Symbiotic Interaction between Humans and Robot Swarms

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by

Jawad Nagi

under the supervision of

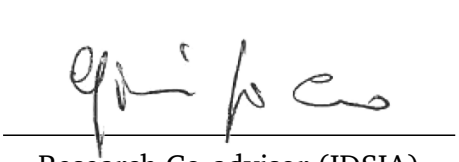
Dr. Gianni Di Caro and Prof. Luca Gambardella

May 2016

Dissertation Committee

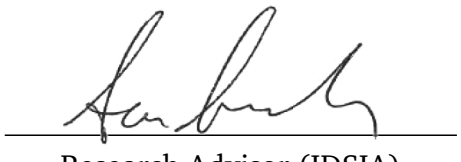
Prof. Alcherio Martinoli	École Polytechnique Fédérale de Lausanne, Switzerland
Prof. Carles Sierra	Universitat Autònoma de Barcelona, Spain
Prof. Cesare Pautasso	Università della Svizzera Italiana, Switzerland
Prof. Fabio Crestani	Università della Svizzera Italiana, Switzerland
Prof. Richard Vaughan	Simon Fraser University, Canada

Dissertation accepted on: 25 May 2016



Research Co-advisor (IDSIA)

Dr. Gianni A. Di Caro



Research Advisor (IDSIA)

Prof. Luca M. Gambardella

PhD Program Director

Prof. Walter Binder

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved PhD research program.

Jawad Nagi (M.Sc., M.Eng.)
Lugano, 25 May 2016

To my beloved parents and sister,

Abstract

Comprising of a potentially large team of autonomous cooperative robots locally interacting and communicating with each other, robot swarms provide a natural diversity of parallel and distributed functionalities, high flexibility, potential for redundancy, and fault-tolerance. The use of autonomous mobile robots is expected to increase in the future and swarm robotic systems are envisioned to play important roles in tasks such as: search and rescue (SAR) missions, transportation of objects, surveillance, and reconnaissance operations. To robustly deploy robot swarms on the field with humans, this research addresses the fundamental problems in the relatively new field of *human-swarm interaction* (HSI). Four groups of core classes of problems have been addressed for *proximal interaction* between humans and robot swarms: interaction and communication; swarm-level sensing and classification; swarm coordination; swarm-level learning.

The **primary contribution** of this research aims to develop a *bidirectional human-swarm communication system* for non-verbal interaction between humans and heterogeneous robot swarms. The guiding field of application are SAR missions. The core challenges and issues in HSI include: *How can human operators interact and communicate with robot swarms? Which interaction modalities can be used by humans? How can human operators instruct and command robots from a swarm? Which mechanisms can be used by robot swarms to convey feedback to human operators? Which type of feedback can swarms convey to humans?* In this research, to start answering these questions, *hand gestures* have been chosen as the interaction modality for humans, since gestures are simple to use, easily recognized, and possess spatial-addressing properties.

To facilitate bidirectional interaction and communication, a *dialogue-based interaction system* is introduced which consists of: (i) a *grammar-based gesture language* with a vocabulary of non-verbal commands that allows humans to efficiently provide mission instructions to swarms, and (ii) a *swarm coordinated multi-modal feedback language* that enables robot swarms to robustly convey swarm-level decisions, status, and intentions to humans using multiple individual and group modalities. The gesture language allows humans to: select and

address single and multiple robots from a swarm, provide commands to perform tasks, specify spatial directions and application-specific parameters, and build iconic grammar-based sentences by combining individual gesture commands. Swarms convey different types of multi-modal feedback to humans using on-board lights, sounds, and locally coordinated robot movements. The swarm-to-human feedback: conveys to humans the swarm's understanding of the recognized commands, allows swarms to assess their decisions (i.e., to correct mistakes: made by humans in providing instructions, and errors made by swarms in recognizing commands), and guides humans through the interaction process.

The **second contribution** of this research addresses swarm-level sensing and classification: *How can robot swarms collectively sense and recognize hand gestures given as visual signals by humans?* Distributed sensing, cooperative recognition, and decision-making mechanisms have been developed to allow robot swarms to collectively recognize visual instructions and commands given by humans in the form of gestures. These mechanisms rely on decentralized data fusion strategies and multi-hop messaging passing algorithms to robustly build swarm-level consensus decisions. Measures have been introduced in the *cooperative recognition protocol* which provide a trade-off between the accuracy of swarm-level consensus decisions and the time taken to build swarm decisions.

The **third contribution** of this research addresses swarm-level cooperation: *How can humans select spatially distributed robots from a swarm and the robots understand that they have been selected? How can robot swarms be spatially deployed for proximal interaction with humans?* With the introduction of spatially-addressed instructions (pointing gestures) humans can robustly address and select spatially-situated individuals and groups of robots from a swarm. A cascaded classification scheme is adopted in which, first the robot swarm identifies the selection command (e.g., individual or group selection), and then the robots coordinate with each other to identify if they have been selected. To obtain better views of gestures issued by humans, distributed mobility strategies have been introduced for the coordinated deployment of heterogeneous robot swarms (i.e., ground and flying robots) and to reshape the spatial distribution of swarms.

The **fourth contribution** of this research addresses the notion of collective learning in robot swarms. The questions that are answered include: *How can robot swarms learn about the hand gestures given by human operators? How can humans be included in the loop of swarm learning? How can robot swarms cooperatively learn as a team?* Online incremental learning algorithms have been developed which allow robot swarms to learn individual gestures and grammar-based gesture sentences supervised by human instructors in real-time. Humans provide different types of feedback (i.e., full or partial feedback) to swarms for

improving swarm-level learning. To speed up the learning rate of robot swarms, cooperative learning strategies have been introduced which enable individual robots in a swarm to intelligently select locally sensed information and share (exchange) selected information with other robots in the swarm.

The **final contribution** is a *systemic* one, it aims on building a complete HSI system towards potential use in real-world applications, by integrating the algorithms, techniques, mechanisms, and strategies discussed in the contributions above. The effectiveness of the global HSI system is demonstrated in the context of a number of interactive scenarios using emulation tests (i.e., performing simulations using gesture images acquired by a heterogeneous robotic swarm) and by performing experiments with real robots using both ground and flying robots.

Acknowledgements

First and foremost, I wish to thank God for giving me the strength and courage to complete my PhD research and this dissertation. During my PhD research, there are many people at the Dalle Molle Institute for Artificial Intelligence (IDSIA) and at the Universitá della Svizzera Italiana (USI), to whom I am indebted to.

To begin with, I express my sincere gratitude towards my PhD co-advisor, Gianni Di Caro, who advised, motivated and supported my ideas right from the beginning of my PhD research at IDSIA. It is highly unlikely that I would have completed my research without his encouragement, patience and continuous support. I would also like to thank Luca Gambardella, my PhD advisor, for giving me the opportunity to pursue my PhD at IDSIA, working on the NCCR Robotics project funded by the Swiss National Science Foundation (SNSF).

When I joined IDSIA in 2011, I had no experience on working with robotics. However, after five years of hard work, and working sometimes until the early hours of the morning to meet conference submission deadlines, I am proud to say that I have achieved my goal. It was an immense pleasure working at IDSIA, and I thank all my colleagues for providing a friendly and competitive environment. I am grateful to Alessandro Giusti, Hung Ngo, Jürgen Leitner, Eduardo Feo Flushing, Frederick Ducatelle, Dan Cireşan, Ueli Meier, Michal Kudelski and Antoine Bautin for guiding and motivating me through out my PhD. It was a pleasure collaborating with them on several research topics. I also thank Jérôme Guzzi and Jacopo Banfi for assisting me with robotic issues. I had great fun in discussing and debating several technical and non-technical topics with all idsiani.

I would especially like to thank Cinzia Daldini, our secretary at IDSIA, and Elisa Larghi, the Dean's secretary at USI, for making my life easier by helping me with all necessary procedures till the very end of my PhD and stay in Lugano. I also wish to show my indebtedness to my dissertation committee, Richard Vaughan, Carles Sierra, Alcherio Martinoli, Fabio Crestani and Cesare Pautasso, for reviewing my thesis and providing useful suggestions. Finally, I thank my parents and my sister for their loving support, who inspired me to achieve the goals that I had kept during my PhD research.

Contents

Chapter 1

Scientific Context, Goals, and Contributions

This chapter serves as the introduction of this dissertation, it presents the scientific goals, objectives, and contributions of this research, and discusses the fundamental research problems in the context of *human-swarm interaction* (HSI).

1.1 Introduction

1.1.1 Robot Swarms for Real-world Applications?

Swarm robotics is a relatively new research area that started developing in the late 1990s. It finds its roots in the field of *swarm intelligence* [Bonabeau et al., 1999], that focuses on the mechanisms employed by animals and social insects for realizing collective behaviours. At the same time, it also grew as a generalization and extension to large multi-robot systems of the behaviour-based approach in robotics [Arkin, 1998]. The field of swarm robotics studies how a large number of autonomous cooperative robots, collectively referred to a *robot swarm* (or in short, swarm), locally interact and communicate with each other and with the environment to produce self-organized coordinated behaviours [Beni, 2005; Sahin, 2005] that can go far beyond the capabilities of single-robot systems, potentially showing super-linear speed-ups.

The key advantages of swarm robotics relies on the capability of producing emerging swarm-level behaviours [Dorigo et al., 2004]. Due to their intrinsic redundancy, robotic swarms naturally provide a diversity of parallel and distributed functionalities, high robustness and flexibility, spatial distribution, fault-tolerance, and adaptivity. Prototypical examples of decentralized and self-

organized systems from nature in which *swarm behaviour* occurs, includes the collective behaviour of animals [Camazine, 2003] such as, the honey bee's nest-building, trail following of ants [Dorigo et al., 1996], construction of the ant and termite mound, bird flocking, and fish schooling. As individuals agents in a natural swarm usually do not need sophisticated knowledge to produce such complex swarm behaviours [Martinoli and Easton, 2003; Martinoli et al., 2004; Ducatelle et al., 2011], this results in a system that is mostly based on relatively simple local behaviours and interactions.

Similarly, in present day robotics technology, a robot swarm is usually composed of a potentially large number of relatively simple and unsophisticated mobile robots. Swarm robots offer low-quality sensing devices, basic locomotion capabilities and limited on-board computation and communication resources as compared to single-robot systems, which are expected to be more sophisticated than an individual swarm robot [Navarro and Matía, 2012; Brambilla et al., 2013; Kumar et al., 2013]. In principle, individual robots in a swarm do not need to be extremely sophisticated, as their coordination, cooperation and synergistic interaction aims to produce a system (swarm) with enhanced abilities and skills.

The evolution of swarm robotics has reached a stage where *miniature robot swarms* lie on one side of the spectrum of distributed robotics, and the other side of the spectrum represents *multi-robot systems*, as illustrated in Figure 1.1. Miniature robot swarms are aimed at achieving lower manufacturing costs with a limited set of features, by leveraging swarm size over the complexity of the system. Commercial miniature swarms include robots such as the, *e-puck*[1] and *Alice* robots, *Jasmine* robots[2] and the iRobot *SwarmBots* [McLurkin and Smith, 2004]. The *Kilobot*[3] is currently the smallest and cheapest swarm robot available, with a retail price of $14 USD. Miniature swarms can have up to 1000 robots, as demonstrated by the Kilobot swarm. As the small size of miniature robot swarms limits the possibility of a flexible design [Navarro and Matía, 2012], fewer features limit the possibility of performing real-world applications.

As miniature robot swarms are potentially unreliable and provide minimalistic performance, in this research the focus is mainly on *mid-range robot swarms*, which are considered the middle ground between multi-robot systems and miniature swarms[4], as illustrated in Figure 1.1. Mid-range robot swarms are seen as more suitable candidates for performing real-world tasks, and feature more complex and sophisticated robots, which are bigger in size and cost higher to produce,

[1] http://www.e-puck.org/
[2] http://www.swarmrobot.org/
[3] https://www.eecs.harvard.edu/ssr/projects/progSA/kilobot.html
[4] Refer to [Tan and Zheng, 2013] for an overview of swarm robotics and multi-robot systems

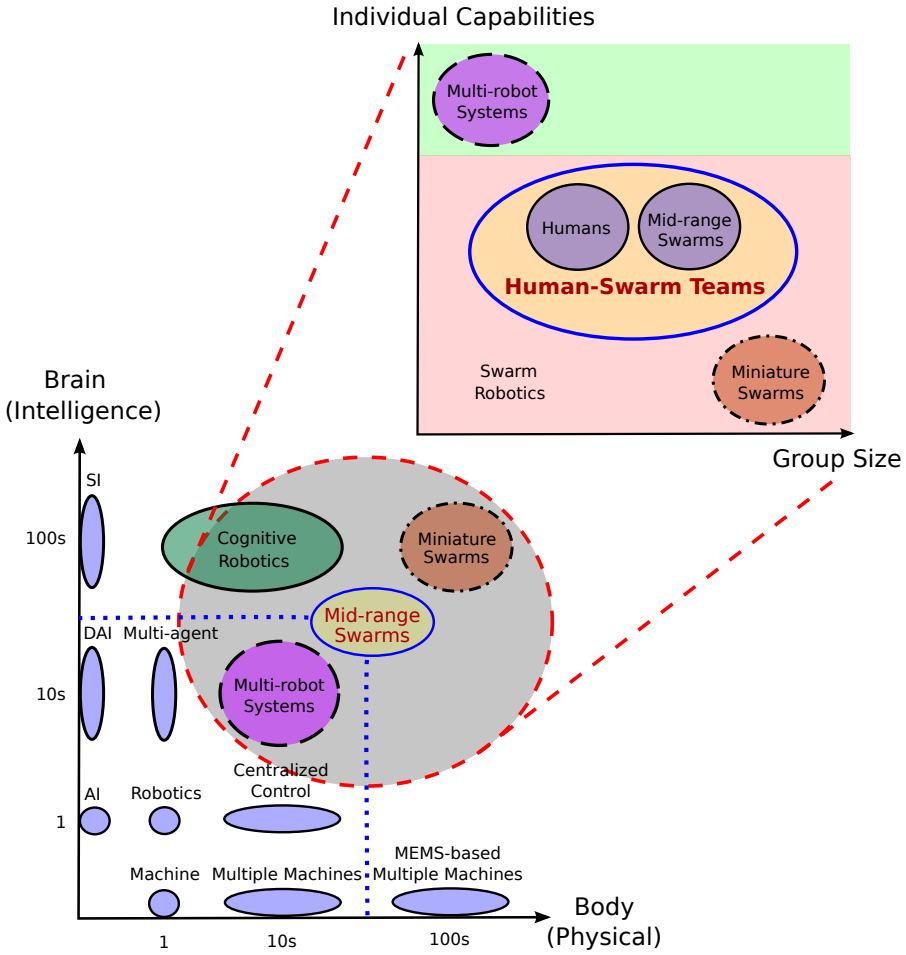


Figure 1.1. The evolution of swarm robotics. *Mid-range robot swarms* are the middle ground between *multi-robot systems* and *miniature robot swarms*.

but still a relatively high number of them can be employed in a single system. Being more capable task solvers, mid-range swarms involve a wider spectrum of features (i.e., sensory-motor skills) for the solution of simpler to advanced tasks. Mid-range robot swarms comprise of a reasonable number of minimum robots (e.g., 5 robots) that can range up to 50 robots or more, and can employ typical swarm intelligence approaches for coordination and cooperation.

Having significant sensing and actuation capabilities, mid-range swarms are considered suitable candidates to support direct interaction with humans and perform real-world tasks in cooperation with humans, which is the central aspect of this research. By including humans in the interaction loop of mid-range swarms, *human-swarm teams* can be formed, as illustrated in Figure 1.1. Robot

swarms offer a clear advantage to interact with humans as peers, due to the combined and enhanced capabilities of a large number of autonomous robots acting in cooperative synergy. This research focuses on developing techniques and strategies for HSI, by integrating the heterogeneous sensory-motor skills and capabilities available from both the human's side and from the swarm's side.

In this research, the main focus is on the application of human-swarm teams for *search and rescue* (SAR) missions, which is the guiding application. As robot swarms have the ability to cover large areas, rescue workers can interact with heterogeneous robotic swarms, and dispatch swarms to visually assess and detect the presence survivors in places where rescue personnel cannot safely reach: burning and collapsed buildings, disaster zones (e.g., fires, avalanches, earthquakes), mountainous terrains and cluttered territory. By coordinating and cooperating with rescue workers using multiple interaction modalities, robots can inform the status and location of survivors once they have been found.

1.1.2 Humans in-the-loop of Robot Swarms

The inclusion of humans in the interaction loop of swarms is a relatively unexplored area. In practice, humans can be included in the interaction loop in many different ways. For instance, humans can script the task, can define the criteria to evaluate the performance, can perceive the environment and the task using a sensory system complementary to the one used by the swarm, and can reason on the task progress using human-specific heuristics and a priori knowledge.

Existing works have explored a wide spectrum of possibilities: humans with a fully supervisory and controlling role [Cummings, 2004; Chen et al., 2011; Kolling et al., 2012], humans treated as a resource to robots [Fong et al., 2003, 2005, 2006], and the case of mixed initiative teams [Hearst, 1999; Bruemmer et al., 2005], in which robots are autonomous and the levels of autonomy and tasking are dynamically assigned on the basis of peer-to-peer teaming with humans [Loper et al., 2009]. *Proximal interaction* [Kolling et al., 2016] with humans was envisioned beneficial for firefighters in SAR scenarios [Naghsh et al., 2008], and in [Couture-Beil et al., 2010a] human operators were able select and command robots. In [Alboul et al., 2008] humans interacted with swarms as a special swarm member that acted as an attractor.[5]

In the context of proximal interactions, a first approach based on the use of *instrumented methods* (i.e., sophisticated devices for supporting interaction) was investigated by Payton et al. [Payton et al., 2001], which demonstrated that hu-

[5]Refer to [Kolling et al., 2016] for a survey of human interaction with robot swarms.

mans are able interact with small-scale robots referred to as *pherobots*, using a handheld remote. The handheld remote consisted of a PalmPilot with an Infrared (IR) emitter that transmitted highly directional and omni-directional wireless signals to the pherobots. Using this instrumented approach, human operators were able to issue commands to individual robots and to the entire swarm, using a narrow (directional) and wide (omni-directional) IR beam respectively. Payton et al. also demonstrated the concept of a world-embedded distributed display [Payton et al., 2003] (where each robot represented a pixel or an annotation on the immediate environment), in which the user interface to interact with the pherobots was adopted as a see-through augmented reality head-mounted display (HMD) worn by human operators. IR signals emitted by the pherobots were received by the HMD, and this information was displayed as a graphical overlay on the human's field of view. Virtual pheromones introduced by Payton et al. signified the use of simple communication and emergent coordinated movement with minimal on-board processing for real-world applications [Payton et al., 2005].

The initial works of Payton et al. [Payton et al., 2001, 2003, 2005] stressed that, without the use of instrumented methods (e.g., handheld devices) [Grieder et al., 2014] and sophisticated interaction (supporting) gadgets, humans might face difficulties in proximally interacting with multiple robots. With the improvements in the sensing devices of mobile robots, the research team at the Autonomy Lab[6] of Vaughan has demonstrated the successful use of *uninstrumented methods* for proximal interaction between humans and multi-robot systems [Pourmehr et al., 2015; Monajjemi et al., 2013]. Uninstrumented methods do not use sophisticated supporting devices from the human's side, instead they rely on more natural and intuitive means of interaction (e.g., robots sense audio/visual signals given by human operators). The Kinect is a good example of a *natural user interface* (NUI) that uses human body movements for interaction.

The Autonomy Lab introduced the use of NUIs [Couture-Beil et al., 2010b; Milligan et al., 2011; Pourmehr et al., 2013b] for selecting and commanding robots from a distributed multi-robot system using different interaction modalities. These modalities include facial engagement and hand gestures (see Section 4.1.1 for more details). Multi-modal interactions have also been investigated [Xavier and Nunes, 2007], in which speech has been used in conjunction with gaze and gestures [Pourmehr et al., 2013a, 2014; Monajjemi et al., 2014].

With the aim of building a HSI system that allows robot swarms to proximally interact and cooperate with humans, and motivated by the use of NUIs, the research reported in this dissertation considers the use of *uninstrumented and non-*

[6] http://autonomylab.org/

verbal communication methods (see Figure 1.2), in which passive markers (i.e., inexpensive coloured gloves worn without any instrumentation) are adopted by human operators. Even if it might appear limited, non-verbal communication consists of wordless (i.e., visual and body language) cues communicated between people [Argyle et al., 1970]. Non-verbal communication represents approximately two-thirds of all human-to-human communication processes, and has the ability to portray a message or instruction with the correct body language and signals [Matsumoto, 2006]. Body language expresses thoughts, intentions, feelings, conscious and unconscious signals, and the mediation of personal space, using physical actions and behaviours [Arygle, 1988], such as: facial expressions, hand gestures, body postures and movements, eye contact (gaze) and touch.

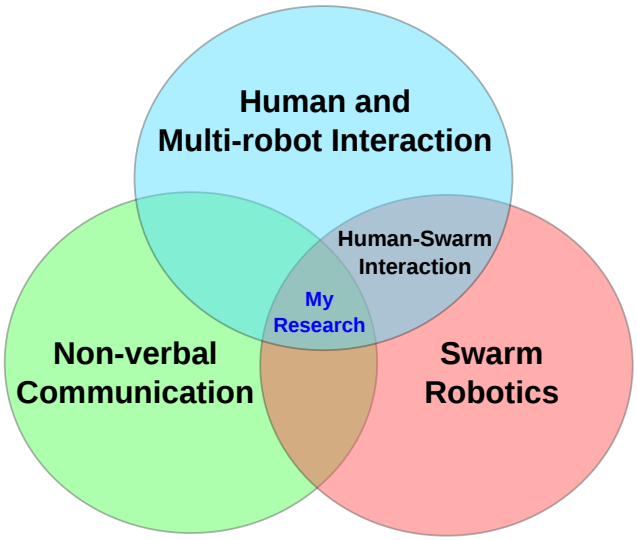


Figure 1.2. Classification of research area.

As a variety of body language cues and behaviours exist that humans can use to interact with multiple robots, we select *hand gestures* as the modality of choice, due to the fact that gestures serve as non-verbal communication symbols, and have the ability to encode human instructions and commands (see Section 1.3).

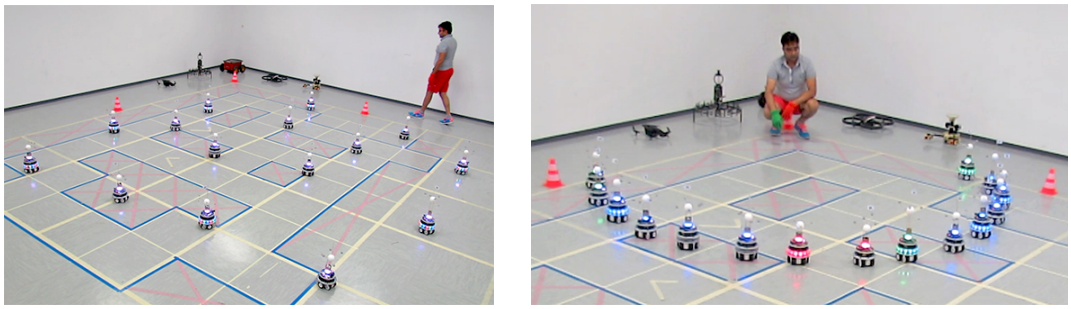
1.2 Considered Human-swarm Interaction Scenario

As highlighted in the previous sections, this research aims at studying interaction and cooperation between humans and mid-range robot swarms, provided that swarms are located within physical proximity (proximal range) of sensing audio/visual signals given by human operators. Humans proximally interact with

robot swarms using hand gestures that are transmitted to robot swarms as visual signals, and sensed by cameras on-board the robots. The guiding application of this scenario are SAR missions. The reference HSI scenario[7] which motivates this research is composed of 6 stages:

1. A human operator enters a dynamic environment (e.g., a room) in which a swarm of robots is present, as shown in Figure 1.3(a). As the human moves within physical proximity of the robots, the human gains the attention of the swarm (e.g., using visual means such as, the waving of arms and hands, through sounds such as a hand clap, a whistle blow).

2. After the human has been detected by the swarm, the robots in the swarm coordinate together to autonomously deploy themselves in positions (locations) which offer a more reliable and clearer view of the human issuing the gestures. Deployment results in the robot swarm physically surrounding the human, as given by the swarm formation in Figure 1.3(b). The deployment stage is useful, however not compulsory.



(a) (b)

Figure 1.3. Illustration of the HSI scenario considered in this research using a swarm of $N = 15$ robots. (a): A human operator entering a room where a swarm of robots are randomly located. (b) A human interacting with a robot swarm.

3. Then, the interaction process initiates with the robot swarm requesting the human to provide a command (i.e., a gesture). The human interacts with the swarm by providing a gesture which encodes a possible mission command (e.g., a command to instruct the swarm to search in a specific direction).

4. Next, the robot swarm cooperatively senses and recognizes (classifies) the given gesture from a set of predefined gestures already learned by the swarm.

[7] A video demonstration illustrating the reference HSI scenario is available at: http://www.jnagi.net/interaction_scenario

5. After the swarm's recognition is complete, the swarm-level decision is communicated as feedback to the human using actuation devices on-board the robots (e.g., coloured lights, speakers). This swarm-level feedback informs the human, if the swarm recognized the command properly or not, and guides the human during the interaction process. When a gesture command is not properly recognized (i.e., due to mistakes made by the human in presenting the gesture or due to recognition errors made by the swarm), the swarm requests the human to present the same gesture again, so that the gesture can be properly recognized the next time.

6. After the given gesture has been properly classified and interpreted by the swarm, the swarm performs the task associated with the recognized gesture.

1.2.1 The Used Robot Platforms

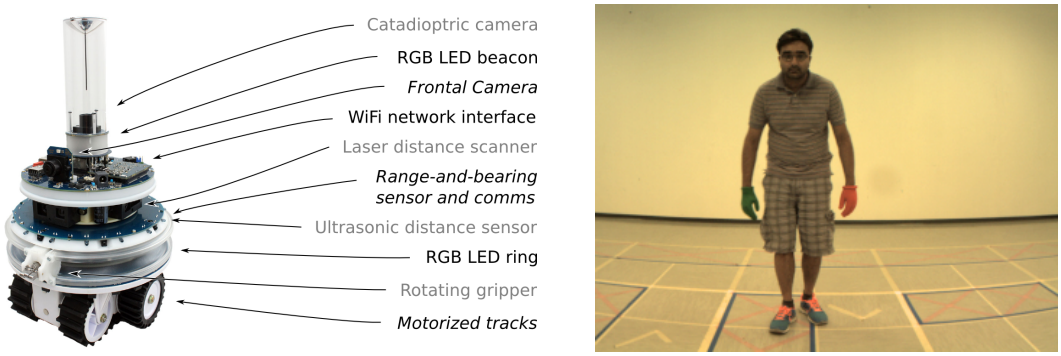
To assemble heterogeneous swarms of robots, we consider the use of two robot platforms, namely the *Foot-bot* robots, that are small ground robots, and the *Parrots*, which are flying robots (drones). In practice, up to 16 Foot-bots and 4 Parrots have been experimentally tested. These two platforms have different capabilities which allows us to investigate a wide range of difference scenarios. The capabilities of both platforms are presented in the next sections.

1.2.1.1 Foot-bot Robots (UGVs)

The Foot-bot is a small unmanned ground vehicle (UGV) that has been derived from the marXbot platform [Bonani et al., 2010], as shown in Figure 1.4(a). The Foot-bot robots were developed within the *Swarmanoid* project [Dorigo et al., 2006, 2013]. Having limited computational power with an on-board ARM 11 processor operating at 533MHz and 128MB of RAM, the Foot-bots are controlled using the multi-robot simulator, ARGoS [Pinciroli et al., 2012] (see Section 6.1), in a Linux-based environment. All computation (i.e., image processing and gesture classification using the frontal camera) is done on-board the Foot-bots.

Foot-bots are equipped with a variety on-board sensing and actuation devices, as shown in Figure 1.4(a): a *frontal camera* that acquires VGA images in a resolution of 640 × 480 pixels (see Figure 1.4(b)), motorized track-based wheels that offer speeds up to 0.3 m/s, a 802.11 wireless (Wi-Fi) network interface, a radio frequency (RF) and infrared (IR) based *range-and-bearing* (RAB) system that allows robots to detect and relatively localize *line-of-sight* neighbouring robots

up to a range of a 5 meters through low-bandwidth wireless communication, a circular ring of 12 RGB coloured LEDs, and a RGB colour beacon.



(a) (b)

Figure 1.4. (a): The Foot-bot robot platform, a small unmanned ground vehicle (UGV) with on-board computational capabilities and a variety of sensing and actuation devices. (b): A RGB image with QVGA display resolution (dimension of 320 × 240 pixels) acquired by the frontal camera of the Foot-bot.

As the RAB system provides structured line-of-sight communication signals with the ability to estimate the angular distance (θ, d) from neighbouring Foot-bots, the RAB system has been adopted for autonomous and coordinated deployment of UGV swarms (see Section 4.3.1).

1.2.1.2 Parrot Drones (UAVs)

The unmanned aerial vehicle (UAV) that has been adopted, is the standard Parrot AR.Drone 2.0 quadcopter, as illustrated in Figure 1.5(a). The Parrots are equipped with: a *frontal camera* that acquires images in a high-definition (HD) resolution of 1280 × 720 pixels using a 92° diagonal wide angle lens (as shown in Figure 1.5(b)), a Wi-Fi network interface for communication, a vertical QVGA camera (with an image resolution of 320 × 240 pixels) for measuring ground-speed, pressure sensors, and ultrasound sensors for altitude measurement.

Equipped with a 1GHz ARM Cortex A8 processor to perform flight operations, the Parrots are also provided with a 800MHz video processor and 1GB RAM for acquiring images. The out-of-the-box Parrots cannot perform any on-board computation (i.e., cannot process or recognize images acquired from the frontal camera). All images acquired by the Parrots are streamed onto a computer (using Wi-Fi), which performs all the necessary processing (see Section 6.1).

(a) (b)

Figure 1.5. (a): A Parrot AR.Drone 2.0 with an indoor hull (top) and its cross-sectional view (bottom). (b): An image with a resolution of 640 × 360 pixels acquired by the frontal camera of the Parrot using a wide angle lens.

The Parrots are controlled in a Linux-based environment using the Robot Operating System (ROS) [Quigley et al., 2009]. In the case of multiple Parrots, each robot is associated to a Linux process that communicates with other processes for information exchange and collective decision-making (see Section 6.1). The frontal camera of the Parrots has been adopted for coordinated deployment and human-relative localization of UAV swarms (see Section 4.3.2).

1.3 Core Challenges in Human-swarm Interaction

The majority of existing solutions that investigate interactions between humans and multiple robots are mostly problem-specific, can hardly scale effectively in the case of large robot swarms both in terms of communication overhead and human workload, and mainly rely upon the use sophisticated interaction devices. As existing approaches are not suitable enough to face such challenges and profit from the advantages of swarms, the multiple dynamic aspects and the lack of precise information in the problem definition of HSI leave space to explore several core issues. With respect to the HSI scenario considered in this research (see Section 1.2), below we identify a number of fundamental research issues and present solutions to tackle these challenges systematically.

In Section 1.4 that follows, we set a number of goals out of these core HSI issues and challenges, and we provide an outline of how these goals have been achieved in this doctoral research.

Proximal Interaction between Humans and Robot Swarms
How can human operators communicate and provide instructions to robot swarms?
Which mechanisms can be used by a swarm of robots to efficiently convey swarm-level feedback to humans?
Which kinds of feedback can swarms convey to humans?

To address these fundamental HSI challenges, a *bidirectional human-swarm communication system* needs to be developed for proximal interaction [Kolling et al., 2016], which allows humans to efficiently provide mission instructions and commands to swarms, and enables swarms to easily convey multi-modal feedback to humans with a high and immediate impact.

Intelligent HSI System with a Human-friendly User Interface
What capabilities and functionalities does the user interface need to have so that human operators can easily interact with robot swarms?
How can the HSI system be made robust so that errors and mistakes made during the interaction can be identified and minimized?

Although some potential works have examined how proximal interaction between humans and multiple robots takes place, little is known on how to best design robust and effective user interfaces for interacting with swarms. We consider that the HSI system should have some important characteristics: to be able to *identify when humans are issuing commands* and basic *reasoning capabilities*.

Collective Sensing and Fusion of Information Acquired by Robot Swarms
How can robot swarms collectively sense and recognize hand gestures given as visual signals by human operators?
Which strategies can be used by robot swarms to efficiently build swarm-level decisions for the recognition of gesture commands?

Robot swarms can function as *distributed sensing* systems to gather (acquire) information from gesture commands. *Cooperative recognition* strategies can allow individual robots in a swarm to effectively fuse information sensed from gestures and build *consensus decisions* for the swarm-level classification of gestures.

