Cooperative Place Recognition in Robotic Swarms

Sarah Brent
Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island
Kingston, RI, USA
sbrent@uri.edu

Chengzhi Yuan
Department of Mechanical, Industrial and Systems Engineering, University of Rhode Island
Kingston, RI, USA
cyuan@uri.edu

Paolo Stegagno
Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island
Kingston, RI, USA
pstegagno@uri.edu

ABSTRACT

In this paper we propose a study on landmark identification as a step towards a localization setup for real-world robotic swarms setup. In real world, landmark identification is often tackled as a place recognition problem through the use of computationally intensive Convolutional Neural Networks. However, the components of a robotic swarm usually have limited computational and sensing capabilities that allows only for the application of relatively shallow networks that results in large percentage of recognition errors. In a previous attempt of solving a similar setup – cooperative object recognition – the authors of [1] have demonstrated how the use of communication among a swarm and a naive Bayes classifier was able to substantially improve the correct recognition rate. An assumption of that paper not compatible with a swarm localization setup was that all swarm components would be looking at the same object. In this paper, we propose the use of a weighting factor to relapse this assumption. Through the use of simulation data, we show that our approach provides high recognition rates even in situations in which the robots would look at different objects.

KEYWORDS

localization, swarm, sensor fusion

ACM Reference Format:
Sarah Brent, Chengzhi Yuan, and Paolo Stegagno. 2021. Cooperative Place Recognition in Robotic Swarms. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3412841.3441954

1 INTRODUCTION

Robotic swarms consist of autonomous relatively inexpensive robots with limited computational and sensing capabilities. Each robot performs simple tasks using only local information and limited communication with its communication neighbors. Thanks to the intrinsic decentralization of the system, many researchers have highlighted the robustness of robotics swarms, as well as the versatility and the ability to parallelize the work. These benefits however come at the price of a higher complexity in the design and analysis of single robot behaviors to achieve a global task from local interactions [2]. Despite this additional challenge, robotic swarms have been proposed for a large plethora of tasks including search and rescue operations [3], exploration [4], information gathering and clean up of toxic spills [5, 6], target search and tracking [7], and even construction [8].

For real world application of these systems, however, swarms still present a number of challenges that are currently being actively researched. Localization of the swarm, acquisition of local information through sensing, communication with a subset of agents in the swarm, and decision making based on the gathered sensor data are aspects that need to be addressed. Particularly while researching sensing and estimation, there are aspects to consider that are specific to swarms. There are constraints on the number of sensor equipment that can be employed due to limits on computational power, energy consumption, and payload. Moreover, the physical dimension of the robots could be small compared to the objects in their environment, and in many cases measurements as images will be collected from very disadvantageous, non-comprehensive points of view.

However, in many aspects of control, localization, and SLAM, robots in a swarm still need to be able to identify an object or a place, be it the target of an action (e.g., [9]) or the current location (e.g., [10]). Many mobile robot localization systems, for example, rely on the presence of known-location landmarks in the environment (e.g., [11–14]). In real-world applications, these landmarks should be naturally part of the environment. Their identification however is often tackled with computationally demanding recognition algorithms based on computer vision or neural networks.

In this paper we propose a cooperative recognition strategy in which each robot uses a Convolutional Neural Network (CNN) trained to recognize the landmarks in the environment. Due to the limited computational capabilities of the robots, however, the CNNs are relatively shallow and provide a relatively high percentage of recognition errors. In a multi-view setup, a suitable strategy to reduce errors is to fuse the results of the individual CNNs.

In single view setups, there is a vast literature on place recognition [15], using cameras (e.g., [10, 16]) and lidar sensors (e.g., [17]). However, there are relatively few authors that have addressed the problem of place recognition with sequence of frames or in general multi-view information. Some algorithms uses sequences of frames with temporal consistency constraints for place recognition [18–22]. This setup significantly differs from a multi-view setting and cannot be straightforwardly applied to our problem. In [23], the authors propose three alternatives for deep networks to use several frames of a sequence in a place recognition task, in which

SAC' 21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/10.1145/3412841.3441954

the model learns to combine single-view CNN descriptors from data. In [24] the authors propose a feature based approach to place recognition using both lidar and camera sensors. However, both these methods are beyond the computational capabilities of robotic swarms.

