Cooperative Sensing and Recognition by a Swarm of Mobile Robots

Alessandro Giusti, Jawad Nagi, Luca Gambardella, Gianni A. Di Caro

Abstract— **We present an approach for distributed real-time recognition tasks using a swarm of mobile robots. We focus on the visual recognition of hand gestures, but the solutions that we provide have general applicability and address a number of challenges common to many distributed sensing and classification problems. In our approach, robots acquire and process hand images from multiple points of view, most of which do not allow for a satisfactory classification. Each robot is equipped with a statistical classifier, which is used to generate an opinion for the sensed gesture. Using a low-bandwidth wireless channel, the robots locally exchange their opinions. They also exploit mobility to adapt their positions to maximize the mutual information collectively gathered by the swarm. A distributed consensus protocol is implemented, to allow to rapidly settle on a decision once enough evidence is available. The system is implemented and demonstrated on real robots. In addition, extensive quantitative results of emulation experiments, based on a real image dataset, are reported. We consider different scenarios and study the scalability and the robustness of the swarm performance for distributed recognition.**

I. Introduction

In this paper we present a distributed algorithm for *sensing* and *classification* tasks using a *swarm of mobile robots*. The use of robotic swarms in recognition tasks can result in an advantage, compared to single robot systems, due to their intrinsic parallelism, spatial distribution, and redundancy of resources. Because of these characteristics, a swarm can act as a distributed sensing system able to concurrently gather perceptual information from different points of view and offering robustness to individual robot failures. Example applications include: use of multiple robots exploiting their different view points and sensing modalities for building inspection (e.g., to detect intruders) or for object inspection in production and manufacturing; agricultural robots cooperatively evaluating the health state of a plant or parcel of land; flying robots visually assessing the presence of survivors or potential dangers during search and rescue operations.

When using swarms, one fundamental difficulty consists in the fact that the advantage provided by the presence of a large number of robots is usually payed back in terms of limited computation and sensing power available onboard. In recognition tasks, to overcome the limitations of the individual robots and let the robots in the swarm synergistically act as a single powerful augmented sensor, effective *coordination* and *communication* mechanisms for information sharing and data fusion are needed.

We focus on a specific problem which is a sort of prototype for recognition problems and which is, at the same time, a fundamental problem for the relatively new field of human-swarm interaction: a swarm of mobile ground robots needs to visually recognize human input given by means of specific *hand gestures* in real-time, e.g., for issuing a command to be executed by the service swarm. The solution that we propose is in some aspects specific for the characteristics of the scenario we consider, but the algorithm for coordination and reaching consensus is of general applicability in the context of distributed recognition tasks (e.g., recognition of objects, audio signals, or movement patterns).

The gesture recognition scenario that we consider, and, more in general, distributed recognition scenarios, present a number of challenges that make them particularly difficult. In particular, some sensing positions, often a large part of the possible ones, do not allow to acquire data that are useful or reliable for classification (e.g., presence of occlusions, excessive distance from signal source, or, like in our case, an angle point of view that does not allow a clear discrimination of the gesture). Even when good quality data can be individually sensed, classification results achievable by a single robot may be quite unreliable (e.g., due to the inaccuracy of the sensing devices or of the on-board image analysis algorithms, which are typical limitations in swarm robotics). Therefore, we propose a solution in which every robot in the swarm cooperatively engages in the (gesture) recognition task; robots in the swarm acquire and process data in parallel, which leads to significantly better performance than that of a single robot that moves and gathers multiple images. Each robot is equipped with a *statistical classifier*, trained on a large dataset of image gestures, which is used to issue *probabilistic classifications* of the hand gesture. *Mobility* is used to optimize robot sensing positions, reduce redundancies in the sensed images, and provide wireless connectivity. *Multi-hop wireless communications* provide the mean to exchange and probabilistically fuse individual gesture classifications among the robots. A *distributed consensus protocol* let the swarm as a whole reach a consensus about the issued gesture, with the converging time depending on parameters defining how *prudent/reliable* the consensus process should be (i.e., how fast the answer from the swarm is needed, depending, for instance, on the urgency of the situation at hand).

The paper is organized as follows. In Section II we discuss related work in different domains. Section III describes statistical classification of hand images at the single robot. The distributed algorithms for cooperative swarm-level recognition are described in Section IV. Section V presents the *foot-bot* [1] robot platform, which we used for real-robot implementation, and the gesture dataset used for training and testing. Section VI reports quantitative evaluation of

A. Giusti, J. Nagi, L. Gambardella, and G. A. Di Caro are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland. {alessandrog,jawad,luca,gianni}@idsia.ch.

system performance using emulation experiments: impact of different parameters affecting the recognition accuracy and swarm's response dynamics are investigated, showing good *accuracy, scalability*, and *robustness*.[1]

II. Related work

Visual gesture recognition has been extensively studied in computer vision research (see [2] for a survey), motivated by a several possible applications [3], including human-robot interaction [4], [5]. Recent research also focuses on the use of depth-aware cameras [6]. In this paper we use a simple 2D appearance-based approach suitable for robots equipped with limited computing power and low-quality cameras, as those typically used in swarms. Unlike most related works, we do not assume that the gesture is directed towards the camera: each robot observes the user from a different viewpoint of view, many of which yield unsatisfactory observations. In this regard, gesture recognition well models a larger class of distributed sensing problems in which observed data quality strongly depends on sensor position.

