Swarm Localization through Cooperative Landmark Identification

Sarah Brent, Chengzhi Yuan, and Paolo Stegagno

University of Rhode Island, ICRobots Lab, Kingston RI 02881, USA,
{sbrent, cyuan, pstegagno}@uri.edu

Abstract. In this paper we propose a landmark-based map localization system for robotic swarms. The proposed system leverages the capabilities of a distributed landmark identification algorithm developed for robotic swarms presented in [1]. The output of the landmark identification consists of a vector of probabilities that each individual robot is looking at a particular landmark in the environment. In this work, this vector is used individually by each component of the swarm to feed the measurement update of a particle filter to estimate the robot location. The system was tested in simulation to validate its performance.

Keywords: swarm, localization, landmark identification, sensor fusion

1 Introduction

A robotic swarm generally consists of relatively inexpensive autonomous robots with limited computational, sensing and communication capabilities. In particular, robots may not have access to global information as their position, the position of their teammates, and commands of a centralized controller. Nevertheless, from the local behavior of the single robots and their interaction with the teammates, a global behavior should emerge to collectively perform some desired task [2]. However, many applications proposed in recent years require some degree of global knowledge, for example search and rescue [3], target search and tracking [4], information gathering and clean up of toxic spills [5], and even construction [6]. In these applications, the knowledge of global position would be beneficial or required for the task execution. Nevertheless, the availability to GPS may be limited by operational conditions as indoor environment.

Many works have addressed the problem of cooperative localization of a multi-robot system in a global frame of reference by using relative measurements between robots (e.g., [7]) and/or relative measurements of landmarks in known locations (e.g., [8]). Reliable relative measurements between robots may be difficult to achieve in practice in a swarm setup, unless dedicated hardware is mounted on the robots (e.g., [9]). Moreover, even if some measurement (e.g., position, bearing, distance) was available through a general purpose sensors as a camera, still the problem of associating the measurements to the specific id of the robot would require dedicated data association algorithms [10].

In this work, we propose a different vision-based approach that takes into account the challenges and limitations posed by a real robotic swarm. First, the robots are equipped with only a general purpose sensor as a camera. Secondly, we assume no relative measurements are available between robots. Although cameras would still allow for relative bearing measurements, due to the swarm setup reliably tagging the robots would not be feasible and we would therefore fall back into a data association problem. However, uniquely identifiable landmarks at known locations are available in the environment, although metric information as relative position or distance from the landmarks is not available. In this setup, each robot could independently perform a landmark based localization.

Single robot (or single sensor) landmark localization has been addressed deeply and broadly for many applications in robotics through either artificial or natural landmarks. An exhaustive review on this topic is outside the scope of this work, that is more focused on multi-robot landmark localization, and the following references should be taken as examples of current state of the art. In [11], authors employ a method that uses a single image from a camera and a minimum of three feature points to recover the camera's viewpoint. In [12], artificial landmarks on the ceiling in an indoor environment are used to localize a two degrees of freedom camera. In [13], authors create and maintain a sparse set of landmarks that are based on biologically motivated feature selection. In [14], the authors use a real-time camera feed from a drone and an AR tag to compute the position of a drone with respect to a point of origin. In [15], the author uses a 2D bar-code landmark for the self localization of mobile robots. In [16], authors propose a method for the global localization problem that uses two landmarks to localize the pose of the robot using bearing angle and distance of landmarks to calculate a possible area of the location of the robot and the particles. In [17], the authors present a landmark matching, triangulation, reconstruction and comparison algorithm that extracts natural landmarks to estimate the position of a robot. In [18], the authors propose a method for robot localization based on the shape and size changes of an object in the robot view as well as trigonometric concepts. In [19], authors deploy a localization method based on artificial as well as natural landmarks employing model based object recognition.

In general, many works proposed on this topic focus not only on the estimation of the robot position, but also on the selection and identification of the landmarks (e.g., [20]). Identification algorithms, however, can often provide wrong results, in particular when many different landmarks are present in the environment, and computational capabilities are limited. In a recent paper [1], we have proposed a system for cooperative identification of landmarks in a robotic swarm, in which the results of single-robot relatively shallow and low-accuracy Convolutional Neural Networks (CNNs) are shared among a robotic team to improve the overall accuracy. The system proposed in [1] is an extension of a previous paper [21] dealing with cooperative object recognition.

In this paper, each robot of the swarm will use the output of the system proposed in [1] to feed a particle filter and estimate its position. This approach is considerably different with respect to state-of-the-art landmark-based cooperative localization systems in literature. In several papers on cooperative localization (e.g., [22–25]) robots use each others as landmarks to improve odometric localization. Other authors [26–28] proposed to use relative position or distance measurements of the landmarks to compute estimates of the robots location via geometric considerations and sensor fusion techniques. In [29], each robot maintains its landmark-based pose estimate using an Unscented Kalman Filter in a common map. In a few papers, specific hardware is developed to use radio [30] or acoustic [9] landmarks and beacons.