Coordination for Spatial Selection of Robots and Deployment of Swarms
How can humans address and select spatially-situated robots from a swarm and how do the robots understand that they have been selected?
How can robots in a swarm efficiently coordinate and spatially deploy themselves for proximal interaction with humans?

The research team at the Autonomy Lab of Vaughan investigated the use of audio and visual interaction modalities for selecting robots from a multi-robot system (see Section 4.1.1). We consider that, *spatially-addressed gestures* can efficiently address (select) spatially distributed robots from a swarm, and individual robots can coordinate and recognize spatial gestures to understand if they have been selected. Deployment techniques should focus on *improving the sensing coverage* of robot swarms (i.e., to sense better quality of information from gesture commands) while *maintaining wireless connectivity* with the swarm network.

Including Humans in-the-loop of Swarm-level Learning
How can swarms learn about the gesture commands given by human operators?
How can humans provide gesture commands for swarm-level learning?

In real-world environments human instructions (gesture commands) arrive in increments (i.e., over the passage of time through interaction with humans). *Online incremental learning* strategies can provide robot swarms the ability to remember previously issued gesture commands (i.e., previous interaction experiences) as well as newly issued gestures. Distributed online learning strategies are required so that robot swarms can efficiently learn gesture commands from humans instructors and teachers in real-time (i.e., learning immediately).

Sharing and Exchanging Learning Information between Individual Robots
What strategies need to be in place so that individual robots in a swarm can collectively learn gesture commands as a team?
How can robot swarms learn gesture commands in bandwidth-limited scenarios?

If individual robots in a swarm learn only their locally sensed information, the presence of a swarm is only exploited in terms of distributed sensing and cooperative recognition. For individual robots to collectively learn gestures in cooperation with other robots in the swarm, *cooperative learning* strategies (see Section 5.4) can allow individual robots to share and exchange learning information with other robots in the swarm. Under communication constraints, mechanisms need to be developed which can select the most important (e.g., novel or representative) learning information available within the swarm network.

The core challenges and issues reported above are common to all multi-robot systems, and are amplified in the case of swarm robotic systems, especially when the scalability of solutions, the limited instantaneous connectivity, and learning are explicitly taken into account.

1.4 Scientific Goals and Contributions

This section presents the main objectives and goals of this dissertation together with the scientific contributions and improvements made to the state-of-the-art. In general, the focus of this research is to develop fully distributed techniques that enable peer-to-peer symbiotic interaction and cooperation between humans and autonomous robotic swarms, and provide highly adaptive solutions to overcome the core challenges presented in Section 1.3. With reference to the considered HSI scenario (see Section 1.2), this research is grounded to real-world applications, which is of both theoretical and practical interest.

1.4.1 Main Goal: Bidirectional Human-swarm Communication

The primary objective of this research is focused on developing a *bidirectional human-swarm communication system* for proximal interaction between humans and heterogeneous robot swarms based on *uninstrumented and non-verbal communication* methods. The purpose of this research is not to develop smart vision-based algorithms for gesture classification, and this is why we consider the use of coloured passive markers (i.e., inexpensive coloured gloves), which robot swarms can detect and recognize. In principle, it is not necessary to use gloves, if sophisticated vision-based gesture recognition techniques are adopted. To develop a bidirectional interaction system that allows human-to-swarm and swarm-to-human communication, existing works in human-robot interaction (HRI) have suggested that, conversations between humans and robots can be efficiently realized using structured dialogues [Chambers et al., 2005; McLurkin et al., 2006] (see Section 2.1.1.2). We explore this opportunity and investigate the use of human-swarm conversational dialogues.

In this research, a *dialogue-based interaction system* is introduced, which consists of: (i) a *grammar-based gesture language* with a vocabulary of non-verbal commands, using which humans can efficiently provide mission instructions to robot swarms, and (ii) a *swarm coordinated multi-modal feedback language*, using which swarms can convey swarm-level status, decisions and intentions to humans. As briefly highlighted in Section 1.1.2, for humans to interact and communicate with swarms, *hand gestures* are selected as the modality of choice, because once compared with other non-verbal cues (i.e., facial expressions, gaze and body postures), gestures have the advantage of being easy to use [Alonso-Mora et al., 2015], they are easily understood and recognizable [Hu et al., 2012], and they can: encode complex human instructions (i.e., represent a variety of tasks that robots can perform) as grammar-based sentences, spatially indicate

directions for robots to move to a location [Abidi et al., 2013], and represent quantities (e.g., numbers), all of which are necessary in HSI (see Section 2.2.1).

For human-to-swarm communication, the grammar-based gesture language provides a set of basic operational commands, which humans can present to robot swarms during SAR missions. For swarm-to-human communication, the swarm-coordinated multi-modal feedback language is based on the use of multiple modalities (i.e., lights, sounds and coordinated movements) in order to: convey to humans the swarm's understanding of recognized gesture commands, detect mistakes made by humans in providing gestures and errors made by swarms in recognizing gestures, and guide humans through the interaction process.

To develop an intelligent HSI system with a human-friendly user interface, we consider the following functionalities and characteristics. Firstly, human instructions that are given using gestures, need be engineered keeping into mind the *psychological and physiological properties* of gestures, such as, naturalness, intuitiveness, comfortableness, recognisability, spatial characteristics, and ease of use [Stern et al., 2006, 2008b,a; Wachs et al., 2008; Stern et al., 2009; Mai et al., 2011; Pfeil et al., 2013]. Secondly, robot swarms should be able to identify when humans are presenting gesture commands, and when random movements (manipulations) of the hands and arms occur–that are not human instructions. This can be achieved by implementing a *human body motion detection system*.

Lastly, it is desirable that swarms possess *basic reasoning and self-assessment capabilities*. This means that, a swarm should be able to intelligently assess without the use of human feedback (external inputs), if it is confident or not in recognizing given human instructions (gesture commands). Self-assessment made by robot swarms should be able to: identify sensing ambiguities, detect errors and mistakes from the human's side and from the swarm's side, and request humans for corrections. The techniques and strategies discussed above are presented in Chapter 2 and have been published in [Nagi et al., 2014b,c].

This set of main goals builds upon the state-of-the-art methods for dialogue-based interaction with multiple robots [Skubic et al., 2004; Chambers et al., 2005; Harris et al., 2005], multi-robot command and control languages [Stoica et al., 2014, 2013], and HSI strategies that adopt the use of gestures [Xavier and Nunes, 2007; Podevijn et al., 2013; Alonso-Mora et al., 2015].

1.4.2 Sub-goals: Functional to Achieve the Main Goal

The four sub-goals presented below provide the secondary contributions of this research, in-line with the main goal. They address the core challenges outlined in Section 1.3 and are functional to achieve the main goal.

1.4.2.1 Distributed Sensing and Cooperative Recognition

This sub-goal aims on developing a distributed sensing and recognition mechanism for *cooperative decision-making* in robot swarms. To achieve this goal, we design a general protocol that enables the swarm-level classification (recognition) of commands defined in gesture language (vocabulary).

To overcome the limitations of individual robots and allow robots in a swarm to utilize their local (individual) sensing capabilities, following the works of Payton and Vaughan (see Section 1.1.2), we find it useful to consider robot swarms as *distributed sensing systems*, that can concurrently gather perceptual visual information (using on-board cameras) from gesture commands while located at different viewpoints. Distributed and parallel sensing mechanisms facilitate the cooperative swarm-level recognition of gestures.

Cooperative recognition is considered as a global swarm task, which can be decomposed into distributed and parallel inter-dependent tasks or sub-problems (that naturally account for distributed and coordinated solutions). These sub-problems can be adaptively solved by every individual robot in the swarm, since every robot has only partial knowledge (i.e., partially viewable information) of the global task. Distributed and parallel information sensed from gestures by multiple robots, can be combined using decentralized data fusion methods that rely on multi-hop messaging passing algorithms. We consider the use of *distributed consensus* mechanisms for building swarm-level decisions (i.e., unified mutual agreements between individual robots in the swarm).

The developed protocol consists of an integrated strategy: a mechanism that recognizes passive markers (i.e., coloured gloves) worn by humans, and a consensus building mechanism for data fusion and multi-hop information propagation. To provide performance control, a "prudence" parameter is introduced in the protocol, using which the accuracy of swarm-level decisions and the time required to build consensus decisions can be specified. This protocol is presented in Chapter 3 and the associated results have been published in [Nagi et al., 2011; Giusti et al., 2012a,b,c; Nagi et al., 2014c, 2015].

This sub-goal provides improvements over the state-of-the-art cooperative distributed vision methods that rely on data fusion mechanisms [Yu and Nagpal, 2009; Jorstad et al., 2010; Kokiopoulou and Frossard, 2010, 2011].

1.4.2.2 Swarm Understanding of Spatial Robot Selection and Deployment

This sub-goal focuses on the development of swarm-level coordination mechanisms for dealing with spatially related issues. We provide techniques and strate-

gies: (i) that enable human operators to efficiently select robots from a swarm, and allow robots in a swarm to understand if they have been selected, and (ii) using which spatially-situated robots in a swarm coordinate and autonomously deploy for proximal interaction with humans.

To select individuals and groups of robots from a swarm, we consider the use of spatially-addressed instructions, namely *pointing gestures*. As human operators may want to select an individual robot or a group (subset) of robots from a swarm, this can be achieved by pointing at the robot(s) that need to be selected. To select individual robots, an index finger can be used to point to the desired robot, while to select a group of robots, two pointing hands can define the range of the group of robots to be selected. The difficult problem is for robots to understand if they have been selected. This is because, many robots may see the given gesture from different viewpoints at the same time, and every individual robot in the swarm has to coordinate with the other robots to reliably decide based on its viewpoint and local gesture observation, if it has been selected or not. Spatial robot selection is performed in a two-stage process. First, robots in a swarm identify the type of selection command (e.g., individual or group selection), and secondly, individual robots solely decide if they have been selected.

As individual robots in a swarm may be positioned in locations that do not offer a good (or clear) view of the issued gesture commands (e.g., due to partial occlusions), individual robots need to coordinate with each other to move to sensing positions that offer better views of gestures. We have considered *distributed mobility strategies* based on mobility rules, using which heterogeneous teams of robot swarms (UGVs and UAVs) can reshape their spatial distribution, and this offers an informed way of human-relative localization. The above mentioned swarm-level coordination mechanisms are presented in Chapter 4 and have been published in [Giusti et al., 2012c; Nagi et al., 2014a,c].

This sub-goal contributes to the state-of-the-art robot selection strategies for *human and multi-robot interaction* (HMRI) and HSI [Monajjemi et al., 2013; Lichtenstern et al., 2013; Pourmehr et al., 2013a; Milligan et al., 2011; Couture-Beil et al., 2010a], and multi-robot deployment techniques [Monajjemi et al., 2013; Guinaldo et al., 2013; Saska et al., 2014; Duan et al., 2014].

1.4.2.3 Cooperative Learning Supervised by Humans

This sub-goal aims on developing *distributed online learning* strategies that allow robot swarms to learn the grammar-based gesture language (vocabulary of commands) supervised by human instructors. In an online learning setting, humans can use their unique sensory-motor capabilities to act as teachers for supervising

the swarm's learning of gesture commands. For instance, after a gesture is presented by a human and it is recognized by the swarm, based on the swarm-level recognition outcome, the swarm conveys multi-modal feedback to the human. After the human has interpreted the swarm's feedback, the human can provide different types of feedback (e.g., full or partial feedback) to the swarm for improving the swarm-level learning and understanding of gesture commands.

As information sharing in robot swarms plays an important role in building synergistic cooperative strategies, we have considered developing *information sharing and selection mechanisms* that can allow robot swarms to collectively and collaboratively learn gesture commands in bandwidth-limited scenarios. To allow *cooperative learning* in robot swarms, information selection strategies need to be based on intelligent criteria, and should rely on principles of sharing and forgetting learning (training) information. The swarm-level learning strategies are presented in Chapter 5 and have been published in [Nagi et al., 2011, 2012b,a; Di Caro et al., 2013b; Ngo et al., 2014; Nagi et al., 2014e,d].

This sub-goal improves the state-of-the-art algorithms for, learning with human feedback [Kakade et al., 2008; Chen et al., 2009; Wang et al., 2010; Crammer and Gentile, 2011], collaborative learning [Predd et al., 2005, 2006a,b, 2009], and the online selection of training information [Zechner and Granitzer, 2009; Lopes et al., 2010; Chen et al., 2013].

1.4.2.4 Building a Complete HSI System

The final goal of this research aims on building a fully functioning HSI system for potential use in real-world interaction applications. This involves a *systemic integration* of all the components and technologies developed in this research, that are discussed in the aforementioned goals.[8] To evaluate the performance of developed HSI system, real robot experiments are performed using heterogeneous teams of robot swarms (i.e., ground and flying robots).

Assessing the efficacy and efficiency of HRI solutions [Goodrich and Olsen, 2003; Steinfeld et al., 2006] for multi-robot systems [Crandall and Cummings, 2007b; Pourmehr et al., 2015] and robot swarms [Harriott et al., 2014; Hayes and Adams, 2014] is an open issue. Although metrics have been proposed for HRI [Steinfeld et al., 2006], proper definitions of metrics in the case of HSI do not exist. Appropriate metrics in context to the considered HSI scenario (see Section 1.2) have been defined in Chapter 6, to assess and evaluate system-level

[8] Videos demonstrating the developed HSI system are available at: http://www.jnagi.net/example_task and http://www.jnagi.net/systemic_integration

parameters: the perceptual reliability of robot swarms, the level of symbiosis and cooperation maintained within swarms, the impact of different learning and communication strategies, the scalability of solutions, and the associated levels of fault-tolerance and adaptivity.

1.5 Impact of Work

The research presented in this dissertation has been performed at Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)[9] associated with Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)[10] and Università della Svizzera Italiana (USI)[11] in Lugano, Switzerland. This research is supported by the National Centre of Competence in Research (NCCR) Robotics, with search and rescue (SAR) missions being the guiding application. The NCCR Robotics project (http://www.nccr-robotics.ch/) consists of research groups in Switzerland that have the common objective of developing new, human-oriented robotic technology. NCCR Robotics is a joint effort between four Swiss institutions: École Polytechnique Fédérale de Lausanne (EPFL), Eidgenössische Technische Hochschule Zürich (ETH Zürich), Universität Zürich (UZH), and the Dalle Molle Institute for Artificial Intelligence (IDSIA). Launched on December 2010, NCCR Robotics spans for a period up to 12 years, and is funded by the Swiss National Science Foundation (SNSF). NCCR Robotics is overlooked by an advisory board committee composed of an international panel of experts.

The first phase of NCCR Robotics which initiated in 2011 and ended in 2015, fully funded this PhD research under the sub-project "*Symbiotic cooperation between humans and robotic swarms*". This project is a follow-up of IDSIA's research in the domain of swarm robotics, a successor to the *Swarmanoid* project[12] and the *Swarm-bots* project[13]. Live demonstrations of the developed technologies have been shown at the AAMAS 2012 conference [Giusti et al., 2012b] and at the annual NCCR review meetings. Experiments have been performed with heterogeneous robotic swarms (UGVs and UAVs). A total of 15 peer-reviewed publications have resulted as a consequence of this research (see Section 7.4). In addition, a number of video demonstrations have been recorded during the course of this research (see Section 7.4.3), which verify the good performance, scalability and robustness of the developed HSI system.

[9] http://www.idsia.ch/
[10] http://www.supsi.ch/
[11] http://www.usi.ch/
[12] http://www.swarmanoid.org/
[13] http://www.swarm-bots.org/

1.6 Dissertation Organization

This dissertation comprises of seven chapters, with this being the Introduction (Chapter 1), and six further chapters follow.

Chapter 2 presents the techniques and strategies developed for bidirectional human-swarm interaction and communication. These include dialogue-based interaction techniques for human-to-swarm and swarm-to-human communication.

Chapter 3 presents a general protocol for the swarm-level classification of commands defined in the gesture language (vocabulary). The developed technologies include: strategies for decentralized data fusion, mechanisms for multi-hop message passing, and techniques for swarm-level decision making.

Chapter 4 provides swarm-level coordination solutions for spatially related issues. These include techniques and strategies for: selecting spatially-situated individuals and groups of robots from a swarm, autonomous and coordinated deployment of robot swarms, and human-relative localization.

Chapter 5 presents algorithms and techniques that allow robot swarms to learn information from humans instructors in real-time. These include: distributed online learning strategies that use human feedback, and cooperative learning mechanisms that rely on information selection and sharing strategies.

Chapter 6 evaluates the algorithms, strategies and techniques developed in Chapters 2 to 5, and presents and discusses the experimental results. Experimental validation is performed in context of the HSI scenario considered in Section 1.2, both in simulation (using emulation tests) and using real robots.

Chapter 7 presents concluding remarks, discusses the key findings, and provides future directions that need to be explored in the domain of HSI.

Chapter 2

Bidirectional Communication between Humans and Swarms

The chapter addresses the main goal of this research outlined in Section 1.4.1. More specifically, a *bidirectional communication system* for proximal interaction between humans and robot swarms is presented in this chapter. To allow robust bidirectional communication, *dialogue-based interaction* is adopted. Figure 2.1 illustrates the bidirectional flow of information between a human and a heterogeneous swarm, namely, *human-to-swarm* and *swarm-to-human* communication.

In the context of human-to-swarm communication, a grammar-based gesture language with a vocabulary of non-verbal commands is developed, using which humans can provide complex mission instructions to robot swarms. The gesture language has the ability to build semantically and syntactically correct sentences of individual gesture commands based on grammatical expressions. The vocabulary allows humans to specify a variety of gesture commands: spatially-addressed commands for selecting robots from a swarm, commands for swarms to perform application-specific tasks, spatial directions for robots to move to specific locations, and commands that encode numerical quantities.

For swarm-to-human communication a *swarm coordinated multi-modal feedback language* is developed, which provides different types of multi-modal feedback to swarms. These multi-modal feedback: convey the swarm's understanding of recognized gesture commands to humans, provide basic reasoning capabilities for swarms to assess their recognition decisions (i.e., to identify sensing ambiguities, and detect mistakes made by humans and errors made by swarms), and guide human operators through the interaction process. Swarms convey multi-modal feedback to humans using actuation systems on-board the robots (e.g., lights, sounds and locally coordinated movements).

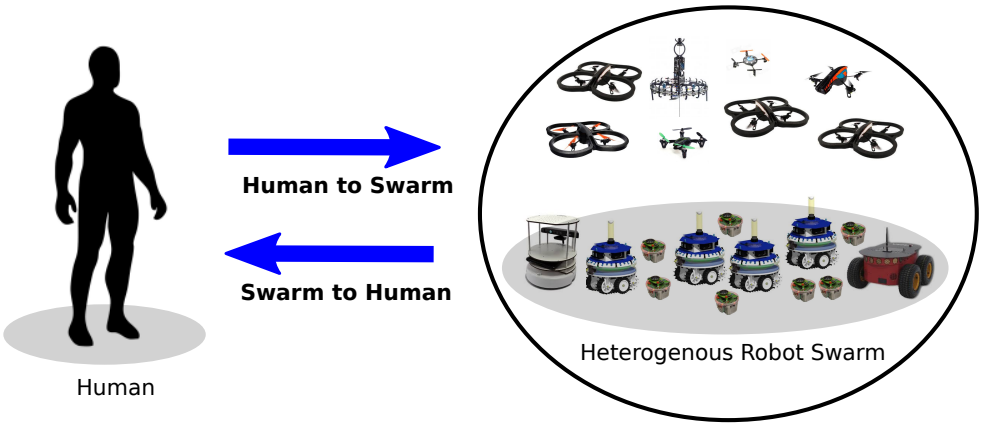


Figure 2.1. Bidirectional flow of information between humans and swarms. *Human-to-swarm communication*: Humans provide instructions to swarms. *Swarm-to-human communication*: Swarms convey feedback to humans.

The research presented in this chapter has been collaborated with a number of experts at IDSIA: Jérôme Guzzi, Alessandro Giusti and Gianni Di Caro. The grammar-based gesture language, the selection of robots using spatially-addressed commands, spatial directions pointed by humans, the swarm coordinated multi-modal feedback language, and the use of spokes-robots for human-to-swarm communication, are based on the creative ideas of the above mentioned individuals. In terms of my contribution, I have implemented and setup the UGV and UAV platforms with the components and technologies presented in this chapter, and have designed and performed experiments with real robots.

2.1 Background and Related Work

Human-robot interaction (HRI) is an extensively studied domain [Murphy, 2004; Goodrich and Schultz, 2007] in which large majorities of works focus on examining interactions between a single human-robot pair [Breazeal, 2004]. Being a relatively new field of research, *human-swarm interaction* (HSI) has received less attention [Xiaohui and Eberhart, 2008], and not much is known about the issues related with interaction between humans and multiple robots [Jones et al., 2010; Rule and Forlizzi, 2012]. Although some existing solutions do provide some capabilities to mitigate complexity that human operators may face in interacting with multiple robots [Saïdi and Pradel, 2006], however it is not entirely clear what kinds of interaction mechanisms and strategies are most appropriate.

The sections below review topics in different domains. The covered topics include: the interaction modalities and methods used by humans to interact and communicate with multiple robots (see Section 2.1.1), and the interaction modalities used by individual and multiple robots to convey feedback to humans (see Section 2.1.2). In addition, for human-to-swarm communication, gesture-based interaction techniques (see Section 2.1.1.1) and dialogue-based interaction mechanisms (see Section 2.1.1.2) are reviewed.

2.1.1 Modalities for Humans to Communicate with Swarms

The majority of existing works that deal with interaction between humans and multiple robots adopt teleoperation strategies [Kira and Potter, 2009; Vasile et al., 2011; Clare and Cummings, 2011; Sycara and Lewis, 2012; Lewis and Sukthankar, 2012; Kolling et al., 2013; Sklar et al., 2013] (supervised control by humans), which rely on the use of sophisticated supporting devices (e.g., handhelds such as joysticks, smartphones and tablet computers) [Coppin and Legras, 2012]. The work conducted by Payton et al. [Payton et al., 2001, 2003] which is discussed in Section 1.1.2, provides a good example of the use of *instrumented methods* (i.e., handhelds and wearables) for interaction with multiple robots. In recent times, an intuitive head-mounted display (HMD) was developed in [Lichtenstern et al., 2013], which had the capability of cycling through real-time video streams originating from the cameras of multiple UAVs. This enabled humans to select a particular robot and inspect the field of view (FOV) of the robot.

Although *instrumented methods* serve as an efficient interaction medium and reduce mental efforts put by human operators, they lack spatially-addressed control of multiple robots. The widespread availability of cost-effective sensors (e.g., cameras and microphones) for ubiquitous computing [Malima et al., 2006; Yin and Xie, 2007] enables multiple robots to sense audio/visual signals given by humans using *uninstrumented methods*. The research team at the Autonomy Lab of Vaughan has demonstrated the successful use of uninstrumented methods for HMRI [Couture-Beil et al., 2010a; Milligan et al., 2011; Pourmehr et al., 2013a; Monajjemi et al., 2013], as discussed in Sections 1.1.2 and 4.1.1. The Autonomy Lab introduced the use of visual interaction modalities such as, gaze, hand gestures, and body postures. Audio modalities such as speech and non-verbal sounds have also been used in conjunction with vision for realizing multi-modal interaction [Stiefelhagen et al., 2004; Xavier and Nunes, 2007; Jones et al., 2010; Pourmehr et al., 2014; Monajjemi et al., 2014].

With respect to non-verbal communication methods (see Figure 1.2), the focus in this research is on how *hand gestures* can be used for interaction with robot

swarms. The next sections presents the related works for gesture-based interaction and control of single and multiple robots, and dialogue-based interaction between humans and robots.

2.1.1.1 Gesture-based Interaction with Robots

In general, hand gestures can be sub-divided into three major groups, namely, *adapters*, *conversational* and *symbolic*. Adapters are hand movements which are not considered gestures, and include manipulations such as, scratching, rubbing, tapping, and touching. Conversational gestures are hand movements that do no refer to actions or words, but instead accompany speech, and are related to the speech. In this research, *symbolic gestures* are of great importance, as they represent hand movements with specific conventionalized meanings.

Being an active area of research, hand gestures have served as interaction modalities for various applications, including, sign language interpretation [Kelly et al., 2010], video games control, virtual reality, and assistive environments [Stefan et al., 2008]. Numerous research efforts have promoted the use of hand gestures as effective interaction modalities for HRI tasks [Mitra and Acharya, 2007; Wachs et al., 2011; Konda et al., 2012; Naseer et al., 2013]. However, hand shape recognition [Duta, 2009; Yin and Zhu, 2006; Huang et al., 2011a; Hu et al., 2012] is a challenging problem [ChaLearn Gesture Challenge], due to the ambiguities associated with different hand poses (orientations) and computer vision problems associated with different lightning conditions. In order for gesture recognition systems to be real-time, robust and deployable in uncontrolled environments [Fang et al., 2007b,a], they need to be able to operate in complex scenes with different backgrounds, under variable lightning conditions [Zhu et al., 2013], while taking into consideration different hand positions/orientations and occlusions [Choras, 2009; Duta, 2009; Hu et al., 2012].

Humans use a broad range of deictic gestures that can direct attention towards collocated objects, people or spaces. To investigate the communication effectiveness of different gestures under various conditions, a set of six dietic gestures: pointing, presenting, touching, exhibiting, grouping and sweeping gestures, were evaluated in [Sauppé and Mutlu, 2014] for a HRI task with a NAO robot. The Autonomy Lab introduced the use of gestures for HMRI applications [Couture-Beil et al., 2010a,b; Milligan et al., 2011]. By detecting human motion (see Appendix B) in predefined zones of the upper body, they defined a set of four distinct *hand waving gestures* (i.e., no wave, left hand wave, right hand wave, two-handed wave) [Monajjemi et al., 2013], which encoded instructions to select and command multiple robots and to gain the attention of

robots [Pourmehr et al., 2013a,b] (see Section 4.1.1). As waving gestures are characterized based on the optical flow of motion in the hands and arms, in practice, this limits the total number of gestures that can be used by humans, and does not fully exploit the potential advantages of gestures, such as the intuitiveness, spatial-addressing properties and shape characteristics.

The majority of uninstrumented methods for interaction with multi-robot systems and robot swarms that use hand gestures and body postures have adopted the Microsoft Kinect sensor [Podevijn et al., 2012, 2013; Gasparri et al., 2012; Lichtenstern et al., 2012; Pourmehr et al., 2013b; Alonso-Mora et al., 2015] for reducing the expertise required by human operators. In [Lichtenstern et al., 2012] a system for controlling a team of flying UAVs was presented, where a human directed gestures to a single UAV (i.e., the elected leader of the swarm) which was equipped with a Kinect. The research group of Dorigo developed a HSI system for gesture-based control of robot swarms [Podevijn, 2012; Podevijn et al., 2012, 2013] that allowed humans to provide commands to swarms using the Kinect sensor. Human operators had the ability to guide groups and teams of robots to designated task completion zones using five basic hand gestures, that encoded the tasks [Podevijn et al., 2012]: steer, split, merge, stop and select. However, the disadvantages of the Kinect are that: multiple Kinect devices located within close proximity of each other generate interference, it is not economically feasible to equip every individual robot in a swarm with a Kinect, and the Kinect is not suitable for outdoor use.

To simplify the task of recognizing gestures without loosing generality, we consider the use of *passive markers* that can naturally supplement symbolic gestures. Human operators wear a pair of inexpensive coloured gloves (i.e., passive markers) as illustrated in Figure 2.3, which serve as suitable color-coded inputs for robot swarms to detect and recognize gestures. Gloves are worn without any instrumentation and are used as part of a computer vision-based approach (see Appendix A). Existing works have also addressed the hand gesture recognition problem by adopting the use of gloves [Parvini et al., 2009; Huang et al., 2011b]. For instance, a multi-coloured glove with a known pattern was designed in [Wang and Popović, 2009], which was used for tracking hands at different orientations and with different finger positions.

2.1.1.2 Interacting with Robots using Dialogues

Within the domain of HRI, a *human-robot conversational dialogue* consists of the entire process in which a human provides a command to a robot, and in turn the robot provides feedback to the human. In a grammar-based language of

operational commands, one command given by the human represents a single word, and multiple commands (i.e., a sequence of words one after the other) comprise of a full sentence, as illustrated in Figure 2.2. As a result, sentences comprise of instructions having more than one word.

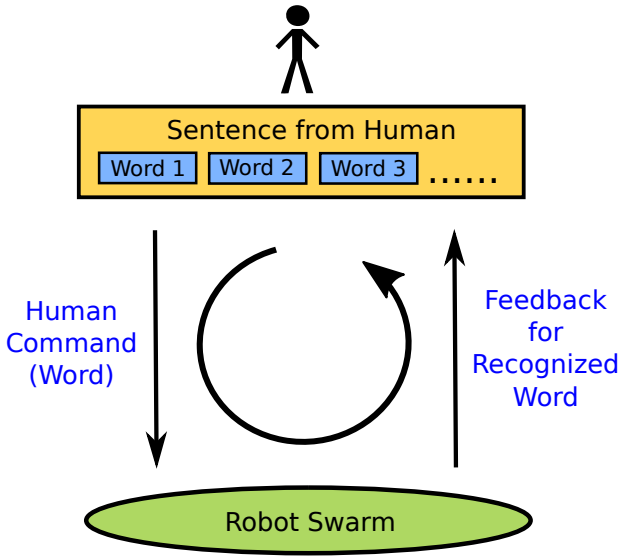


Figure 2.2. Dialogue-based interaction between a human (who provides a sentence using single words/individual gestures) and a robot swarm (which provides swarm-level feedback regarding the recognition outcome of each word).

With respect to the considered HSI scenario (see Section 1.2), a dialogue between a human and a robot swarm can take place as follows: a human provides a single word (individual gesture) to the swarm, and the swarm provides appropriate feedback to the human (see Section 2.1.2) after recognizing the given gesture. This process repeats for every word (individual gesture) given by the human in a sentence, as depicted in Figure 2.2. A dialogue is considered complete when the human has presented the full sentence, and all words (gestures) in the sentence have been correctly recognized by the swarm. Such a structured dialogue enables bidirectional communication between humans and swarms.

Since the dawn of artificial intelligence in the 1960s, *human-machine interaction* (HMI) using verbal and non-verbal dialogues [Weizenbaum, 1966] has been an active area of research [Wahlster and Kobsa, 1986]. Dialogue-based interaction [Allen et al., 2001; Fong et al., 2003] between humans and robots can be realized using three main communication channels: spoken (or verbal), written, and non-verbal modes of communication. However, as human language is

frequently ambiguous, humans often refers to objects/entities in terms that are sometimes incomprehensible to robots. This can make the complete understanding (i.e., clarification) of human instructions even more difficult for robots (e.g., instructions may be interpreted as having more than one meaning). Considering these challenges, a limited number of research efforts have investigated the use of dialogues for interaction with single robots [Jones and Rock, 2002], multiple robots [Chambers et al., 2005], and swarms [McLurkin et al., 2006].

A spatial language for human-robot dialogues was reported in [Skubic et al., 2004], which described how linguistic spatial descriptions and information can be used by humans in a natural style. In recent times, two gesture-based languages, namely, the UGV Language (UGVL) [Stoica et al., 2013] and the UAV Language (UAVL) [Stoica et al., 2014] were introduced for controlling the behaviour of multiple ground and aerial robots respectively. However, the UGVL and UAVL methods adopted an instrumented method with the use of a Biosleeve device [Wolf et al., 2013] (a sophisticated interaction device with multiple sensors) which is worn on the arm of the human operator.

To the best of our knowledge, currently no dialogue-based interaction system exists that can address and command a swarm of robots using spatially-addressed instructions and gestures, while providing swarm-level feedback to humans. As it is too costly to implement iterative methods for building a natural language system that can constantly add/update new contents incrementally [Dow et al., 2005], we have developed a *grammar-based gesture language* with a vocabulary of commands in Section 2.2.1, which follows syntax rules for combining the semantic meanings of individual gestures and provides humans more flexibility and variety for communicating with robot swarms.

2.1.2 Modalities for Swarms to Convey Feedback to Humans

During dialogue-based interaction, it is crucial for human operators to receive appropriate feedback from robot swarms, in order to be fully aware of the current intentions, status and state of the swarm. For instance, after an individual robot or swarm has recognized a human command (e.g., gesture), the swarm should inform the human regarding the recognized command using a feedback mechanism. As feedback from robots plays an important role during human-robot conversational dialogues [Skubic et al., 2004; Deits et al., 2013], to convey to humans the response (feedback) of a single or a group of robots, research efforts have used the actuation systems and devices on-board small mobile robots.