Fusion of the observations from multiple wide-baseline static cameras is a technique used in many applications, including body pose [7] and head pose [8] recovery, with the multiple views providing valuable inputs for reconstruction of 3D information. However, the coordinated use of large amounts of vision sensors raises the need of distributed information processing and optimized communications [9]. For instance, in many cases distributed camera networks make use of local image processing, to limit data exchange to manageable amounts. In our system we follow the same approach, also considering that we deal with a short-range, unreliable, and very low-bandwidth communication system.

Our swarm needs to settle on a decision in a fully distributed way. The problem is known as *distributed consensus*, and its theoretical framework has been subject to a large amount of research (see [10] for an overview), also in the context of distributed sensing [11]. Our system adopts a relatively simple consensus protocol, which we empirically prove to be nevertheless effective even in the considered challenging communication scenarios. Collaborative perception in swarms is also discussed in [12], where the outputs of IR reflectance sensors mounted on mobile robots are fused in a distributed way to asses the shape of an object.

In our protocol we also exploit mobility for better sensing. The interplay between sensing and mobility for groups of networked sensors is explored from the theoretical point of view in [13] for target tracking applications. A single mobile robot is considered in [14] for viewpoint planning and optimization in visual object detection and recognition.

Vision-based interaction between a human and multiple mobile robots (equipped with relatively powerful laptops and external cameras) is demonstrated in [15], where each robot independently attempts to detect a frontal gaze from the human, and reacts according to a simple collective inference mechanism. The study is quite preliminary, demonstration-oriented, and as such lacks of a detailed analysis.

III. Single Robot Recognition of Hand Gestures

The hand gesture recognition process starts with the single robots detecting the human performing the gesture by using their on-board camera. Each robot *processes the grabbed image* in order to extract the information regarding the gesture, and then issues a classification of the gesture based on a predefined set of possible known gestures. The output of this process is the *opinion* of the robot for the gesture, expressed in the form of a numerical vector assigning a *posterior probability* to each possible gesture. The process is iterated through the repeated grabbing of new images of the same (or new) gestures. Robot opinions are sent out and spread throughout the robot network in order to allow the swarm as a whole to share the individually sensed information in terms of probabilistic opinions. A *distributed consensus* process (described in Section IV) controls the spreading and fusion of the individual opinions in order to rapidly and cooperatively reach a robust swarm-level consensus of the hand gesture, and act accordingly. In the following of this section we describe the hand-gesture recognition process for the single robot, that is, the actions that a robot perform to process the image and produce an opinion vector.

In order to simplify the image recognition task without losing generality, we assume that the human wears a hand glove with a known characteristic color. Once the hand is detected in the image, the robot interprets its shape, thus obtaining a classification vector $\mathbf{C}$, the robot opinion, which contains a probability for each gesture. Throughout the paper, we consider a set of $K = 6$ gestures in terms of *finger counts*, in which zero (closed hand) to five fingers are shown, as illustrated in Figure 1. Gestures may be shown in any rotation and in any finger combination.

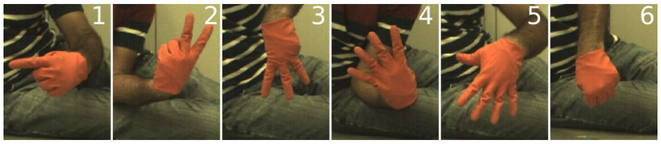

Fig. 1: The six finger-count gestures.

The single robot recognition problem is difficult because the image to be processed may be taken from any point of view (angle and distance) with respect to the hand, the hand orientation is unknown, and due to the limited camera resolution and processing power of the foot-bot robots (see Section V). For instance, Figure 2 shows that, when the gestures are viewed from non optimal (i.e., non perpendicular) angles, they become difficult to distinguish even for a human observer.

To deal with the single robot recognition task, we adopt a basic computer vision approach based on *segmentation*, *feature extraction*, and *supervised classification*. We accept that a single robot might have poor recognition performance, especially when observing the hand from a bad view point. Our main goal is to obtain a satisfactory accuracy at the

[1] Supplementary material available on the web includes videos, the image dataset, and MATLAB code for experiment replication: `http://www.idsia.ch/~gianni/SwarmRobotics/iros2012/`

swarm level in a reasonably short time by integrating the opinions from multiple robots (some of which will hopefully be in a good position), as discussed in Section IV. In this perspective, the core property of the robot-level recognition technique is not accuracy, but the ability to return a classification vector well representing the ambiguities in the data and the difficulty of the instance. Therefore, if the robot position does not allow a precise assessment of the gesture, its opinion should show a set of more or less uniform probabilities (*high information entropy*), while if the robot is highly confident of its assessment, this should be reflected in an opinion vector with one class peak (*low information entropy*). These properties are experimentally validated in Section VI-A.

Color-based Segmentation. Once an image is acquired, the first processing step consists in *segmenting* the hand, by exploiting the characteristic color of the glove with a standard color-based segmentation approach in the HSV color space [16]. The largest *connected component* in the resulting binary image is identified as the hand, and used to compute quantitative *features*.

Feature Computation and Selection. Numerical features are computed using a total of *40 geometrical and invariant properties* frequently used in literature for similar recognition tasks, which include[2] roundness, eccentricity, extent, area moments of inertia and Hu invariant moments. The weight for each feature was determined by its *information gain*, and the top 20 features were retained for classification.

Probabilistic Classification The resulting 20-element feature vectors are classified by means of a non-linear *Support Vector Machine* (SVM) with an RBF-*Gaussian* kernel. The SVM was trained using a subset of the images in the dataset described in Section V-B. A disjoint subset was used in the experiments for testing. The training set includes samples from all points of view: this results in a classifier which can deal with observations from any viewpoint.

