickly recovers the correct estimate of its relative configurations, as confirmed by the sample distributions in the third snapshot. Note that when the green robot is kidnapped, the belief on its configuration is still updated on the basis of odometric information, even though the underlying framework is momentarily non-rigid due to the lack of the measurements of the green robot.

Extension 1 contains video clips of these experiments. See http://www.dis.uniroma1.it/~labrob/research/mutLoc.html for additional material, including other experiments.

8. Discussion and comparison with other possible approaches

This section provides first a short discussion on some general aspects of the proposed localization system and then
Fig. 10. Second experiment. Top: snapshots of the scene and sample distributions computed for $R_2$ (red), $R_3$ (green) and $R_4$ (blue) by the mutual localization system of $R_1$ (yellow, circled); the small triangles represent the ground truth and the black circles represent the obstacles. Bottom: norm of the position error and orientation error affecting the configuration estimates of $R_2$, $R_3$ and $R_4$.

8.1. Discussion

The experiments of the previous section highlight the conditions needed in order to have an effective localization. The presented method requires that every robot sends its measurements to its communication neighbors and that these measurements are sufficiently synchronized. This is not a technical issue, since with the current communication technology it is possible to synchronize two computers in a network with an accuracy of 1 ms or less.

The number $n$ of robots must not be known in advance, and may indeed vary during the operation. In fact, whenever a new robot enters in the communication neighborhood of $R_i$, and the PMR algorithm is able to formulate a hypothesis on its configuration, a new particle filter is initialized for that robot.

Also, rigidity of the framework underlying the problem instance is not necessary for the proposed localization system to work. In fact, when the framework is non-rigid, PMR will still find a partial solution that provides configuration estimates for robots belonging to its rigid component (see the discussion at the end of Section 5.3). In addition, the filtering stage will exploit odometric measures to propagate current estimates, even for those robots for which PMR is unable to provide an estimate.

Finally, the proposed localization method effectively rejects outliers, since they are either removed by belief-based pruning (if they are inconsistent with the current belief) or they remain unlabeled, hence ‘inactive’, at the end of PMR.

8.2. Comparison with other possible approaches

As mentioned in the introduction, it appears that the problem formulated in this paper (Problem 1) has not been
The state of each particle consists of the state of the robot feature-based SLAM is performed using a particle filter. The distribution presented by Montemerlo and Thrun (2003), in which we adopt the strategy of FastSLAM with unknown data associations. At each step, one could 'best' data associations at each step. To do this, one could consider multiple robots, and allowing the incorporation of mutually exclusive and negative information.

In order to perform a comparison, we have run the FastSLAM-like mutual localization method on the same experiments presented in Section 7. To achieve a rough equivalence to our method from a computational viewpoint, we first implemented the FastSLAM-like method with 100 particles. For example, in the first experiment the maximum value of $|N|$ is 3 and this would lead to 300 EKFs, comparable to the 300 particles used by our method. Figure 13 shows the performance of the FastSLAM-like method on the first experiment data. Clearly, localization is not successful, since the orientation estimates are affected by significant errors.

In view of the above results, we increased the number of particles used by the FastSLAM-like method to 1000. Considering that each particle includes a number of EKFs equal to the number of observed features, this brings the computational load to 3000 EKFs in the first experiment and 5000 EKFs in the second experiment, i.e. 10 times the load of our method. As shown by Figure 14, the increase in the number of particles was beneficial, since the performance of the FastSLAM-like method becomes comparable to that of our method (compare with the error plots in Figure 9).

![Fig. 11. The effect of the batch least-squares refinement (below) on a sample solution produced by PMR during the second experiment (above). Mutual localization is performed by $R_1$, shown as a yellow triangle, with its observed points marked by yellow circles. The red, green, blue triangles and asterisks represent, respectively, the estimated configurations of $R_2, R_3, R_4$ and their observations. A zoom around $R_2$ is provided to highlight the slight refinement.](image-url)
Still, the convergence of the FastSLAM-like method once the symmetry is broken is considerably slower.

Results of the FastSLAM-like method (still with 1000 particles) in the second and third experiments are reported in Figures 15 and 16, respectively (to be compared with Figures 10 and 12, respectively). In spite of the much higher computational load, the FastSLAM-like method cannot produce reliable estimates in these two experiments due to their increased difficulty. For example, while our method achieves good performance throughout the third experiment, solving the kidnapping issue in few seconds, the FastSLAM-like method never recovers the actual arrangement of the group, although the correct configuration of some robots is occasionally reconstructed (e.g. in the second experiment, $R_3$ after $t = 25$ s).