Similarly, object classification has a vast literature in single view setup [25], but relatively few works in multi-view setups or multi-sensor setups. In [26–28], the authors use various approaches to seek to reconstruct the cluttered parts of an environment in order to discern the subject from the background using a multi-view approach. In [29, 30], a method for selection of an optimal number of images that are taken from different perspectives of a 3D object is used for recognition. In [31, 32], the authors employ a network of smart cameras which jointly classify the observed object. In [33, 34], methods were developed to classify objects based on sound features and visual information. The commonality between all these approaches is that they employ a broad point of view to select the informative features of the observed objects, which is different from a robotic swarm setup in which each robot may take only partial information of the observed place.

In one paper [1], the authors proposed a study on multi-sensor object classification on a series of objects in which the individual classification results were fused together through an iterative naive Bayes classifier. The main drawback of this approach is that the robots were assumed to be looking all at the same object. Therefore, all individual results were fused together and all team members would eventually converge to the same result. In a robot localization scenario, however, it is likely that different team members would look at different objects. This would happen, for example, if there is a significant distance between some robots, or if the robots are looking in different directions.

In this study, we relax that assumption by employing a weighted naive Bayes classifier (WNBC) in which the weights that each robot assigns to other robots' results received through communication depend on the distance on the communication graph, and the magnitude of the relative yaw. As the overall performance of the system depends on this relationship between the weights and these two parameters, we have performed a series of numerical simulation to optimize our system and highlight the effect of this relationship. The main contribution of this paper is to provide a novel place recognition algorithm specifically designed for 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 problem of cooperative place recognition. Section 3 will describe the proposed system. In Section 4, we propose a description of the simulation platform and implementation. Section 5 will conclude the paper.

2 PROBLEM SETTINGS

Let us consider a swarm system $A = \{A_1, A_2, ..., A_n\}$ of n agents. The generic robot A_i, $i = 1, \ldots, n$ lives in a 3D environment populated with a set $\Omega = \{\omega_1, \omega_2, ..., \omega_m\}$ of m objects ω_l, $l = 1, \ldots, m$. In general, robots are not aware of their global positions in any fixed world frame $W = O - XYZ$, nor they have access to each other's relative position. However, we assume that the robots are able to communicate with each other within a certain range r. Hence, we can define the communication graph as an ordered pair $G = (\mathcal{N}, \mathcal{E})$ consisting of nodes $\mathcal{N}$ (the robots) and edges $\mathcal{E}$. An edge $e = \{i, j\}$ is an unordered pair such that if $\{i, j\} \in \mathcal{E}$, robots A_i and A_j can communicate. 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. It is not within the scope of this paper to study the problem of controlling the swarm so that this assumption is verified. However, there are in literature (e.g., [35]) connectivity maintenance swarming algorithm that can guarantee hat this assumption is verified.

Each agent A_i is given a set of exteroceptive sensors and gathers a measurement z_i of an object $\omega^i \in \Omega$, where the superscript i identifies the specific object observed by robot A_i. In fact, contrary to the assumptions of [1], different robots can potentially observe different objects in the environment. In the following, we assume that all the robots will be equipped with the same exteroceptive sensor: a camera. However, this assumption is easily generalizable to different exteroceptive sensors. In addition, each robot A_i will be able to measure its own yaw angle ϕ_i in the world frame W through a magnetometer. In the following, we will indicate with $Z = \{z_i, i = 1, 2, ..., n\}$ the set of exteroceptive measurements collected by all the robots, and with $\Phi = \{\phi_i, i = 1, 2, ..., n\}$ the set of all yaw angles. Collectively, we will indicate with $Z_\Phi = \{Z, \Phi\}$ the set of all exteroceptive and yaw measurements.

To formally introduce the problem that we will address in this work, we define n random variables $O^i(\omega)$, $i = 1, \ldots, n$, that represent the objects observed by A_i, $i = 1, \ldots, n$:

$$O^i(\omega) = O^i = l \Leftrightarrow \omega^i = \omega_l \tag{1}$$

We then define the probability $p(O^i = l) = p(O^i)$ as the probability that $\omega^i = \omega_l$.