None of the above mentioned papers is compatible to our assumptions, due to the use of relative and/or metric measurements, or additional hardware. Lidar is often used to address this type of problem, but there are several issues with using lidar. First, low-cost lidars usually have limited field of view. More expensive lidars can have up to several tens of meters of field of view but are not compatible with a swarm setup. Moreover, the use of lidars in localization usually requires a complete occupancy map of the environment, while we only assume that we know the coordinates of the landmarks in the world frame. From these considerations, we decided to use cameras as the only sensor for our localization algorithm, since for a camera a landmark will be recognizable from multiple distances and even with the interference of possible foreign smaller objects. To the best of our knowledge, this paper is the first to address explicitly the problem of vision-based localization with cooperative landmark identification in robotic swarms.

The rest of the paper is organized as follows. Section 2 will introduce the problem settings, including the robot model and sensor equipment, the communication graph as well as the formal definition of the localization problem of the swarm using cooperative place recognition. Section 3, will describe the methodology used including the system architecture, a recall of the cooperative landmark identification, as well as a description of the particle filter. In Section 4, a description of the simulation platform used to validate the localization system. Finally Section 5 will conclude the paper.

2 Problem Setting

Let $A = \{A_1, A_2, ..., A_n\}$ be a multi-robot system consisting of n agents. The generic robot A_i, $i = 1, \ldots, n$ moves in a 3D environment populated with a set $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$ of m objects ω_l, $l = 1, \ldots, m$. Its configuration $\mathbf{c}_i(k) = [\mathbf{c}_i(k) \;\; \phi_i(k)]^T$ at time step k in a world frame $W = O - XYZ$ is described by the position $\mathbf{q}_i(k) = [x_i(k) \; y_i(k)]^T \in \mathbb{R}^2$ and orientation $\phi_i(k) \in SO(2)$ of a frame attached to a representative point. A_i is modeled as a unicycle:

$$\begin{bmatrix} x_i(k) \\ y_i(k) \\ \phi_i(k) \end{bmatrix} = \begin{bmatrix} x_i(k-1) \\ y_i(k-1) \\ \phi_i(k-1) \end{bmatrix} + \begin{bmatrix} T\cos\phi_i(k) & 0 \\ T\sin\phi_i(k) & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} v_i(k) \\ \eta_i(k) \end{bmatrix}, \tag{1}$$

where v_i(k) and $\eta_i(k)$ are the linear and angular velocities respectively, and T is the duration of the time step. In general, robots are not aware of their global positions in W, nor they have access to each other's relative position or bearing. However, we assume that the robots are able to communicate with each other within a certain range r. Hence, we define the communication graph as an ordered pair $G(k) = (\mathcal{N}, \mathcal{E}(k))$ consisting of nodes $\mathcal{N}$ (the robots) and edges

$\mathcal{E}(k)$. Note that in general the communication graph is time variant. An edge $e = \{i, j\}$ is an unordered pair such that if $\{i, j\} \in \mathcal{E}(k)$, robots A_i and A_j can communicate at time step k. This implies that the underlying communication graph is undirected, i.e., if A_i communicates with A_j, then conversely A_j can communicate with A_i. We also will be operating under the assumption that the communication graph is connected, i.e., there is a path between any two nodes of the graph. However, it should be noted that if the communication network gets disconnected, the swarm will continue working as two or more separate subgroups and will perform the algorithm on the respective sub-networks. It is outside the scope of this paper to study the problem of controlling the swarm so that this assumption is verified. However, it can be achieved assuming that the robots move according to a connectivity maintenance swarming algorithm as the one proposed in [31].

Each A_i is equipped with and odometry module, that provide measurements $\mathbf{u}_i(k) = [\bar{v}_i(k)\ \bar{\eta}_i(k)]^T$ of the linear and angular velocity at all time steps. Each A_i is also equipped with a camera and gathers images $z_i(k)$ of an object $\omega^i(k) \in \Omega$, where the superscript i identifies the specific object observed by robot A_i at time step k. In general, different robots can potentially observe different objects in the environment at the same time. In addition, each robot A_i will be able to collect measurements $\bar{\phi}_i(k)$ of its own yaw angle $\phi_i(k)$ in the world frame W through a magnetometer. In the following, we will indicate with $Z(k) = \{z_i(k), i = 1, 2, ..., n\}$ the set of exteroceptive measurements collected by all the robots at time step k, and with $\Phi(k) = \{\bar{\phi}_i(k), i = 1, 2, ..., n\}$. Collectively, we will indicate with $Z_\Phi(k) = \{Z(k), \Phi(k)\}$ the set of all camera and yaw measurements.