The SVM classifier returns the posterior probabilities of the six classes, i.e., the 6-dimensional opinion vector $\mathbf{C}$, propagated and used in the swarm as discussed below.

IV. Cooperative Recognition and Distributed Consensus

Let $R = \{r_1, r_2, \cdots, r_N\}$ be the set of the N robots comprising the swarm. In our scenario, when the swarm is idle or is performing some routine activity, a subset of the robots is continuously searching for the presence of a human and its willingness to communicate. This is indicated by the presence of an orange-colored glove: when a robot detects it in its field of view, it locally broadcasts a special starting message which is then relayed to the whole swarm in a multi-hop fashion using a line-of-sight wireless interface; each robot then moves to suitable observation positions and enters a *Recognition* state, on which the present work is focused.

When in the *Recognition* state, each robot $r \in R$ performs four parallel activities: sensing, communication, data integration, and mobility, which are described in the following.

A. Sensing

Each robot in the swarm iteratively acquires and processes video image frames; for each frame, the robot attempts to localize the hand and classify its shape, thus generating an *opinion*, as described in Section III. The robot generates and broadcasts a new opinion for each acquired frame. An opinion has two components: a) the classification vector of K probabilities $\mathbf{C} = [P(i_1), P(i_2), \cdots, P(i_K)]$, and b) a *weight* w for the opinion, meant to account for potential *redundancies* between opinions of nearby robots. The rationale behind the precise meaning of w and its calculation are described in the rest of this subsection.

When the distribution of the robots in the swarm is not uniform, it might happen that small groups of robots are very close to each other. Therefore, the individual opinions of these robots need to be discounted to account for their redundancy, since they observe very similar images, such as they will likely produce very similar opinions. The weight w of their opinions is then decreased compared to the weight of the opinions of more isolated robots. In this way, we avoid to over count opinions referring to the same point of view, favoring at the same time the realization of a consensus truly based on sensing diversity (e.g., if a group of robots are all located in a bad position and all issue the same wrong classifications, these, if not properly discounted, can produce a large wrong bias in the entire swarm decision process).

Due to the geometry of the problem, we consider that the redundancy between observations of two close by robots mainly depends on the angle between them relative to hand position.[3] Note that we assume that the true orientation of the hand (i.e., the direction it is pointing towards) is unknown to the robots: in fact, estimating such orientation, without knowing the actual gesture performed, is by itself a problem of comparable difficulty to recognizing the gesture.

Weight calculation is performed as follows. Let $d(r_1, r_2)$ denote the angular distance between two robots r_1 and r_2. We assume that there is a minimum angular distance d_m, such that, when $d(r_1, r_2) > d_m$, the observations of r_1 and r_2 can be considered as independent (in the experiments d_m was set to 30°). The weight w_r of a robot r only depends on the number and the positions of its neighbors closer than d_m. Let $Q(r) = \{q_1, q_2, \cdots q_{|Q|}\}$ be the set of such neighbors (referred to as *sensing neighbors*), and let $|Q(r)|$ denote its cardinality. We define w_r according to the following formula:

$$w_r = \frac{1}{\left(1 + \sum_{q \in Q(r)} c\left(d\left(r, q\right)\right)^2\right)} \ , \qquad (1)$$

where $c(\cdot)$ is a linear function of the distance between r and its neighbor, defined by $c(d_m) = 0$ and $c(0) = \rho$, with $0 \leq \rho \leq 1$. The parameter ρ indicates the expected redundancy between observations of two robots at exactly the same position and its value is experimentally set to $\rho = {}^2\!/_3$ (see Section VI-B). The use of Eq. (1) to define w has several

[2] The full list of shape features and the feature selection approach are detailed in the supplementary material available on the web.

[3] Observations of the hand from the same angle and different distances result in scaled instances of similar shapes, so we ignore the radial component of robot positions for computing the weights.

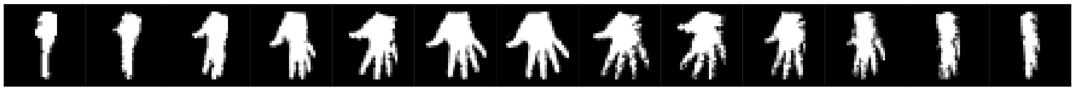

Fig. 2: Segmentation results for images of the same gesture acquired from different points of view.

nice properties: *a*) function $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N$ mapping the position of robots to weights is continuous; *b*) if a robot $q \in Q(r)$ moves farther from robot r, w_r increases, such that r's opinion gains importance; *c*) it can be proven that the optimization of a robot's own weight has as consequence the equalization of the distances between the robot and its neighbors. According to (1), a robot computes its weight by only knowing the angular distance to each sensing neighbor. Using foot-bots, this is possible with the measures provided by the range and bearing device (see Section V).

B. Mobility

While in the *Recognition* state, robots *move* in order to optimize the swarm distribution for the recognition task.