Problem Statement 1. *The problem of identifying the objects ω^i out of the set Ω through the measurements Z_Φ, is the problem of assigning to each ω^i a potentially different label $l^i = cl^i(Z_\Phi)$ out of the set $L = \{1, \ldots, m\}$ on the basis of the measurements Z_Φ, where $cl^i(Z_\Phi) = l^i$ if and only if ω^i is recognized to be ω_{l_i}.*

A common policy to solve Problem 1 is to assign to the observed objects $\omega^i, i = 1, \ldots, n$, the labels that maximize the probabilities $p(O^i = l^i|Z_\Phi)$ given the measurements Z_Φ:

$$cl^i(Z_\Phi) = \underset{l^i \in L}{\operatorname{argmax}}\, p(O^i|Z_\Phi) \tag{2}$$

Whenever a labeling policy is given in the form of equation (2), it is called a Bayes classifier. The crux of it is for each robot in the system to estimate $p(O^i|Z_\Phi)$, and then make a decision. One of the main focuses of this work is to define a distributed way for each robot A_i to estimate $p(O^i|Z_\Phi)$, as defined in the following Problem 2:

Problem Statement 2. *The problem of identifying the objects $\omega^i, i = 1, \ldots, n$ out of the set Ω is the problem of computing the n vectors of probabilities $p(O^i|Z_\Phi), i = 1, \ldots, n$, given the exteroceptive measurements Z and the yaw measurements Φ of all robots.*

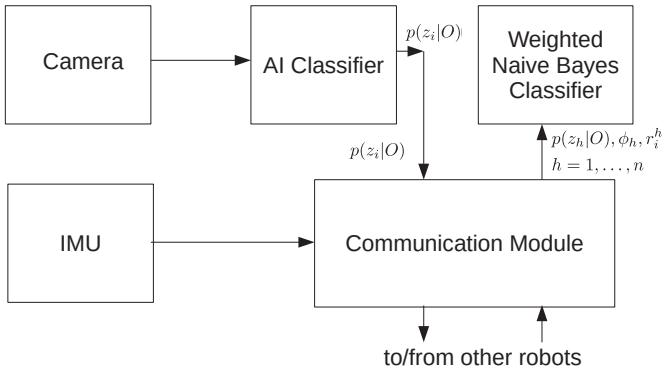

Figure 1: Block scheme of the system running on robot A_i.

Note that in Problem 2, contrarily to classical multi-view identification and distributed estimation problems, the robots are not required to reach a consensus. Instead, each robot can reach a potentially different solution, even though each robot will use the same data including its own measurements and the measurements from all other robots.

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, is passed through an AI classifier (Section 3.2), a CNN, to determine which object A_i is observing. This step is done independently by each robot. The output of the classifier is the m-vector of probabilities $P(z_i|O^i)$ that A_i obtains a measurement z_i given that it is observing ω_l, $l = 1, \ldots, m$. This information is then provided to the communication module that broadcasts it to the communication neighbors of A_i together with the measured yaw angle ϕ_i. The communication module also implements a multi-hop communication algorithm so that each robot A_h in the team can receive the probability vectors and yaw angle measurements of A_i, even the ones that are not directly communicating with A_i itself.

As all robots do the same, A_i's communication module also receives the probability vectors $P(z_h|O^h)$ and the yaw angles ϕ_h, $h = 1, \ldots, n$, $h \neq i$ of all other robots in the swarm. With an appropriate communication protocol (Section 3.3), A_i also computes an estimate r_i^h 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 in the multi-hop communication algorithm for a message sent from robot A_h to reach robot A_i, and is equivalent to the graph length of the shortest path that connects nodes i and h in the communication graph G.

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, Section 3.4) together with the yaw angles ϕ_i, ϕ_h, and the estimated communication distances r_i^h. This information is used by the WNBC to iteratively compute $P(O^i|Z_\Phi)$.

3.2 Single-Robot Place Recognition

In the scope of this work we are using a standard single-view recognition algorithm, a convolutional neural network (CNN) on the Tensorflow platform. CNN's are frequently used with image data for recognition purpose [36]. First we have created a training and a testing dataset on the simulated world that we have used to demonstrate our cooperative algorithm. We have 7055 images that we use for our training dataset and 3036 that are used for the testing dataset. The training dataset was used to learn the weights of a 5-layered CNN with 19 different categories. In order to estimate how many epochs to use to train the neural network we look at the ROC graph [37, 38]. The ROC graph, formally called the receiver operating characteristic curve, shows the performance of the classification model at all of the classification thresholds. It plots the true positive rate and the false positive rate. This was fundamental to assess that the CNN was giving equal weight to all the input values, as well as to avoid overfitting. The choice to use a relatively shallow CNN was dictated by the limited computational capabilities of the hardware this algorithm is meant for, i.e., the onboard computer of the robots. After learning the network, the testing dataset was used to evaluate the single robot recognition capabilities. The results showed a single robot correct recognition rate of 84%. This probability is determined by considering each image as a given set of "grades" or weights that tells us how fitting the given image is to each class of landmarks. This grade is divided by the sum of the grades of all categories in order to obtain a measurement of the probability $p(z_i|O^i)$.