Finally, we define n random variables $O^i(\omega, k), i = 1, \ldots, n$, one for each A_i, that represents the objects observed by $A_i, i = 1, \ldots, n$ at time step k:

$$O^i(\omega, k) = O^i(k) = l \Leftrightarrow \omega^i(k) = \omega_l \tag{2}$$

Then, the probability $p(O^i(k) = l) = p(O^i(k))$ is the probability that $\omega^i(k) = \omega_l$.

The problem that we will address in this work is formally introduced as:

Problem 1. *The problem of localizing the agents in the world frame W is the problem of computing an estimate $\hat{\mathbf{c}}_i(k), i = 1, \ldots, n$ of their configurations $\mathbf{c}_i(k), i = 1, \ldots, n$ at each time step k given all exteroceptive and yaw measurements $Z_\Phi(s), s = 1, \ldots, k$ and all odometry measurements $\mathbf{u}_i(s), s = 1, \ldots, k$ at all time steps up to k.*

3 Methodology

3.1 System Architecture

The block scheme of the system running on each robot A_i is depicted in Fig. 1. The image collected by the camera, A_i's measurement $z_i(k)$, is passed through an AI classifier to determine which object A_i is observing. The output of the classifier is the m-vector of probabilities $P(z_i(k)|O^i(k))$ that A_i is observing ω_l.

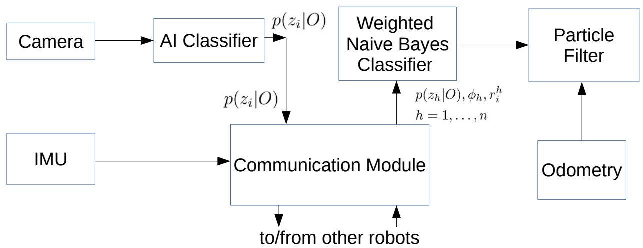


Fig. 1: Block scheme of the system running on robot A_i.

This information is then provided to the communication module that broadcast it to A_i's communication neighbors together with the measured yaw angle $\phi_i(k)$.

As all robots do the same, A_i's communication module also receives the probability vectors $P(z_h(k)|O^h(k))$ and the yaw angles $\phi_h(k)$, $h = 1, \ldots, n$, $h \neq i$ of all other robots in the swarm. The communication module also implements a gossiping algorithm so that each robot A_i in the team can receive the probability vectors and yaw angle measurements of all the robots in the swarm, even the ones that are not directly communicating with A_i itself. With an appropriate communication protocol, A_i also compute an estimate $r_i^h(k)$ of the communication distance between itself and the generic robot A_h, $h = 1, \ldots, n$, $h \neq i$. The communication distance is the number of communication steps that are needed for a message sent from robot A_h to reach robot A_i at time step k, and is equivalent to the graph length of the shortest path that connects nodes i and h in the communication graph $G(k)$.

The probability vector computed by the A_i's AI classifier, as well as the ones received by the other robots are passed to the Weighted Naive Bayesian Classifier (WNBC) together with the yaw angles $\phi_i(k)$, $\phi_h(k)$ and the estimated communication distances $r_i^h(k)$. This information is used by the WNBC to iteratively compute $P(O^i(k)|Z_\Phi(k))$. A_i uses this information computed locally to feed the measurement update of a particle filter that produces the estimate $\hat{\mathbf{c}}_i(k)$. The odometry measurements are used in the time update.

3.2 Cooperative Landmark Identification

This subsection recalls the concepts presented in [1] to perform the instantaneous cooperative landmark identification needed to feed the particle filters. Since the algorithm is instantaneous, for the sake of clarity in this section we will drop the time dependency represented by (k) in all variables. Each robot uses a standard single-view recognition algorithm, a convolution neural network (CNN) on the Tensorflow platform. CNN's are frequently used with image data for recognition purpose. To set up the individual CNN's, we have created a training and a testing dataset on the simulated world that we will be using to demonstrate our cooperative algorithm. In that world, we have deployed 17 buildings that are used as landmarks for the localization problem. Each dataset includes tens of images for each building collected from different points of view. A 5-layered CNN was learned using the training data set. The low number of layers in the relatively shallow CNN was mandated by the limited computational capabilities of the hardware this algorithm is meant for, the onboard computer of the robots. Then, we evaluated the single robot recognition capabilities with the testing dataset, resulting in a single robot correct recognition rate of 77%. The output of our CNN is a probability vector $p(z_i|O^i)$, which describes the probability that the observed object is of a certain known type.