A robot selects its radial position with the goal of gathering better quality observations. This is achieved by positioning at a distance $d_o = 1.5$m, which maximizes single robot accuracy, as shown in Section VI. The angular position cannot be similarly optimized, because the direction θ of best observation (i.e., $\theta = 0$) is unknown. Instead, a robot exploits tangential moves to increase the weight of its opinion. This is realized by maximizing the angular distance from its closest neighbor, which has the effect of increasing the amount of information collectively gathered by the swarm, as the overall result is a uniform angular spacing of the robots (see Figure 3). In practice, at each control step, a robot estimates the radial and tangential components, and moves in the direction of the resultant vector that optimizes the two objectives. This simple approach results in an emerging collective behavior in which robots tend to equalize their angular distances along a semi-circle centered on the gesture. Performance improvements resulting from such mobility strategy are quantified in Section VI-C.

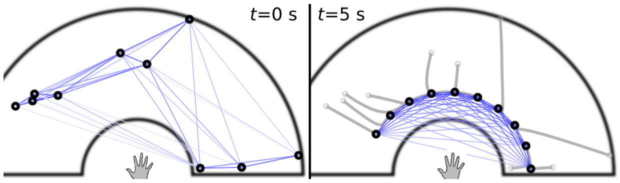

Fig. 3: Initial random positions (left) and final positions (right) of 10 robots implementing the mobility strategy of Section IV-B. Robot trajectories are represented as gray lines. Thin blue lines show distance-dependent link quality between robot pairs (a light line means high packet losses).

C. Multi-hop communications

After an opinion is built, it is sent out and propagated to the rest of the swarm encoded in an `OPINION` message. In our scenario, robots are equipped with IR-based line-of-sight communication devices: messages are locally broadcast using such devices, and relayed by the neighbors in a multi-hop fashion, eventually propagating to the whole swarm.

Each robot stores the set of all unique received `OPINION` messages, as well as its own opinions. This represent the robot's view of swarm's observations. Due to multi-hop communication delays or packet losses, not all robots necessarily share the same set of opinions at the same time.

D. Data fusion

While robots are in the *Recognition* state, the swarm continuously generates and exchanges opinions, thus accumulating evidence about the input that the user is providing. When enough evidence is accumulated in favor of a gesture class i', that is, a consensus is being formed in the swarm, a final decision regarding the gesture must be collectively taken. Following the assessment of the swarm decision, all robots enter the corresponding *Action(i')* state associated to gesture i' through the mechanisms described in the following section. In our fully-decentralized approach, no particular robot has the responsibility of triggering the swarm-level decision. Instead, *any* robot may initiate such process at any moment by fusion of all the pieces of information it is aware of. How this is done in practice is described below.

All classification vectors available to a robot (either the received external opinions and/or the own opinions) are treated as noisy measures. In order to implement a computationally fast yet effective fusion of this information, each robot performs a linear composition of the available classification vectors: each new vector is summed up, component-wise, to the previous ones, weighted by its associated weight w. This process happens incrementally, as soon as a new opinion is available at the robot. In this way, for each gesture class, the weighted average of its opinion probability is calculated, resulting at time t in a *local decision vector* $\mathbf{D}(t)$ that additively summarizes all available information. Note that vector $\mathbf{D}(t)$ is not normalized and its six components sum to the total weight of all opinions received so far. Figure 4 illustrates the way the decision vector is built, as well as the meaning of the λ measure of confidence (see below).

If the information in the decision vector indicates that the probability of one class is significantly greater than the probability of the other classes, and, at the same time, enough sample opinions have been collected, the robot can issue its own *final* decision regarding the gesture. More specifically, let i' be the index of the largest component of $\mathbf{D}(t)$, that is, the class in favor of which most evidence is available at time t to the robot, and i'' the index of the second largest component of $\mathbf{D}(t)$. As a measure of *confidence* about the true class being i' we use $\lambda(t) = \mathbf{D}(t)_{i'} - \mathbf{D}(t)_{i''}$. λ is proportional to the minimum amount of evidence which needs to be acquired in favor of any other class in order to change the result. In order to determine whether there is enough evidence in favor of i' for taking a decision, the robot compares the computed $\lambda(t)$ to a fixed threshold λ_s, which is a swarm-level parameter encoding the tradeoff between recognition speed (small λ_s) and accuracy (large λ_s). If $\lambda(t) > \lambda_s$ the robot can sent out its own final decision.

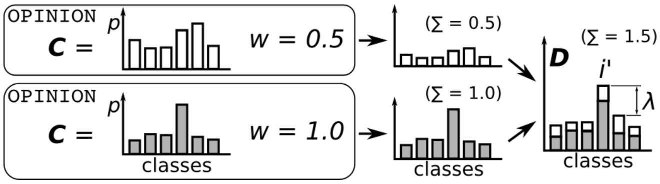

Fig. 4: Building of the decision vector and meaning of λ.

E. Swarm-level decision making

At any robot of the swarm, the occurrence of one of two conditions triggers a swarm-level decision: a) λ exceeding λ_s (*evidence trigger*), b) the time since entering the *Recognition* state exceeding a swarm-level parameter T (*time trigger*). If one of the two conditions hold, then a robot d broadcasts a $\texttt{DECISION}(i'_d, \lambda_d)$ message, and immediately makes a transition to the *Action(i'_d)* state. Setting different values for the parameters λ_s and T determine different response dynamics of the swarm: how the swarm is *prudent* or *fast* defining a classification and, in turn, actuating the associated action. For instance, setting both parameters to small values determine a fast but potentially inaccurate classification response. Setting T to a large value means to allow the swarm to gather enough statistical evidence before triggering an action. These parameters can be tuned in accordance to the requirements of the application scenario under consideration.

Depending on its current state, a robot r receiving a $\texttt{DECISION}(i'_d, \lambda_d)$ message reacts in different ways, based on its current values of i'_r and λ_r.