A first analysis of feedback from a small mobile ground robot (UGV) was investigated in [Mohammad and Nishida, 2007], in which the *e-puck* robot [Mon-

dada et al., 2009] was used. The results reported in [Mohammad and Nishida, 2007] and its successive works [Mohammad and Nishida, 2008] indicate that, actuation systems such as, speech, lights (e.g., LEDs) and robot movements (i.e., wheels), are the most effective interaction modalities that small mobile robots can use for conveying feedback to humans. The results in [Mohammad and Nishida, 2008] explicitly indicated that, using any type of feedback provides a statistically significant improvement as compared to cases with no feedback. No interaction modality is considered superior over another, as modalities are generally problem-specific. The research group of Dorigo [Podevijn, 2012; Podevijn et al., 2012, 2013] has investigated the use of feedback from a swarm of Foot-bots (see Section 1.2.1). Self-organized mechanisms for conveying visual feedback to humans were presented in [Podevijn et al., 2012, 2013] which adopted the use of coloured LEDs and coordinated multi-robot movements.

Following these works, we consider the use of *multiple individual and group modalities*, which includes, coloured lights, synthetic speech and robot movements, for swarms to convey feedback to humans (see Section 2.3).

2.2 Human-to-Swarm Communication

This section introduces a gesture language for human-to-swarm communication using dialogue-based interaction. This vocabulary of gesture commands has been designed keeping into mind *search and rescue* (SAR) scenarios (see Section 1.1.2) and is aimed for use with UGV and UAV swarms.

2.2.1 Gesture Language for Issuing Human Instructions

An iconic *grammar-based gesture language* with a vocabulary of non-verbal commands has been developed for humans to communicate and interact with robot swarms. Individual gestures represent the symbols (i.e., grammatical expressions) in the language. The concept of a non-verbal communication language with grammatical expressions emerges from the context of human-robot conversational dialogues (see Section 2.1.1.2). As it seems natural to emulate structured dialogues that are evident in humans, we consider that the following characteristics and functionalities are crucial for the gesture vocabulary:

(a) *Flexibility and Variety*: Human commands (gestures) can be represented using one hand or a combination of both hands. The choice of using one and two-handed gestures provides variety and flexibility to human operators, as

simple instructions can be associated with one handed commands, while complex instructions may require the use of both hands.

(b) *Intuitiveness*: As the vocabulary comprises of multiple gesture commands, gestures need to be intentionally engineered so they can be easily comprehended by robot swarms. Gesture commands need to be instinctive, so that they represent similarities with sign language, and at the same time they should be able to encode a variety of tasks that robot swarms can perform.

(c) *Spatiality*: For humans to address/select robots from a swarm, gestures that possess spatial properties are essential. More importantly, individual robots in a swarm should be able to accurately interpret if spatially-addressed gestures are presented to them. In this context, the spatial relationship between humans and robot swarms needs to be investigated.

(d) *Extensibility*: The vocabulary needs to be designed such that, it is relatively easy to expanded/increase the number of gesture commands (so that new gesture symbols that are "different" enough from previous ones can constantly be added for handling new tasks).

Operationally, the grammar-based language is implemented as a Finite State Machine (FSM) with four *interaction states*, referred to as *semantic gesture classes* (see Figure 2.3). These interaction states comprise of the entire gesture language, and are introduced next in Section 2.2.2.

2.2.2 Semantic Gesture Classes

This section introduces the four semantic gesture classes which consist of four different levels of instructions and commands that can be given by human operators to robot swarms. Every semantic gesture class corresponds to a single interaction state in the gesture language. Figure 2.3 illustrates the gesture language with the four semantic gesture classes of commands. A trained statistical classifier is used for learning (see Chapter 5) and recognizing (see Sections 3.4) gestures in the language. Individual gestures in every semantic class are designed in such a way that they different in terms of shape characteristics and flexibility (one or two-handed) when compared to gestures in all other semantic classes.

In every semantic class, gestures are selected such that they are sufficiently expressive and highly discriminative compared to gestures in other semantic classes. The first semantic gesture class (or first interaction state) explicitly deals with the *selection of individuals and groups* of spatially-situated robots from a

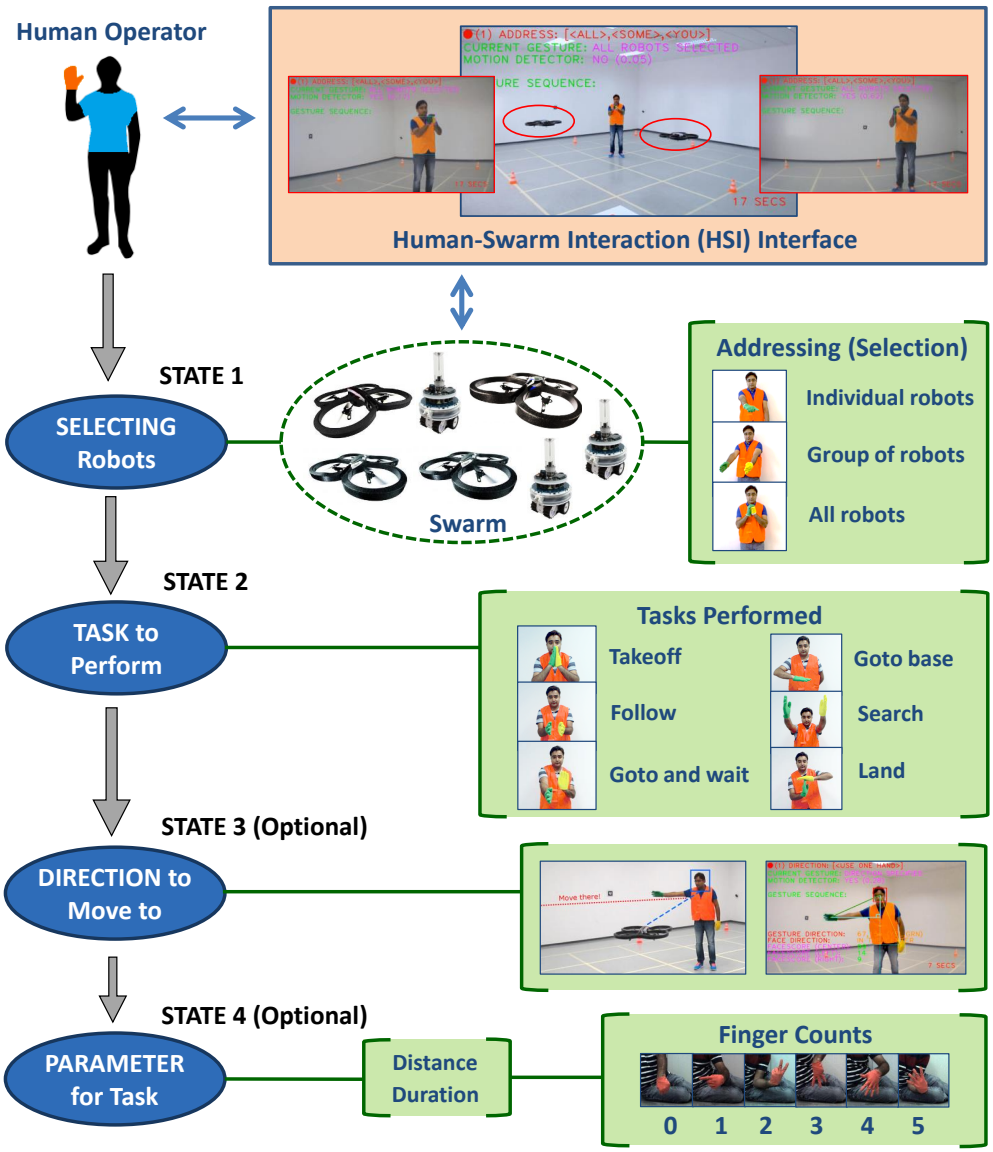


Figure 2.3. A grammar-based gesture language (vocabulary) for dialogue-based interaction between humans and robot swarms.

swarm. As addressing is an important means to select robots and trigger their attention, we consider that, when interacting with a robot swarm, all commands begin with an addressing terminal, as presented later in Section 2.2.3.1. In this way, the first command issued by humans is for selecting robots.

The second semantic class of gestures represents potential SAR commands that humans can provide to swarms to perform predefined tasks. The third and

fourth semantic classes are parametric, and humans may choose to use them depending upon the complexity of the task. For instance, if the task requires the selected robots to move to a specific direction, the third semantic class is required. In addition, if additional information is required to complete the task, then the fourth semantic class which represents application-specific parameters needs to be used. The four semantic gesture classes are introduced below.

1. Selecting Spatially-situated Robots (C_{sel}): These commands are used to select a specific number of robots and direct (instruct) the selected robots to perform a task. Spatially-addressed commands provide an intuitive way of selecting and gaining the attention of individuals and groups of robots from a swarm. From a swarm of robots, three types of spatial robot selections are considered [Nagi et al., 2014c]: (i) individual robots, (ii) a group (subset or team) of robots, and (iii) all robots in the swarm, as illustrated by the spatial pointing gestures in Figure 2.4(a). Section 4.2 introduces the algorithms that enable spatial robot selection and Section 4.1.1 presents the related work. After robots have been selected, the selected robots can then be instructed to perform a specific task, using the second semantic gesture class presented below.

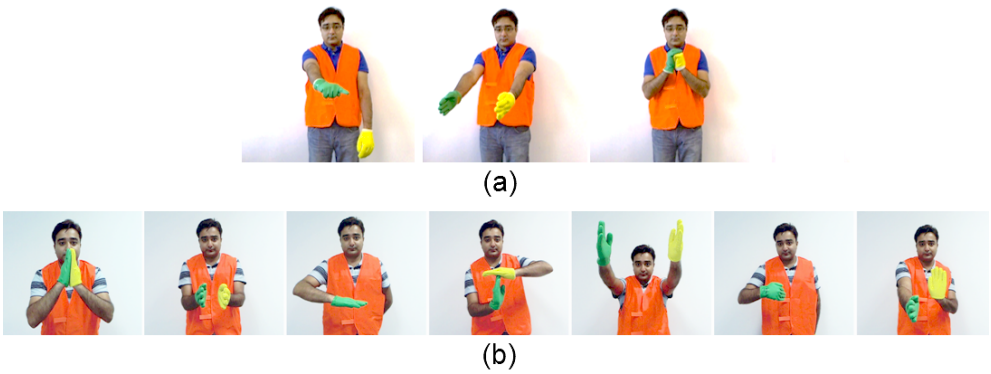


Figure 2.4. Gesture commands in the grammar-based language. (a) Spatial gestures for robot selection. Left to right: Individual selection, group selection, and all robot selection. (b) Gestures for performing SAR tasks. Left to right: Take off, follow, go to base, land, search, follow me, go to and wait.

2. Application-specific Commands (C_{app}): To support a variety of SAR missions that heterogeneous teams of robots (UGVs and UAVs) can perform, gestures are intentionally designed to be intuitive (i.e., gestures represent similarities with sign language) [Stern et al., 2006, 2008b,a; Wachs et al., 2008; Stern et al., 2009] so they can easily be remembered by human operators. Figure 2.4(b) illustrates

seven gesture commands which could represent the tasks: take off, follow, go to base, land, search, follow me, and go to and wait.

3. Spatial Directions (C_{dir}): Tasks may require robots to move to a specific direction or location. Humans can provide spatial directions for robots to manoeuvre (e.g., to move to a space, object or specific location) by pointing the hand and arm in a specific direction, as illustrated in Figure 2.5. For simplicity, we consider that spatial directions given by humans consist of four *cardinal compass directions*: north (N), east (E), south (S), and west (W), and four *intercardinal directions*: north-east (NE), south-east (SE), south-west (SW), and north-west (NW). Using these compass directions, humans can provide up to 8 unique directions, in which a single direction represents a circular area of 45° within a circular plane of $[0, 360°]$. Direction-specific feedback in Section 2.3.2 discusses the approach using which spatial directions are estimated by robots.

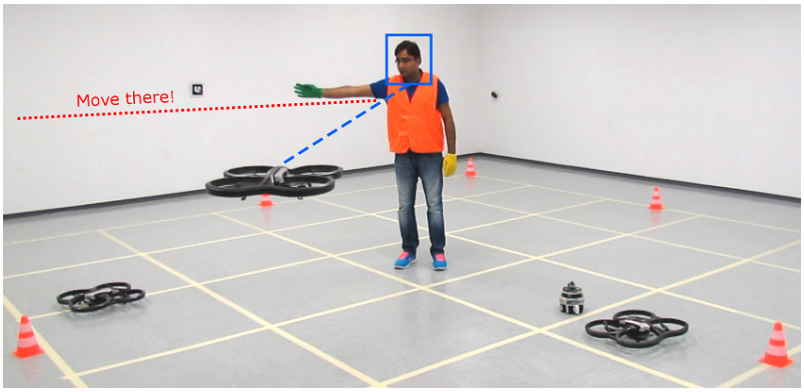


Figure 2.5. A human operator providing a spatial direction (to an airborne UAV) to maneouver to, using the direction pointed by the arm and the hand.

4. Numerical Quantities (C_{num}): The simplest way using which application-specific parameters (i.e., quantities) can be provided, is by encoding gestures into *finger counts*, as illustrated in Figure 2.6 with $K = 6$ gestures that represent finger counts from 0 to 5. With a combination of both hands, atleast 11 distinctive quantities (e.g., finger counts 0 to 10) can be easily represented. Task-parameters encode specific measurement units which include, distance (e.g., in metres), duration (e.g., in minutes) and speed (e.g., in m/s).

The gesture language consists of $K = 16$ gestures (see Figures 2.4 and 2.6) that are distributed among the four semantic classes [C_{sel}, C_{app}, C_{dir},C_{num}]. The next sections present how individual gestures in the four semantic classes are combined to build full sentences of human commands.

Figure 2.6. Finger count gestures used for representing quantities. Left to right: Finger counts from 0 to 5.

2.2.3 Building Syntactically Correct Sentences

Having the ability to support a variety of operations, the gesture language builds iconic sentences by combining single gestures (words) in a semantically and syntactically correct manner. Every sentence contains at least one selection (C_{sel}) and one task (C_{app}) command. Depending upon the complexity of the task, humans may provide additional instructions, which includes spatial directions (C_{dir}) and application-specific parameters (C_{num}). As a result, sentence composition follows the simple rule that, sentences are always organized in the sequence C_{sel}–C_{app}–[C_{dir}]–[C_{num}], where the commands in the brackets are optional.

This form of simple grammar allows to easily check the correctness of sentences, and words (individual gestures) can be syntactically combined to build sentences while retaining the semantic/symbolic meanings of the individual gestures. Section 2.3.2 presents the trained statistical classifiers used to recognize the gestures the four semantic classes.

2.2.3.1 Grammar Definition

This section presents the grammar developed for sentence composition. This grammar expresses relatively complex human instructions by combining the individual gestures given in Figures 2.4 and 2.6 and spatial directions shown in 2.5. The Backus–Naur Form (BNF) family of meta-syntax notations [Backus et al., 1960; Naur et al., 1963] has been adopted to express *context-free grammar*, since it encodes grammar intended for human consumption. Based on the gesture language in Figure 2.3, complete sentences in the BNF are as follows:

```
<sentence> ::= <addressing> <task>          (complete sentence)

<addressing> ::= all robots                 (task directed to all robots)
<addressing> ::= some robots <subset>       (select only a few robots)
<addressing> ::= you                        (directed to individual robots)
```

```
<subset> ::= (robots within spatial cone defined by hands)

<task> ::= take off
<task> ::= go to base
<task> ::= follow <person>
<task> ::= land <where>
<task> ::= follow me
<task> ::= go to and wait <where>
<task> ::= search <where> <duration>

<where> ::= <direction> <distance>
<where> ::= <landmark>

<duration> ::= <number>
<distance> ::= <number>

<landmark> ::= here
<landmark> ::= base

<person> ::= <person-id>
<person> ::= <direction>
<person-id> ::= <number>

<direction> ::= (relative direction pointed by human)

<number> ::= (number of fingers)
```

2.2.3.2 Grammar Terminals

The grammar includes the following terminals, which correspond to actual gestures. Gestures on the right-hand side of a single non-terminal need to be clearly distinguishable (below, they have been regrouped into a single row):

```
allrobots  somerobots  you
takeoff  gotobase  follow  land  gotoandwait  search
here  base
(relative direction pointed by user)
(number of fingers)
```

2.2.3.3 Grammar Generality

The grammar-based language has been designed keeping into mind SAR missions. However, this grammar can be easily adopted for other application scenarios, and could be easily extended. As only the blue rules in Section 2.2.3.1 are application-specific, a meaningful grammar for different applications can easily be built by substituting the blue rules with other application-specific rules. Depending upon the complexity of the application scenario, additional semantic gesture classes can also be included into the language.

2.2.3.4 Examples of Valid Sentences

Using the CFG defined in Section 2.2.3.1, a few examples of sentences (that do not violate syntax and semantic rules) are:

`allrobots goto <dir> 5`	(all robots go to <direction pointed by user> for 5m)
`you land here`	(pointed robot lands at current position)
`somerobots follow 3`	(pointed robots follow person with id #3)
`you and you follow right`	(pointed robots follow person at given direction)
`allrobots search left 10 4`	(all robots search in direction left at 10m for 4mins)

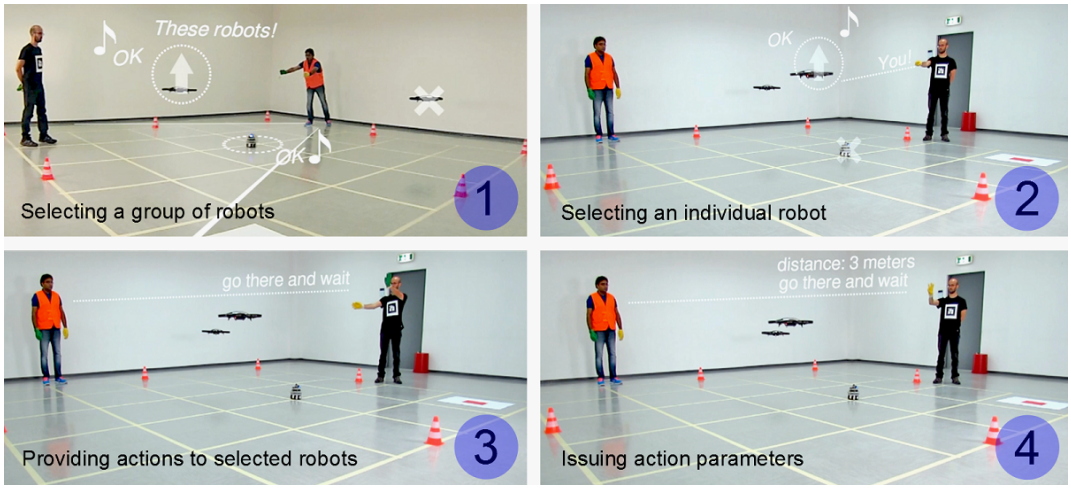


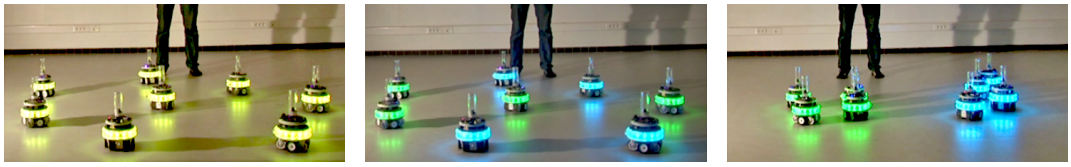
Figure 2.7. A real-world demonstration scenario (images from top left to bottom right) in which two human operators provide a sentence of gesture commands to a heterogeneous robotic swarm composed of 2 UAVs and 1 UGV.

Figure 2.7 depicts a real-world scenario, in which two human operators provide a sentence to select and command robots from a heterogeneous swarm.

First, one human operator selects a group of two robots. Next the second operator, selects an individual robot. Next, all selected robots are instructed by the second operator to go and wait at the pointed direction at a distance of 3 metres.[1]

2.3 Swarm-to-Human Communication

During dialogue-based interaction with robot swarms, it is crucial for robots in a swarm to provide feedback to human operators. Although feedback from robots in a swarm can be wirelessly transmitted and displayed on handheld devices (e.g., smartphones and tablets), the focus in this research is on the use of uninstrumented interaction methods, as introduced in Section 1.1.2. Since small mobile robots mainly have a limited set of on-board actuation devices, we consider the use of *visual mechanisms* (i.e., coloured lights and locally coordinated movements) and *audio capabilities* (i.e., playback of sounds and synthetic speech) which are on-board swarm robots such as the Foot-bots (see Section 1.2.1.1).



(a) (b) (c)

Figure 2.8. Visual feedback conveyed by a robot swarm (sequence from left to right) [Podevijn et al., 2012; Podevijn, 2012]. (a): A human commands a swarm to split into two groups. (b) Groups are formed and visual feedback is given using LEDs of different colors. (c): The two groups are separated using locally coordinated movements (visual feedback).

As visual feedback from multiple robots has demonstrated to be effective in conveying information to humans (see Section 2.1.2), coloured lights are simple and easy to use and can encode different types information. For instance, if a human provides a gesture that requires a robot swarm to split into two groups, coloured lights and locally coordinated movements can convey to humans the two groups of robots, as illustrated in Figure 2.8 [Podevijn et al., 2012]. To convey application-specific feedback, audible feedback such as predefined sounds and synthetic speech are known to be easily understood by humans. As an example, for a robot to convey to a human that it's battery is running low, the robot

[1] A video demonstrating the real-world scenario in Figure 2.7 is available at: http://www.jnagi.net/demonstration_scenario

can play a sound file (i.e., synthetic voice or speech) in a dying (fading out) voice. In addition, coloured LEDs can be augmented with speech such that, the more critical the battery power gets, the faster the LEDs blink.

As the combined use of multiple mechanisms (audio and visual) is more effective in terms of motivation of a human's response, *multi-modal feedback* robustly conveys swarm-level status, decisions, and intentions to humans. To reduce the cognitive load of human operators which see and hear feedback from swarms, three types of swarm-to-human multi-modal feedback are developed:

- The first feedback is responsible for conveying a swarm's understanding of recognized gesture commands to humans (see Section 2.3.1).
- The second feedback provides basic reasoning capabilities and conveys to humans self-assessment decisions made by swarms. This feedback identifies sensing ambiguities, detects mistakes made by humans and errors made by swarms, and requests humans for corrections (see Section 2.3.2).
- The third feedback is used for guiding human operators through the interaction process (see Section 2.3.3).

The combined and coordinated use of these feedback builds a *swarm coordinated multi-modal feedback language*, which allows swarm-to-human interaction and communication with a high and immediate impact. The three multi-modal feedback are introduced in the next sections.

2.3.1 Feedback Showing Swarm Understanding

This section presents a multi-modal feedback which conveys to humans a swarm's understanding of recognized (classified) gesture commands during dialogue-based interaction. In context of the gesture language defined in Section 2.2.1, feedback from robot swarms varies depending on the semantic gesture class (see Section 2.2.2). In principle, multi-modal feedback from a swarm of robots can be given in three possible ways:

1. An individual robot is selected as the representative of the swarm. This individual robot, better known as a *spokes-robot*, communicates swarm-level feedback to humans on behalf of the swarm.
2. Only a random subset of spatially-situated spokes-robots (in the swarm) convey swarm-level feedback to human operators.

3. The entire robot swarm (i.e., all robots) conveys feedback to humans.

A spokes-robot can be beneficial in situations when all robots in the swarm need to convey the same information (feedback) to humans. With the use of a spokes-robot, a swarm's consumption of energy and resources (e.g., communication bandwidth) can be minimized. In principle, a spokes-robot can be selected from a swarm using at least two possible strategies:

(a) *Relative Location with respect to Humans*: The robot that is the most closest (nearest) to the human, or the robot that has the most frontal view of the human (i.e., the human's face).

(b) *Actuation Capabilities*: Actuation devices on-board robots can convey different types of feedback. For instance, some robots in the swarm may have coloured LEDs or beacons, others might only have speakers, while others may have a combination of two or more (multi-modal) such capabilities.

Other possible strategies to select spokes-robots may include: the color of the LEDs or random selection. In the following section we describe how the two aforementioned strategies are used for conveying multi-modal feedback (to humans) regarding the swarm's understanding of recognized gesture commands.

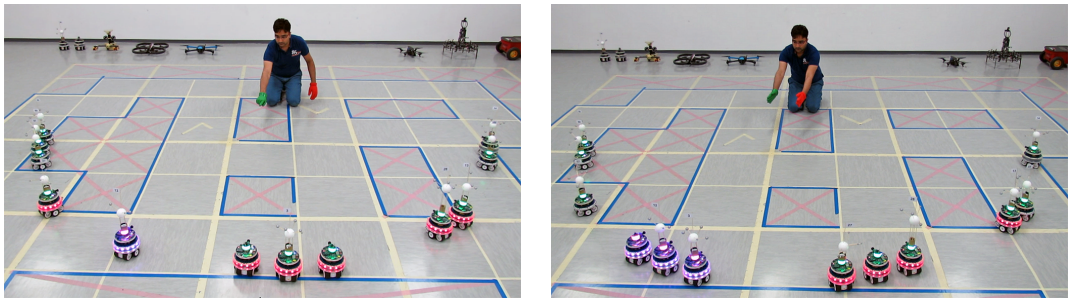
2.3.1.1 Multi-modal Feedback for Interaction States

We develop four multi-modal feedback for each of the four semantic gesture classes (or interaction states) given in Section 2.2.2. These feedback convey the swarm-level classification of gesture commands to humans. Although there are many techniques using which robots in a swarm can provide these feedback, we consider one possible way in which visual and audible mechanisms are carefully chosen and combined together. The four multi-modal feedback that are discussed below take into consideration when UGVs are used vs. UAVs.

1. Spatial Selection Feedback: Spatially distributed robots in a swarm that are selected by humans, use visual modalities (including coloured lights and coordinated robot movements) to convey to humans that they have been selected. After robots in a swarm recognize an <addressing> symbol from the grammar (see Section 2.2.3.1), such as:

```
<addressing> ::= allrobots
<addressing> ::= somerobots <subset>
<addressing> ::= you
```

the selected robot(s) (which maybe: an individual spokes-robot, multiple spokes-robots or the entire swarm) provide *selection feedback* to the human using coloured lights. If an individual or a group of robots is selected with the C_{sel} command (see Section 2.2.2), the use of spokes-robots is the best choice. For instance, if an individual robot is selected, this individual is the only spokes-robot. If a group of robots is selected, then all robots in the selected group are considered spokes-robots. Otherwise, if all robots in the swarm are selected, then the entire swarm communicates feedback to the human.



(a) (b)

Figure 2.9. Spatial selection feedback from a robot swarm using colored LEDs. (a): Feedback from a spokes-robot robot. (b): Feedback from a group of robots.

Selected robots convey multi-modal feedback to humans based upon the actuation capabilities of the robot platforms (see Section 1.2.1). In the case of UGVs, individuals and groups of selected UGVs blink (flash) their colored LEDs in a different color than the non-selected robots, as illustrated in Figure 2.9, in which the LEDs of the selected robots change to pink color.[2] Groups of selected UGVs use locally coordinated movements to physically move close to one another for visually expressing a group, while non-selected UGVs move a little farther away or remain at their places. When UAVs located on the ground are selected, they first make a beep sound, and then start to fly above the same location where they were positioned. Also, when UAVs are selected, after selection the selected UAVs fly at higher altitudes while the non-selected UAVs usually fly a little lower.

Algorithms 2 and 3 in Section 4.2.1, depict the mechanisms using which selected individuals and groups of robots convey feedback to humans. In practice, line 21 in Algorithm 2 and line 23 in Algorithm 3 allows selected individuals and groups respectively, to change the colors of their LEDs. If all robots in the swarm are selected, then all robots change their LEDs to the same color.

[2]A video demonstrating spatial selection feedback from UGV swarms is available at: http://www.jnagi.net/individual_and_group_selection

2. Application-specific Feedback: To convey the recognition outcome of task related commands, we employ a spokes-robot which conveys swarm-level decisions on behalf of the swarm. We consider the use of audio mechanisms by using the speakers on-board the robots. After robots in a swarm identify a <task> symbol that has been defined in the grammar (see Section 2.2.3.1):

```
<task> ::= land <where>
<task> ::= follow <person>
<task> ::= search <where> <duration>
```

a spokes-robot provides audible (vocal) feedback, by speaking the name of the recognized application-specific command. In this case, the spokes-robot is selected as the robot which is the most nearest to the human. This audible feedback from the spokes-robot serves as an acknowledgement for the human.

3. Direction-specific Feedback: For a swarm of robots to convey direction-specific feedback (i.e., illustrate recognized spatial directions) to humans, multi-modal feedback is adopted with the use of audio and visual modalities. After a swarm identifies a relative <direction> (or location) pointed by the human, based on the defined grammar:

```
<where> ::= <direction>
<person> ::= <direction>
<direction> ::= (relative direction pointed by user)
```

robots in the swarm convey the recognized direction (pointed by the human) using their coloured LEDs, and a spokes-robot provides audible feedback by speaking the name of the recognized direction (e.g., north, south-west). As the Footbot (UGV) platform is equipped with a circular ring of 12 multi-colored LEDs around the circumference of the body (see Section 1.2.1.1), LED colors corresponding to the direction where the human is pointing at are set to a specific color (e.g., red), as illustrated in Figure 2.10. Since such LEDs are not available on the Parrots (UAV platform; see Section 1.2.1.2), audible feedback given by the spokes-robot is sufficient to convey the recognized direction to humans.

To convey direction-specific feedback, robots need to estimate spatial directions pointed by humans.[3] We provide an explanation of how spatial directions are estimated by individual robots in a swarm. Firstly, robots in a swarm need to

[3]A video that demonstrates how spatial directions are estimated by robot swarms is available at: http://www.jnagi.net/spatial_directions

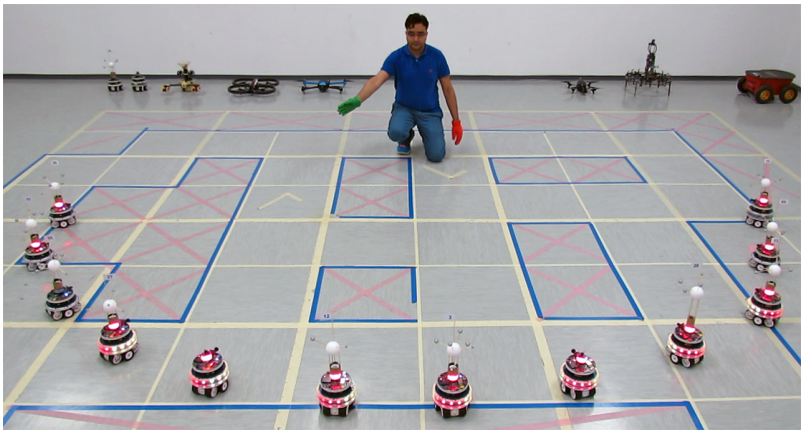
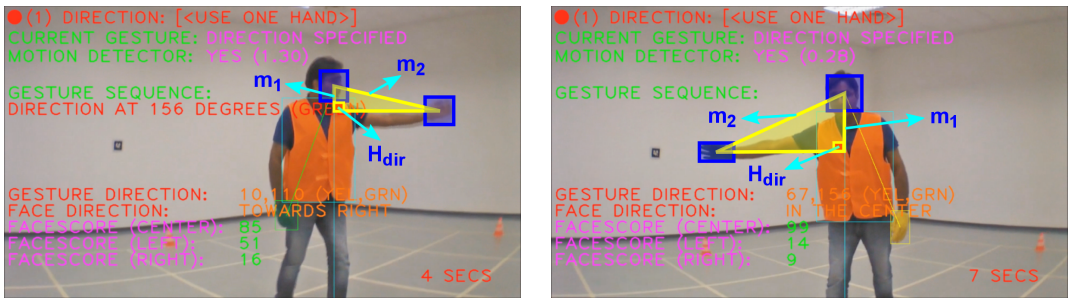


Figure 2.10. A human provides a direction by pointing his hand towards the left, where the robot swarm needs to move. The swarm recognizes the command and provides feedback using coloured LEDs to indicate the recognized direction.



(a) (b)

Figure 2.11. A human providing a spatial direction to an airborne UAV. Measures from the upper body are used for identifying the direction of the pointing hand.

to identify their relative angular position with respect to the human's location. At this aim, operations on the upper body are performed to detect the face in a $[0, 180°]$ semi-circular plane using the face detection approach in Section 4.3.2.1. Face detection results in computing a bounding box $b = (f_x, f_y, f_{width}, f_{height})$ around the face, as illustrated in Figure 2.11. Using the coordinates of the bounding box, the *face centroid* $c_{face}(\mathbf{x}, \mathbf{y})$ coordinates are computed as: $\mathbf{x} = ((f_x + f_{width})/2)$ and $\mathbf{y} = ((f_y + f_{height})/2)$. Next, a straight line m_1 is computed from the y-coordinate of $c_{face}(\mathbf{x}, \mathbf{y})$ to $y = 0$ of the image plane, and another straight line m_2 is computed from $c_{face}(\mathbf{x}, \mathbf{y})$ to the centroid of the pointing hand c_{hand}. In the final step, the inner angle between m_1 and m_2 is calculated as the *hand orientation* $H_{dir} = (\text{atan2}(c_{hand} - c_{face}) \times 180/\pi)$ within one complete revo-

lution [0 : 360°] (i.e., circumference of 2π radians), as illustrated in Figure 2.11 with a human providing spatial directions towards the right and left of a UAV.