Note that with this approach it is possible to easily include other types of sensor. In fact, after the CNN computes $p(z_i|O^i)$, the rest of the system does not need to know the data type that originated the specific probability vector. Therefore, to extend the system to incorporate additional types of sensors, it is only necessary to train a new CNN with the data collected by that sensor.

3.3 Communication

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 R_i's measured probabilities and yaw angle. However, every member of the team eventually needs to receive $p(z_i|O^i)$, $i = 1, \ldots, n$, to compute $P(\omega^i = \omega_l|Z_\Phi)$. Therefore, each robot enacts a multi-hop communication approach comprising multiple communication steps to spread the information among the team. 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 with S_i^j. The format of S_i^j is the following:

$$S_i^j = \begin{bmatrix} p(z_i|O^i)^T & r_i^j & \phi_i & i \end{bmatrix}^T, \tag{3}$$

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 while the communication in the team is happening as specified in the following.

Let be ID_i^k a set such that $h \in ID_i^k$ if A_i has received at least one message S_h^j for any $h = 1, \ldots, n$ between communication step 0 and communication step k. This means that if $h \in ID_i^k$ then A_i has

Algorithm 1: The pseudocode of the communication algorithm running on A_i.

1 $ID_i^0 = \{i\}$
2 broadcast $S_i^i = \begin{bmatrix} p(z_i|O^i)^T & r_i^i = 0 & \phi_i & i \end{bmatrix}^T$
3 **while** $ID_i \neq \{1, 2, \ldots, n\}$ **do**
4 **if** A_i *receives* S_h^j **then**
5 **if** $h \notin ID_i^k$ **then**
6 $ID_i^{k+1} = ID_i^k \bigcup \{h\}$
7 $S_h^i = S_h^j$
8 $r_h^i = r_h^j + 1$
9 broadcast S_h^i
10 **else**
11 ignore S_h^j
12 **end**
13 **else**
14 wait
15 **end**
16 **end**

received $p(z_h|O^h)$ and ϕ_h from a previous communication message from at least one robot in the team. The pseudocode of the multi-hop communication algorithms executed by each A_i is presented in Algorithm 1. It includes the following steps:

- **line 1:** The algorithm is initialized by setting the ID_i^0 to include only i.
- **line 2:** Then A_i sends the message S_i^i (containing its own measurements) to its communication neighbors once; in this step, $r_i^j = r_i^i = 0$.
- **lines 3-9:** if A_i receives a message S_h^j from A_j with the measurements from A_h, and $h \in ID_i^k$, i.e., A_i has never received the measurements of A_h (line 5) then A_i creates a new message S_h^i identical S_h^j (line 7), increases the estimated communication distance by 1 (line 8) and broadcasts S_h^i to its neighbors.
- **line 11:** if instead A_i had already received A_h's measurements from another robot, the message S_h^j is ignored.

When all robots in the team performs this algorithm, each robot will receive the probability vectors and yaw angles from all other robots. Moreover, an estimate of the communication distance between A_i and any other A_j, $j \in ID_i^k$ will be available to A_i.

3.4 Weighted Naive Bayes Classifier

The final goal of A_i is to compute the m-vector of probabilities $p(O^i|Z_\Phi)$. Here we describe first the distributed Naive Bayes Classifier (NBC) approach proposed in [1], and then we introduce a weighting factor to take into account the reliability of the information provided by the other robots. 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 $p(O^i|Z) = p(O^j|Z), \forall i, j = 1, \ldots, n$. Although the classification method defined by equation (2) is simple, characterizing the conditional probability $p(O^i|Z)$ is not trivial. We begin by applying Bayes rule, and $p(O^i|Z)$ can be rewritten as:

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

We can recursively apply the definition of conditional probability, thus the numerator of the right-hand side of equation (4) can be factorized as:

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

Equation (5) can be computed recursively using the measurements one at a time. However, the characterization of the dependency between the measurements can still prevent the actual computation of each factor. In the traditional naive Bayes classifier the measurements are assumed to be conditionally independent from each other. Considering that measurements come from different robots at different locations, we can thus exploit the conditional independence of the measurements z_i assumption and simplify the above equation (5).