If r currently is in *Recognition* state:

- if $\lambda_d \geq \lambda_r$, the robot *adopts* the incoming decision by setting $i'_r \leftarrow i'_d$ and $\lambda_r \leftarrow \lambda_d$; the message is forwarded and a transition to state *Action(i'_d)* is performed;
- else, the robot *overrides* the decision by discarding the incoming message and using its own information to set up a decision, since the swarm is already issuing one: it broadcasts a new $\texttt{DECISION}(i'_r, \lambda_r)$ and makes a transition to the *Action(i'_r)* state.

If robot r is in an *Action* state, it reacts to an incoming decision as follows:

- if $\lambda_d > \lambda_r$, the decision is adopted: $\lambda_r \leftarrow \lambda_d$ and the message is forwarded; if $i'_d = i'_r$ the robot does not need to change its current action, while, if $i'_d \neq i'_r$, then $i'_r \leftarrow i'_d$ and the robot makes a transition to the new *Action(i'_d)* state, replacing its current action with one which is supported by more evidence;
- if $\lambda_d \leq \lambda_r$ decision is ignored and message is discarded.

Note that the propagation mechanism for DECISION packets ensures that eventually the whole swarm converges to the same decision even when different robots independently and asynchronously take different decisions in different parts of the swarm, with the winning decision being the one with the largest confidence. Decision packets *compete* with each other: decisions with large λ values override and interrupt the propagation of weaker decisions. The currently selected decision is periodically rebroadcast by the robots in an *Action* state, in order to ensure that robots of the swarm that were potentially disconnected at the time the original decision was spread out are kept up to date. The assumption is that by mean of robot mobility these disconnected robots will eventually get in touch with the rest of swarm.

V. Implementation

The described system was implemented on real robots using the foot-bot platform. In addition, a simulation environment was built for running the quantitative experiments.

A. Robot Platform

The foot-bot robot (Figure 5) is a small mobile platform, directly derived from the *marXbot* [1], which is specifically designed for swarm robotics applications [17]. The robot is based on an on-board ARM-11 processor and is programmed in a Linux-based operating environment. We use a subset of the platform's capabilities, namely: a *frontal camera*, which acquires 512×384 RGB images; *motorized track-based wheels* for mobility at speeds up to 0.3 m/s; an IR-based *range-and-bearing* sensor and actuator system, which allows a robot to detect its *line-of-sight* neighbors up to a range of a few meters, estimate their distance and bearing, and send them messages through a low-bandwidth (100 Bits/s), low-reliability, communication channel.

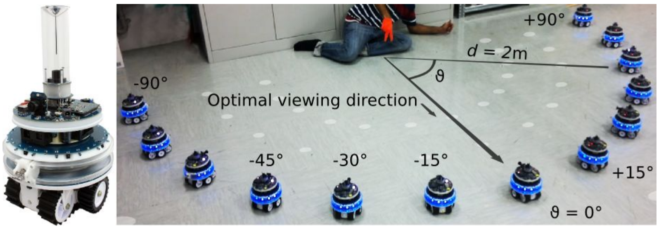

Fig. 5: The foot-bot robot and acquisition setup ($d = 2$m).

B. Data Acquisition for Training and Test Sets

A swarm of 13 foot-bots was used to acquire a large dataset of gestures from different points of view with known ground truth. One-third of the dataset (*training set*) was used to train, off-line, a single SVM classifier which is then used by all robots during the experiments. The rest of the dataset was used to perform the simulated experiments reported in Section VI (*test set*). A small fraction of the dataset was also discarded, in order to ensure that if a frame appears in the training set, frames taken shortly before or after (which may be very similar) do not appear in the testing set.

Robots were placed at evenly-spaced angles of 15° covering a half-circle centered on the person showing the gesture (see Figure 5). The gesture was performed as if directed to a viewer precisely in front (at the zero angle). Therefore,

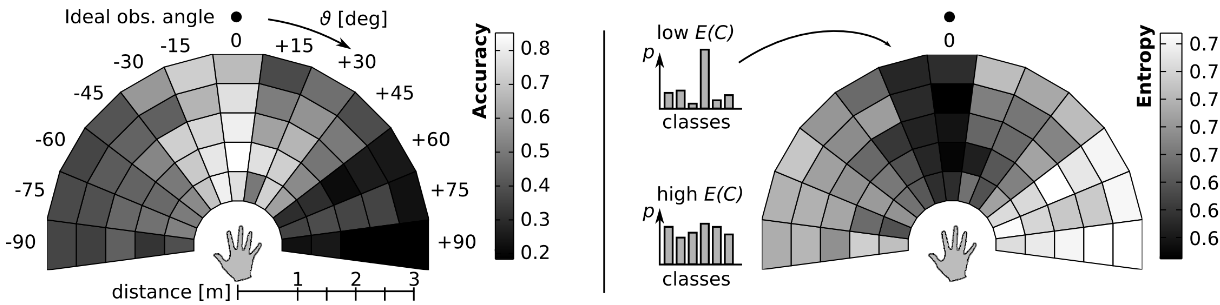

Fig. 6: Average accuracy (left) and average entropy (right) of classification vectors obtained from different viewpoints.

the central robot is in the optimal point of view, whereas the remaining robots see the gesture from angled vantage points.