Each A_i communicates its computed $p(z_i|O^i)$ over the network, together with its measured yaw angle ϕ_i. This means that the communication neighbors of R_i will receive its measured probabilities and yaw angle. However, every member of the team eventually needs to receive $p(z_i|O^i)$, $i = 1, \dots, n$ to compute $P(\omega^i = \omega_l|Z_\Phi)$. Therefore, the robots enact a gossiping algorithm to spread the information among the team. This means that at a certain point, a generic robot A_j will send to its communication neighbors the data from A_i in a message that we denote as S_i^j, whose format is:

$$S_i^j = \left[p(z_i|O^i)^T \; r_i^j \; \phi_i \; i \right]^T, \tag{3}$$

where $p(z_i|O^i)$ and ϕ_i are the communicated data, and i is the indication of the owner of the measurements. r_i^j is an estimate of the communication distance between A_i and A_j, and is computed by using a counter that is increased every time that the message is rebounced by a robot to its communication neighbors.

The final goal of A_i is to compute the m-vector of probabilities $p(O^i|Z_\Phi)$. At this aim, a distributed Naive Bayes Classifier (NBC) was proposed in [21]. In [1], a weighting factor was introduced to take into account the possibility that robots may look at different objects at the same time. In the NBC, the yaw information is not used to compute $p(O^i|Z_\Phi) = p(O^i|Z)$, and the probability vectors computed by all robots converge to the same value $p(O^i|Z) = p(O^j|Z), \forall i, j = 1, \dots, n$. To compute $p(O^i|Z)$, we begin by applying Bayes rule, and $p(O^i|Z)$ can be rewritten as:

$$p(O^i|Z) = \frac{p(O^i)p(Z|O^i)}{p(Z)} \tag{4}$$

By recursively applying the definition of conditional probability, the numerator of the right-hand side of equation (4) can be rewritten as:

$$\begin{aligned} p(O^i)p(Z|O^i) &= p(O^i)p(z_i, i = 1, \dots, n|O^i) \\ &= p(O^i)p(z_1|O^i)p(z_2|O^i, z_1) \dots p(z_n|O^i, z_1, \dots, z_{n-1}) \end{aligned} \tag{5}$$

By assuming conditional independence of the measurements z_i, equation (5) can be simplified as:

$$p(O^i)p(Z|O^i) = p(O^i) \prod_{j=1}^{n} p(z_j|O^j) \tag{6}$$

Each A_i recursively computes equation (6) by maintaining at all communication steps b an estimate of $P(O^i|Z_i^b)$, where $Z_i^b = \{z_q, \forall q \in ID_i^b\}$ is the set of all measurements received by A_i up to the communication step b. Every time that A_i receives a new message S_h^j such that $h \notin ID_i^b$, it will update its current estimate incorporating the new measurements:

$$p(O^i|Z_i^b, z_h) = \frac{p(O^i)p(Z_i^b, z_h|O^i)}{p(Z_i^b, z_h)} = \frac{p(O^i)p(Z_i^b|O^i)p(z_h|O^i)}{p(Z_i^b)P(z_h)} \tag{7}$$

This algorithm relies on the assumption that A_i and A_h are collecting measurements of the same object ($\omega^i = \omega^h$). In the setting of this work, however, we have rejected this assumption as it is not compatible with a real world scenario. Therefore, we define the following random variable

$$R_h^i = \begin{cases} 1 & \text{if } \omega^i = \omega^h \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

whose value is 1 if A_i and A_h are collecting measurements of the same object, and zero otherwise.

Introducing R_h^i, and considering that if $R_h^i = 0$ the measurement z_h carries no information on the object observed by robot A_i, we can write:

$$\begin{aligned} p(O^i|Z_i^b, z_h) &= p(O^i|Z_i^b, z_h, R_h^i)p(R_h^i) + p(O^i|Z_i^b, z_h, \bar{R}_h^i)p(\bar{R}_h^i) = \\ &= p(O^i|Z_i^b, z_h, R_h^i)p(R_h^i) + p(O^i|Z_i^b, \bar{R}_h^i)p(\bar{R}_h^i) = \\ &= p(O^i|Z_i^b, z_h)p(R_h^i) + p(O^i|Z_i^b)p(\bar{R}_h^i). \end{aligned} \tag{9}$$

Introducing equation (7) into (9):

$$\begin{aligned} p(O^i|Z_i^b, z_h) &= \frac{p(O^i)p(Z_i^b|O^i)p(z_h|O^i)P(R_h^i)}{p(Z_i^b)p(z_h)} + p(O^i|Z_i^b)p(\bar{R}_h^i) = \\ &= p(O^i|Z_i^b)\frac{p(z_h|O^i)p(R_h^i)}{p(z_h)} + p(O^i|Z_i^b)p(\bar{R}_h^i). \end{aligned} \tag{10}$$