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

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

$$\begin{aligned} p(O^i|Z_i^k, z_h) &= \frac{p(O^i)p(Z_i^k, z_h|O^i)}{p(Z_i^k, zh)} \\ &= \frac{p(O^i)p(Z_i^k|O^i)p(z_h|O^i)}{p(Z_i^k)P(z_h)} \end{aligned} \tag{7}$$

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 setup of this work, however, A_i and A_h may be looking at different objects. To relax this assumption, we define the following random variable R_h^i

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

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:

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

Introducing equation (7) into (9):

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

Note that in Equation (10) the term $p(z_h)$ is a normalization factor α such that

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

therefore:

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

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 landmark. As the robots do not have direct access to their relative distance, they can use the estimated communication distance (which also provides an estimate of their Cartesian distance) to compute the following:

$$p(R_h^i|r_i^h) = \frac{1}{(r_i^h + 1)^\lambda}, \tag{14}$$

where λ is positive parameter used to increase or decrease the weight of the other robot's measurements. When λ is small (< 1), the measurements from robots that are further away will have higher weights. In the limit that $\lambda = 0$, $p(R_h^i|r_i^h)$ converges to 1. This is the unweighted case. For large λ ($\gg 1$), $p(R_h^i|r_i^h)$ converges to zero. In principle, it is possible to use this properties also to limit the number of communication steps to a specific value r_{max}. In fact, for $\lambda > 0$, after a given number of communications steps, new incoming measurements will be assigned an almost zero weight. Therefore, it is possible to compute r_{max} such that those measurements are never communicated. This would be important to limit communication in large swarms of hundreds or thousands of agents, making te system more scalable. Clearly, r_{max} depends on λ and on the approximation that is possible to accept.

Similarly, if the two robots look in different directions (i.e., have a relative orientation near 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

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

In our implementation, $p(R_h^i|\phi_i - \phi_h)$ is a Gaussian function with zero mean and $\pi/3$ covariance. This choice was made so that small values of $\phi_i - \phi_h$ would provide $p(R_h^i|\phi_i - \phi_h) \simeq 1$, while larger yaw differences would result in lower weights. However, other functions can lead to equivalent results. Finally, equations (14) and (15) can be combined into the following:

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

Figure 2: Two views of the simulated world in ROS Gazebo used to evaluate the performance of the place recognition system.

4 SIMULATIONS

In order to test the proposed algorithm, we used a ROS Gazebo simulation with ten small (~20 *cm*) robots moving in a complex environment (Figure 2) consisting in a street with nineteen unique buildings and other landmarks (e.g., trees, a playground, a mailbox). The location of each building is assumed to be known in advance. Therefore, from the position and orientation of a robot it is therefore possible to predict which landmark it would be facing, and viceversa it is possible to divide the configuration space of the robots into nineteen cells, one for each landmark, from which a measurement of that landmark would be collected. Each robot is equipped with a simulated camera and IMU sensor, and limited communication distance is simulated through the knowledge of the robots' position. A 1s communication delay between transmission and reception of a message is also introduced.

In a typical simulation, the robots are divided into two or more groups (clusters). Robots of the same cluster collect images of the same landmark (therefore are in the same cell in the configuration

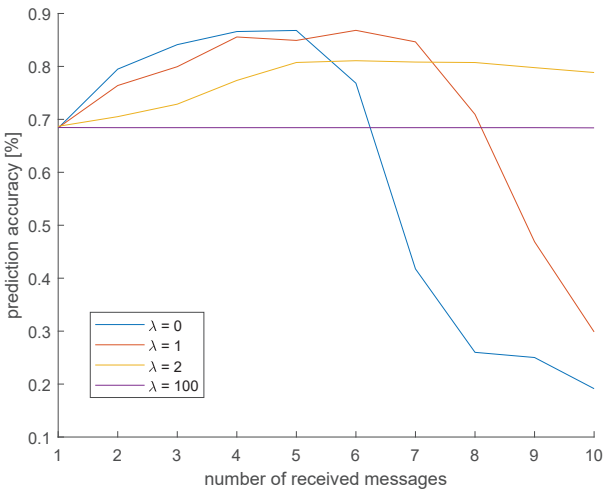

Figure 3: Correct recognition rate as a function of the number of communications received.