Each of the 13 robots has acquired and stored $\approx$ 190 unprocessed, timestamped images in 95s, while the user was showing a single known gesture and slowly changing the hand orientation and other characteristics of the gesture (e.g., finger combinations). After a 5s pause signaled by robots, acquisitions for the next class began, with the procedure being repeated for each of the 6 classes. In total, the swarm could acquire about $13 \times 190 \times 6 = 14820$ images in ≈ 10 minutes. The procedure was repeated 5 times in total, once for a different distance between the robots and the human operator (3.0m, 2.5m, 2.0m, 1.5m, 1m). This resulted in the final dataset comprising roughly 74000 images acquired by the swarm in a total of 65 different viewpoints.

VI. Experimental results

In the following of this section we report the experimental results for single-robot and swarm-level accuracy in gesture recognition, as well as the impact of different aspects such as the value of the prudence parameter λ_s, the number of robots in the swarm, and the use of mobility.

The swarm experiments, carried out in simulation using the MATLAB environment, are actually *emulation* experiments, since classifications are always based on the real images included in the test set described in Section V-B. Once a ground truth gesture has been assigned, a simulated robot positioned at (d, θ) in the polar plane centered on the human hand, 'sees' an image which is randomly selected from the subset of the test images that were acquired from the viewpoint closest to (d, θ) (for the same gesture).

A. Single-Robot Classification Accuracy and Entropy

We first study the recognition accuracy of a single robot as a function of its position. These experiments show how effective is the trained classifier. Results are reported in Figure 6 (left). Robots positioned in central locations (i.e., close to $\theta = 0°$, the direction the gesture was directed to) provide good recognition accuracies (up to 81%). The performance systematically degrades with the increase of the angle with respect to the hand position and with the increase of the radial distance. Performance of robots at the very radial periphery is extremely poor, due to the bad viewpoint.

In Figure 6 (right) we report, in addition to classification accuracy also the *entropy* of the output classification vectors. In fact, since the impact of an opinion vector $\mathbf{C}$ in the distributed consensus process is determined, other than by its weight, also by the relative differences among $\mathbf{C}$'s components, the entropy of $\mathbf{C}$ can precisely quantify how much information is carried in the opinion. The values reported in the figure refer to the normalized entropy $H(\mathbf{C})$, which is low only when the opinion strongly favors some classes over others. For example, $H(\mathbf{C} = \{0, 0, 1, 0, 0, 0\}) = 0$, and $H(\mathbf{C} = \{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}) = 1$. Figure 6 shows that classification vectors resulting from angled viewpoints exhibit, on average, a larger entropy than classification vectors from good viewpoints. As a result, in the consensus process, the opinions of robots in good positions have a larger effect in determining the winning class with respect to other opinions, which is precisely what we want. Similarly, correct opinions have classification vectors with an average entropy $H = 0.671$, significantly lower than the average entropy of wrong classification vectors, $H = 0.763$.

B. Effect of Weighting in Opinion Fusion

In this section we report the results of an experiment designed to study the effect of the weighting scheme for the opinion vectors described in Section IV. We consider a scenario with three robots, a, b and c, all at the same distance d from the hand. Robots a and b are very close to each other and share the same angular position $\theta_{\mathrm{a}} = \theta_{\mathrm{b}} = \theta'$, whereas robot c is at a much larger angular distance, $\theta_{\mathrm{c}} = \theta' \pm 30°$ (i.e., it is isolated from the other two). Each robot outputs one classification vector. These vectors are fused, with the weight of robot c fixed to the maximal value, while the weight of the grouped robots a and b is varied: $w_{\mathrm{c}} = 1$ and $w_{\mathrm{a}} = w_{\mathrm{b}} = w_{ab} \in [0, 1]$. In this way, we can study the impact of different weightings for robots a and b, which report correlated opinions. For each experiment trial the swarm decision vector $\mathbf{D} = w_{\mathrm{ab}}\mathbf{C}_{\mathrm{a}} + w_{\mathrm{ab}}\mathbf{C}_{\mathrm{b}} + w_{\mathrm{c}}\mathbf{C}_{\mathrm{c}}$ is computed, and the result is considered successful if the largest element of $\mathbf{D}$ corresponds to the true class of the gesture.

The real robot scenario is emulated by using as observations for a, b and c, three images belonging to the testing set and acquired during the same time interval. In this way, we approximate the simultaneous acquisitions of the same scene from the three robots. The procedure is repeated for all triplets of observations in the testing set and for different values of θ'. The resulting average classification accuracy is computed as a function of the weight w_{ab}. Experiment results are reported in Figure 7a. The thin red line shows that the average accuracy peaks for $w_{ab} \approx 0.7$. The improvement