In equation (10) the term $p(z_h)$ is a normalization factor α such that

$$\sum_l \frac{p(O^i = l|Z_i^b)p(z_h|O^i = l)}{p(z_h)} = 1, \tag{11}$$

therefore:

$$p(O^i|Z_i^b, z_h) = \alpha\, p(O^i|Z_i^b)p(z_h|O^i)P(R_h^i) + p(O^i|Z_i^b)p(\bar{R}_h^i). \tag{12}$$

Considering that

$$P(\bar{R}_h^i) = 1 - P(R_h^i) \tag{13}$$

the final step consists in computing the probability $P(R_h^i)$ that $\omega^i = \omega^h$. In general, $p(R_h^i)$ may depend on several factors and we do not have a standard way to compute it. In this work, we assumed that $p(R_h^i)$ depends on the distance and the relative orientation between A_i and A_h, and that these two factors are independent from each other. This is based on two considerations. First, the further apart the robots are, the less likely they are to be observing the same object. As the robots do not have direct access to their relative distance, they can use the estimated communication distance (which also provide an estimate of their cartesian distance) to compute the following:

$$p(R_h^i | r_i^h) = \frac{1}{{r_i^h}^{\frac{r}{H}}} \tag{14}$$

where H is the average size of the objects in the world. Similarly, if the two robots look in different directions (i.e., have a relative orientation close to π), they are unlikely to be watching the same object. The relative orientation can be computed through the use of the yaw measurements ϕ_i, ϕ_h, therefore we have considered

$$p(R_h^i | \phi_i, \phi_h) = p(R_h^i | \phi_i - \phi_h) \tag{15}$$

Finally, equations (14) and (15) can be combined into the following:

$$p(R_h^i) = p(R_h^i | \phi_i, \phi_h, c_i^h) = p(R_h^i | \phi_i, \phi_h) p(R_h^i | c_i^h) \tag{16}$$

Note that by applying equation (12) instead of the NBC, each robot will compute its own personalized vector of probabilities that takes into account the relevance of other robot measurements to its individual identification process.

3.3 Particle Filter for Localization

Each A_i maintains a particle filter to estimate its configuration in the world frame W. Therefore, the state of each particle is an estimate $\hat{\mathbf{c}}_i^\delta(k) = [\mathbf{q}_k^\delta \ \phi_k^\delta]^T = [x_k^\delta \ y_k^\delta \ \phi_k^\delta]^T, p = 1, \ldots, \Delta$, including position and orientation of the robot in W, where Δ is the number of particles maintained. The filter consists in a time update, performed every time that a new odometry measurement $\mathbf{u}_i(k)$ is available, and a measurements update, performed every time that a new probability vector $P(O^i(k)|Z_\Phi(k))$ is computed. Note that the magnetometer readings $\bar{\phi}_i(k)$ where already used to compute $P(O^i(k)|Z_\Phi(k))$, so they will not be used in the particle filter to avoid reusing the same measurements twice.

3.3.1 Time Update

The time update of each particle follows the motion model of the unicycle (1):

$$\begin{aligned} \phi_k^\delta &= \phi_{k-1}^\delta + T(\bar{\eta}_i(k) + \nu_\eta^\delta) \\ x_k^\delta &= x_{k-1}^\delta + T(\bar{v}_i(k) + \nu_v^\delta)\cos((\phi_k^\delta + \phi_{k-1}^\delta)/2) \\ y_k^\delta &= y_{k-1}^\delta + T(\bar{v}_i(k) + \nu_v^\delta)\sin((\phi_k^\delta + \phi_{k-1}^\delta)/2) \end{aligned} \tag{17}$$

where ν_η^δ and ν_v^δ are samples from the noise affecting the odometry measurements, assumed to be Gaussian with zero mean and known covariance.

3.3.2 Measurement Update

For the measurement update, we use a map of the environment in which the facade of each building ω_l observable by the robots is represented as a segment specified through the coordinates of a starting $\mathbf{a}_{ls}^W = [x_{ls}^W \ y_{ls}^W]^T$ and a final $\mathbf{a}_{lf}^W = [x_{lf}^W \ y_{lf}^W]^T$ point in W. This representation is compact, utilizes very low memory, and easy to implement as it requires very low information. It is also easily expandable to more complex environments.

For each particle δ, it is necessary to compute the expected measurement, i.e., what building the robot would see if it was in the location of that particle. First, the segments are rotated and translated in the frame attached to the robot:

$$\mathbf{a}_{l*}^{\delta} = R(\phi_k^{\delta})^T(\mathbf{a}_{l*}^{W} - \mathbf{q}_k^{\delta}),\ * = s, f,\ l = 1, \ldots, n, \tag{18}$$

where the superscript δ in $\mathbf{a}_{l*}^{\delta}$ indicates that the point is expressed in the frame δ identified by the particle, and $R(\phi_k^{\delta})$ is the 2D rotation matrix of an angle ϕ_k^{δ}.