The hand orientation H_{dir} is estimated by the robot which has the most frontal-view of the face (see Section 4.3.2.2), and this robot broadcasts H_{dir} to rest of the swarm. In this way, the entire swarm does not need to take part in estimating the hand direction. Alternatively, a mixed HSI solution using an instrumented-natural interface can be adopted. For instance, if every robot is equipped with a GPS device, then spatial directions pointed by humans can be selected robustly using spatial coordinates.

4. Numeric Feedback: To convey numeric feedback to humans, multi-modal feedback is adopted with the use of a spokes-robot which is selected as the robot closest to the human. After a swarm identifies a <number> given by the human, defined in the grammar rules:

```
<duration> ::= <number>
<distance> ::= <number>
<person-id> ::= <number>
<number> ::= (number of fingers)
```

the spokes-robot conveys the recognized number of finger counts K (where $0 \leq K \leq 10$ for finger counts given using both hands), by blinking its coloured beacon K consecutive times and speaking the number K.[4]

2.3.2 Feedback for Self-assessment and Error Correction

The multi-modal feedback presented in this section conveys to humans the self-assessment decisions made by swarms, if the swarm is confident or not in recognizing given gestures. Self-assessment made by swarms relies on basic reasoning capabilities: identifying sensing ambiguities, detecting mistakes made by humans in providing gestures and recognition errors made by swarms, and requesting humans for corrections (i.e., if a swarm is not confident in recognizing a gesture, the swarm requests the human to present the same gesture again).

In conventional closed-loop interaction systems, when a robot does not properly interpret (recognize) a command given by a human, the human provides feedback to the robot, using which the robot learns to better understand and robustly recognize the same command the next time. We consider that a more convenient way is by allowing robot swarms to assess the performance of their own

[4] A video demonstrating the use of visual feedback for finger count recognition is available at: http://www.jnagi.net/finger_count_recognition.

recognition (classification) decisions. Robot swarms can autonomously decide without any human intervention (i.e., without any external input or feedback) if they are confident or not in recognizing given commands (i.e., to assess swarm-level decisions without human inputs). For swarms to assess their recognition performance without human feedback is a challenging problem, which has not been addressed in existing works.

2.3.2.1 Formulation of Self-assessment Feedback

To equip robot swarms with basic reasoning capabilities, we consider that the swarm-level recognition outcome for classifying (recognizing) a gesture command produces two possible outcomes: *confident* or *not confident*. This implies that, after a swarm has recognized a given gesture command, the swarm may or may not be confident regarding its recognition outcome.

When a swarm realizes that it is not confident in recognizing a gesture command, it requests the human to provide the same gesture again, in an effort to recognize it confidently the next time. This process repeats until the swarm is confident in recognizing the given gesture. For instance, if a human provides a numerical quantity based on the count of fingers (see Section 2.2.2) and the swarm is not confident in recognizing the number of finger counts, the swarm requests the human to provide the same number of finger counts again. This mechanism ensures that, sensing and recognition errors made by swarms due to poor environmental conditions and mistakes made by human operators when issuing commands are robustly detected, and proper measures are taken by the swarm to continue with the interaction process.

With the aim to develop an intelligent and human-friendly HSI system, we introduce a strategy that provides four possible swarm-level recognition outcomes for the classification (recognition) of a single gesture command. In this strategy, "confident or certain" decisions are categorized as *properly recognized* or *inappropriate*, and "not confident or uncertain" decisions are further classified as *not properly recognized* or *undefined*, as introduced below.

(i) *Confident*: When a swarm is confident (certain), this implies that, either the given command has been properly recognized or the given command is inappropriate in the current circumstance. When a command is *properly recognized*, in order to speed up the interaction, no feedback is provided to the human. However, when the swarm identifies that the command is *inappropriate*, this reveals that given gesture command has been misplaced according to the predefined grammar rules in the gesture language (see

Section 2.2.3.1). This simply means that, the word (individual gesture) issued by the human is incorrectly positioned in the sentence. As an example, when a swarm requests a human to provide a gesture to select robots, but instead the human issues a command to perform a task (see Section 2.2.2), the command given by the human is inappropriate. When this happens, a spokes-robot (closest to the human) provides feedback to inform the human regarding his/her mistake, and requests the human to provide a gesture that belongs to the correct semantic gesture class (interaction state).

(ii) *Not Confident*: When a swarm is not confident (uncertain), this indicates that the given command is not properly recognized or the given command is undefined (not defined) in the gesture language. When a gesture command is *not properly recognized*, a spokes-robot provides feedback to request the human to present the same gesture again more carefully and clearly, so it can be recognized correctly the next time. However, when a swarm identifies that a given command is *undefined*, this implies that the given gesture does not exist in the predefined set of commands. In such situations, a spokes-robot provides feedback to the human operator to indicate that the human maybe unaware of the predefined commands in the vocabulary. Then, the interaction process starts from the beginning with the swarm requesting the human to provide a gesture for robot selection.

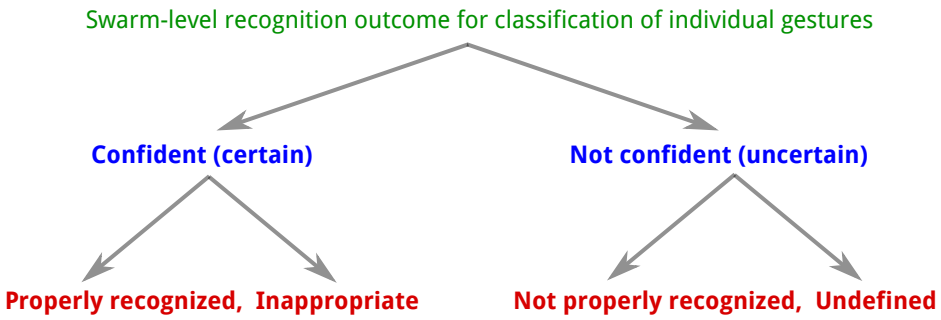


Figure 2.12. The swarm-level decision (for the recognition of an individual gesture) results in a *confident* or *not confident* decision. These decisions are further categorized into four swarm-level recognition outcomes: properly recognized, inappropriate, not properly recognized and undefined.

These four swarm-level recognition outcomes are illustrated in Figure 2.12, which form a new state in the HSI system, known as the *assessment state*. Figure 2.13 depicts the assessment state with respect to the gesture vocabulary in Figure 2.3 that has four interaction states.

Multi-modal self-assessment feedback from robot swarms is reliably conveyed to humans with the aid of blinking lights (used by the entire swarm) and synthetic speech (used by a spokes-robot). Visual feedback is provided such that all robots in the swarm blink their colored LEDs: once if the swarm properly recognizes the given command, twice if the swarm does not properly recognize the command, thrice if the command is inappropriate, and four times if the command is undefined. The spokes-robot (closest to the human) provides audible feedback by speaking the name of the recognized command and if the human needs to provide a correction (i.e., provide the same gesture again).

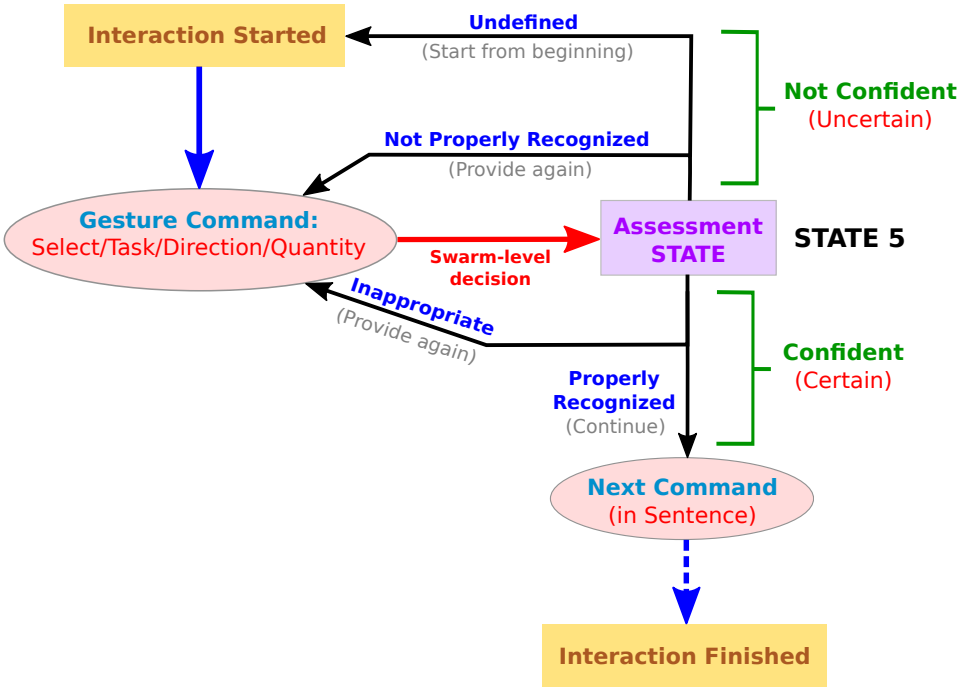


Figure 2.13. Illustration of the assessment state with the four swarm-level recognition outcomes, in relation to the gesture vocabulary in Figure 2.3.

2.3.2.2 Implementation of Self-assessment Feedback

To comprehend the way in which the self-assessment feedback works, an introduction of offline learning methods is necessary, as presented in Section 5.2. We consider the use of a statistical multi-class SVM classifier that is trained using gesture images. Multi-class SVMs aim on maximizing the margin (hyperplane) between K classes in a multi-dimensional feature space. A properly trained SVM classifier can identify (recognize) a given gesture from a predefined set of K

trained gestures. After prediction of a gesture sample, SVMs output a probability vector that corresponds to the K gesture classes.

To implement the self-assessment feedback on robots, a *multi-classifier scheme* is adopted in which different SVM classifiers are trained using the four semantic gesture classes of commands in Section 2.2.2. In particular, every robot in the swarm is equipped with multiple SVM classifiers to recognize (classify) an individual gesture. The gestures in Figures 2.4 and 2.6 comprise of the entire gesture vocabulary, and are used for training the multiple classifiers on-board the robots. Every robot in the swarm is equipped with 6 statistical classifiers:

(a) ***Classifier*** C_{F1}: This is a multi-class supervised classifier trained using the $K = 3$ robot selection gestures in Figure 2.4(a).

(b) ***Classifier*** C_{F2}: A multi-class classifier trained using the $K = 6$ application-specific (i.e., SAR) gestures in Figure 2.4(b).

(c) ***Classifier*** C_{F3}: This is a binary classifier with $K = 2$ classes, in which one class represents spatial directions (see Figure 2.11) and the other class represents numerical quantities (i.e., the 6 finger counts in Figure 2.6 grouped into one class). This classifier determines if a given gesture is a direction (pointed by the arm and hand) or a numeric quantity.

(d) ***Classifier*** C_{F4}: A classifier trained using the $K = 6$ finger count gestures in Figure 2.6, in which finger counts represent integers from 0 to 5.

(e) ***Classifier*** C_{F5}: This classifier is a combination of classifiers C_{F1} and C_{F2} having $K = 10$ classes, and is trained on gestures for selecting and commanding robots (i.e., all gestures in Figure 2.4). This classifier identifies if a gesture represents a command to select robots or a command to perform a task.

(f) ***Classifier*** C_{F6}: A combination of classifiers C_{F3} and C_{F4} that has $K = 7$ classes. One class represents spatial directions and the remaining six classes represent the $K = 6$ finger counts. This classifier determines if a gesture represents a direction or numeric quantity. If the recognition outcome of classifier C_{F6} is a quantity (i.e., a finger count), then classifier C_{F4} is used to identify the number of finger counts.

Using the six statistical classifiers, a *deductive reasoning mechanism* based conditional logic is implemented, which provides output rules and reliably associates a predicted gesture to one of the four swarm-level outcomes: properly recognized, not properly recognized, inappropriate, undefined. The pseudocode in

Appendix C.1 illustrates the working principle of the deductive reasoning mechanism using the six trained classifiers $[C_{F1}, \ldots, C_{F6}]$.[5]

To classify an individual gesture, every robot makes use of a combination of two classifiers in $[C_{F1}, \ldots, C_{F6}]$. Consider that the two classifiers are *classifierA* and *classifierB*, and C_A and C_B respectively represent the number of classes (gestures) in the two classifiers. With the use of two classifiers, the prediction outcome of an individual gesture results in two classification vectors, $classifierA_{out}$ and $classifierB_{out}$, as illustrated in Appendix C.1. From the prediction outcome of $classifierA_{out}$, we identify *probA1* as the gesture class with the highest probability (score), and *probA2* as the class with the second highest probability. Similarly, *probB1* and *probB2* are identified from the prediction outcome of $classifierB_{out}$. Using these four probability measures [*probA1,probA2,probB1,probB2*], two *normalized confidence measures probA* and *probB* are computed. The *average probability difference* P_{avg} between $classifierA_{out}$ and $classifierB_{out}$ is calculated as a simple average: $P_{avg} = (probA + probB)/2$, as illustrated in Appendix C.1.

For any probabilistic values in $classifierA_{out}$ and $classifierB_{out}$, the value of P_{avg} always lies in a closed interval $[0, 1]$. To make reliable swarm-level decisions, we select $P_{avg} = 0.5$ (the middle point of the interval $[0, 1]$) as the baseline between undefined and not properly recognized decisions. When $P_{avg} > 0.5$, the recognition results of $classifierA_{out}$ and $classifierB_{out}$ have a comparatively high difference between each other (i.e., the mean confidence varies a lot between both classifiers), which indicates that the swarm is not confident and the gesture is classified as *not properly recognized*. However, when $P_{avg} <= 0.5$ the mean confidence computed from both the classifiers is significantly low, this indicates that the swarm's decision highly uncertain and the gesture is classified as *undefined*.

2.3.3 Feedback to Guide Interaction

The multi-modal feedback presented in this section is conveyed by robot swarms to guide human operators through the different states in the interaction system (see Figures 2.3 and 2.13). We consider that, conveying the current state of the HSI system to humans during dialogue-based interaction provides a human-friendly user interface. Multi-modal (audio and visual) mechanisms are adopted which includes the use of coloured lights, sounds, and synthetic speech. In context of the HSI scenario considered in this research (see Section 1.2), three types of multi-modal feedback can be conveyed by swarms as presented below.

[5] Depending upon the application requirements and conditions, additional classifiers can be added to produce a more complex reasoning process.

1. Start and End of Interaction Process: During dialogue-based interaction it is desirable for human operators to know when the interaction system has been launched and when the interaction process has ended, as illustrated in Figure 2.13. The starting of the interaction process informs humans to prepare (get ready) for interaction, and the ending of the interaction process signifies that the given commands have been properly recognized and understood by the swarm, and the task associated with the recognized sentence is going to be performed. For instance, when the interaction system starts, all robots switch on their LEDs or beacons. This useful to identify individual robot malfunctions and failures (e.g., low battery, issues with wireless connectivity, sensory-motor problems). Robots with problems do not turn on their LEDs, and this is an effective way to identify faulty robots prior to the start of the interaction process.

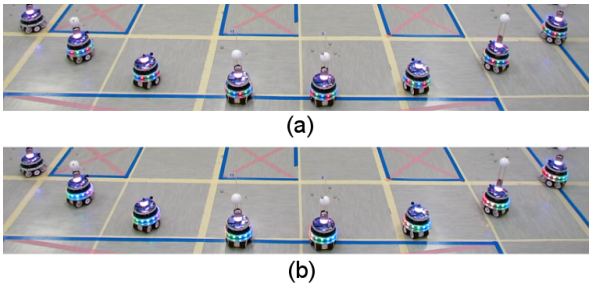


Figure 2.14. Top: LEDs arranged in RGB (red, blue and green) colors to convey that the interaction process has started. Bottom: LEDs arranged in a color spectrum represent that a full sentence of commands has been successfully recognized and the interaction process is going to finish.

To robustly convey the start and end of the interaction process, we consider the use of multi-modal feedback using audio and visual modalities. Combinations of blinking and fading multi-coloured RGB LEDs arranged in different patterns allow humans to easily distinguish between the start and the end of the interaction, as illustrated in Figure 2.14. Complementary to visual feedback, audio feedback is given using a spokes-robot which is located closest to the human. When the interaction system is launched the spokes-robot speaks that the interaction has started. After all gestures (words) in a full sentence have been properly recognized, the spokes-robot speaks the names of all gestures in the sentence (in the order they were recognized), indicating that the interaction has finished.

2. Detecting Human Operators for Interaction: After the interaction system is launched, the next step requires the robot swarm to search for a human operator located within physical proximity (proximal range). Individual robots in a

swarm need to search for the human, because it is likely that the human may not be present in the field of view (FOV) of every individual robot. In principle, human operators are detected by identifying and recognizing the passive markers (i.e., coloured gloves) that they wear (see Appendix A).

Multi-modal feedback is adopted with the use of audio and visual mechanisms. When searching for a human operator, individual robots in a swarm rotate 360° in their current position/location, the coloured LEDs of all robots change to white color and the coloured beacon of all robots blinks in yellow color, as shown in Figure 2.15(a). After a human has been detected, individual robots stop rotating and fixate the FOV of their cameras on the human, and the beacons of all robots stop blinking and change to green color. Audible feedback (sounds and synthetic speech) is provided in conjunction with coloured lights and coordinated movements such that, a spokes-robot speaks that the swarm is currently searching for a human, and it also speaks when a human operator (wearing coloured passive markers) has been successfully recognized.

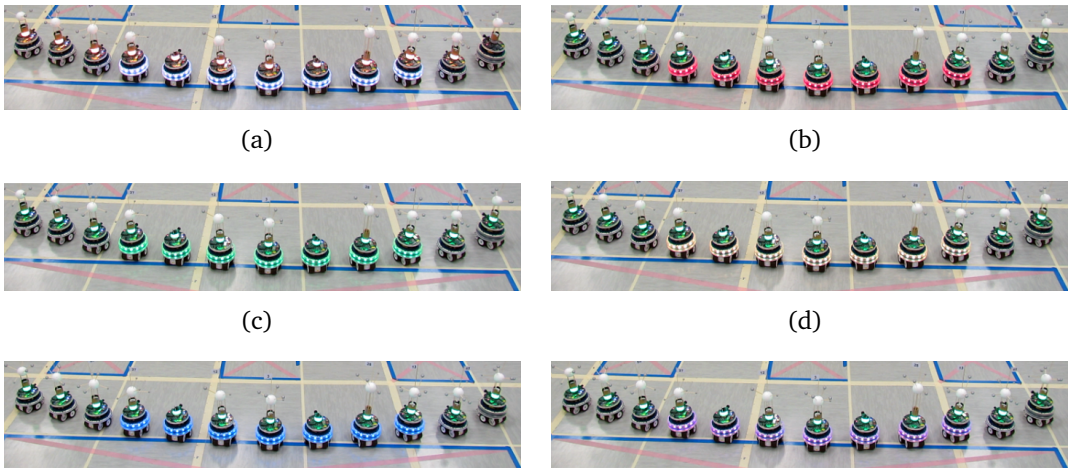


Figure 2.15. A swarm of robots with different LED colors conveying to humans the different interaction states (given in Figures 2.3 and 2.13).

3. Semantic Classes (States) in HSI System: The HSI system consists of 5 interaction states when the self-assessment state is included (see Figures 2.3 and 2.13). Each of these interaction states are communicated to human operators using multi-modal audio and visual mechanisms. Multi-modal feedback allows humans to closely follow the HSI system, and easily interpret the intentions of swarms and the commands (semantic gesture classes) requested by swarms.

To convey visual feedback, the entire swarm takes part in expressing the current interaction state. All robots in a swarm change their LEDs to different colors, which allows humans to easily differentiate between the 5 interaction states. In conjunction with visual feedback, a spokes-robot provides vocal feedback to humans by speaking the name of interaction state (e.g., state for: robot selection, performing tasks etc). As an example, when a robot swarm requests a human to provide a gesture for robot selection, the LED colors of all robots change to red (see Figure 2.15(b)) followed by a spokes-robot that speaks to the human to provide a gesture for selecting robots. Similarly, when a swarm requests a human to present a gesture: to perform a task, to move to direction, as a application-specific parameter (numerical quantity), all robots change their LEDs to green, yellow and blue colors respectively, as shown in Figures 2.15(c), (d) and (e). When a swarm provides self-assessment feedback to humans (see Section 2.3.2) the LEDs change to purple color, as shown in Figure 2.15(f).[6]

2.4 Summary of Experimental Results

The experimental results and discussion of this chapter are presented in Section 6.5. With the low computational capabilities of swarm robots, the results for human-to-swarm communication illustrate that, words (individual gestures) and sentences in the grammar-based gesture language can be efficiently learned by robot swarms, and the interaction time on the Foot-bot platform is reasonable with respect to the swarm size (i.e., the size of the swarm accounts to the amount of information relayed within the swarm network). In the context of swarm-to-human communication, the results indicate that swarms of relatively large sizes (i.e., swarms of 10 robots or more) can reliably classify the two types of not confident (uncertain) decisions: undefined and not properly recognized. Overall the results signify that, larger swarms yield better swarm-level classification performance, however the time required to interact is also slightly higher.

2.5 Summary of Contributions

This chapter presented a bidirectional human-swarm interaction and communication system to fulfil the main goal of this research given in Section 1.4.1. This chapter is divided into two sections, human-to-swarm communication strategies in Section 2.2 and swarm-to-human communication methods in Section 2.3.

[6] A video illustrating visual feedback for different interaction states is available at: http://www.jnagi.net/interaction_states_feedback

For non-verbal human-to-swarm communication, a grammar-based gesture language with a vocabulary of commands is introduced using which humans communicate and provide mission instructions to robot swarms. This vocabulary allows humans to: select spatially distributed robots from a swarm using spatially-addressed gestures, provide potential SAR commands for robot swarms to perform tasks, specify spatial directions for robots to move to, and present numerical quantities (finger counts). This self-contained gesture language provides a number of advantages. Firstly, it serves as a direct interaction protocol between humans and swarms. Secondly, it provides a basic set of encoded control commands. Lastly, the gesture language allows humans to build iconic sentences composed of words (individual gestures) that encode mission instructions.[7]

Within the context of swarm-to-human communication, a swarm coordinated multi-modal feedback language is developed which consists of three types of multi-modal feedback. These feedback provide multiple functionalities: (i) the ability to guide humans during the interaction process, (ii) an intelligent HSI system and a human-friendly interface with basic reasoning capabilities (i.e., to identify mistakes made by humans and errors made by swarms), and (iii) the use of spokes-robots which minimize the use of energy, resources and bandwidth.[8]

In summary, the grammar-based language and the swarm coordinated multi-modal feedback language provide robust and effective dialogue-based interaction and bidirectional communication between humans and robot swarms.

[7] A video demonstrating the grammar-based gesture language is available at: http://www.jnagi.net/gesture_language

[8] A video demonstrating the swarm coordinated multi-modal feedback language is available at: http://www.jnagi.net/swarm_coordinated_language

Chapter 3

Swarm-level Classification of Gestures: A General Protocol

This chapter presents a general protocol that allows robot swarms to sense and recognize (classify) mission instructions issued by human operators, as outlined by the sub-goal in Section 1.4.2.1. The main purpose of this protocol is the swarm-level classification of commands defined in the gesture language (see Section 2.2.1), For robot swarms to classify gesture commands, we introduce a *distributed sensing and cooperative recognition* mechanism, as shown in Figure 3.1.

Operatively, gestures given by humans using passive markers (i.e., coloured gloves) are separated from the image background using *color-based segmentation* (see Appendix A). If a human operator remains still for a short period of time and no *human body motion* is detected during this period, this gives an indication to the swarm that the human is presenting a gesture, as described in Appendix B.

When no human motion is detected, a robot swarm performs *distributed and parallel sensing* of the given gesture, as presented in Section 3.2. After visual information from the gesture has been acquired by the swarm, the *cooperative recognition and decision-making protocol* introduced in Section 3.4 is adopted for the swarm-level classification of the gesture. Every robot in the swarm classifies the given gesture based on its individual viewpoint and produces a local *opinion* regarding the classified gesture. Opinions generated by individual robots are disseminated through the swarm network using multi-hop communication. After individual robots receive opinions from all other robots in the swarm, opinion fusion is performed at the individual-robot level. To efficiently fuse individual robot opinions, decentralized data fusion algorithms are employed for building a *distributed consensus*. After a consensus is built, the swarm-level decision-making mechanism provides the swarm-level recognition outcome for the given gesture.

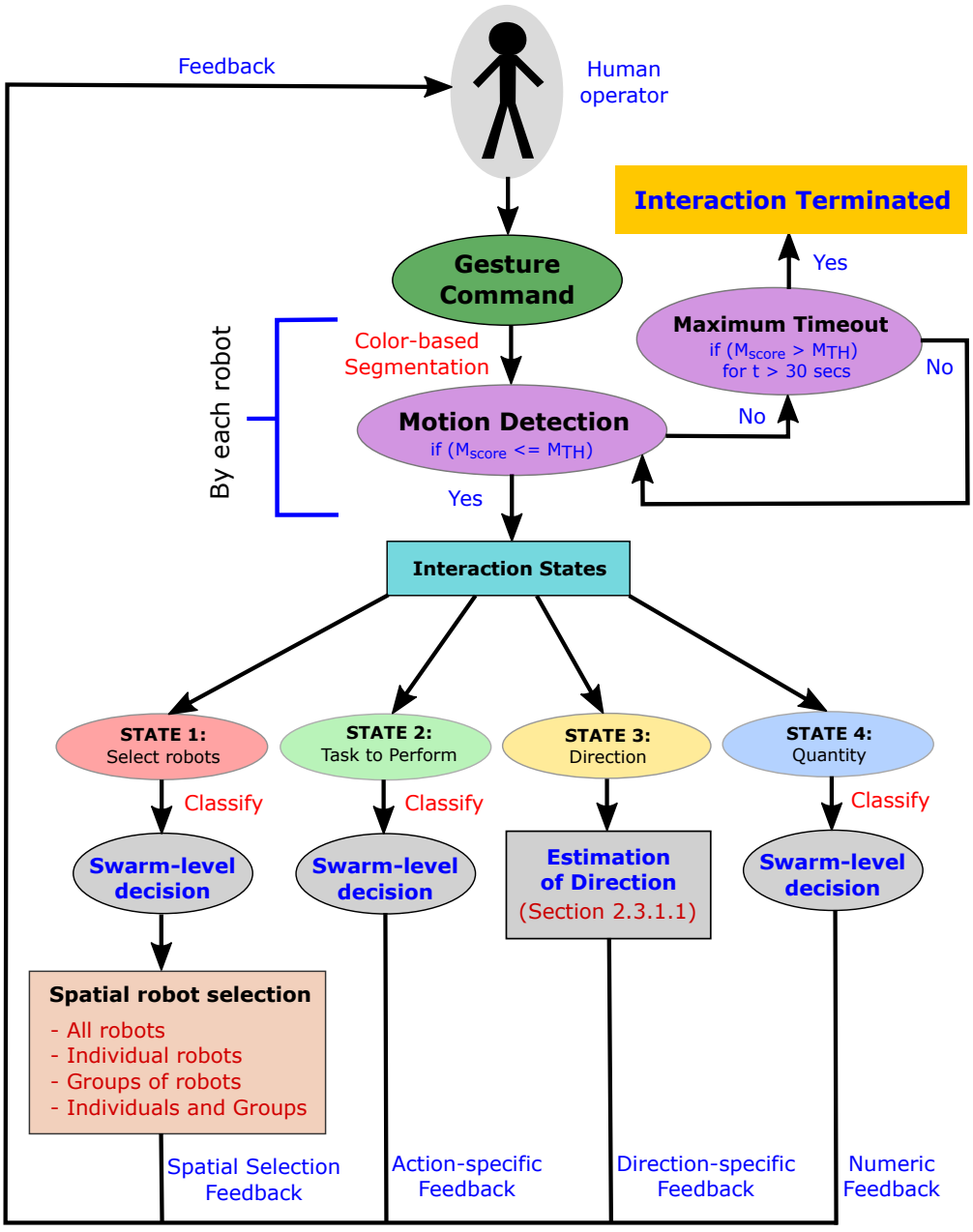


Figure 3.1. Distributed sensing and cooperative recognition (swarm-level classification and decision-making) of commands defined in the gesture vocabulary. The strategies developed for selecting spatially-situated robots from a swarm are presented in Chapter 4.

In this chapter, I gratefully acknowledge the help and assistance of Hung Ngo, Alessandro Giusti and Gianni Di Caro, because without their cooperation and collaboration it would not have been possible to develop the cooperative recognition protocol. The cooperative recognition protocol has been devised by Gianni Di Caro, and has been implemented by Alessandro Giusti. Different approaches for data fusion have been investigated with the assistance of Hung Ngo, including the design and implementation of a general data fusion algorithm for building consensus decisions. My contribution in this chapter has been to support the above mentioned individuals in evaluating the performance of the developed algorithms and techniques (performing experiments).

3.1 Background and Related Work

This section reviews related works in different domains. The covered topics include, distributed and cooperative sensing mechanisms used by multi-robot systems and multi-camera networks to acquire (collect) information from dynamic environments (see Section 3.1.1), and data fusion methods that combine information sensed by multiple sensors/robots (see Section 3.1.2). Consensus algorithms are reviewed for sensor networking applications and multi-agent systems.

3.1.1 Distributed and Cooperative Sensing

Information readily acquired using sensors on-board multiple robots (e.g., cameras) has been formulated as a distributed and cooperative sensing problem by Payton et al. [Payton et al., 2001, 2003] and the research team at the Autonomy Lab of Vaughan [Couture-Beil et al., 2010a; Milligan et al., 2011; Pourmehr et al., 2013a; Monajjemi et al., 2013]. Existing works that deal with problem of sensing information using multi-robot systems make use of centralized or distributed mechanisms. Centralized sensing systems for interaction between humans and multiple robots generally adopt the Kinect sensor [Podevijn et al., 2012, 2013; Gasparri et al., 2012; Lichtenstern et al., 2012] as discussed in Section 4.1.1. Since centralized systems make use of a single sensor which offers good quality of reliable information, distributed sensing systems [Winfield, 2000; Fleck and Straqer, 2008; Perez et al., 2013] use multiple sensors (e.g., multiple robots) situated in different positions (locations) in the environment, in which every sensor (robot) has a different viewpoint to the entity of interest (e.g., gesture).

Visual sensing systems comprise of an integrated network of stationary cameras attached to walls/ceilings (with pan-tilt-zoom capabilities) or mobile camera

networks (e.g., cameras on-board mobile robots), that are capable of processing and fusing images of a scene from a variety of viewpoints. Distributed sensing in multi-camera networks has been used for many applications: monitoring and surveillance [Bramberger et al., 2006; Remagnino et al., 2004], object detection [Sankaranarayanan et al., 2008], pose estimation [Wu and Aghajan, 2008; Aghajan et al., 2008] and gesture recognition [Wu and Aghajan, 2006; Aghajan and Wu, 2007]. Section 5.1.1 provides details on multi-camera networks.

Due to the economical cost of sensing devices in recent times, distributed and cooperative sensing [Dietl et al., 2001; Yang et al., 2007] in multi-robot systems [Gerkey et al., 2003; Gerkey and Matarić, 2004] has gained much importance. We consider that, with the use of data fusion methods (see Section 3.1.2), a swarm of robots equipped with cameras can collectively act as a single sensor.

3.1.2 Data Fusion using Consensus Building Methods

Information sensed distributively and/or in parallel from multiple sensors (e.g., multiple robots), requires fusion. Initial studies on the statistical consensus theory [DeGroot, 1974] investigated techniques for finding agreements between different experts (sensors), which directly relates to the problem of fusing uncertain sensor readings. This framework was later adopted in several other research works [Berman et al., 1989; Benediktsson and Swain, 1992].

Consensus algorithms serve as fundamental tools in wireless sensor networks (WSNs). A swarm of mobile robots that form an ad-hoc network, can distributively and cooperatively sense information from a single entity of interest (e.g., a gesture) while located at different viewpoints in the environment. In decentralized data fusion, a variety of *distributed consensus* strategies exist for sensor networks [Olfati-Saber and Shamma, 2005], multi-agent [Olfati-Saber and Murray, 2004] and multi-robot [Stroupe et al., 2001; Fagiolini et al., 2008] systems. Fusion of observations from multiple static cameras has been adopted in many perception applications: object recognition and image classification [Schriebl et al., 2009; Naikal et al., 2010; Kokiopoulou and Frossard, 2006], pose estimation [Jorstad et al., 2010] and collective map building [Aragues et al., 2012], with multiple viewpoints providing valuable inputs for reconstructing 3D information and to overcome the limits (i.e., occlusions, range) of each sensor. Handling multiple successive observations from every sensor is not a standard feature in consensus algorithms. Due to this reason, Kalman filters that provide dynamic updates to iteratively build consensus decisions [Olfati-Saber, 2007] have been introduced for single-target [Olfati-Saber and Sandell, 2008] and multi-target [Soto et al., 2009] tracking. An overview of common consensus algorithms

and applications is reported in [Olfati-Saber et al., 2007].