The superior performance of our method can be explained by the fact that the mutual exclusive structure of the solution (i.e. the fact that the same measurement cannot be associated with different robots) is directly exploited in the multiple registration phase, whose complexity is quadratic with $|N_i|$ in normal operation. The successive filtering phase ‘forgets’ the mutual exclusion by keeping a bank of independent particle filters. However, as already mentioned, mutual exclusion implicitly stands, since the input data for the filters are the solutions of the multiple registration process. Moreover, the increased dimension of the measurements (augmented with the orientation by PMR) allows the particle filters to converge very quickly.

9. Source code release

A C++ code that implements the mutual localization algorithm described in this paper has been released within the MLAM ROS stack, which is available at http://www.ros.org/wiki/mlam. Note that the code is released under the BSD license, therefore it is provided ‘as is’ and we cannot be held responsible for any damage or harm caused by the use of our software.
Fig. 13. First experiment with the FastSLAM-like method using 100 particles. Norm of the position error (top) and orientation error (bottom) affecting the configuration estimates of $R_2$ (red), $R_3$ (green) and $R_4$ (blue).

Fig. 14. First experiment with the FastSLAM-like method using 1000 particles. Norm of the position error (top) and orientation error (bottom) affecting the configuration estimates of $R_2$ (red), $R_3$ (green) and $R_4$ (blue) by the FastSLAM-like method.

Fig. 15. Second experiment with the FastSLAM-like method using 1000 particles. Norm of the position error (top) and orientation error (bottom) affecting the configuration estimates of $R_2$ (red), $R_3$ (green) and $R_4$ (blue) by the FastSLAM-like method.

Fig. 16. Third experiment with the FastSLAM-like method using 1000 particles. Norm of the position error (top) and orientation error (bottom) affecting the configuration estimates of $R_2$ (red), $R_3$ (green) and $R_4$ (blue) by the FastSLAM-like method.

10. Conclusions
We have presented and experimentally validated a novel mutual localization algorithm for the case of anonymous position measurements. As explained in the paper, this is a challenging and practically relevant operating scenario that has received little attention in the literature.
The method is based on a two-step approach: a probabilistic multiple registration algorithm, which provides all data associations that are consistent with the relative robot measurements and the current belief, as well as the robot orientations; and a dynamic filtering stage, which incorporates odometric data into the estimation process. We designed the proposed method proceeding from a detailed formal analysis of the implications of anonymity on the mutual localization problem.

Experimental results on a team of differentially driven mobile robots have shown the effectiveness of the approach, and in particular its robustness against false positives and negatives that may affect the robot measurement process. We have also provided an experimental comparison that shows how the proposed method outperforms more classical approaches that may be designed building on existing techniques. Finally we have made available the source code of the proposed method within the MLAM ROS stack.

We are currently working on the development of a similar localization approach for groups of 3D agents (i.e. in \( SE(3) \)) and on its application to swarms of flying robots, possibly equipped with more limited (e.g. bearing-only plus IMU) sensors. In this direction, some preliminary results have been recently presented by Stegagno et al. (2011) and Cognetti et al. (2012).

Notes
1. Here, ‘similar’ must be intended in the sense of the specific attribute used to detect robots, be it shape, color, etc.
2. Since in this section we are considering a static version of the problem we shall omit all of the \( t \) superscripts. Note also that, for compactness, symbols introduced and used here but not in the rest of the paper are not included in Table 1.
3. Strictly speaking, the two problems are not equivalent due to the fact that we deal with sets of labeled points.
4. If both matchings had passed belief-based pruning (for example, because the current belief on the configurations of \( R_j \) and \( R_k \) was uniform), the algorithm would have created two branches, ultimately leading to multiple different solutions. This kind of situation occurs in the early stages of the first and second experiments, see Section 7.
5. See http://www.dis.uniroma1.it/labrob/software/MIP.

Funding
This work was partially supported by the PRIN program of the Italian Ministry of Education, University and Research (MEMO-NET Project).

References


Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Video</td>
<td>Three experiments of mutual localization using anonymous measurements</td>
</tr>
</tbody>
</table>