The segment, now expressed in the frame δ, identifies a line in the form $ax + by + c = 0$. The coefficients are computed as:

$$\begin{aligned} a &= (y_{lf}^{\delta} - y_{ls}^{\delta}) \\ b &= -(x_{lf}^{\delta} - x_{ls}^{\delta}) \\ c &= -x_{ls}^{\delta}(y_{lf}^{\delta} - y_{ls}^{\delta}) + y_{ls}^{\delta}(x_{lf}^{\delta} - x_{ls}^{\delta}) \end{aligned} \tag{19}$$

To check if the particle is oriented in the direction of the building, we compute the intersection of the line $ax + by + c = 0$ with the line $y = 0$, obtaining $x = -c/a$.

Clearly, if the resulting $x < 0$, the particle is not oriented towards that particular building. Moreover, if the point $[-c/a\ 0]$ is between $\mathbf{a}_{ls}$ and $\mathbf{a}_{lf}$, the particle is oriented towards that l-th building.

At this point, there is still the possibility that the particle is behind the facade of the building. To check if this is the case, we compute a point $[x_n\ y_n]^T$ along the positive normal of the segment $[-(y_{lf}^{\delta} - y_{ls}^{\delta})\ (x_{lf}^{\delta} - x_{ls}^{\delta})]^T$ and we check that the condition $c(ax_n + by_n + c) > 0$ is verified. If this is the case, the particle is oriented towards the l-th building from the correct direction, therefore we update its weight w_k^{δ} of a given particle by multiplying its current weight w_{k-1}^{δ} by $p(O^i(k) = \omega^l | Z_{\Phi}(k))$, that is, the probability that the robot is oriented towards the l-th building according to the measurements fused through the weighted Bayes classifier:

$$w_k^{\delta} = w_{k-1}^{\delta} * p(O^i(k) = \omega^l | Z_{\Phi}(k)) \tag{20}$$

Note that, being a probability, $0 \leq p(O^i(k) = \omega^l | Z_{\Phi}(k)) \leq 1$. Therefore, the weights of the particles will decrease a little if $p(O^i(k) = \omega^l | Z_{\Phi}(k)) \simeq 1$, and will decrease by several orders of magnitude if $p(O^i(k) = \omega^l | Z_{\Phi}(k)) \to 0$. However, if a particle is not associated to any building, as for example when the particle is outside the mapped area, following this algorithm it would eventually have the largest weight. To avoid this paradoxical situation, particles that are not associated to any building automatically receive very low weights. Moreover, since misidentifications are still possible, the resampling of the particles is not performed every measurements update, so that a single misidentification will not cause the removal of correct particles. The mean of the particles of the filter running on A_i is eventually selected as the estimate:

$$\hat{\mathbf{c}}_i(k) = \text{mean}_{\delta}(\hat{\mathbf{c}}_i^{\delta}(k)), i = 1, \ldots, n. \tag{21}$$

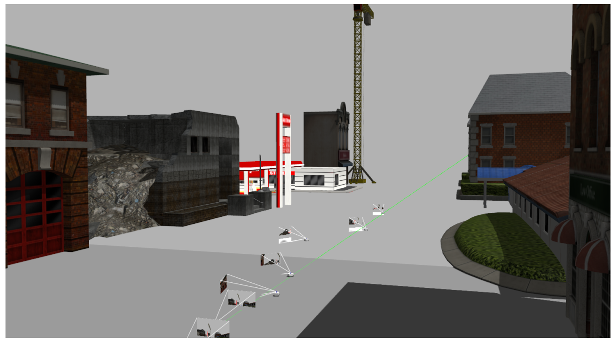
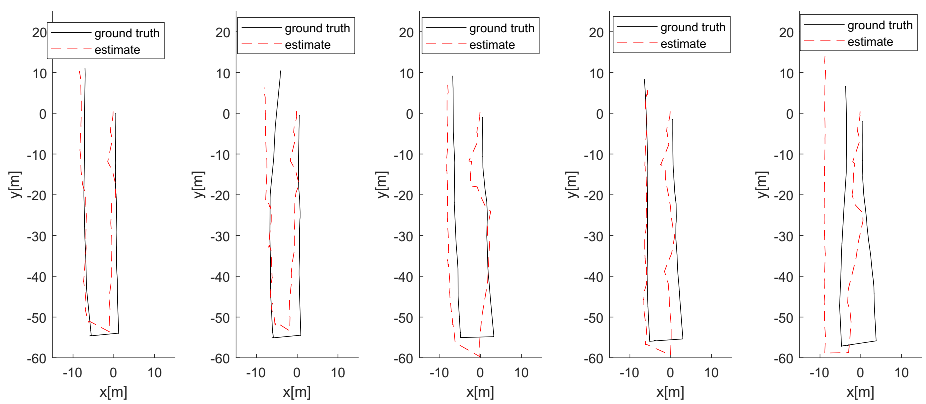


Fig. 2: Simulated city in Gazebo.