For vision-based classification tasks, distributed camera networks have been used to build consensus decisions [Aghajan et al., 2008; Aghajan and Cavallaro, 2009]. A distributed face recognition system was presented in [Kokiopoulou and Frossard, 2010, 2011], in which multiple cameras participated to build a fully-distributed multi-class classifier that took advantage from the joint information contained in face observations acquired from multiple viewpoints. The standard approach for fusing vision-based data from multiple cameras requires computing features from acquired images, which are then aggregated and centrally classified, as presented in [Yu and Nagpal, 2009] for human action detection.

Ensemble learning (or *ensembles of classifiers*) [Polikar, 2006] has emerged as one of the promising approaches for data fusion [Erdem et al., 2005; Parikh and Polikar, 2007] which makes use of heuristics such as: mean, weighted average, and majority voting. As every sensor (robot) suffers a loss after learning the truth of its prediction, in ensemble systems, the weights of the fused decision are updated taking into account the incremental performance of each classifier [Polikar et al., 2001]. Ensemble methods such as, bagging [Breiman, 1996] and boosting [Quinlan, 1996], work by combining relatively *weak learners* and have been extended to online versions [Oza, 2005]. Refer to [Polikar, 2006, 2007] for theoretical insights on ensemble-based learning.

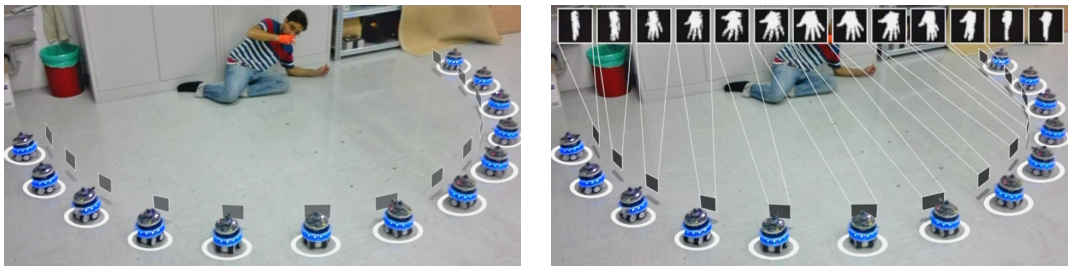
The focus of research in this dissertation lies on the use of robot swarms as distributed and cooperative sensing systems for the collective recognition of gesture commands. For instance, a single robot positioned at a bad viewpoint (i.e., a location in which gestures are not clearly distinguishable even to human observers) faces difficulties in accurately classifying gestures. In a swarm of robots, where some robots maybe located at good sensing viewpoints while others in bad viewpoints, we consider that data fusion mechanisms (see Section 3.4.3) can reliably allow robot swarms to cooperatively recognize gesture commands.

3.2 Robot Swarms as Distributed Sensing Systems

Following the works of Payton et al. and Vaughan in Sections 1.1.2 and 3.1.1, we consider robot swarms as *distributed and parallel sensing systems*, in which the sensing capabilities of all robots in the swarm (i.e., the on-board robot cameras) are focused on the task of recognizing gesture commands given by humans.

When using robot 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 computational power and low-quality sensing

devices (see Section 1.2.1.1). To overcome the limitations of the individual robots, we consider that robots swarms can synergistically act as a *single power augmented sensor* by distributively fusing information acquired from individual robots. Considering a swarm as an array of distributed $R = \{r_1, r_2, \ldots, r_N\}$ robots, where r represents an individual robot and N represents the number of robots in the swarm (swarm size), every robot in $r \in R$ acquires an image i_t of a given gesture at time t, as illustrated in Figure 3.2 using a swarm of $N = 13$ robots. Figure 3.2(b) depicts a distributively sensed gesture from 13 different viewpoints. The 13 segmented gestures (i.e., black and white gesture images) are obtained using the color-based segmentation approach in Appendix A.



(a) (b)

Figure 3.2. Distributed sensing of a gesture using a swarm of $N = 13$ robots. (a): A human presenting a gesture to the swarm. (b): Illustration of a segmented gesture (see Appendix A) acquired by a swarm from 13 different viewpoints.

Although distributed sensing allows robot swarms to quickly acquire a large amount of information (i.e., gesture observations), it however presents a number of challenges. In particular, a large portion of the possible sensing positions do not allow robots in a swarm to acquire good quality of gesture samples due to the, presence of occlusions, angled viewpoints (i.e., bad sensing positions), and excessive distance from the signal source (i.e., the human). To overcome these issues, spatially-aware swarm deployment techniques have been developed in Section 4.3, which enable individual robots to move to sensing positions that offer better views of gestures [Batalin and Sukhatme, 2002; Li et al., 2007].

3.3 Cooperative Recognition Problem Formulation

First, we consider the gesture recognition problem using an individual robot. For a single robot to learn and recognize gestures, a standard vision-based approach is adopted as illustrated in Figure 3.3. A single robot performs: *color-based seg-*

mentation (see Appendix A) to separate gestures from the image background, *feature extraction* to compute meaningful features from the segmented gesture images (see Section 5.2.1), and *supervised classification* to learn and recognize gestures (see Sections 5.2 and 5.3). Experimental results showing the classification performance of a single robot are presented in Section 6.4.4. Since the single robot recognition problem is based on standard techniques (as explained in Figure 3.5), we focus on gesture recognition using a swarm of robots.

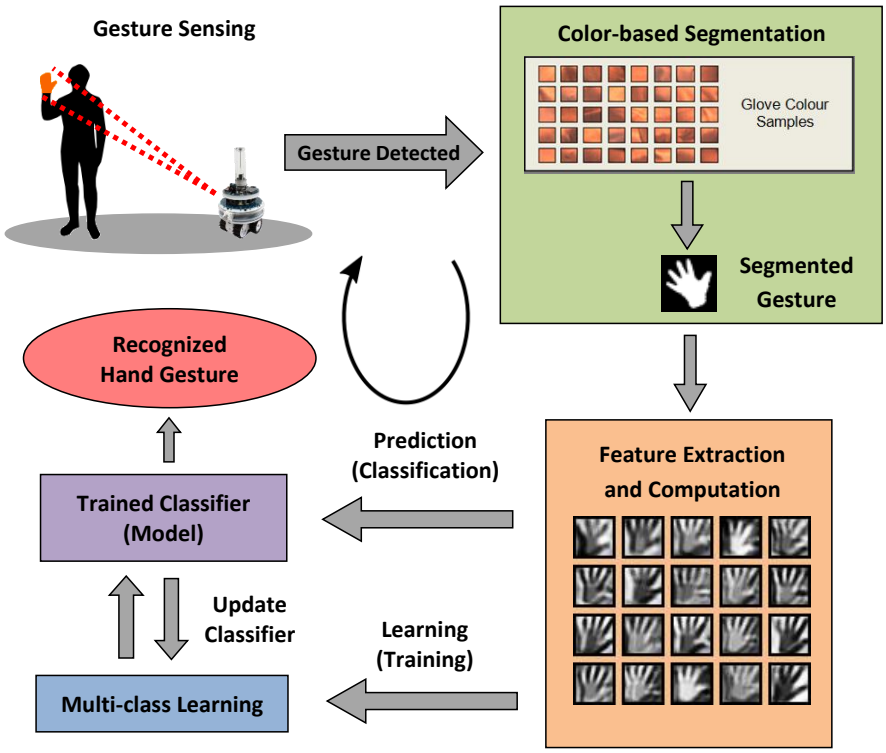


Figure 3.3. A standard vision-based approach for learning and classifying gestures using a single robot.

The cooperative recognition and decision-making process[1] involves: acquisition and processing of visual signals (gestures), formation of opinions, spreading and dissemination of opinions through multi-hop communication, decentralized data fusion using distributed consensus mechanisms, and swarm-level decision-making, as illustrated by the flow of information in Figure 3.4. When the interaction process starts, the robot swarm requests the human to provide a gesture

[1] A video of the developed cooperative recognition protocol using a swarm of robots is available at: http://www.jnagi.net/cooperative_recognition

command. After a gesture is presented, the swarm transitions to the *InformationGathering*() state (see Figure 3.4), in which visual information regarding the gesture is acquired (sensed) and processed by all robots in the swarm. The last state is the *CollectiveDecision*() state, which issues the swarm-level recognition of outcome of the given gesture command.

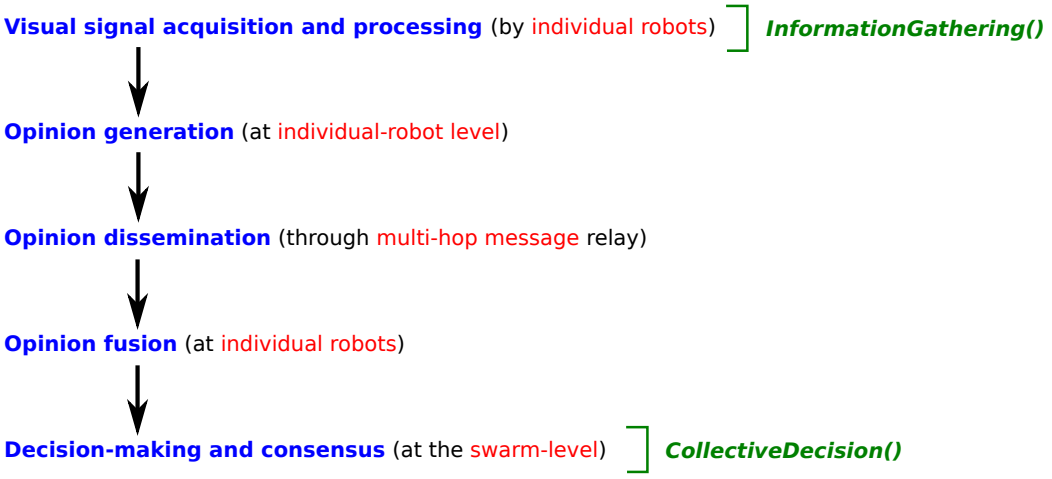


Figure 3.4. Cooperative recognition and decision-making by a robot swarm.

The cooperative recognition and decision-making problem considered in this research is formulated in a general form that requires making minimal assumptions regarding prior knowledge or available infrastructures. Considering the dialogue-based interaction scenario illustrated in Figure 2.2, a human provides a full sentence of commands to a robot swarm, in which the sentence is composed of 2 to 4 words (or individual gestures; see Section 2.2.3). For every gesture (word) given by the human in a sentence, the swarm recognizes the gesture and communicates appropriate swarm-level feedback to the human (see Section 2.3).

We consider the cooperative recognition of a single word (individual gesture) in a full sentence as follows: Let E be a spatially-situated entity of interest, namely a gesture (or more appropriately the segmented hand mask obtained from a given gesture) which represents the object of the cooperative recognition task. The entity E is assumed to be persistent within a finite time period ΔT: it does not change appreciably for (at least) a time interval ΔT. During ΔT, E can be (repeatedly) sensed by a set $R = \{r_1, r_2, \ldots, r_N\}$ of N robots composing the swarm. The robots are scattered in random positions throughout the environment (see Figure 1.3(a)) and may move during ΔT.

Each robot $r \in R$ can acquire observations of entity E using an appropriate robot sensor (i.e., a robot acquires images of a human who is presenting gestures

using its on-board camera). The quality of the sensed observations depends on the positions of the robots (see Section 4.3), both relative to E (e.g., at a distance or a relative angle) and in relation to local environmental conditions (e.g., occlusions, light sources and sounds). Based on individual robot observations, the swarm as a whole needs to classify (recognize) E from a predefined set of K gesture classes (see Section 2.2.2). Every observation requires a constant acquisition time $\Delta T_{\text{obs}} < \Delta T$, and each robot can asynchronously perform repeated observations during ΔT. In this way, information about E can be progressively accumulated and used by every robot to form a local (individual) assessment, hereafter termed as an *opinion*, regarding the gesture class to which E belongs.

Every robot in the swarm is equipped with a *statistical classifier* which supports multi-class classification (see Section 5.2 for more details). This classifier is used to build individual robot opinions regarding E. The classifier is *aspecific*, meaning that, it can operate on gesture observations sensed from any position, albeit with varying accuracy depending on the sensing position with respect to E. Each individual observation is independently classified and results in a posterior probability vector, referred as a *classification vector*: a normalized numerical vector $\mathbf{c} = \{c_1 \dots c_K\}$, $\sum_{i=1}^{K} c_i = 1$ that assigns a *posterior probability* to each of the K possible gesture classes, as discussed in Section 5.2 for Support Vector Machine (SVM) classifiers. To allow individual robots to build an opinion about E's class, classification vectors generated by each robot can be used and combined with the classification vectors produced by the other robots in the swarm.

3.4 Protocol for Cooperative Recognition

As swarm robots are equipped with wireless communication interfaces, a typical robot swarm forms a *mobile ad-hoc network* (MANET) [Royer and Toh, 1999; Tseng et al., 2002; Perkins, 2008]. The global topology of such a network is assumed to be unknown, dynamic, and not necessarily always fully connected. However, certain assumptions on connectivity might be required to ensure the convergence of the consensus, as presented in Section 3.4.5. Robots only make use of *local message broadcasting*, such that, a message sent by a given robot r only reaches a subset of the swarm (i.e., the neighbours of r). The goal of the swarm is to exploit the ad-hoc network to reach, in a fully distributed way, a *distributed consensus* about the class of E which results in a swarm-level classification decision, that balances time constraints with the accuracy of the decision. We investigate the finite-time capability of robot swarms to effectively building cooperative decisions regarding an entity of interest E. Once a decision has been

built, the swarm as a whole can act according to the meaning associated to E and its classification (i.e., perform the task encoded in the recognized gesture).

In context of the cooperative recognition problem statement, we investigate the behaviour of a robot swarm within the time period ΔT, that is, following the triggering at time T_{start} of the recognition process. Triggering is performed by appropriate attention gaining mechanisms (e.g., a hand clap or a whistle blow; see Section 1.2). Following triggering, robots enter (possibly at different times) the *InformationGathering*() state, during which each robot in the swarm collects observations and shares information (opinions) with the other robots regarding E. In the *InformationGathering*() phase, each robot $r \in R$ engages in three parallel activities: sensing, communication and data fusion (see Section 3.4.3). When a robot r has gathered enough statistical evidence in favour of a specific gesture class among the K possible classes, it makes a transition to a *CollectiveDecision*(i) state ($i = \{1 \dots K\}$) and issues a *decision*: from r's point of view, in which E belongs to class i with a specified confidence level. In this way, r's decision and the associated confidence are communicated to the neighbouring robots and then propagated throughout the swarm in multi-hop fashion (see Section 3.4.2), with the aim to reach a *swarm-level consensus*, namely a swarm-level decision. Multiple robots can asynchronously make a transition to the *CollectiveDecision*() state, such that different decisions can exist at the same time within the swarm: conflicts are resolved at the node (i.e., the robot) where they arise, with the *winning decision* having the "highest confidence" being propagated further. Using this strategy, it can be ensured that the whole swarm unanimously converges to a single decision within a finite amount of time (see Section 3.4.4).

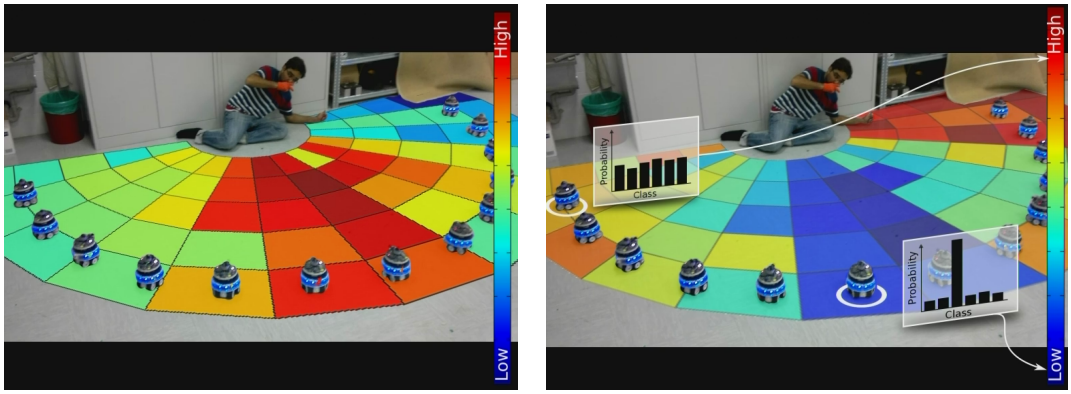
3.4.1 Opinion Formation

In the *InformationGathering*() state, each robot $r \in R$ iteratively acquires and processes observations about the entity of interest E. Each observation consists of a K-dimensional classification vector $\mathbf{c} = \{c_1, \dots, c_K\}$. At any given time t, a robot r can rely on certain number $w_t \geq 0$ of observations $\{\mathbf{c}^1, \dots, \mathbf{c}^{w_t}\}$, gathered since T_{start} (with $t - T_{\text{start}} \leq \Delta T$). The *opinion* $\mathbf{o}^r(t)$ produced by robot r at t is defined as the sum of the classification vectors generated up to t:

$$\mathbf{o}^r(t) = \sum_{f=1}^{w_t} \mathbf{c}^f \tag{3.1}$$

The elements of $\mathbf{o}(t)$ sum to w_t, the total number of observation it combines: $\sum_{i=1}^{K} \mathbf{o}_i^r(t) = w_t$. An individual robot r does not attempt to explicitly

estimate the *reliability* associated to its classifications (opinions), which depends on the position-quality relationship. It is expected that a robot in a bad position will likely produce unreliable opinions. However, a properly trained statistical classifier is expected to correctly reflect in the output classification vectors the ambiguities present in the input data. The classification vectors associated to bad observation positions, on average, have a more flat distribution of values as compared to the classification vectors associated with good viewpoints.



(a) (b)

Figure 3.5. Performance of a single robot using an acquired image dataset of $K = 6$ finger count gestures (see Section 6.2). (a): Position-dependent classification accuracy of gestures from an individual robot placed in $13 \times 5 = 65$ different viewpoints using an offline trained SVM classifier. (b): Entropy computed from the SVM classification vectors in (a).

For instance, the performance of a single robot is illustrated in Figure 3.5 using a dataset of images which comprises of $K = 6$ finger count gestures (see in Section 6.2). Figure 3.5(a) depicts the classification accuracy of an individual robot as a function of robot position (i.e., position-dependent classification performance) using an offline trained SVM classifier (see Section 5.2). The SVM classification performance is the gesture recognition accuracy of an individual robot from $13 \times 5 = 65$ different viewpoints. A flat value probability distribution indicates that none of the K possible classes are associated to a probability which is significantly higher than the others, implicitly indicating that an intrinsic uncertainty lies within the classification vector. To quantify this aspect, the notion of *normalized entropy* for a classification vector $\mathbf{c}$ is computed:

$$H(\mathbf{c}) = -\frac{1}{\log_2(K)} \sum_{i=1}^{K} \mathbf{c}_i \log_2(\mathbf{c}_i) \tag{3.2}$$

where $H(\mathbf{c})$ takes values in $[0,1]$, and is low only when the probability of one or a few classes is much higher than the other classes. For example, in a $K=3$ class scenario, for $\mathbf{c}=[1\ 0\ 0]^T$, $H(\mathbf{c})=0$, while for $\mathbf{c}=[1/K\ 1/K\ 1/K]^T$, $H(\mathbf{c})=1$ (in this case, the vector $\mathbf{c}$ contains no useful information).

We select the entropy of the classification vector as an indirect measure of the reliability or uncertainty of a robot, to empirically define the *impact* that a robot's opinion has in the consensus building process. Figure 3.5(b) illustrates the entropy of an individual robot by using the same experimental setting in Figure 3.5(a). It is observed that, the entropy of the SVM classification vectors at good viewpoints is low, while in bad viewpoints the entropy is high. This measure of impact is used for prioritizing the dissemination of high-impact opinions in communication-constrained environments. Taking into account that an opinion vector needs to be normalized to a value of 1, so it can be treated as an empirical probability distribution, the *impact of an opinion* $\mathbf{o}(t)$ is defined as:

$$I(\mathbf{o}(t);w_t)=w_t(1-H(w_t^{-1}\mathbf{o}(t))), \tag{3.3}$$

where the parameter w_t represents the number of observations $\mathbf{o}(t)$ is based on. The impact function $I(\mathbf{o})$ has the following properties:

- $0\leq I(\mathbf{o})\leq w$, where $I(\mathbf{o})=0$ iff $\mathbf{o}=\{w/K,w/K,w/K\}$: An opinion has zero impact if it is not informative.
- $I(\mathbf{o})=w$ iff all but one component of $\mathbf{o}$ are null: An opinion has a large impact if it results from many informative classification vectors that agree with each other.
- $I(n\mathbf{o})=n\,I(\mathbf{o})$: The opinion resulting from n identical classification vectors has n times the impact of the opinion resulting from a single classification, which quantifies the reasonable assumption that an opinion resulting from multiple agreeing observations should weigh more than an opinion resulting from only one of such observations.

The process of opinion formation is illustrated in Figure 3.6, in which a multi-class SVM classifier is used by every robot to recognize a given gesture, which results in local classification vectors. Opinions originating from robots positioned in bad positions with respect to the sensing of the entity E, have, on average, a lower impact value than opinions obtained from good positions (and an equivalent number of observations). According to the definition of the impact function in eq. (3.3), this is due to the combination of two effects: (i) individual classification vectors resulting from bad positions have on average, larger entropy values,

(ii) the same classification vectors are more contradictory to each other, which results in the equalization of values associated to each class in the opinion vector.

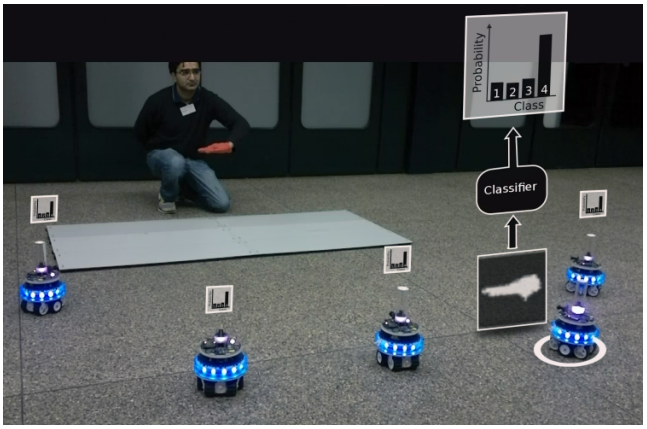


Figure 3.6. The process of *opinion formation*. After individual robots independently classify a gesture, individual opinions, namely classification vectors (associated with the gesture recognition outcome of every robot), are produced.

In spite of the nice characteristics of the impact function, it may happen that a robot r_w repeatedly generates *wrong* classifications, which are all consistent with each other. In this case, the (wrong) opinion of r_w becomes stronger and stronger and has the potential to negatively affect the swarm-level consensus decision. To robustly deal with this issue, efficient means of information dissemination are established: concurrently with r_w all the other robots in the swarm build and circulate their opinions, and when most of robots have settled on a correct decision, they will effectively outweigh the robots with the wrong decision.

3.4.2 Multi-hop Spreading of Opinions

Robots in a swarm continuously update their opinion by iteratively sensing observations and generating classification vectors accordingly. To allow fast and accurate decision-making at the swarm-level, each time a robot revises its current opinion it can use the communication network to rapidly spread the opinion and share it with the other robots in the swarm.

In an ideal communication environment, each robot can effectively broadcast its opinion to the rest of the swarm immediately after the opinion has been updated, so that every robot in the swarm is always informed about the most recent opinion that has been spread through the swarm network. As the flow of data packets between robots is a routing problem, when no external infras-

tructure is present, a *multi-hop* information dissemination protocol is required to spread messages throughout the ad-hoc network, and proper network control strategies are needed to minimize traffic jamming, packet losses and collisions, energy consumption, and to cope with local bandwidth limitations.

The major challenges faced in the design and implementation of communication protocols for robot swarms includes addressing the issues of resource management and bandwidth limitation [Wieselthier et al., 2002], as swarms may have a large number of robots. To address these issues, we introduce a general approach for dealing with robot communications for swarm recognition tasks. This approach provides simplicity and robustness in challenging communication environments, and is able to effectively work in conditions: when relatively large bandwidth is available, and under strict bandwidth-limited conditions.

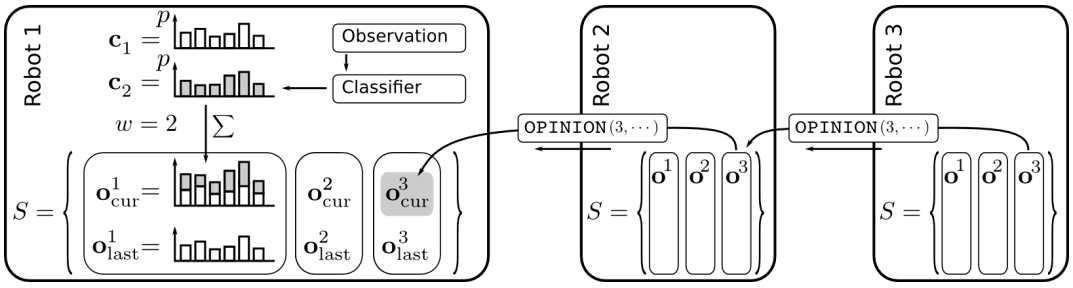


Figure 3.7. Overview of robot data structures for managing opinions. In this example, robot #1 has just acquired and processed a second observation. The updated opinion is saved in it's data structures as $\mathbf{o}^1_{\text{cur}}$. Robot #1 has previously broadcast the opinion obtained from its first classification vector ($\mathbf{o}^1_{\text{last}}$). Now, robot #1 receives from robot #2 a previously unseen opinion of robot #3, which is saved as $\mathbf{o}^3_{\text{cur}}$. Robot #2 previously received the same opinion in a message from robot #3, but it did not reach robot #1.

Robots exchange their opinions using `OPINION` messages, and each robot $r \in R$ maintains a set $S \subseteq R$ of all robots it knows about (i.e., all robots from which at least one opinion has been received in the past), including r itself. For each robot $s \in S$, the following information is maintained in a data structure (i.e., the memory) of robot r:

- $\mathbf{o}^s_{\text{cur}}$, the most recent known opinion of s, and the time t^s_{cur} when it was generated.
- n^s, the number of times an opinion $\mathbf{o}^s_{\text{cur}}$ has been received in an `OPINION` message.

- $\mathbf{o}^s_{\text{last}}$, the last opinion of robot s that robot r has relayed by broadcasting an OPINION message.

The data structures associated to S represent a *robot's view* of the swarm's opinions, as illustrated in 3.7. Not all robots share the same view, as updated opinions do not necessarily reach all robots immediately.

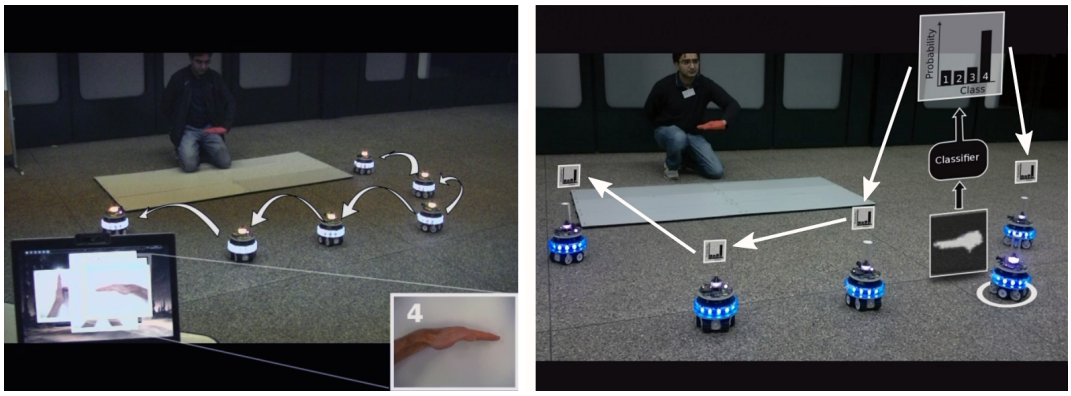


Figure 3.8. The multi-hop spreading and dissemination of opinions (generated by every individual robot) within a swarm of robots.

At initialization time, S is empty. When robot r performs its first sensing, which results in a classification vector $\mathbf{c}^1$, an entry for r is added to S, and its values are set as follows: $\mathbf{o}^r_{\text{cur}} \leftarrow \mathbf{c}^1$, $t^r_{\text{cur}} \leftarrow t$, $n^r \leftarrow 0$, $\mathbf{o}^s_{\text{last}} \leftarrow \mathbf{0}$. Whenever a new observation f is processed and a new classification vector $\mathbf{c}^{\mathbf{f}}$ is generated, the data structure is updated by setting: $\mathbf{o}^r_{\text{cur}} \leftarrow \mathbf{o}^r_{\text{cur}} + \mathbf{c}^f$, $t^r_{\text{cur}} \leftarrow t$, $n^r \leftarrow 0$. Figure 3.8 illustrates the multi-hop spreading of opinions in a swarm of robots.

3.4.2.1 Receiving OPINION Messages

The OPINION messages contain three pieces of vital information: (i) the robot r' originating the message (i.e., the robot who generated the opinion–may not be the last sender of the message), (ii) the time[2] t' at which the opinion was last updated by r', and (iii) the opinion vector itself $\mathbf{o}'$. Such a message is denoted with the notation OPINION$(r', t', \mathbf{o}')$. When robot r receives a message OPINION$(r', t', \mathbf{o}')$, the robot executes the following actions:

- if $r' \in S$, t' is compared to $t^{r'}_{\text{cur}}$:
  - if $t' < t^{r'}_{\text{cur}}$, the incoming opinion is out of date, the message is ignored.

[2]Robots are not synchronized: t' is only used as a robot-specific message *sequence number*.

  - if $t' = t^{r'}_{\text{cur}}$, the opinion is already known, but $n^{r'}$ is incremented.
  - if $t' > t^{r'}_{\text{cur}}$, the opinion is new, then the data needs to be updated: $\mathbf{o}^{r'}_{\text{cur}} \leftarrow \mathbf{o}'$; $t^{r'}_{\text{cur}} \leftarrow t'$, $n^{r'} \leftarrow 1$.

- else, an entry for r' is added to S, and its values are initialized accordingly: $\mathbf{o}^{r'}_{\text{cur}} \leftarrow \mathbf{o}'$, $t^{r'}_{\text{cur}} \leftarrow t'$, $n^{r'} \leftarrow 1$, $\mathbf{o}^{s}_{\text{last}} \leftarrow \mathbf{0}$.

3.4.2.2 Propagating `OPINION` Messages based on Information Importance

The set S and the associated data structures always represent the most up-to-date view of the swarm's state as seen from r. To disseminate this information to the rest of the swarm, robot r *periodically broadcasts* `OPINION`$(s \in S,\ t^{s}_{\text{cur}},\ \mathbf{o}^{s}_{\text{cur}})$ messages. If $s = r$, then the robot is communicating its own current opinion to the rest of the swarm; else, the robot is relaying information from another robot, which was previously received. After an `OPINION`$(s \in S, t^{s}_{\text{cur}}, \mathbf{o}^{s}_{\text{cur}})$ message is sent, the robot keeps track of this by setting $\mathbf{o}^{s}_{\text{last}} \leftarrow \mathbf{o}^{s}_{\text{cur}}$. The frequency for broadcasting `OPINION` messages, and how many opinions to include in each message, depends upon the available bandwidth.

To decide which opinion(s) to include in each local broadcast, if new updates from other robots or from the robot itself are available, the robot evaluates all the data it has locally accumulated and makes an intelligent selection from it. This selection strategy aims to transmit all critical information to allow the swarm reach a robust consensus, while at the same time taking into account bandwidth limitations. The selection is made on the basis of an estimate of the *amount of novel information that the propagation of the opinion will provide to the rest of the swarm*. At this aim, two functions $I_1(s)$ and $I_2(s)$ are designed to heuristically estimate this amount of information, which we introduce as the *information importance* of an opinion. At robot r, priority is always given to the opinion(s) with the highest measure of information importance (ties are resolved randomly):

- $I_1(s) = I(\mathbf{o}^{s}_{\text{cur}} - \mathbf{o}^{s}_{\text{last}})$, where s indicates a robot in S (which includes r) and the function I is defined by eq. (3.3) $I_1(s) = (w^{s}_{cur} - w^{s}_{last})H\big[(w^{s}_{cur} - w^{s}_{last})^{-1}(\mathbf{o}^{s}_{cur} - \mathbf{o}^{s}_{last})\big]$. $I_1(s)$ measures the impact of the information contained in $\mathbf{o}^{s}_{\text{cur}}$ the most up-to-date knowledge that r has about the opinion of s, relative to the information associated to $\mathbf{o}^{s}_{\text{last}}$ the last opinion of s which was included in an `OPINION`$(s, \ldots)$ message broadcast by r. If r has never in the past broadcast an opinion of s, then $\mathbf{o}^{s}_{\text{last}} = \mathbf{0}$ and $I_1 = I(\mathbf{o}^{s}_{\text{cur}})$. If robot r has already propagated $\mathbf{o}^{s}_{\text{cur}}$, then $\mathbf{o}^{s}_{\text{last}} = \mathbf{o}^{s}_{\text{cur}}$, and this consequently results in $I_1(s) = 0$.