space), while robots in different clusters collect images of different landmarks. The distance between clusters and robot within the same cluster varies from one simulation to another, but in general robots are deployed so that the communication graph is always connected. This is to ensure that we collect homogeneous data. If the robots were split into multiple unconnected groups, the algorithm would work independently in each subgroup, but the communication graph would not have enough depth to stress-test it. The formation shape also varies from simulation to simulation, to change the connectivity of the communication graph. Sometimes a line formation is used (as the one shown in Figure 4), sometimes a double line, and sometimes more general formations.

Over the course of a simulation, each robot A_i collects an image, computes the probability measurement $p(z_i|O^i)$ with the CNN, collects other robots measurements using the communication algorithm described in Section 3.3, and applies the weighted naive Bayes classifier described in Section 3.4, until all communications have been received. Every few seconds and a small motion, a new set of images are collected and the cooperative recognition is repeated. With this methodology, 118 unique configurations were collected over five simulations in which the robots covered the whole environment. Considering that each experiment is performed with ten robots, we have a total of 1180 data point. The result of each step of the iterative weighted naive Bayes classifier of each robot was recorded together with the robots' location and the number of other robots' measurements incorporated in each step.

In order to evaluate the effect of the parameter λ in equation (14) the weighted naive Bayes classifier was run with this dataset for $\lambda = 0$, $\lambda = 1$, $\lambda = 2$, and $\lambda = 100$. The case $\lambda = 0$ is the same as the unweighted case, and was performed for comparison. In the case $\lambda = 100$ all weights are trivially $\simeq 0$, therefore it corresponds to the single robot recognition. It was done as a sanity check and to gain an understanding of the advantages provided by the cooperation among the robots.

To evaluate the performance of the system we have plotted in Figure 3 the correct recognitions percentage against the number of incorporated messages for the four values of λ. The results indicate that the choice of λ has a strong effect on the results. As the number of communications from other robots in the swarm increases, the probability of correctly identifying which landmark they are observing increases up to a certain critical number of incorporated messages. This is true for all values of λ, with the exception of $\lambda = 100$, which we trivially note remains constant for all message numbers due to the near-zero weights assigned to the measurements of all other robots. Note that the single robot correct recognition rate of 68% is considerably lower with respect to the recognition rate computed through our testing dataset in the CNN training and testing process. This may be due to the fact that the motion in swarm brings the robots to more varied orientations than what was included in the original testing set.

Conversely, in the unweighted case where $\lambda = 0$, all robots' $p(z_h|O^i)$ are given weight 1. Since all robots converge to the same result, but not all robots are observing the same landmark, it is natural that the final correct recognition rate at ten messages is low. The initial good results are due to incorporating measurements from nearby robots that are more likely to observe the same landmark. Now, we contemplate the case where the robots start to weigh other robots' predictions $p(z_h|O^i)$. In particular, predictions $p(z_h|O^i)$ from robots that are further away are given lower weights. In the case for $\lambda = 1$, we see that the final prediction accuracy remains high, around 90% for up to seven messages before it starts to deteriorate rapidly down to 30% accuracy for ten messages.

Lastly, in the case where $\lambda = 2$, for early messages, meaning from the first five robots that are closer, we observe that this weight does not afford enough importance to close neighbors, and therefore the overall prediction accuracy is not as high as in the cases $\lambda = 0$ and $\lambda = 1$. However, as the messages from robots that are further away are received, these are weighted less, so the recognition accuracy is not compromised remaining at around 80%.

These results illustrate the importance of weighting the probability vectors to assign more importance to the robots in our cluster, and less importance to the robots in clusters further away. This also leads us to contemplate potentially being able to dynamically change the weighting policy during operation.

In Figure 4, we show two visualizations of the results of simulations for $\lambda = 2$. The robots (triangles) are distributed in a line in front of three landmarks. Each landmark is represented as a segment of a specific color in the plots, representing the facades of the corresponding buildings. The color of each robot is matched with the color of the landmarks that it recognizes after incorporating all ten communicated messages. The two configurations differs mainly for the yaw angles of the robots. This data visualization shows that with the proposed algorithm the robots are capable of splitting the group in clusters, that is not possible using the unweighted naive Bayes classifier. However, as shown in the second configuration, mispredictions are still possible.