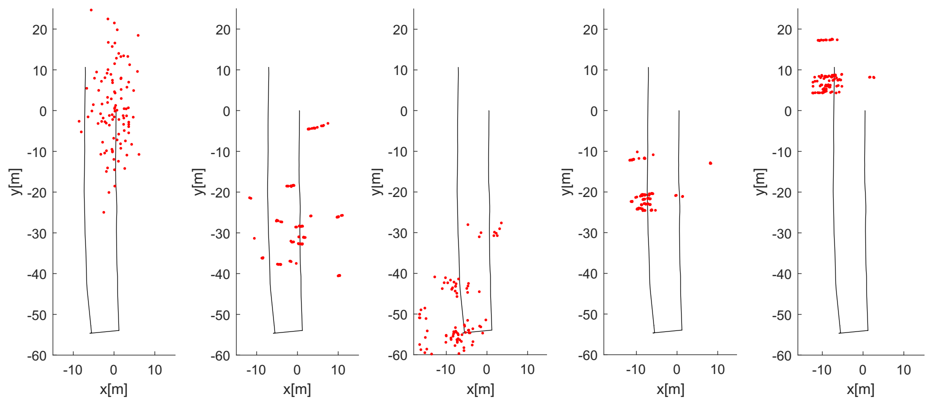
(a) Robot 1 (b) Robot 2 (c) Robot 3 (d) Robot 4 (e) Robot 5

Fig. 3: The estimated robot trajectories (red dashed lines) and the ground truth data (black solid lines) for all five robots in a typical experiment.

4 Simulations

We have tested the proposed algorithm in simulation using a complex environment in the Gazebo/ROS framework. In the simulated world, we have placed five small ($\sim$20 cm) robots and 17 unique buildings that act as landmarks. A view of the simulated world is provided in Fig. 2. In a typical simulation, the robots move in the world roughly in a line formation running the cooperative localization system. At the same time, ground truth data are collected.

The results of a typical experiment are presented in Figures 3 and 4. Figure 3 reports the estimated robot trajectories (red dashed lines) and the ground truth data (black solid lines) for all five robots. The plots show how the estimates correctly track the actual paths of the robots. Figure 4 reports the particle distribution (red dots) for five time instants at 0%, 25%, 50%, 75% and 100% of the simulation time for Robot 1. The black solid lines show the ground truth. From this plot, it is possible to see how the particles are initially scattered and slowly converges towards the ground truth.



(a) t = 0% (b) t = 25% (c) t = 50% (d) t = 75% (e) t = 100%

Fig. 4: The particle distribution (red dots) for five time instants at 0%, 25%, 50%, 75% and 100% of the simulation time for Robot 1. The black solid lines show the ground truth.

5 Conclusions

In this paper, we have presented a landmark-based localization system that uses the output of a cooperative identification module to feed a set of independent particle filters. The proposed system achieves a collaborative localization setup without the need for knowledge, either in the form of an estimate or of a measurement, of relative positions and/or distance. Moreover, this result is obtained using only a general purpose sensor as a camera, without the need for additional hardware. These features makes the proposed system particularly suited for robotic swarms.

In the future, we plan to test the proposed localization system on real robots with larger data sets. This will also stress out the computational requirements of the algorithms. We foresee that larger maps may require adjustments in the measurement update steps to avoid checking all buildings in the maps, but rather selecting only a few candidate buildings in the proximity of each particle. Moreover, we will study more in depth the problem of particles not associated to any building, and we are planning to incorporate directly in the CNN's a case in which no landmarks are recognized.

6 Acknowledgments

This work was supported in part by the National Science Foundation under Grant CMMI 1952862.

References

1. S. Brent, C. Yuan, and P. Stegagno, "Cooperative place recognition in robotic swarms," in *Proceedings of the 2021 Annual ACM Symposium on Applied Computing*, ser. SAC '21, Virtual Conference, 2021.