- $I_2(s) = (1/2)^{n^s} I_1(s)$ defines the importance of an opinion taking into account both its absolute importance value and the effective need to communicate with neighbours. This is obtained by scaling I_1 with a factor $(1/2)^{n^s}$ which represents a rough estimate of the ratio between the number of neighbouring robots that are expected to have *never* received the opinion. The higher this ratio, the more important it is to include the opinion in the next broadcast message, to guarantee the spreading of information. This estimate is based on the assumption that, each time an opinion $\mathbf{o}^s$ is received by r (with n^s representing the total number of opinions received), the number of r's neighbours that have not received the same information is reduced by half, based on the fact that some neighbours within the same communication range of r may have received the same broadcast. For r's own opinions that have not yet been transmitted, $n^s = 0$ and $I_2(s) = I_1(s)$. Instead, if $s \neq r$, then $n^s \geq 1$ and $I_2(s) \leq I_1(s)$.

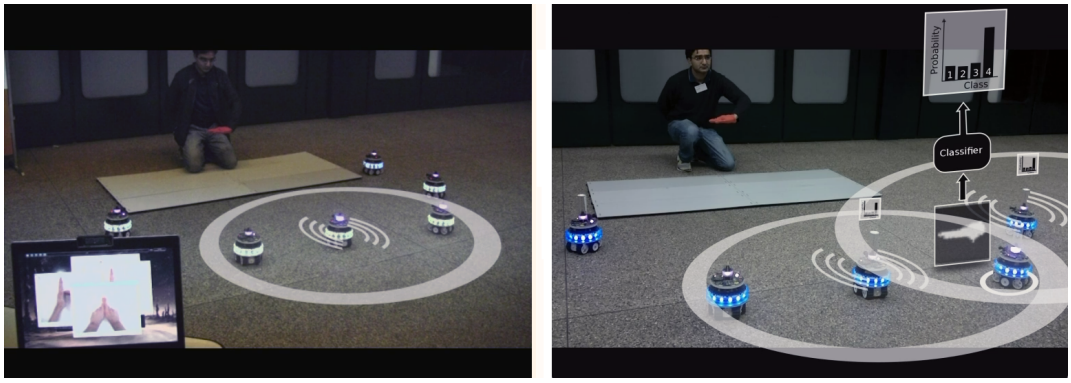


Figure 3.9. The opinion propagation mechanism that takes into account the *information importance* in selecting opinions to broadcast to neighbouring robots. In turn, the neighbouring robots relay and propagate these opinions in a multi-hop fashion to the entire swarm, as shown in Figure 3.8.

The use of importance estimates I_1 and I_2 for opinion selection result in an improved multi-hop message propagation behaviour: since important opinions are associated to higher values of $I_{\{1,2\}}$ compared to the less important ones, they get a higher selection priority and can more rapidly spread throughout the swarm. In this way, the available network bandwidth is effectively used to transmit messages that contain the most novel and important information. However, this does not mean that opinions with low information importance values do not get spread throughout the swarm. Instead, the prioritization mechanism only reduces the *frequency* at which opinions are propagated. Figures 3.8 and 3.9 illustrate the propagation of opinions to neighbouring robots using this strategy.

To explain this further, consider the case of a robot r in a bad sensing position, which will naturally result in classification vectors with high entropy values. As r's opinions $\mathbf{o}^s$ will get a low score from the selected measure of importance, both the robot r itself and other robots that have received these opinions will unlikely include them in their OPINION messages. However, if the classification vectors generated by r are consistent with each other over time, the importance value of r's opinion constantly grows as more observations are acquired and classified. In fact, assuming that r is not able to send its opinion in the past (i.e., $\mathbf{o}^s_{\text{last}} = \mathbf{0}$), $I(\mathbf{o}^s_{\text{cur}})$ increases proportionally to w^s_{cur}. Eventually, as more observations are gathered, this raises the information importance of $\mathbf{o}^s$ to a value that will trigger its inclusion in the next OPINION messages. A similar process happens on the robots where the opinions are relayed. In practice, as long as an opinion is supported by a sufficiently large number of similar observations, it is always propagated with the propagation frequency depending both on the number of observations and the entropy value of the classification vectors (i.e., the lower the entropy, the higher the frequency).

3.4.3 Decentralized Data Fusion

While robots are in the *InformationGathering*() state, they continuously generate and exchange updated opinions, locally accumulating evidence about the entity to be classified. When robot r has enough evidence accumulated in favour of a class i', the robot makes a transition to the *InformationGathering*(i') state and issues class i' as its *local decision*. This triggers the swarm-level decision-making process during which a unanimous consensus needs to be rapidly reached regarding E's class, as discussed in Section 3.4.4. In a fully decentralized approach, no particular robot has the responsibility to trigger the swarm-level decision process. Instead, *any* robot r can initiate such a process at any time, based on the *fusion* of all the pieces of information (i.e., opinions) it has available at that time.

Data fusion at robot r is realized through the use of a *local decision vector* $\mathbf{D}(t)$, which additively combines all the opinion vectors that are available at the current time t, including both the robot's own opinion and the opinions received from the other robots in the swarm:

$$\mathbf{D}(t) = \sum_{s \in S} \mathbf{o}^s_{\text{cur}} \tag{3.4}$$

Every time an OPINION message is received or modified by a robot r, $\mathbf{D}(t)$ is updated as shown in Figure 3.10. Section 3.4.3.1 presents a variety of different data fusions approaches that have been investigated in this research. The data

fusion approach in eq. (3.4) is referred to as the *linear opinion pools* method, which avoids *double counting* of redundant information [Bailey et al., 2012]. This is desirable, as observations are not independent (see [Genest and Zidek, 1986] for more information on fusing multiple probability distributions).

Both the individual components and the L_1 norm $\|\mathbf{D}(t)\|_{L_1} = \sum_{i=1}^{K} \mathbf{D}_i(t)$ of the $\mathbf{D}(t)$ vector grow over time as more opinions are received/generated and more classification vectors (observations) contribute towards each opinion (if the observation process continues indefinitely, $\lim_{t\to\infty} \|\mathbf{D}(t)\|_{L_1} = +\infty$). The value of the kth component of the vector $\mathbf{D}(t)$ is the result of both: the total number of observations that have been taken into account and the sum of the weights associated to class k for each accounted opinion (i.e., how likely is k the true class associated to the entity E). In other words, $\mathbf{D}(t)_k$, $k = 1 \ldots K$ represents the cumulative amount of *evidence* which is available to robot r at time t, to support the assertion that k is E's true class. Figure 3.10 illustrates an individual robot fusing received opinions to form $\mathbf{D}$.

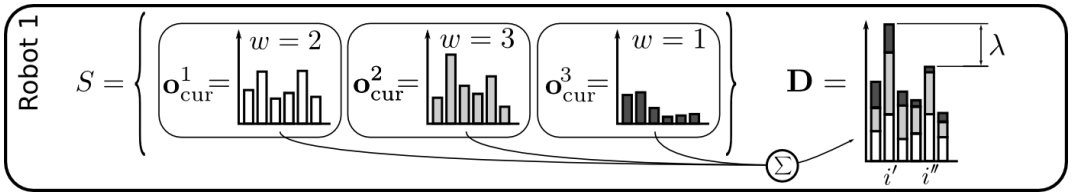


Figure 3.10. The process using which a *local decision* is generated by an individual robot and the illustration of how parameter λ is calculated.

By inspecting $\mathbf{D}(t)$, every robot can make a *local decision* about E's class with class i', that is associated to the highest value among $\mathbf{D}(t)$'s K elements. The decision made by every robot is local and needs to be integrated and compared with the local decisions made by the other robots in the swarm. As different robots might issue different local decisions, the issue which arises is that: when should the local decision be made and spread out to the swarm?

To trigger a local decision, two general criteria need to be satisfied by robot r: (i) the evidence in favour of the selected class i' is significantly larger than the evidence in favour of any of the other $K-1$ classes, (ii) a sufficient amount of information has been collected, such that, it is very unlikely that the decision will change if additional opinions were to be collected. The satisfaction of these two criteria is quantified through the *measure of confidence* $\lambda(t)$, that E's true class is the class i' that has the largest amount of gathered evidence. More specifically, the components in $\mathbf{D}(t)$ with the largest and second largest values, which are i' and i'' respectively, are used to define the measure of confidence:

$$\lambda(t) = \mathbf{D}(t)_{i'} - \mathbf{D}(t)_{i''} \tag{3.5}$$

where λ (see Figure 3.10) quantifies the amount of additional evidence that would be needed in favour of class i'' to let it become the current candidate decision. An even larger amount of additional evidence would be needed for any other class $j \in K,\ j \neq i', i''$. If the value of the local $\lambda(t)$ indicates that enough evidence has been gathered in favour of class i', a robot triggers the generation of a local decision and enters the new state *CollectiveDecision*(i'). The mechanisms of this process are presented in Section 3.4.4. Figure 3.11 depicts individual robots in a swarm fusing opinions to produce a unified swarm-level consensus decision. In the next section, we present a general data fusion algorithm for swarm-level consensus building and provide different data fusion methods that can be easily used with this algorithm.

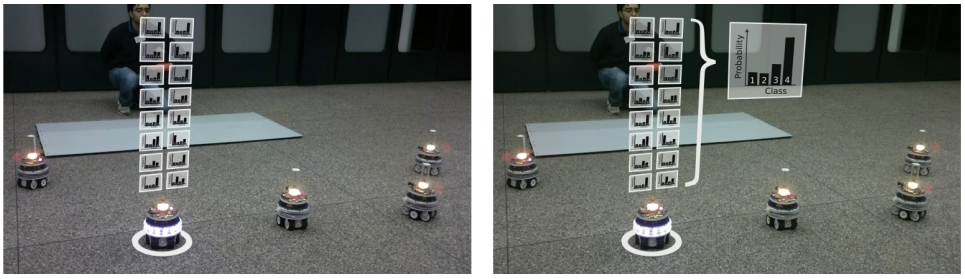


Figure 3.11. Opinions received by an individual robot and the fusion of all received opinions into a swarm-level consensus decision.

3.4.3.1 Different Data Fusion Approaches

To ensure that the data fusion process is robust, effective and scales well with large sized swarms, we introduce a general algorithm that ensures guaranteed convergence in building unified consensus decisions and facilitates the integration of different *data fusion methods*. This algorithm is inspired from *ensemble-based learning* (see Section 3.1.2), in which different data fusion methods have different working mechanisms derived under different assumptions. Some fusion methods may perform well in some domains or some input regions, while others may perform poorly in other domains or regions. The research area most closely related to the cooperative recognition problem in Section 3.3 deals with issuing consensus predictions using *expert advice* [Cesa-Bianchi and Lugosi, 2006]. In this paradigm, multiple classifiers (on-board robots in a swarm) develop an adaptive aggregation rule to intelligently fuse their individual opinions, thereby exploiting the *wisdom of the swarm* [Giusti et al., 2012c; Nagi et al., 2015].

Consider a scenario in which, a human presents a gesture, and the swarm recognizes it and produces a consensus decision. The human presents another gesture and the swarm performs a consensus and recognizes it, and this process repeats for every new gesture given by the human. Every time a gesture is presented by a human, the following events take place: the trained classifier of every individual robot issues an opinion based on its local observation and expertise, after which an overall swarm-level consensus decision made through an aggregation rule. Next, the human reveals the true outcome (i.e., the actual label that corresponds to the given gesture) to the robot swarm using audio or visual signals (e.g., speech, gestures). Lastly, the data fusion aggregation rule is updated based on the performance of the individual opinions.

Algorithm 1: A General Data Fusion Algorithm for Consensus Building

```
//Initialization
{c_{0,r} = 1}_{r=1}^{N} //Consensus weight
//Main interactive learning loop
for t = {1, 2, ..., T} do
    Receive new observation x_t ∈ R^d
    Output prediction probability vector c
    // BEGIN swarm consensus phase
    Exchange c among N robots in the swarm
    // On receiving all c's
    Compute consensus probability vector c_c
    // END swarm consensus phase
    Output consensus label ŷ_t = arg max_{i=1,...,K} (c_c^i)
    Observe feedback y_t ∈ {1, ..., K}
    Update consensus weight {c_{t,r}}_{r=1}^{N}
    Update K-class learning parameters using y_t
end
```

The update rule aims to find the best aggregation in terms of the cumulative prediction mistakes, as quickly as possible. A general approach for data fusion is introduced in Algorithm 1 which has been designed keeping in mind a wide range of data fusion algorithms (for multi-class recognition) that can be easily plugged in. Without loss of generality, Algorithm 1 is executed on every robot r in the swarm. The algorithm works with any supervised classifier whose prediction output is a classification vector $\mathbf{c}$ over the K trained classes (see Section 5.2).

The issued prediction is always different for every robot in the swarm, as every robot observes inputs from different viewpoints.

Initially, each robot r is assigned a unit consensus weight $c_{0,r} = 1$. Given a set of $\{1,\ldots,T\}$ samples issued by a human, as more samples are presented, learned, and classified, robots with more accurate predictions obtain higher weights and robots with more mistakes receive diminishing weights. Swarm-level consensus decisions are built using the weighted prediction vector, $\mathbf{c}_c$, where $\mathbf{c}_c^i = \sum c_{t,r}\mathbf{c}_r^i$ for each class $i = \{1,\ldots,K\}$. After the actual/true label is revealed by the human, each consensus weight is updated based on the current loss of its learner. We consider three data fusion methods that can be easily used in a multi-class setting with Algorithm 1: (i) averaging methods [Giusti et al., 2012c; Nagi et al., 2012b], (ii) frequency counting (counting the number of correct predictions) [Nagi et al., 2014d], and (iii) aggregation rules [Nagi et al., 2015].

Averaging Methods: One of the most simplest and effective data fusion method involves in computing an *element-wise average* over the classification vector probabilities for each class. If $\mathbf{c}^r$ represents the classification vector for robot r in a swarm of N robots, then $\gamma = \sum_{r=1}^{N} \mathbf{c}^r$ represents the consensus outcome as the mean (average) of all the classification vectors (i.e., $\mathbf{c}_r$'s). The *weighted arithmetic mean* (or weighted average) is another approach for tasks that require data fusion. Weighted average is computed by calculating the component-wise weighted sum of all the classification vectors that are use to build the consensus decision. Every classification vector is weighted according to a specific individual weight (i.e., this weight can be an individual robot's recognition performance for all the samples it has predicted so far):

$$\gamma = \arg\max_{\mathbf{c}} \left(\frac{\sum_{i=1}^{N} \left(\mathbf{w}_t^{\mathbf{r}_i} \cdot \mathbf{c}^{\mathbf{r}_i} \right)}{\sum_{j=1}^{N} \mathbf{c}^{\mathbf{r}_j}} \right) \tag{3.6}$$

where $\mathbf{w}_t^{\mathbf{r}_i}$ represents the individual weight of each robot at time t, and $\mathbf{c}^{\mathbf{r}_i}$ is the classification vector of robot r in a swarm of $r = \{1,\ldots,N\}$ robots.

Frequency Counting: This data fusion approach calculates the frequency of correct predictions for each class and weighs the classification vector of each robot $\mathbf{c}_r$ with this frequency (or weight) before performing weighted averaging (i.e., frequency counting is used in conjunction with weighted average). The recognition confidence of every robot at time t is calculated as an *online accuracy rate* $oa_t^{\mathbf{r}_i} = (\sharp^{correct predictions}/_{\sharp samples})$. To obtain a weight for each robot at

time t (which serves as a measure of a robot's online learning performance) a *normalized online accuracy weight* is computed:

$$\mathbf{w}_t^{\mathrm{r}} = \left(oa_t^{\mathrm{r}} \cdot \sum_{j=1}^{N} oa_t^{\mathrm{r}} \right) \tag{3.7}$$

where $\mathbf{w}_t^{\mathrm{r}}$ is scaled between a closed interval $[0,1]$ and $\mathbf{w}_t^{\mathrm{r}}$ can be directly inserted into eq. (3.6).

Aggregation Rules: Based on a set of complex update rules, aggregation methods provide efficient ways to fuse information predicted by multiple classifiers/robots. A variety of aggregation approaches exist that are suitable for use with Algorithm 1. The Weighted Average Algorithm (WdAA; [Kivinen and Warmuth, 1999]) is one such approach that makes use of a multiplicative update rule $c_t = c_{t-1}e^{-\delta_t}$ with δ_t being the prediction loss on the current observation. Other aggregation approaches for data fusion include the Weak Aggregating Algorithm (WkAA; [Kalnishkan and Vyugin, 2005]) and the Aggregating Algorithm (AA; [Vovk and Zhdanov, 2009]). The WkAA employs a time- and loss-dependent update rule $c_t = c_{t-1}e^{-\delta_t/\sqrt{t}}$, while the multi-class extension of the AA algorithm uses a more involved update rule.

3.4.3.2 Dealing with False Positives

During cooperative sensing and recognition, we assume that a few robots are located in good sensing positions. When majority of robots in a swarm produce wrong decisions, this results in the swarm not properly recognizing the given gesture, referred to as a false positive (FP). FP swarm-level decisions occur in situations when the human is positioned in a bad position with respect to the majority of robots. Techniques that have been adopted to eliminate FPs include:

- Exploiting the entropy of generated opinions to identify the relative position of a human with respect to individual robots (i.e., determine if robots are located in good or bad sensing positions), as illustrated in Figure 3.5.
- Opinions generated from robots in bad sensing positions have high entropy (i.e., probability vectors are flat), and these robots receive lower weights in the consensus building process.
- Self-weighting the consensus weight in Algorithm 1. The consensus weight can be diminished if majority of the robots are in bad positions.

3.4.4 Swarm-level Decision Making

To determine if there is enough evidence in favour of class i' with the highest value in $\mathbf{D}(t)$ (see eq. (3.4)) a robot compares its $\lambda(t)$ value (in eq. (3.5)) to a fixed threshold λ_s, which is a swarm-level *prudence* parameter encoding the desired trade-off between the *recognition speed* (small λ_s) and the *classification accuracy* (large λ_s). If the *evidence-trigger* satisfies the condition $\lambda(t) > \lambda_s$ (i.e., *confidence* > *threshold*) as shown in Figure 3.12, the robot can send out its decision indicating i' as the true class for entity E. In real-world scenarios, a robot swarm is required to settle on a decision in a finite (and possibly short) amount of time. To avoid excessively and indefinitely prolonging the classification process, a local decision is automatically triggered at a robot when the time since the robot entered the *CollectiveDecision*() state exceeds the value of the swarm-level parameter T, the *time-trigger*.

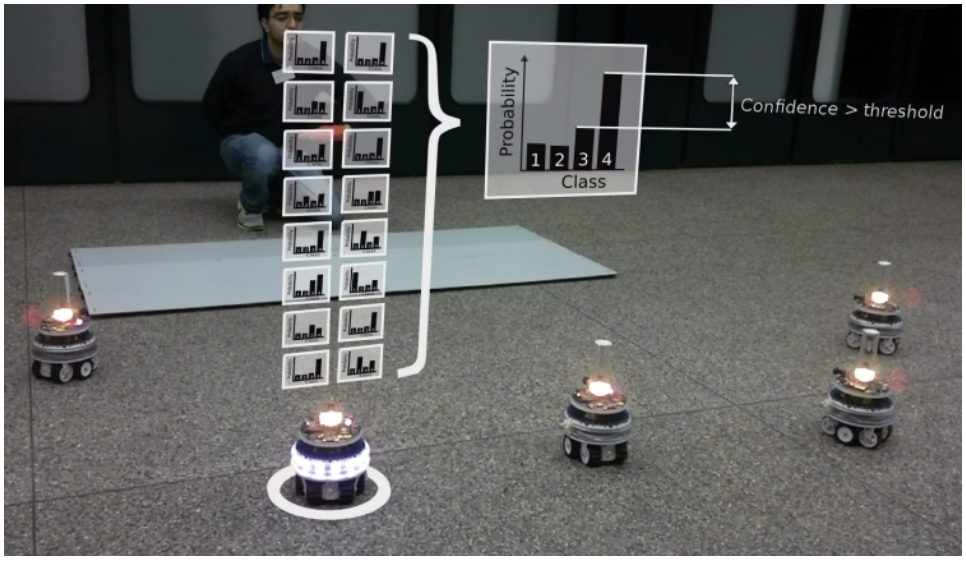


Figure 3.12. Calculation of the *measure of confidence* λ given in eq. (3.5). The confidence represents $\lambda(t)$ and the threshold represents λ_s, such that $\lambda(t) > \lambda_s$.

At robot r, the occurrence of either an evidence-trigger or a time-trigger determines the transition to the *CollectiveDecision*(i'_d) state, and the broadcast of a DECISION(i'_d, λ_d) message to announce to the rest of the swarm the candidate decision i'. The values assigned to parameters λ_s and T shape the response dynamics of the swarm: how *prudent* or *fast* the swarm is in issuing a classification. These parameters can be tuned in accordance to the requirements of the application. For instance, in urgent situations a fast response from the swarm may

be desirable, while in other cases a slow and more accurate response may be required. Setting both parameters to small values determines a fast but potentially inaccurate classification response, and setting T to a large value means to allow the swarm to gather enough statistical evidence before triggering an action.

Multiple robots can asynchronously send out their local decisions at the same time. In any case, once at least one `DECISION` message starts to be spread out, all robots in the swarm are forced to move into the *CollectiveDecision*() state when they receive the message, so that the swarm as a whole can rapidly settle on a *common classification decision*. When robot r receives a `DECISION`(i'_d, λ_d) message, it reacts in different ways depending on its local status and information and based on the comparison between the received confidence λ_d and the confidence λ_r associated to the local best guess.

If r is still in the *InformationGathering*() 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 then further relayed and a transition to the *CollectiveDecision*(i'_d) state is made.
- else, the robot first makes a transition to the *CollectiveDecision*(i'_r) state, and then *overrides* the decision by discarding the incoming message and using its own information to set up a decision; as a result, it broadcasts a new message `DECISION`(i'_r, λ_r).

If robot r is already in the *CollectiveDecision*() state (i.e., this implies that it has already settled upon a decision, i'_r):

- if $\lambda_d > \lambda_r$, the new decision i'_d is adopted: $\lambda_r \leftarrow \lambda_d$ and the message is forwarded further; in the case $i'_d = i'_r$ the robot does not need to change its current decision, while if $i'_d \neq i'_r$, it sets $i'_r \leftarrow i'_d$ to replace its current decision with i'_d which is supported by more evidence, and it makes a transition to the new *CollectiveDecision*(i'_d) state.
- if $\lambda_d \leq \lambda_r$, the robot does not forward the message and ignores the (less confident) decision.

After entering into the *CollectiveDecision*(i'_r) state and adopting a decision, robot r periodically rebroadcasts the `DECISION`(i'_r, λ_r) message to ensure effective information propagation (e.g., robustness to communication losses) and to keep all robots consistently up to date.

According to the mechanisms aforementioned above, decision packets *compete* with each other: decisions assessed with high confidence (i.e., with large λ) override and interrupt the propagation of weakly assessed decisions. It is therefore possible that, a robot after receiving and adopting a decision can change its decision shortly afterwards, if a different different decision with a larger λ is received. The above rules for the propagation of `DECISION` packets ensure that, in a connected network eventually all robots *converge* to the same decision even when different robots issue different decisions at different times, with the "winning decision" being the one with the highest confidence.

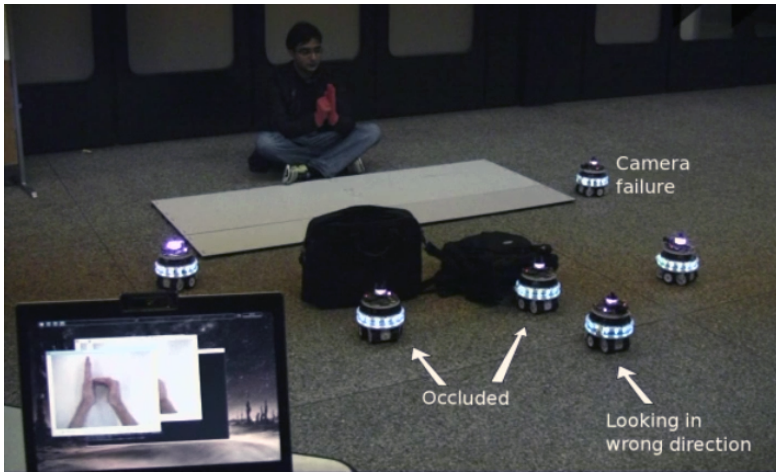


Figure 3.13. Robustness of the cooperative recognition protocol to occlusions, camera failure, and wrong viewpoints.

After adopting a classification decision, a robot needs to start *acting* according to the classified command. However, as a robot's current decision can be overridden due to the propagation of multiple decisions, the robot waits a short time before starting the associated task. The Propositions in Section 3.4.5 provide a technique to define a delay: to wait for the minimum time that guarantees under certain assumptions that no further decision messages will be received.

After a robot swarm has mutually converged upon a swarm-level decision in the *CollectiveDecision*() state, it transitions to the *Action*() state for the winning class i'. In the *Action*() state, the task associated with the recognized gesture is performed. The cooperative recognition and decision-making protocol is robust towards individual robot failures, robots out of communication range and robots with occluded field of views, as illustrated in Figure 3.13 where the two unaffected robots (i.e., the two robots on the extreme left and right of the human) continue to function as normal.

3.4.5 Properties and Complexity

The cooperative recognition and decision-making system implements an epidemic protocol for disseminating temporally-aggregated opinions and is combined with an optimized flooding mechanism for enforcing a swarm-level consensus decision. Each of these steps are analysed separately to derive time and communication complexity bounds.

The key assumption made is that, the network topology and communication model guarantee that a single message propagated by flooding can reach to all nodes in the network within a finite time delay $T_{\max}$. Assuming a dynamic topology and an imperfect communication model, under broad conditions it can be proven that $T_{\max}$ exists, and its value is related to N (which defines the swarm size) and the topology.

Proposition 1: (Time complexity of the opinion dissemination phase): *Assume that T_{max} exists; additionally, assume that each robot in the swarm makes a single observation at time t_0, generates a corresponding opinion, then stops sensing and only relays opinion messages. Under such assumptions, all robots will be aware of all opinions by time $t_0 + 2T_{max}$.*

The above proposition implies that, at time $t_0 + 2T_{\max}$ the average of all opinions known to a robot, is the same as the average of all opinions generated by all robots in the swarm.

Proposition 2: (Time complexity of decision propagation): *Let t_1 be the time when robot r_1 issues the first decision d_1 (of class i_1 and confidence λ_1). r_1 is the first robot entering the CollectiveDecision() state. If T_{max} exists: (i) by time $t_1 + T_{max}$ all robots in the swarm are in the CollectiveDecision(i) state, with the decision i that can take different values for different robots; (ii) by time $t_1 + 2T_{max}$ all robots are in the same CollectiveDecision(i_b) state, where i_b might differ from i_1. No further changes occur afterwards.*

From Proposition 2 the following can immediately be derived:

Proposition 3: *Under the same assumption as Proposition 2, any decision which is not reverted by time $t_1 + 2T_{max}$ will not be reverted any more, as it is the final unanimous swarm decision.*

The proofs for Propositions 1, 2 and 3 are based on the fact that, all communication within the swarm represents instances of either *one-to-all* or *all-to-all* flooding problems, for which strict time bounds can be derived [Topkis, 1989].

3.4.5.1 Memory and Computational Complexity

In terms of memory requirements, the opinion dissemination approach in Section 3.4.2 requires saving two opinion vectors $\mathbf{o}_{\text{cur}}$ and $\mathbf{o}_{\text{last}}$ for each known robot in the swarm. Decision propagation only requires robots to store information about the last adopted/generated decision. Given a swarm of N robots and K classes, the memory complexity is therefore $O(KN)$.

The opinion dissemination stage dominates the computational complexity of the approach. At each iteration of its internal control loop, every robot is required to evaluate each known opinion (which are N at most) and compute its importance (using functions I_1 or I_2 in Section 3.4.2.2) which finally determines the most important opinion. As all known opinions must be summed up to build the local decision vector, the computational complexity of the actions performed by every individual robot is $O(KN)$.

3.5 Summary of Experimental Results

The experimental results of this chapter are presented in Section 6.4 together with the discussion. The results investigate the effect of different parameters and techniques used in the cooperative recognition protocol. The impact of the prudence parameter λ_s is studied relative to the swarm-level classification accuracy, the time taken to reach consensus decisions, and the swarm size. This parameter is beneficial as it controls the trade-off between: the amount of evidence to gather before making swarm-level decisions, and the time taken to reach consensus decisions. The effect of communication losses in small and large sized swarms is investigated with different packet loss rates. Various strategies for opinion selection and aggregation (message prioritization) have been compared, with the proposed strategies providing the best performance. The performance of different data fusion approaches has been compared with respect to the sensing positions of robots and the size of the swarm. The recognition performance of individual robots is investigated with respect to their sensing positions and the opinion impact is characterized based on measures of entropy.

3.6 Summary of Contributions

This chapter introduced a general protocol for the swarm-level classification of gesture commands to fulfil the sub-goal outlined in Section 1.4.2.1. This cooperative recognition protocol provided a robust and scalable solution to address the

distributed sensing and decision-making problem in robot swarms. The protocol together with its major contribution is summarized below.

Visual information from a given gesture is acquired by individual robots in a swarm using distributed sensing mechanisms. After information processing (i.e., color-based segmentation and feature extraction), individual robots classify a given gesture command based on their viewpoints from the gesture. After classification, local opinions are produced by individual robots regarding the recognition outcome of the gesture. These opinions are propagated and disseminated within the robot swarm using multi-hop message-passing algorithms. After every robot receives opinions from the other robots in the swarm, a *distributed consensus* is built at the individual-robot level (based on position-dependent estimates of reliability) for fusing opinions between robot members. The consensus decision provides the swarm-level classification outcome for the given gesture.

A tunable prudence parameter is introduced in the protocol to balance (control) the trade-off between the *time taken to reach consensus decisions* (convergence speed) and the *amount of evidence collected* from multiple gesture observations (gathered by a swarm from a single gesture shown for a duration of time). Using the prudence parameter humans can specify if swarm-level decisions require urgency (i.e., a fast response) or high classification accuracy. This protocol can result in an advantage in many practical and real-world scenarios when computation and communication resources and rapid response time are an issue.

Chapter 4

Swarm-level Coordination: Swarm Understanding of Robot Selection and Spatially-aware Deployment

The swarm-level coordination mechanisms presented in this chapter enable spatially distributed robots in a swarm to understand if they have been selected, and allow individual robots to deploy themselves in dynamic environments for proximal interaction with humans, which is one of the sub-goals in Section 1.4.2.2.

Based on the developed cooperative sensing and recognition system in Figure 3.1, after a swarm-level consensus identifies the spatially-addressed robot selection command (i.e., to select individual robots, groups of robots, or all robots in the swarm) given by the human, a second classification stage is used for the coordination and identification of the selected robots (see Section 4.2). Spatially-aware deployment strategies make use of local mobility rules for the coordinated deployment of heterogeneous swarms (UGVs and UAVs). These mobility rules allow individual robots to move to sensing positions that offer better viewpoints of gestures, and at the same time provide human-relative localization.

In this chapter, I am very grateful to Jacopo Banfi for his help with the spatial selection of robots, and I acknowledge with appreciation Alessandro Giusti and Gianni Di Caro for their support and assistance with the swarm deployment strategies. Jacopo Banfi assisted in the planning and implementation of video demonstrations which illustrate the selection of spatially-situated individuals and groups of UAVs. Alessandro Giusti has implemented the mobility strategy for the deployment of UGVs, and has performed experiments in simulation and with real robots. My contributions include: the design and implementation of the algorithms using which humans select spatially-situated individuals and groups of

robots, performing experiments in simulation and with real robots to investigate robot selection performance, and implementing mobility strategies for human-relative localization and the deployment of multiple UAVs.

4.1 Background and Related Work

This section reviews related works in different domains. The covered topics include, the interaction modalities and techniques used for selecting individuals and groups of robots from multi-robot systems, strategies using which single and multiple robots understand that they have been selected (see Section 4.2), and mechanisms that allow distributed mobile sensing systems to autonomously deploy and move to better sensing positions (see Section 4.3).