Finally, in Figure 5 we provide the full confusion matrix for the case of $\lambda = 2$ and all ten incorporated messages in the form of a heatmap. The figure shows that most buildings are correctly recognized in a majority of cases. Therefore, we believe the system proposed in this paper is suitable to be used to feed the measurement

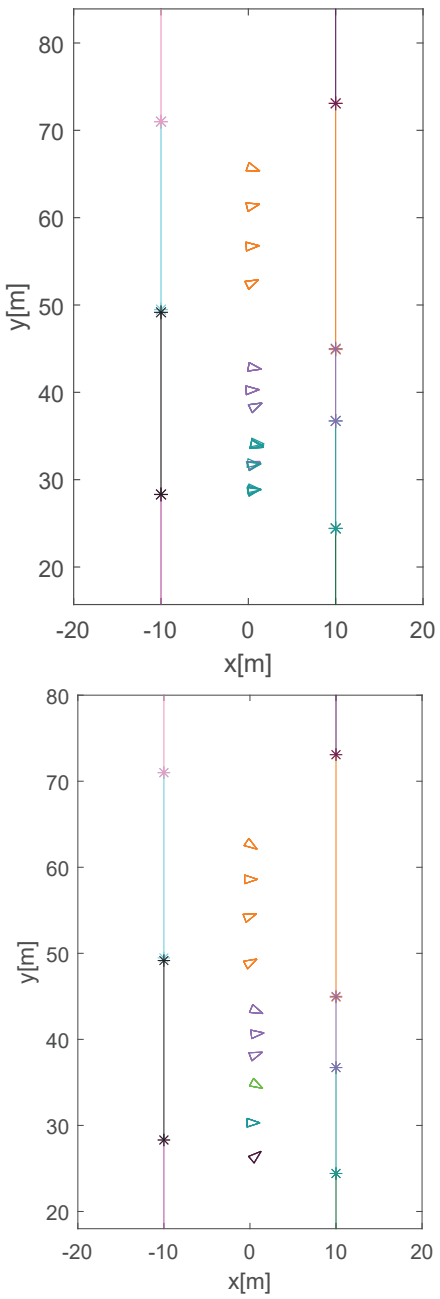

Figure 4: Two examples of swarm configuration collected during an experiment in which the robots where divided in three clusters. The two configurations differ mainly for the yaw of the robots.

update of a landmark-based localization scheme, as outlined in Section 5.

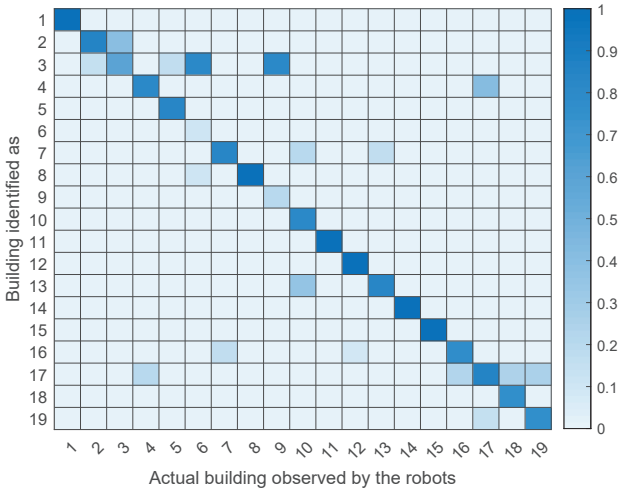

Figure 5: Confusion matrix for $\lambda = 2$ and ten received messages.

5 CONCLUSIONS

In this paper we have presented and formalized a methodology to perform cooperative place recognition in a robotic swarm. The ultimate goal of this project is the application of the presented system to provide measurements in a landmark-based Bayesian localization scheme. The proposed solution relies on a weighted naive Bayes classifier to fuse the solution of individual shallow CNNs. The simulation results have shown good results in general, and a significant improvement with respect to the employment of an unweighted naive Bayes classifier. With respect to the unweighted version, the main advantage relies in the ability of the robots to converge to different solution, thus accounting for the situation in which the overall group is split into clusters of robots observing different landmarks. However, the results have also shown a strong sensitivity of the algorithm with respect to design parameters.