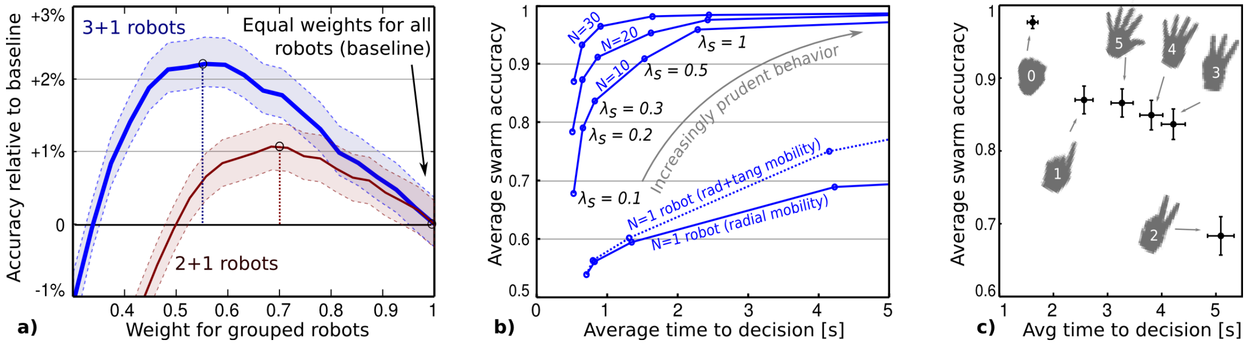

Fig. 7: ***a***) Effect of opinion weighting on accuracy. *Thin red line*: 2 robots group + 1 isolated robot ($w = 1$). *Thick blue line*: 3 robots group + 1 isolated robot ($w = 1$). 95% confidence intervals are reported. ***b***) Tradeoff between accuracy and time to decision for different values of λ_s and swarm sizes N. Each data point is averaged over 60 replicas and 6 gestures. The dotted line shows the case of one robot radially and tangentially moving during all the experiment in order to cover multiple viewpoints. ***c***) Average accuracy and time to decision for each of the 6 gestures. Deciding on difficult gestures takes longer time, as is the case for class 2 (which can result from many different combinations of (two) fingers).

over the baseline (all robots have the same weight) is limited but statistically significant for $p < 0.01$ under the Wilcoxon paired signed-rank test. Using the peak value $w = 0.7$ and solving for ρ in Eq. (1) with one single neighbor at distance 0, yields $\rho \approx 2/3$, which is the value we set ρ in the experiments of the following sections. Repeating the same experiments with a triplet of robots in the same position and a fourth isolated robot (thick blue line), the average accuracy peaks when the weight of the correlated robots is set to $w = 0.55$ (with a +2.5% improvement over the baseline), which is consistent with the value $\rho \approx 2/3$ in Eq. (1).

C. Swarm-level accuracy vs. λ, swarm size, and mobility

The simulation/emulation experiments described in the following are aimed to study swarm-level recognition accuracy and response time depending on the value of different parameters of the distributed consensus protocol. In all experiments, robots' opinions are produced once per second, which is the performance attained on the foot-bot platform, accounting for image acquisition, processing and classification. Robot communications are simulated with parameters matching the characteristic of the foot-bot's IR communication system: packets are received with a 0.1s delay; packet loss probability is modeled as a piecewise linear function $\pi(d)$ of the distance d between two communicating robots: $\pi(d \leq d_{min} = 1\text{m}) = 0.2, \pi(d \geq d_{max} = 4\text{m}) = 1$, and $\pi(d_{min} < d < d_{max})$ is the line segment between $(d_{min}, \pi(d_{min}))$ and $(d_{max}, \pi(d_{max}))$.

For each tested scenario (i.e., a set of parameter values), we perform a large number of simulation trials with different random positioning of the robots and different random sampling of observations from the test set. Each simulation results in one of three outcomes: *success* (all robots reach an *Action* state for the correct class), *failure* (all robots reach a *Action* state for the same, wrong class), or *no consensus* (none of the previous outcomes is true at $t = 2T$). For each experiment, we record the time to decision, defined as the earliest time in which all robots are in the *Action* state for the same class (or $2T$ in case of *no consensus*). Two performance measures are computed for each scenario: the *average accuracy*, i.e., the fraction of experiments with *success* outcome, and the *average time to decision*.

1) Effect of λ_s: The swarm-level parameter λ_s determines how much statistical evidence a robot needs in order to initiate a swarm-level decision. Figure 7b shows that decisions taken on the basis of less evidence (small λ_s) are less accurate but issued faster compared to more prudent decisions (large λ_s), i.e., decisions taken only when a very solid evidence is available. Moreover, the figure shows that, with the increasing of its size N, the swarm improves both accuracy and time to decision, and the value of λ_s has a reduced impact on performance (e.g., for $N = 10$ swarm accuracy ranges from 0.68 ($\lambda_s = 0.1$) to 0.95 ($\lambda_s = 1$), while for $N = 30$ it ranges from 0.88 to 0.99). A single robot, even when keeping moving both radially and tangentially, in order to cover as many viewpoints as possible (dotted line), has a poor performance compared to a robot swarm, which can exploit its parallelism to acquire data in a much faster way.

2) Effect of Swarm Size: Increasing the number N of robots in the swarm has two main positive effects on the distributed consensus protocol: a larger number of different opinions are available, and swarm connectivity is improved due to increased robot density. The general impact of an increased swarm size on recognition accuracy is shown in Figures 7b and 8a in relation to the value of λ and the use of mobility. As expected, a larger swarm always results in higher accuracies. The results reported in Figure 8b illustrate the specific impact of N on communications, showing that larger swarms are quite robust to very high packet loss rates.