2. L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, 08 2015.
3. M. Bakhshipour, M. J. Ghadi, and F. Namdari, "Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach," *Applied Soft Computing*, vol. 57, pp. 708 – 726, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494617301072
4. M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422 – 434, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889015001876
5. N. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *Journal of hazardous materials*, vol. 154, pp. 880–7, 07 2008.
6. M. Kayser, L. Cai, S. Falcone, C. Bader, N. Inglessis, B. Darweesh, and N. Oxman, "Design of a multi-agent, fiber composite digital fabrication system," *Science Robotics*, vol. 3, no. 22, 2018. [Online]. Available: https://robotics.sciencemag.org/content/3/22/eaau5630
7. L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018. [Online]. Available: https://doi.org/10.1177/0278364918760698
8. B. Wang, S. Rathinam, and R. Sharma, "Landmark placement for cooperative localization and routing of unmanned vehicles," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 33–42.
9. Y. Lin, P. Vernaza, J. Ham, and D. D. Lee, "Cooperative relative robot localization with audible acoustic sensing," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3764–3769.
10. A. Franchi, G. Oriolo, and P. Stegagno, "Mutual localization in multi-robot systems using anonymous relative measurements," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1302–1322, 2013. [Online]. Available: https://doi.org/10.1177/0278364913495425
11. P. Zhang, E. E. Milios, and J. Gu, "Underwater robot localization using artificial visual landmarks," in *2004 IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 705–710.
12. D. Chen, Z. Peng, and X. Ling, "A low-cost localization system based on artificial landmarks with two degree of freedom platform camera," *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pp. 625–630, 04 2015.
13. S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual slam," *Robotics, IEEE Transactions on*, vol. 24, pp. 1054 – 1065, 11 2008.
14. L. Jayatilleke and N. Zhang, "Landmark-based localization for unmanned aerial vehicles," in *2013 IEEE International Systems Conference (SysCon)*, 2013, pp. 448–451.
15. H. Kobayashi, "A new proposal for self-localization of mobile robot by self-contained 2d barcode landmark," in *2012 Proceedings of SICE Annual Conference (SICE)*, 2012, pp. 2080–2083.
16. S. Han, J. Kim, and H. Myung, "Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1745–1756, 2013.
17. D. C. K. Yuen and B. A. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 217–226, 2005.

18. S. A. Bangash and A. Ghafoor, "Vision based mobile node localization using a landmark," in *The 5th International Conference on Automation, Robotics and Applications*, 2011, pp. 255–259.
19. G. Jang, S.Kim, W. Lee, and I. Kweon, "Robust self-localization of mobile robots using artificial and natural landmarks," in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694)*, vol. 1, 2003, pp. 412–417 vol.1.
20. R. Sahdev and J. K. Tsotsos, "Indoor place recognition system for localization of mobile robots," in *2016 13th Conference on Computer and Robot Vision (CRV)*, June 2016, pp. 53–60.
21. P. Stegagno, C. Massidda, and H. H. Bülthoff, "Distributed target identification in robotic swarms," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, Salamanca, Spain, 2015, p. 307–313. [Online]. Available: https://doi.org/10.1145/2695664.2695922
22. C. Kim, H. Yang, D. Kang, and D. Lee, "2-d cooperative localization with omnidirectional mobile robots," in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2015, pp. 425–426.
23. K. Song, C. Tsai, and C. Huang, "Multi-robot cooperative sensing and localization," in *2008 IEEE International Conference on Automation and Logistics*, 2008, pp. 431–436.
24. L. Wang, X. Cai, W. Fan, and H. Zhang, "An improved method for multi-robot cooperative localization based on relative bearing," in *2008 IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 1862–1867.
25. Z. Li, R. Acuna, and V. Willert, "Cooperative localization by fusing pose estimates from static environmental and mobile fiducial features," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, 2018, pp. 65–70.
26. P. Wu, Q. Luo, and L. Kong, "Cooperative localization of network robot system based on improved mpf," in *2016 IEEE International Conference on Information and Automation (ICIA)*, 2016, pp. 796–800.
27. H. Kang, J. Kawk, C. Kim, and K. Jo, "Multiple robot formation keeping and cooperative localization by panoramic view," in *2009 ICCAS-SICE*, 2009, pp. 3509–3513.
28. C. Jennings, D. Murray, and J. J. Little, "Cooperative robot localization with vision-based mapping," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, 1999, pp. 2659–2665 vol.4.
29. S. Xingxi, H. Bo, W. Tiesheng, and Z. Chunxia, "Cooperative multi-robot localization based on distributed ukf," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 6, 2010, pp. 590–593.
30. C. Kim, D. Song, Y. Xu, J. Yi, and X. Wu, "Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1161–1173, 2014.
31. R. F. Carpio, L. Di Giulio, E. Garone, G. Ulivi, and A. Gasparri, "A distributed swarm aggregation algorithm for bar shaped multi-agent systems," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 4303–4308.