4.1.1 Selecting Single and Multiple Robots

To allow individual robots in a swarm to understand that they have been addressed, a substantial amount of work on selecting and commanding robots from a multi-robot system has been investigated by the research team at the Autonomy Lab of Vaughan [Milligan et al., 2011; Monajjemi et al., 2013] using uninstrumented methods. To gain the attention of multiple robots, a *gaze detection* approach (i.e., to capture the movements of the eyes) was developed in [Couture-Beil et al., 2010a,b] that relied on the mechanisms of face engagement. Many other works of the Autonomy Lab have adopted gaze [Pourmehr et al., 2013b,a; Monajjemi et al., 2013] as a means to initiate interaction between humans and multi-robot systems. We consider that, gaze detection is costly and unreliable, especially when using low-quality cameras that are available on swarm robots.

As humans are accustomed to directing other people's attention by pointing towards an entity of interest, recent studies have identified that spatially-addressed *pointing gestures* act as a directive (instruction) for robots [Abidi et al., 2013]. Since pointing behaviours are considered a natural means for machines to recognize human intentions [Sato et al., 2007], machine vision approaches have proven to be useful for learning and recognizing pointing gestures [Nickel and Stiefelhagen, 2007; Martin et al., 2010]. For instance, if a human needs to select a specific number of robots (i.e., individual robots or a group of robots) from a swarm and instruct them perform a task, *spatial pointing gestures* [Payne et al., 2006; Kwon and Gross, 2007; Kurdyukova et al., 2012; Folmer and Morelli, 2012] can allow humans to address spatially-located robots. In [Pourmehr et al., 2013b] pointing gestures were investigated for HMRI, by mounting Kinect sen-

sors onto multi-robot system comprising of two Pioneer P3-DX robots. Based on these findings, we consider that, humans can select individual robots from a swarm simply by pointing at the individual robots (see Section 4.2.2).

As spatial relationships are considered one of the key elements for proximal interaction between humans and multiple robots [Wang et al., 2011; Pourmehr et al., 2013a], the spatial relation between a human and an individual robot is defined by the *angular distance* (θ, d) between the human and the robot. The Autonomy Lab adopted the use of pointing instructions to select and command multiple robots [Couture-Beil et al., 2010a; Milligan et al., 2011; Pourmehr et al., 2013b], and these techniques were inspired from similar existing works [Daily et al., 2003; Skubic et al., 2007; Naghsh et al., 2008; Micire et al., 2009]. More specifically, in [Milligan et al., 2011] a human was required to draw a circle (with the hand) around a desired area, in front of the group of robots to be selected [Pourmehr et al., 2013b]. Tracking a moving hand trajectory and determining if the face is within the hand's circular motion (i.e., if the face is within the circumference of the drawn circle) is a complex recognition process. Alternatively, we consider that the range of the robot group to be selected can be defined as a *spatial cone* between two pointing gestures (see Section 4.2.3).

4.1.2 Deploying Robot Swarms in Dynamic Environments

Being aware of a human's location is a precondition in human-centered computing for applications such as, HRI, human action recognition, and intruder detection. The deployment of mobile robot swarms is directly related to the coverage problem in distributed mobile sensing systems [Liu et al., 2005; Li et al., 2007] (see Section 3.1.1). A number of works have investigated multi-sensor positioning for improving the coverage of mobile sensor networks [Capkun and Hubaux, 2005; Guestrin et al., 2005; Krause et al., 2006] and multi-robot systems [Guinaldo et al., 2013]. Vision-based sensing has been adopted for recognizing humans and their intentions [Teixeira et al., 2010; Espès et al., 2013; Duan et al., 2014]. Positioning of multi-robot systems with on-board cameras has been investigated for cooperative distributed vision [Navarro-Serment et al., 2004] and object recognition [Westell and Saeedi, 2010] tasks. Deployment of aerial robot swarms [Saska et al., 2014; Purohit et al., 2014] has also been explored in recent times. As the aforementioned strategies are problem specific, we introduce *mobility rules* that use the RAB system on the Foot-bots (see Section 1.2.1.1) for the coordinated deployment of robot swarms (see Section 4.3.1).

In recent times, vision-based human localization has been gaining much attention [Gay-Bellile et al., 2010; Kim et al., 2011]. To localize a human from the

viewpoint of multiple airborne UAVs, the Autonomy Lab presented vision-based approaches for Simultaneous Localization and Mapping (SLAM) using marker-based and feature-based methods [Monajjemi et al., 2013, 2014; Pourmehr et al., 2014; Sadat et al., 2014]. Other localization approaches have adopted laser rangefinders [Duan et al., 2014] and ultra-wide band (UWB) technology [Espès et al., 2013]. We consider estimating the angular position ϕ between a human's face and a robot (using robot cameras), which is considered a more natural way of human-relative localization (see Section 4.3.2).

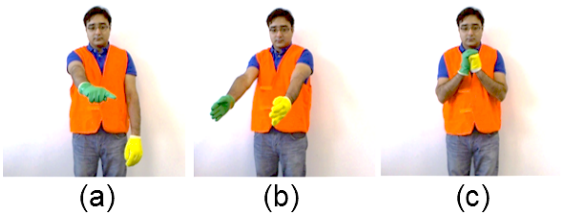


Figure 4.1. Spatially-addressed pointing gestures for selecting robots from a swarm. (a): Individual selection. (b): Group selection. (c): All robot selection.

4.2 Selecting Spatially-situated Robots from a Swarm

For humans to efficiently select spatially distributed robots from a swarm, and for the robots to robustly understand that they have been selected, we consider the use of spatially-addressed pointing gestures. Considering a multi-class classification problem, a supervised classifier (e.g., a SVM; see Section 5.2) is trained offline to classify the predefined set of $K = 3$ spatial robot selection gestures given in Figure 4.1. Every individual robot $\{r_1, r_2, \ldots, r_N\}$ in a swarm is equipped with a multi-class classifier S_{multi}, and each robot uses this classifier to independently predict a spatial gesture by computing $r_{\hat{y}} = \text{argmax}(c)$, where $r_{\hat{y}}$ is the class among the $K = 3$ gestures that has the highest probability in the output classification vector $r_{\mathbf{c}}$. The cooperative recognition protocol in Section 3.4 is used for data fusion, which allows individual robots to reach a swarm-level consensus regarding the presented gesture. The swarm-level decision identifies a presented gesture among one of the $K = 3$ predefined gestures shown in Figure 4.1.

4.2.1 Swarm Understanding of Spatially-addressed Gestures

When a robot swarm recognizes a given gesture as a spatially-addressed command for selecting robots, then the issue for the robot swarm is to understand, to which robot(s) the spatial gesture has been addressed/directed. The gesture

sample that was initially used by the robot swarm to identify that a robot selection command was presented, this gesture sample is used again by the swarm to identify the selected robot(s). In principle, this is achieved using a *cascaded classification* scheme, in which classifier S_{multi} with $K = 3$ gesture classes firstly identifies the robot selection command given by the human (i.e., to select individuals, groups or all robots), and secondly a binary classifier on-board every robot coordinates with the other robots in the swarm to identify the selected robot(s). The binary classifier with $K = 2$ gesture classes is trained to recognize, if the human is pointing towards the robot(s) that need be selected, or if the human is pointing towards the robot(s) that are not desired (unintended) to be selected.

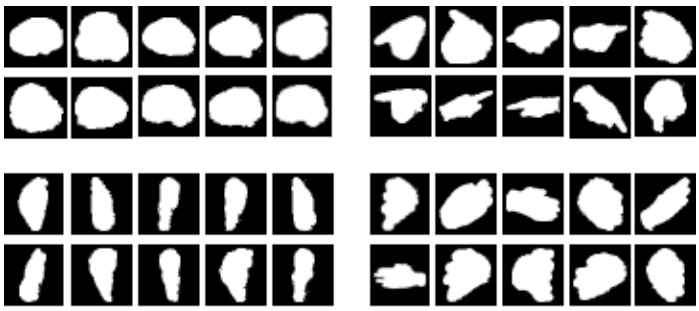


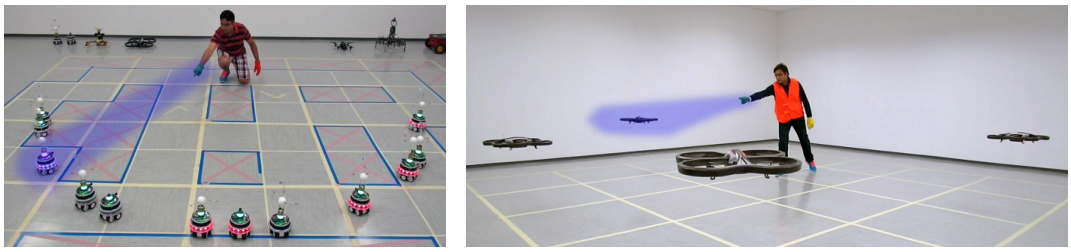
Figure 4.2. Segmented gestures on the top and bottom correspond to the spatial gestures in Figures 4.1(a) and (b) for individual and group selection. Top left: Finger pointing towards an individual robot to be selected (class 1). Top right: Finger pointing in a direction where the individual to be selected is not located (class -1). Bottom left: Hand pointing towards a group of robots to be selected (class 1). Bottom right: Hand pointing in a direction where the group to be selected is not located (class -1).

A binary classification problem with $K = 2$ classes has two output labels $y_i \in \{-1, +1\}$. Class label 1 represents that the human is directly pointing towards the robot(s) that need be selected, and class label -1 represents that the human is pointing towards the robot(s) that are not desired to be selected. Figure 4.2 illustrates the segmented hand masks of the spatial robot selection gestures in Figures 4.1(a) and (b). The segmented gestures for selecting individual robots are shown on the top of Figure 4.2 and the segmented gestures for selecting a group of robots (which use both hands) are given on the bottom of Figure 4.2. All images on the left of Figure 4.2 represent class label 1 and all images on the right represent class label -1. All images on the top of Figure 4.2 are used to train classifier S_{ind} for individual robot selection, and all images on the bottom are used to train classifier S_{grp} for group selection.

The recognition outcome after classifying a gesture with the binary classifiers S_{ind} and S_{grp} results in an output classification vector $\mathbf{c}$ that has 1×2 elements. To identify the selected individuals and group of robots, every robot uses its local results from $r_{\mathbf{c}}$ together with the algorithms presented in the next sections.

4.2.2 Incrementally Selecting Individual Robots

When a human presents the individual selection gesture given in Figure 4.1(a), spatially-located individual robots need to understand if they have been selected. Figure 4.3 illustrates a human providing a one-handed spatial pointing gesture to select an individual robot from a swarm of UGVs (left) and UAVs (right). If the task requires to select more than one individual robot, an *incremental selection* approach is adopted in which additional individuals are selected one-by-one, as illustrated in Algorithm 2 that runs distributively on every robot in the swarm.



(a) (b)

Figure 4.3. Selecting an individual robot from a swarm using the robot selection gesture in Figure 4.1(a). (a): Selecting a UGV. (b): Selecting an airborne UAV.

In incremental selection a human points to an individual robot and selects it, and then the human points to another robot and selects it. This selection process repeats until a condition is satisfied which enforces the incremental selection mechanism to terminate. This condition is specified by humans: if the hand that issues the gesture (i.e., green glove) is higher than the other hand (i.e., yellow glove) the incremental selection process continues until the gesture issuing hand goes lower than the other hand. In practice, the centroid of the green and yellow gloves, $c_{grn}(\mathbf{x}, \mathbf{y})$ and $c_{ylw}(\mathbf{x}, \mathbf{y})$ respectively, are compared with respect to the x-y coordinates of the image plane, as given by lines 23–25 in Algorithm 2.

For individual robots to understand if they have been selected, we adopt a *distributed election* approach which is inspired from existing works [Couture-Beil et al., 2010a,b]. In this approach, every robot that classifies the given gesture as class 1 (i.e., finger directly pointing towards the robot) using its local classifier

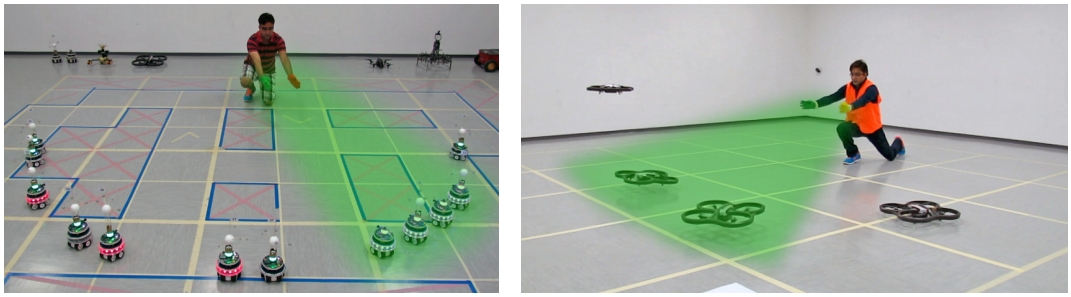
Algorithm 2: Incremental Selection of Individual Robots

```
1  indtotal → 0; //Initialization
2  if (consensus == individual) then
3      repeatloop:
       //Motion detection (Appendix B)
4      for every acquired image do
5          if (M_score < M_TH) then
6              break; //Exit loop and proceed
7          end
8      end
       //Individual selection score
9      if (argmax(r_c) == 1) then
10         Individual score: r_indscr = |c_i1 − c_i2| × (argmax{c_i1, c_i2})
           //For all robots in swarm, broadcast and receive
11         for i = 1 : N do
12             Send ordered pair (r_indscr, r^i_id) to robot r^i
13             Receive (r_indscr, r^i_id) broadcast from ith robot
14         end
15     end
       //Robot with highest score
16     After N − 1 pairs of (r_indscr, r_id) are sent and received by every robot
17     Compute: r^win_ind = arg max_{r_indscr} {(r^1_indscr, r^1_id), ..., (r^N_indscr, r^N_id)}
18     indtotal++; //Total individuals selected
       //Swarm-to-human communication
19     //Check if robot is selected individual
20     if (r^win_ind == r^i_id) then
21         //Change LED colors of robot r^i_id (Section 2.3.1)
22     end
       //Check height of glove
23     if (c_grn(y) > c_ylw(y)) then
24         goto repeatloop;
25     end
26 end
```

$r_{S_{ind}}$, computes an *individual selection score* $r_{indscr} = |c_{i1} - c_{i2}| \times (\text{argmax}\{c_{i1}, c_{i2}\})$ using its local classification vector $r_{\mathbf{c}}$. The value of r_{indscr} and the robot identification number r_{id} are broadcast as an ordered pair (r_{indscr}, r_{id}) in a multi-hop fashion to all robots in the swarm. To identify the selected individual, every robot uses its list of received ordered pairs (including its own pair) and builds a distributed election using $r_{ind}^{win} = \arg\max_{r_{indscr}} \{(r_{indscr}^1, r_{id}^1), \ldots, (r_{indscr}^N, r_{id}^N)\}$, where N represents the number of robots in the swarm that predict the gesture as class 1, and r_{ind}^{win} corresponds to the robot that wins the election. Robot r_{ind}^{win} changes the colors of its LEDs to convey selection feedback to humans (see Section 2.3.1).

4.2.3 Simultaneously Selecting a Group of Robots

When a human operator issues the group selection gesture given in Figure 4.1(b), spatially-situated robots that are located within close proximity of each other need to understand that the human wants to select these robots as a group. Figure 4.4 illustrates a human operator providing a two-handed spatial pointing gesture to select a group of robots from a swarm of UGVs (left) and UAVs (right). To select a group of robots, a *simultaneous selection* approach is adopted.



(a) (b)

Figure 4.4. Selecting a group of robots from a swarm using the gesture in Figure 4.1(b). (a): Selecting a group of 4 UGVs. (b): Selecting a group of 2 UAVs.

In simultaneous selection, robots that are located within the confined spatial area between the two pointing hand gestures are members of a group. In other words, both pointing gestures define the boundaries of a *spatial cone* in which the group of robots is located. For robots to understand that they have been selected as a group, Algorithm 3 runs distributively on every robot in the swarm.

To be robust, the group selection process ensures that only the robots that lie within the spatial cone (defined by the boundaries of both the hands) get selected as a group. To select the members of a group, robots in a swarm go through a

Algorithm 3: Simultaneous Selection of a Groups of Robots

1 //Initialization
2 $grptotal \rightarrow 0$;
3 $min_{dist} = 40$; //Minimum distance
4 $min_{grpscr} = 0.65$; //Minimum score
5 $\vec{r}^{win}_{grp}$<list> $\rightarrow Empty$;
6 **if** ((consensus == group) && ($M_{score} < M_{TH}$)) **then**
//**Select robots located within spatial cone**
7 **if** ((argmax($r_{\mathbf{c1}}$) == "1") *and* (argmax($r_{\mathbf{c2}}$) == "1")) **then**
//**Compute metrics**
8 1. Centroids of both hands: $c_{grn}(\mathbf{x},\mathbf{y})$ and $c_{ylw}(\mathbf{x},\mathbf{y})$
9 2. Euclidean distance between hands:
10 $d = \sqrt{(c_{grn}(\mathbf{x}) - c_{ylw}(\mathbf{x}))^2 + (c_{grn}(\mathbf{y}) - c_{ylw}(\mathbf{y}))^2}$
11 3. Group score: $r_{grpscr} = (\text{argmax}(r_{\mathbf{c1}}) + \text{argmax}(r_{\mathbf{c2}}))/2$
//**Check gloves separation and group score**
12 **if** (($d >= min_{dist}$) *and* ($r_{grpscr} >= min_{grpscr}$)) **then**
13 Add robot r^i_{id} into list $\vec{r}^{win}_{grp}$
14 $grptotal++$; //Total robots in group
15 **end**
16 **end**
//**For all robots in swarm, broadcast and receive**
17 **for** $i = 1 : N$ **do**
18 Send $\vec{r}^{win}_{grp}$ (list of selected robot ids) to robot r^i
19 Receive $\vec{r}^{win}_{grp}$ broadcast from ith robot
20 **end**
//**Swarm-to-human communication**
21 **for** $i = 1$:length($\vec{r}^{win}_{grp}$) **do**
//**Check if robot is in selected group**
22 **if** ($\vec{r}^{win}_{grp}(i) == r^i_{id}$) **then**
23 //Change LED colors of r^i_{id} (Section 2.3.1)
24 break; //Exit loop
25 **end**
26 **end**
27 **end**

two-stage selection process as given in Algorithm 3. In the first stage, all robots that classify both the pointing gestures as class 1 (i.e., the two hands directly pointing towards a robot) using their local classifier $r_{S_{grp}}$, are selected as possible candidates for the group. In the second stage, the candidate robots compute the centroids of both the hands $c_{grn}(\mathbf{x},\mathbf{y})$ and $c_{ylw}(\mathbf{x},\mathbf{y})$, based on their viewpoint from the gesture. The centroids of both hands are used by all candidate robots to compute the Euclidean distance d between the two hands (gloves).

The group score $r_{grpscr} = (\text{argmax}(r_{\mathbf{c1}}) + \text{argmax}(r_{\mathbf{c2}}))/2$ is only computed by the candidate robots. Individual robots positioned at viewpoints which: (i) observe a distance $d >= min_{dist}$ between both hands and (ii) obtain a group score $r_{grpscr} >= min_{grpscr}$, get selected as members of the group, as given by line 12 in Algorithm 3. The group selection parameters min_{dist} and min_{grpscr} are selected using a trial and error approach. Good values have been experimentally found as $min_{dist} = 40$ and $min_{grpscr} = 0.65$. The selected group of robots $\vec{r}^{win}_{grp}$ change the colors of their LEDs to convey spatial selection feedback to human operators (see Section 2.3.1).

4.3 Spatially-aware Swarm Mobility

When robots in a swarm do not know where they are located in the environment with respect to the human issuing the gesture commands, the correct understanding of gestures becomes a challenging task for the swarm. As briefly highlighted in Section 1.2, swarm deployment is not compulsory as humans can move directly in front of the majority of robots to present gestures. However, it is common that some robots in a swarm are unable to detect gestures, due to other robots obstructing their field of view. It is therefore considered better for robots to be fully aware of the location of humans prior to the start of the interaction.

As the quality of sensed information in distributed mobile sensing systems strongly depends on multi-sensor (multi-robot) positioning (see Section 3.1.1), we consider exploiting individual robot mobility to improve the overall sensing coverage and spatial distribution of robot swarms. To develop coordinated mobility strategies for spatially-aware deployment some challenges need to be addressed. Firstly, if the size of the swarm is large positioning becomes difficult, as robots may obstruct the field of view of other robots (i.e., partial occlusions). Secondly, the angular distance (θ,d) between a human and every robot needs to be determined when deciding (selecting) positions that offer good and bad viewpoints of the human who presents the gestures. Estimating the angular distance is dependent on the type of robot platform (e.g., UGV, UAV).

Since distributed mobility strategies have the potential to guide robots to sensing positions that offer good coverage of humans, we consider developing *spatially-aware swarm mobility* strategies that are platform dependent and provide human-relative localization. As briefly discussed in Section 1.2.1, with the Foot-bots (UGVs) the RAB system is adopted (see Section 4.3.1), and with the Parrots (UAVs) the frontal camera is used (see Section 4.3.2).

4.3.1 Coordinated Deployment of UGV Swarms

When a UGV swarm is deployed at random positions in a room and a human operator approaches to proximally interact with the swarm as given in Figure 1.3(a), first the human issues an instruction to gain the attention of a swarm. After Foot-bots in the swarm interpret the human's intention, the swarm prepares for interaction. This involves in reshaping the spatial distribution of the swarm for which the following goals need to be simultaneously achieved:

(a) Sensing observations from good viewpoints in front of the human.

(b) Increasing mutual information collectively gathered by the swarm.

(c) Maintaining wireless connectivity within swarm network.

In general, *goal (a)* implies moving closer to the human or to a new relative position from the human, assuming that the function that relates a robot's position to the sensing quality is (partially) known. The new target position needs to be chosen taking into account the distance from the human, the distance to the neighbouring robots, and to avoid obstructing the field of view of the neighbours.

In *goal (b)*, it is assumed that the correlation among sensed observations acquired by any two robots monotonically decreases as the distance in between the robots increases. Under this assumption every individual robot needs to maximize its distance to its closest neighbours, to increase the amount of mutual information gathered by the swarm. This avoids robots occluding each other and prevents robots from being too close to their neighbours.

In *goal (c)*, robots are required to stay within a given maximal distance from each other in order to maintain a connected network topology (i.e., a minimum network connectivity) from the M closest neighbouring robots. As the network topology must be equivalent to a connected graph (according to the wireless range of the Foot-bot robots), the needed wireless connectivity for multi-hop communication needs to be taken in consideration. With the aim to reach goals (a)-(c), robot swarms are deployed using a set of *local mobility rules*:

Radial Positioning (Rule 1): This rule controls the *radial position* of individual robots. With the goal of gathering better quality observations, robots aim on reaching distance d_o from the target (i.e., human). The optimal distance that needs to be maintained between a human and a Foot-bot is found to be $d_o =$ 1.5m, which maximizes single-robot recognition accuracy.

Tangential Positioning (Rule 2): This rule controls the *tangential movements* of individual robots with the aim to increase the amount of information collectively gathered by the swarm (i.e., observations acquired by any two neighbouring robots do not have a high correlation). This is achieved by maximizing the angular distance (θ,d) of every robot with respect to its neighbours and the human (i.e., human anchors the coordination) which reduces redundancy in the robots' observations (i.e., observations sensed by any two neighbouring robots do not have a high correlation).

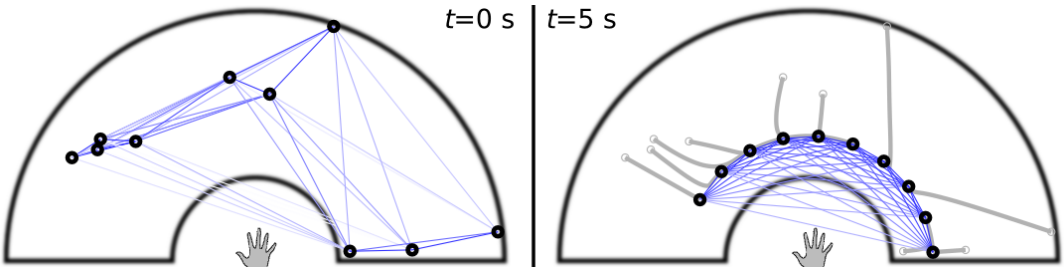


Figure 4.5. Simulated deployment using $N = 10$ robots. Robot trajectories are represented as gray lines. The thin blue lines show distance-dependent link quality between robot pairs (using line-of-sight communication). Left: Initial random positions. Right: Positions after implementing mobility rules 1 and 2.

Unfortunately, *Rules 1 and 2* result in a topology that negatively affects robot connectivity in goal (c) when line-of-sight communication mechanisms are used. When robots are deployed along a semi-circle as shown in Figure 4.5 (right), the closest neighbour to every robot usually occludes robots one step further away, effectively blocking communication links. This results in less efficient multi-hop communication as messages need more hops to propagate to the swarm. To counter this effect, a rule for line-of-sight communications is introduced:

Line-of-sight Communication (Rule 1b): This rule replaces *Rule 1* in the case of line-of-sight communication and tackles the above mentioned issue using a probabilistic approach. The target distance d_0 is defined as $d = d_o + \eta,\ \eta \sim$

$\mathcal{U}(-a,+a)$, in which $\mathcal{U}$ denotes an uniform probability distribution and a is a parameter defining the expected spatial deviation ($a = 0.5$m).

At every control step, every Foot-bot in the swarm estimates its radial and tangential components using the local mobility rules (which are augmented with a line-of-sight obstacle avoidance mechanism) and moves in the direction of the resultant vector. As a consequence of such local rules, UGVs tend to position themselves at regular angular intervals (i.e., in a uniform angular spacing) along a semi-circle centered in front of the human. Figure 4.6 illustrates the spatial distribution of a swarm of 6 Foot-bots before and after deployment.

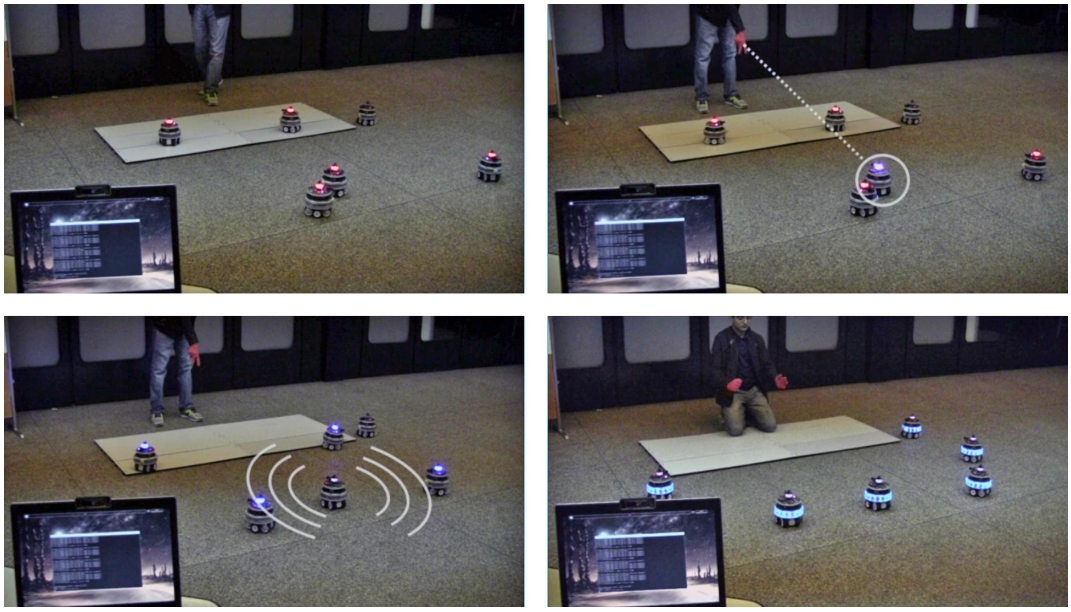


Figure 4.6. Spatially-aware deployment using a swarm of $N = 6$ Foot-bots. Image sequence from top left to bottom right. Top left: A human operator enters a room in which robots are randomly located. Top right: The human provides an attention gaining instruction to the swarm and an individual robot identifies this instruction. Bottom left: The swarm initiates the deployment process. Bottom right: The swarm settles upon its decided spatial configuration.

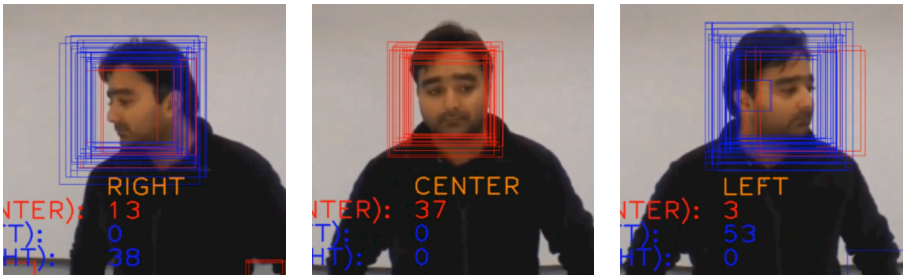
4.3.2 Human-relative Localization of UAV Swarms

For robot swarms to be fully aware of a human's location, the direction in which the human is facing needs to be determined. This is achieved by estimating the pose of the human's face. To estimate the *face pose*, the first step involves face detection. After a face has been detected, relative measures from different

face poses are computed using a *face score system*. The face scores are used determine the angular position ϕ and the distance d between a human and a UAV. The estimated angular distance (ϕ,d) is used as a measure for coordinated UAV deployment and human-relative localization, as presented in the next sections.

4.3.2.1 Face Detection

Face detection is an active topic in computer vision due to its significant role in many real-world applications such as, face recognition, gaze detection, and pose estimation. However, detecting faces is a challenging problem due to the factors associated with illumination conditions, facial expressions, and camera position. In general, face detection provides a normalized and user-centric view of humans. In the context of HRI, face detection identifies the direction and visual orientation in which humans are facing and is functional in determining the relative angular and radial position of humans from the viewpoint of robots.



(a) (b) (c)

Figure 4.7. Face pose estimation using the frontal camera of an airborne Parrot. Face windows (bounding boxes) around the detected face of a human. Identified face poses: (a) Right, (b) Center, (c) Left.

Face detection is performed using the OpenCV implementation of the Viola-Jones face detector [Viola and Jones, 2004], which computes a *face window* (or bounding box) around a detected face. As face detectors are insensitive to small changes in orientation and position, they have the ability to compute multiple face windows around a detected face [Couture-Beil et al., 2010a,b], as shown in Figure 4.7. The number of detected face windows (bounding boxes) represent the *recognition confidence* of a face detection classifier. The larger the number of detected windows, the more confident the classifier is in detecting a face and vice versa. We use the recognition confidence to estimate the face pose.

In practice, the recognition confidence is obtained by setting the OpenCV face detection parameter *minNeighbors* (which specifies the number of neighbours

each candidate window should retain) to the minimal value, which identifies all groups and subgroups of neighbouring windows clustered around the face. The recognition confidence from OpenCV's face detector is used to build a *face score system*, which is introduced in the next section.

As robots can easily loose a detected face and detect false positives (FPs), a Kalman Filter is adopted for smoothing face detection estimates. Using a nearest neighbour strategy with the Mahalanobis distance as an estimate (i.e., computing the covariance of the detected face windows), the best window to use for face tracking is determined. In addition, the face centroid $F_{cent}(\mathbf{x}, \mathbf{y})$ is computed by averaging the centroid of all detected face windows.

4.3.2.2 Face Score System

Inspired by the well known AdaBoost technique [Viola and Jones, 2001] that implements a robust face detector capable of detecting frontal-views of faces, we use a combination of two face detectors for detecting faces from frontal and lateral views. In practice, we consider two pretrained Haar feature-based cascade classifiers (OpenCV face detectors) which are used by every UAV in the swarm. One Haar classifier FC_f is trained on the frontal-views of the face profile and the other classifier FC_s is trained on the lateral-views (left and right views) of the face profile, as illustrated in Figure 4.7. The red coloured face windows show detections from FC_f (i.e., the frontal-view classifier) and the blue coloured face windows are the outcomes of FC_s (i.e., the lateral-view classifier). For every image i acquired by the frontal camera of a Parrot, four relative *face measures* $Fm = \{Fm_f, Fm_{ff}, Fm_s, Fm_{sf}\}$ are computed:

(i) Fm_f (frontal-view): Computed by running classifier FC_f on image i.

(ii) Fm_{ff} (frontal-view flipped): Image i is flipped horizontally 180° to obtain i_h which is processed by classifier FC_s.

(iii) Fm_s (lateral-view): Computed by running classifier FC_s on image i.

(iv) Fm_{sf} (lateral-view flipped): Obtained by running classifier FC_s on i_h.