3) Effect of Mobility and Communication Losses: Figure 8 reports the average accuracy obtained with and without mobility for different swarm sizes and packet loss rates. In the experiments, to have approximately the same amount of observations per robot in each scenario, λ_s is set to a very large value, (i.e., robots never take a decision before the time triggering threshold T is reached). The figure shows the positive effects of mobility on accuracy (statistically significant according to the Wilcoxon paired signed-rank test,

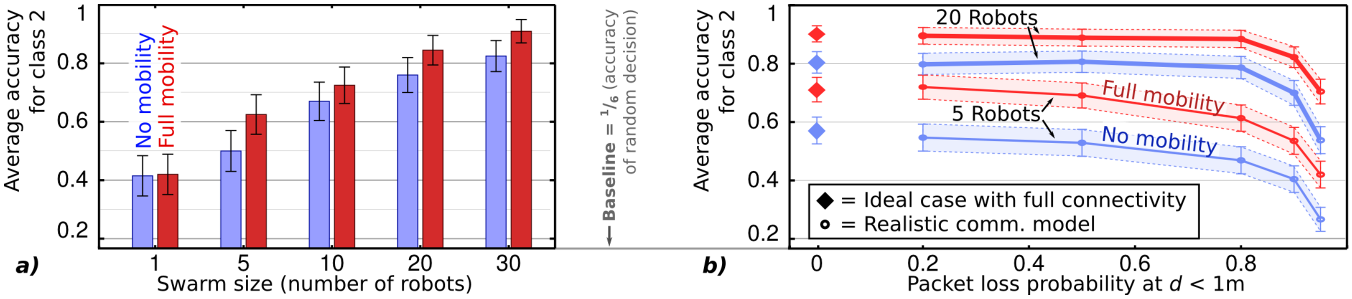

Fig. 8: ***a***) Effect of mobility and swarm size on swarm accuracy. Each data point is the average of 200 trials considering only gesture class 2 (the hardest to detect). 95% confidence intervals are reported. T and λ_s are set respectively to 3s and ∞. In the no mobility case, robots do not move after the initial random deployment. ***b***) Effect of different packet loss probabilities. Both static and mobile swarms (size 5 and 20) are investigated. As in a), only gesture class 2 is considered. Each data point is the average of 450 trials. 95% confidence intervals are reported, $T = 4$s, $\lambda_s = \infty$.

for $p < 0.01$). Mobility also improves communication as it tends to group robots (see Figure 3) which ensures a more efficient multi-hop propagation of messages. Figure 8b shows the swarm's resiliency to unreliable communications. Large swarms show no significant decrease in performance up to 80% of packet loss probability.

VII. Conclusions and Future work

We have introduced an algorithmic protocol for the cooperative recognition of hand gestures by a swarm of mobile robots. The protocol, based on distributed sensing, consensus, and multi-hop information sharing, is not limited to gesture recognition, but can be immediately adapted and applied to different scenarios in which a group of robots need to cooperatively sense and classify an entity of interest.

The system was implemented both on a swarm of real mobile foot-bot robots and in a simulation environment using real images. In a number of different quantitative experiments we have shown that system's recognition accuracy scales effectively with the increase of the number of robots in the swarm, and is robust to heavy communication losses. These are fundamental target properties for swarm systems. Moreover, we have shown that the swarm can reliably converge to a unanimous decision when enough evidence is available in the swarm in favor of a class. Recognition accuracy and response speed can be effectively and smoothly balanced using a single parameter, λ. Finally, we introduced simple criteria to exploit and control robot mobility with a twofold objective: to maximize the mutually collected information and to improve wireless connectivity, both resulting in significant improvements in recognition accuracy.

We are currently focusing on online approaches, in which training gestures are provided incrementally and interactively by a human instructor. The challenge is to let the swarm cooperatively learning out of a few training samples. Moreover, in these same context, we are exploring the use of simple yes/no feedback signals for gesture training, which are easy for a human to provide in an interactive scenario.

References

[1] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4187–4193.

[2] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 37, no. 3, pp. 311–324, 2007.

[3] J. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[4] X. Yin and M. Xie, "Finger identification and hand posture recognition for human-robot interaction," *Image and Vision Computing*, vol. 25, no. 8, pp. 1291–1300, 2007.

[5] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition," in *Proceedings of the 3rd IEEE International Conference on Signal & Image Processing and Applications (ICSIPA)*, 2011.

[6] *Chalearn Gesture Challenge*, http://http://gesture.chalearn.org/.

[7] H. Aghajan, C. Wu, and R. Kleihorst, "Distributed vision networks for human pose analysis," *Signal Processing Techniques for Knowledge Extraction and Information Fusion*, pp. 181–200, 2008.

[8] C. Wu and H. Aghajan, "Head pose and trajectory recovery in uncalibrated camera networksâRegion of interest tracking in smart home applications," in *Proceedings of the International Conference on Distributed Smart Cameras (ICDSC)*. IEEE, 2008, pp. 1–7.

[9] R. Chellappa, W. Heinzelman, J. Konrad, D. Schonfeld, and M. Wolf, "Special Section on Distributed Camera Networks: Sensing, Processing, Communication, and Implementation," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2513–2515, 2010.

[10] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[11] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.

[12] S. Kornienko, O. Kornienko, C. Constantinescu, P. M., and P. Levi, "Cognitive micro-Agents: individual and collective perception in microrobotic swarm," in *Proceedings of the IJCAI Workshop on Agents in Real-Time and Dynamic Environments*, 2005.

[13] R. Olfati-Saber and P. Jalalkamali, "Coupled Distributed Estimation and Control for Mobile Sensor Networks," *IEEE Trans. on Automatic Control*, vol. 57, no. 9, p. 1, 2012.

[14] K. Shubina and J. Tsotsos, "Visual search for an object in a 3D environment using a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547, 2010.

[15] B. Milligan, G. Mori, and R. Vaughan, "Selecting and commanding groups in a multi-robot vision based system," in *Proceedings of the Int. Conf. on Human Robot Interaction (HRI)*, 2011, pp. 415–416.

[16] R. Gonzalez and R. Woods, *Digital image processing*. Prentice Hall, pp. 954, 2008.

[17] M. Dorigo *et al.*, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, 2012 (to appear).