Based on these considerations, future works will include not only the application in a localization scheme, but also the study of an adaptive law for the mentioned parameters. Moreover, we plan to include different sensors into the system, for example a lidar sensor. Finally, we will work towards transitioning from simulated environment to real-world application.

ACKNOWLEDGMENTS

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

REFERENCES

[1] 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*, ser. SAC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 307–313. [Online]. Available: https://doi.org/10.1145/2695664.2695922

[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] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, 2019. [Online]. Available: https://robotics.sciencemag.org/content/4/35/eaaw9710
[5] E. Zahugi, M. Shanta, and T. Prasad, "Oil spill cleaning up using swarm of robots," *Advances in Intelligent Systems and Computing*, vol. 178, pp. 215–224, 01 2013.
[6] N. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *Journal of hazardous materials*, vol. 154, pp. 880–7, 07 2008.
[7] 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
[8] 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
[9] R. Dasgupta, S. O'Hara, and P. Petrov, "A multi-agent uav swarm for automatic target recognition," 01 2005, pp. 80–91.
[10] 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.
[11] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *Robotics and Automation, IEEE Transactions on*, vol. 13, pp. 251 – 263, 05 1997.
[12] X. Xu, Y. Luo, and H. Hao, "Vision-based mobile robot localization using natural landmarks," in *2012 International Conference on Systems and Informatics (ICSAI2012)*, 2012, pp. 2012–2015.
[13] Pifu 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.
[14] D. Heo, A. Oh, and T. Park, "A localization system of mobile robots using artificial landmarks," in *2011 IEEE International Conference on Automation Science and Engineering*, 2011, pp. 139–144.
[15] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb 2016.
[16] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.
[17] J. Guo, P. Borges, C. Park, and A. Gawel, "Local descriptor for robust place recognition using lidar intensity," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2019.
[18] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
[19] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1643–1649.
[20] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with smart," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1612–1618.
[21] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 1180–1187.
[22] T. Naseer, W. Burgard, and C. Stachniss, "Robust visual localization across seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018.
[23] J. Facil, D. Olid, L. Montesano, and J. Civera, "Condition-invariant multi-view place recognition," 02 2019.
[24] J. Collier, S. Se, and V. Kotamraju, "Multi-sensor appearance-based place recognition," 05 2013.
[25] X. Shen, "A survey of object classification and detection based on 2d/3d data," 2019.
[26] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2012–2019.
[27] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon, "Automatic line matching and 3D reconstruction of buildings from multiple views," in *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, ser. International Archives of Photogrammetry and Remote Sensing, vol. 32, Part 3-2W5, Munich, Germany, Sep. 1999, pp. 69–80. [Online]. Available: https://hal.inria.fr/inria-00590111
[28] A. Mittal and L. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," vol. 2350, 05 2002.
[29] S. Huang, Y. Chen, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu, "Perspectivenet: 3d object detection from a single rgb image via perspective points," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8905–8917. [Online]. Available: http://papers.nips.cc/paper/9093-perspectivenet-3d-object-detection-from-a-single-rgb-image-via-perspective-points.pdf
[30] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese, "Objectnet3d: A large scale database for 3d object recognition," vol. 9912, 10 2016, pp. 160–176.
[31] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606–1624, 2008.
[32] N. Naikal, A. Y. Yang, and S. S. Sastry, "Towards an efficient distributed object recognition system in wireless smart camera networks," in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–8.
[33] D. McGibney, T. Umeda, K. Sekiyama, H. Mukai, and T. Fukuda, "Cooperative distributed object classification for multiple robots with audio features," in *2011 International Symposium on Micro-NanoMechatronics and Human Science*, 2011, pp. 134–139.
[34] Piyush P., R. Rajan, L. Mary, and B. I. Koshy, "Vehicle detection and classification using audio-visual cues," in *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2016, pp. 726–730.
[35] 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.
[36] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017, pMID: 28599112. [Online]. Available: https://doi.org/10.1162/neco_a_00990
[37] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Image classification for ground traversability estimation in robotics," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham: Springer International Publishing, 2017, pp. 325–336.
[38] A. Giusti, J. Guzzi, D. C. Cireşan, F. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2016.