The four face measures, namely, the frontal-view, frontal-view flipped, lateral-view, and lateral-view flipped, each represent the number of detected face windows (see Section 4.3.2.1). To represent Fm in terms of a meaningful representation, a set of three *face scores* $\{S_c, S_r, S_l\}$ are computed using $S_c = Fm_f + Fm_{ff}$, $S_r = Fm_s$ and $S_l = Fm_{sf}$. The three face scores $\{S_c, S_r, S_l\}$ are used for estimating the face pose in image i. Large values of a face score indicate that the face

is detected with a high confidence and vice versa for low scores. For instance, when a robot is positioned directly in front of a human (i.e., frontal-view), the value of S_c is larger than S_r and S_l. However, if a robot is positioned towards the left or right of the human (i.e., lateral-views), the value of S_l or S_r respectively (depending upon the side) is larger than S_c. If all three scores are below the threshold S_{TH}, we consider that the human is not present in field of view of the robot or the face (human) is too far away to be reliably detected.

The relative distance between a human and a robot is computed as the *average area of face windows*, using $d = \sum_T^i F_A(i)/T$, where $F_A = [Fm_{area}(i), \ldots, Fm_{area}(T)]$ represents the total area of all face windows and $T = (Fm_f + Fm_{ff} + Fm_s + Fm_{sf})$ denotes the total number face windows. Large values of d indicate that the robot is near to the human and small values indicate that the robot is far away.

4.3.2.3 Learning Face Pose Estimates

The face scores and the relative human-robot distance $\{S_c^i, S_r^i, S_l^i, d^i\}$ computed from a single image i are referred to as *face pose features*. The face pose features represent the estimated angular distance (ϕ,d) between a human's face and a UAV. We consider the face pose estimation problem as a supervised learning task, in which the objective is to predict the face pose ϕ in a $[0, 180^\circ]$ semi-circular plane in front of the human operator. To learn and predict face poses, we adopt the Locally Weighted Projectron Regression (LWPR) algorithm [Vijayakumar et al., 2002] which belongs to a family of online incremental learning methods that perform piecewise linear function approximation using regression. As the LWPR is a non-parametric local learning system that makes use of a mixture of locally linear kernalized regressors, it learns a non-linear regression function with 2nd-order online methods and makes use of samples (observations) arriving incrementally over the course of time. For more details regarding the LWPR refer to [Klanke et al., 2008; Glaude et al., 2011; Vijayakumar et al., 2005].

Consider a supervised non-linear regression task in which $\mathbf{x}_i = \{S_c^i, S_r^i, S_l^i, d^i\}$ represents a set of face pose features computed from image i and y_i denotes the face pose ϕ_i (i.e., the target label) in i. Given a set of N training samples as input-output tuples ($\{\mathbf{x}, \ldots, \mathbf{x}_N\}$,$\{\phi, \ldots, \phi_N\}$), the LWPR learns the relationship (mapping) between the face pose features and the face pose for every sample in N. For a set of M testing and validation samples $\{\mathbf{x}, \ldots, \mathbf{x}_M\}$, the task of the LWPR algorithm is to predict the face pose of every testing sample $\{\phi, \ldots, \phi_M\}$. To learn and predict face poses using the LWPR, a dataset of face images is acquired using a swarm of $N = 4$ airborne Parrots (see Section 6.1). The experimental results for face pose estimation using a single UAV are reported in Section 6.6.2.2.

4.3.2.4 Localization of Humans

Using information from the face pose features $\{S_c, S_r, S_l, d\}$ and the predicted face pose ϕ (which are computed from every acquired image), UAVs in a swarm can deploy and localize relative to the location of human operators. Considering a swarm of $R = \{1, 2, \ldots, N\}$ UAVs for $r \in R$, the goal of every Parrot r is to move to a target position that optimizes the swarm's spatial distribution. This is achieved by using the following set of *local mobility rules*:

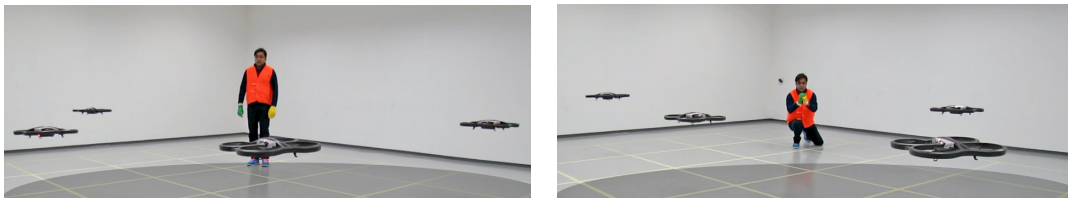
Radial Positioning (Rule 1): With the goal of gathering better quality observations, the *radial position* of each Parrot is selected at angular intervals such that the human is surrounded in a $[0, 180°]$ semi-circular plane. At every control step t, the angular distance (r^t_ϕ, r^t_d) between the human's face and a robot is computed. The angular distance (r^t_ϕ, r^t_d) is used as feedback for the UAV's attitude controller to simultaneously steer the roll and pitch. This allows the robot to manoeuvre itself $180°/N$ degrees apart from the other robots while maintaining an optimal distance $d = 2$m between itself and the human. At the swarm-level, this results in the maximization of the angular distance of every robot with respect to its closest neighbours. This approach works well as long as a minimum distance of $d = 1.5$m is enforced between neighbouring UAVs.

Tangential Positioning (Rule 2): With the aim of increasing the amount of mutual information collectively gathered by a UAV swarm, the predicted face pose (at every control step) r^t_ϕ is used by a UAV to manoeuvre its *tangential position* by steering the yaw angle. As soon as a UAV detects the human's face, it fixates its position in the direction facing towards the human.

Altitude Positioning (Rule 3): When interacting with UAVs that are located on the ground, it is natural for humans to bend their body and tilt their head down. However, when UAVs are airborne, the goal of each UAV is to maintain a fixed altitude with respect to the height of the human operator [Nagi et al., 2014b,a]. To achieve this, at every control step t a Parrot checks it's *elevation component* and maintains a fixed altitude with respect to the human's height. This manoeuvre is performed by constantly minimizing the Euclidean distance between the face centroid $F_{cent}(\mathbf{x}, \mathbf{y})_t$ (see Section 4.3.2.1) and the centroid of acquired image.

At every control step, each Parrot estimates its angular, radial, and elevation components using the local mobility rules and steers its heading in the direction

provided by the resultant vector while maintaining a fixed altitude. The combined application of these rules enables UAVs in a swarm to position themselves along a semi-circle surrounding the human, as illustrated in Figure 4.8.



(a) (b)

Figure 4.8. Spatially-aware swarm deployment and human-relative localization using a swarm of $N = 4$ airborne Parrots.

4.4 Summary of Experimental Results

The experimental results and discussion of this chapter are given in Section 6.6. The results investigate: (i) the performance of the algorithms and techniques using which spatially-situated robots in a swarm understand if they have been selected or not, and (ii) the effect of deployment and mobility strategies on the swarm-level gesture recognition performance. In the context of robot selection, individual and group selection scores are investigated with respect to surrounding non-selected robots. An inversely proportional relationship is found between the selection accuracy and the size of the swarm. In the case of large swarms, selection accuracy decreases with the increase in swarm size. In the case of deployment, mobility strategies reshape the spatial distribution of the swarm and provide better gesture recognition performance compared to situations with no deployment (i.e., when individual and swarm-level sensing positions are not optimized). In addition, different mobility strategies have been compared with respect to the swarm-level recognition accuracy, the swarm size, and the communication capabilities of the Foot-bot platform.

4.5 Summary of Contributions

This chapter presented swarm-level coordination mechanisms to fulfil the subgoal outlined in Section 1.4.2.2. Strategies that allow humans to select spatially distributed individuals and groups of robots from a swarm were introduced with the use of spatially-addressed gestures, and the developed algorithms enable

robots in a swarm to understand if they have been selected or not. For spatial selection, individual robots in a swarm calculate an individual or group score. The individual and group scores provide a relative measure and determine if a human is pointing (providing a spatial gesture) towards an individual robot or a group of robots. Robots that obtain the highest scores are chosen as the selected individual or group member. The distributed mobility strategies enable spatially-aware deployment of heterogeneous robot swarms (UGVs and UAVs) for proximal interaction with humans, and provide human-relative localization in context of the considered HSI scenario (see Section 1.2).

Chapter 5

Learning as a Swarm

For robot swarms to recognize gesture commands given by humans (see Section 3.4), first the robots have to learn the commands defined in the gesture language (see Section 2.2.1), before they can be classified. The focus of this chapter is on the development of *supervised learning* strategies that allow robot swarms to distributively and collectively learn gestures in real-time supervised by humans instructors, which is one of the sub-goals outlined in Section 1.4.2.3.

This chapter is organized as follows. First, we investigate the use of *offline learning* methods by using a dataset of gesture images for training (i.e., building a classifier), as shown in Figure 5.1(a). The red and blue coloured samples (acquired by an individual robot in a swarm) represent a binary (two-class) classification problem. Although offline (batch) approaches are very efficient and provide good learning and classification performance with a swarm of robots, the main limitation of offline methods is that no new knowledge can be added/updated into the trained classifier. In this context, we direct attention towards *online incremental learning* methods as shown in Figure 5.1(b). In online learning, samples arrive incrementally over the course of time and are used for training. Every time new samples arrive they are used for retraining the current classifier. In this way, new knowledge is incrementally updated into the classifier model.

To include humans in the loop of online learning, we introduce the learning strategy in Figure 5.1(c). The scenario depicted in (c) is as follows: a human provides a gesture to a swarm, the swarm classifies the gesture and conveys feedback to the human based on the swarm-level recognition outcome of the gesture. Based on the swarm's feedback, the human provides the swarm with the label of the given gesture sample, which is used by individual robots in the swarm to update their classifiers. The entire process from the human presenting a gesture to the robots updating their classifiers, is termed as an *interaction round*.

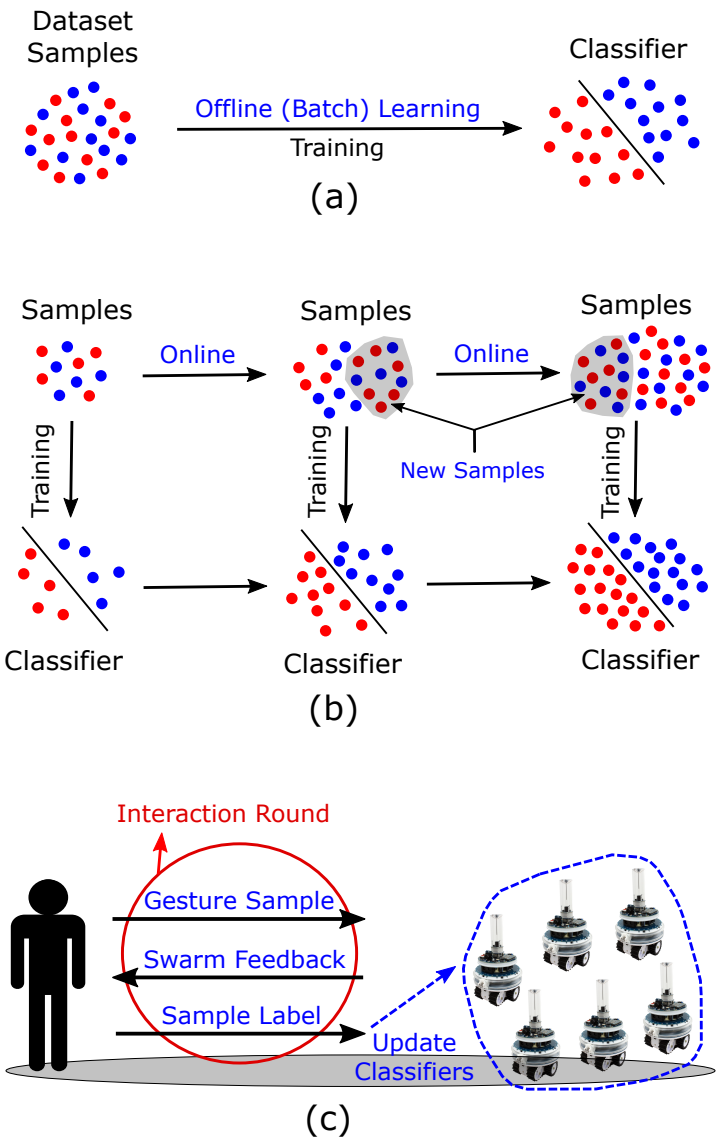


Figure 5.1. Supervised learning strategies for robot swarms for learn gesture commands. A sample represents a gesture image acquired by an individual robot. (a) Offline/batch learning using $K = 2$ gesture classes. (b) Online incremental learning with $K = 2$ classes. (c) Online learning using feedback from humans.

The *cooperative learning* strategy in Figure 5.2 uses the learning approach in Figure 5.1(c). The only difference of Figure 5.1(c) with the cooperative learning is that, individual robots in a swarm share and exchange acquired gesture samples with each other (using information selection and sharing strategies) before building a swarm-level classification decision (see Section 5.4).

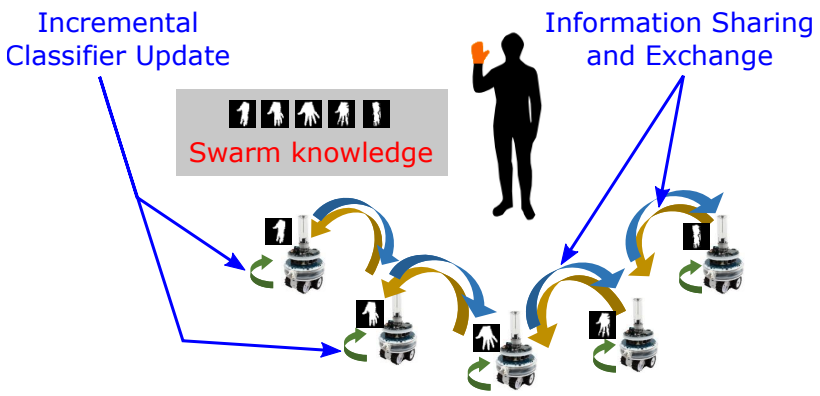


Figure 5.2. Distributed cooperative learning with a swarm of $N = 5$ robots using information selection and sharing strategies. By incrementally sharing knowledge, individual robots learn information sensed by other robots in the swarm.

The research presented in this chapter has been collaborated with a number of colleagues and experts at IDSIA: offline learning with support from Dan Cireşan, Ueli Meier, Jürgen Schmidhuber and Frederick Ducatelle, online learning with guidance from Hung Ngo and Eduardo Feo Flushing, and cooperative learning in collaboration with Alessandro Giusti and Gianni Di Caro. The Convolutional Neural Network (CNN) is one of the adopted offline learning method, and is based on Dan Cireşan's implementation. Frederick Ducatelle assisted with the first implementation of the gesture classification algorithm on the Foot-bot platform, and also supported in testing the algorithm with small sized swarms. The Confidence-Weight Swarm Learning (CWSL) algorithm has been developed in collaboration with Hung Ngo. Strategies for cooperative learning have been formulated by Gianni Di Caro, and have been implemented by Alessandro Giusti. My contributions in this chapter include: investigating the use of different types of learning algorithms (suitable for swarm learning) with advice and collaboration from different experts, and designing and performing experiments.

5.1 Background and Related Work

This section reviews related works in different domains. The covered topics include, distributed learning in wireless sensor networks (WSNs) and multi-camera systems (see Section 5.1.1), supervised online learning strategies that use feedback from humans (see Section 5.1.2), and collaborative training (learning) strategies in multi-classifier and ensemble-based systems (see Section 5.1.3).

5.1.1 Learning in Sensor Networks and Multi-camera Systems

Distributed and decentralized learning in wireless sensors networks (WSNs) has been actively studied in recent years [Mihaylov et al., 2010; Iyengar and Brooks, 2012] with the research group of Poor [Predd et al., 2005, 2006a,b, 2009] providing an in-depth analysis. As it is difficult to assume a specific topology of mobile sensor networks (i.e., topology changes over time), the success of distributed learning depends upon network topology and the learning architecture. The research group of Poor [Predd et al., 2006c, 2009] identified that distributed learning strongly depends on the modelling of communication links [Predd et al., 2009] in network topologies (see Section 5.1.3).

Visual sensing networks also known as *distributed smart cameras* [Rinner and Wolf, 2008; Tabar et al., 2006; Fleck and Straqer, 2008; Song et al., 2011], allow capabilities of distributed visual sensing [Kulkarni et al., 2005; Aghajan et al., 2008; Aghajan and Cavallaro, 2009] with robust and reliable communication between cameras [Kim and Medioni, 2008; Wang et al., 2012a]. One practical application of multi-camera systems is 3D information reconstruction [Peissig et al., 2002] using distributed computer vision techniques [Chellappa et al., 2010; Tron and Vidal, 2011] for tasks that require multi-target tracking and to overcome the limitations of occlusions and range in individual cameras. A broad survey on visual sensing networks is available in [Soro and Heinzelman, 2009].

Distributed learning in multi-camera networks is a task in which information is sensed and learned in parallel from multiple observation points to solve a single joint-task, also referred to as *multi-view learning* [Thomas et al., 2006; Chiu et al., 2007]. Multi-pose or multi-view learning [Muslea et al., 2002; Shan et al., 2006] has been adopted for a number of visual learning tasks: object recognition [Sun et al., 2009; Westell and Saeedi, 2010], hand gestures [Chen, 2009; Kirishima et al., 2010], faces [Jones and Viola, 2003; Li et al., 2004], humans and pedestrians [Zhao et al., 2012; Zheng et al., 2011], and road traffic signs [Timofte et al., 2009]. Existing works for distributed learning in multi-camera networks mainly focus on supervised learning and classification problems [Shan et al., 2006; Kokiopoulou and Frossard, 2010]. A survey on multi-view learning is available in [Sun, 2013; Xu et al., 2013].

Multi-robot systems with on-board cameras have recently started to receive attention: self-organized camera networks for multi-robot deployment [Canedo-Rodriguez et al., 2013] and distributed target sensing and recognition using multiple UAVs [Schwager et al., 2011]. We consider that, a swarm of mobile robots equipped with cameras [Cui et al., 2007; Aghajan and Cavallaro, 2009] can collectively learn and predict gesture commands from a broad visual scene.

5.1.2 Online Incremental Learning

To constantly learn new information from dynamic environments research studies have focused on developing online variants of supervised learning algorithms, which has resulted in widely dispersed learning methods [Raducanu and Vitria, 2007; Steil and Wersing, 2006; Cruz et al., 2008; Kawewong et al., 2011; Kankuekul et al., 2012]. Online styles of learning or *online learning* methods, are well justifiable for multiple robots because remembering past samples [Leistner et al., 2008] is useful for learning problems associated with incremental data acquisition [Argall et al., 2009], which is the case in most real-world environments. In online learning no assumptions regarding the distribution of the data are made, when each datum arrives it is used for training and is discarded—not to be used later in another iteration of the training (learning) process.

The intuitive working principle of online learning algorithms [Crammer et al., 2006; Cesa-Bianchi and Lugosi, 2006] relies on balancing the two conflicting goals in making model updates: the new model should provide a smaller loss on the current training sample while not forgetting much of the information learned from the previous training samples (i.e., a small divergence with the old model). Online learning frameworks [Cesa-Bianchi and Lugosi, 2006; Vovk, 2001; Azoury and Warmuth, 2001] aim to make as few mistakes as possible on any sequence of given samples. For instance, the online learning Passive-Aggressive (PA) [Crammer et al., 2006] algorithm finds a new weight vector that is closest in the ℓ^2-norm sense to the old one, under the constraint that its hinge loss on the current sample is zero. For more insightful discussions on online learning refer to [Giraud-Carrier, 2000; Shen, 1996].

The main advantage of online learning algorithms is that they provide the capability to include humans in the loop of the learning process (see Figure 5.1(c)). We consider that, humans can supervise the learning of robots [Rouanet et al., 2011, 2013]: gestures one after the other are given by humans to robots, and these gestures are incrementally learned in real-time by the swarm. In this way, during online learning human instructors and teachers can select the gesture observation sequence to show to swarms in an adversarial manner. A number of research studies have used human feedback for learning with robots [Austermann and Yamada, 2008; Alissandrakis and Miyake, 2009; Lang et al., 2010; Giovannangeli and Gaussier, 2010]. Prominent works that use human feedback mainly rely on scalable *active learning* strategies [Joshi et al., 2010, 2012].

The 2nd-order online learning algorithms that use linear models have shown to achieve good performance for supervised learning tasks. These algorithms maintain extra *confidence information* in the form of a covariance matrix [Cesa-

Bianchi et al., 2005; Dredze et al., 2008; Cesa-Bianchi et al., 2009] which is used to guide and adapt the direction and magnitude of weight updates during the online learning process. Confidence-Weighted (CW) learning algorithms [Crammer et al., 2008, 2009; Wang et al., 2012b] are good examples of 2nd-order learning methods. CW learning methods belong to a family of linear classifiers that maintain a multivariate Gaussian distribution over the weight vectors and exclusively deal with linearly separable data. CW learning is motivated from the insight that low-confidence feature weights should be updated more aggressively than high-confidence ones. Existing works have shown that CW learning [Crammer et al., 2012, 2013] outperforms other popular learning algorithms such as 1st-order online learning methods [Bottou and LeCun, 2004; Bottou and Bousquet, 2007]. We consider the use of CW learning as it allows to effectively fuse multiple predictions using confidence information (see Appendix D).

5.1.3 Learning with Cooperation and Collaboration

Cooperative and collaborative learning strategies [Dillenbourg, 1999] allow multiple sensors/robots to learn as a team [Panait and Luke, 2005] and require the share and exchange of information during the learning process. Potential works that investigate cooperative learning for visual recognition tasks select data to be learned using margin-based classifiers [Wang et al., 2007; Olvera-López et al., 2010; Liu and Motoda, 2013], which aim to make robust selection among the available (possibly large) samples and address memory and computational costs. Some methods are based on selecting input patterns that are located near to the boundaries of a classifier's decision margin [Wang et al., 2007].

The majority of existing cooperative learning approaches select samples for training [Zechner and Granitzer, 2009; Lopes et al., 2010; Plaza and Plaza, 2010] and contribute in speeding up the performance and the learning rate [Chen et al., 2013; Liu and Motoda, 2013]. A few works have addressed the issue of the online selection of training samples [Garcia-Pedrajas, 2009; Fu et al., 2011]. In [Torralba et al., 2004, 2007] an approach for cooperative learning is presented, in which visual features computed from multi-viewpoint images [Sun et al., 2006] are used as the shared information for object recognition.

Collaborative learning has also been investigated for sensor networking applications [Predd et al., 2006c, 2009]. A distributed approach for collaborative learning put forward by the research group of Poor [Predd et al., 2005, 2006a] decomposes a training set into smaller batches (subsets) and subsequently parallelizes the learning process (by assigning distinct sensors to each of the training batches). If the dataset is not partitioned, three collaborative learning methods

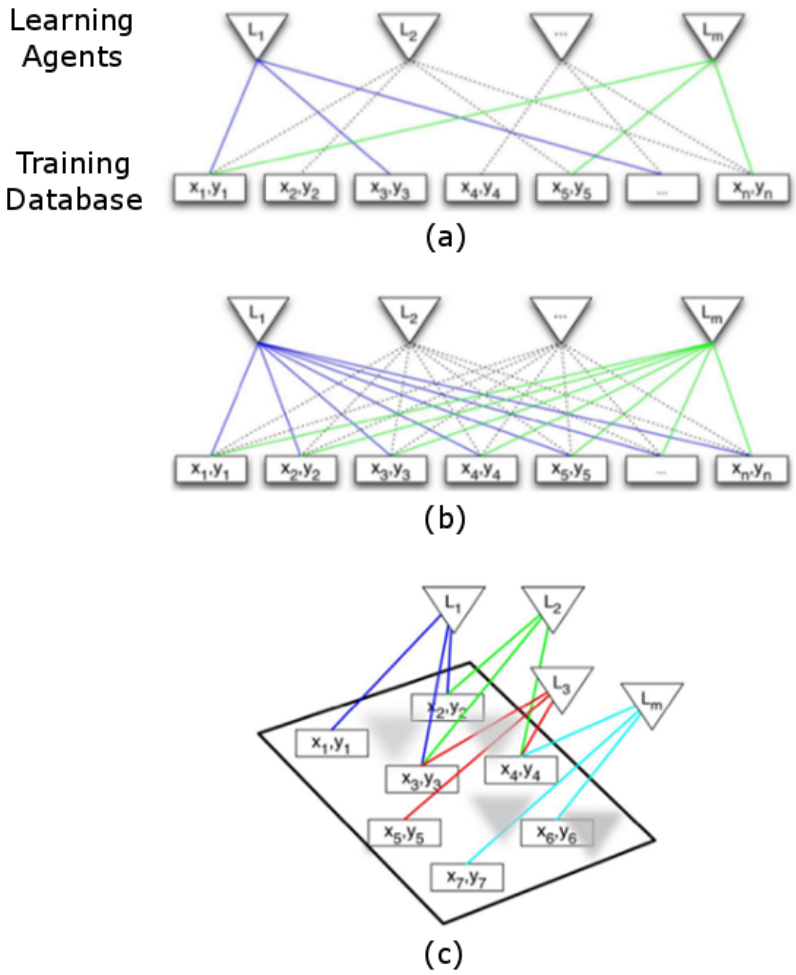


Figure 5.3. Data partitioning methods for collaborative learning [Predd et al., 2009] in sensor networks: (a) A bipartite graph model. (b) A centralized ensemble. (c) Topology dependent learning.

exist [Predd et al., 2009]: (i) a *bipartite graph* in which every sensor has access to some of the training samples in the dataset, (ii) a *centralized ensemble* in which every sensor can access all the samples in the dataset, and (iii) *topology dependent* in which sensors only access samples that are nearby (in close proximity) with respect to topological characteristics. Figure 5.3 illustrates the three collaborative learning methods in which the learning agents represent the sensors. If the dataset is partitioned, an ensemble of classifiers (see Section 3.1.2) is used with *public database* (training set) that is available to every sensor [Predd et al., 2006a,c], as illustrated in Figure 5.4. In this setting, every sensor learns only local knowledge (i.e., partial-view information) of the entire training set.

In our case, the above mentioned approaches are computationally expensive when using robots that have limited on-board computational capabilities. We introduce a heuristic approach for cooperative learning in robot swarms which is based on information selection and sharing strategies (see Section 5.4).

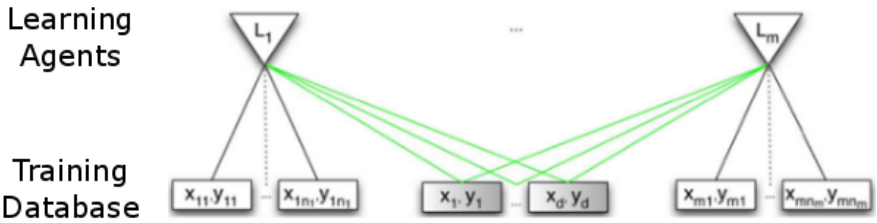


Figure 5.4. An ensemble with a public database [Predd et al., 2009] in which every sensor has access to all samples in the training set.

5.2 The Starting Point: Offline Learning

The most simplest and straightforward approach in which a swarm of robots can learn information is by using an *offline learning* method, also known as batch learning. Prior to the learning phase, a large dataset of gesture images is gathered by a swarm of robots from multiple viewpoints, as presented in Section 6.2. As every robot in the swarm is equipped with a local supervised classifier, every robot uses a subset of images from this dataset to train its local classifier. In this way, every robot in the swarm independently learns gesture commands from multiple viewpoints. Individual robots with classifiers trained offline use the cooperative recognition protocol in Section 3.4 for the swarm-level classification of gestures.

We consider the use of two supervised classifiers: Convolutional Networks (CNNs) [Nagi et al., 2011; Cireşan et al., 2011, 2010] and Support Vector Machines (SVMs) [Giusti et al., 2012c; Vapnik, 2013]. CNNs are based on neural architectures and SVMs are statistical margin-based classifiers. The CNN is a *deep learning* framework that possess the capability to compute features from images and perform classification. Figure 5.5 illustrates the Max-Pooling Convolutional Neural Network (MPCNN) and the free parameters used for training in each layer. For a segmented gesture image in Figure A.1, the activations of the first convolution and max-pooling layers (C1 and MP1 respectively) for 20 maps, are illustrated in Figures 5.6(a) and (b) respectively. As activations pass through the deep feedforward network they are down-sampled in the following layers such that a feature vector of $F = 300$ elements is obtained in the fully-connected layer, which is used for classification in the last layer, as shown in Figure 5.5.

Due to the fact that, statistical margin-based classifiers such as SVMs are not capable of computing features from images, we introduce feature extraction methods in Section 5.2.1 that can be used for multi-class classification with SVMs. The classification performance of SVMs strongly depends on the feature extraction method and the quality of the learned features (i.e., different types of feature computation methods may result in a better or worse SVM performance).

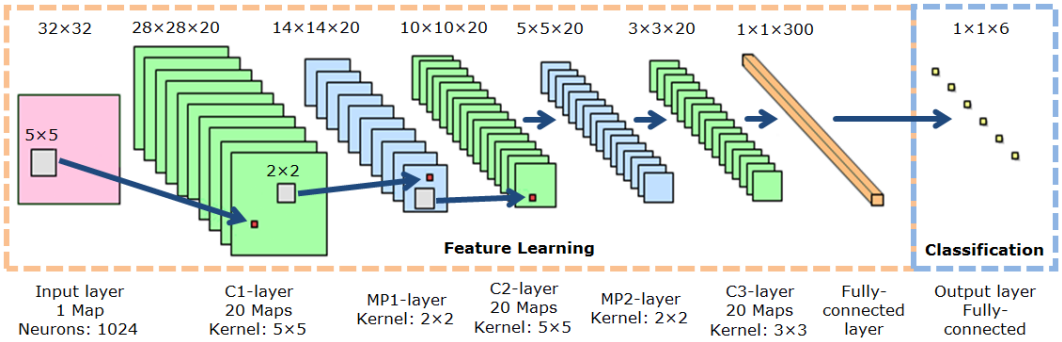


Figure 5.5. The architecture of a deep learning Max-Pooling Convolutional Neural Network (MPCNN) using alternating convolutional and max-pooling layers. Parameters used for MPCNN training are shown in every layer of the network.

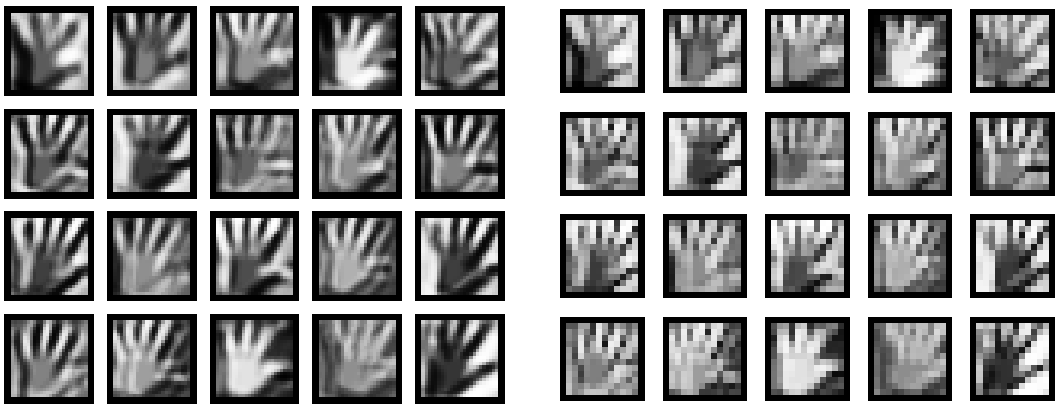
Besides conducting classifications multi-class SVMs also compute posterior probabilities for each class. Based on the analytic concept of generalization and certainty, for a problem of K gesture classes SVMs aim to estimate the probability of each class in the data $\vec{\mathbf{x}}$ such that, $p_i = p(y = i|\vec{\mathbf{x}})$ for $i = \{1, 2, ..., K\}$. Given that r_{ij} is an estimate for the output probability of pairwise classifiers between class i and class j, and p_i is the probability of the ith class, a *one-against-one* (OAO) approach solves the following optimization problem:

$$\text{minimize}_p \left\{ \frac{1}{2} \sum_{i=1}^{K} \sum_{j;j=1} \left(r_{ji} p_i - r_{ij} p_j \right)^2 \right\} \tag{5.1}$$

subject to the constraints $\sum_{i=1}^{K} p_i = 1$ and $p_i \geq 0$, where r_{ij} is defined as $r_{ij} \approx p(y = \{i, j\}, \vec{\mathbf{x}})$, such that $r_{ij} + r_{ji} = 1$. Refer to [Vapnik and Vapnik, 1998; Platt, 1998; Vapnik, 2013] for more details of SVMs.

The advantage of offline learning methods is that, they require all gesture images (samples) to be shown only once and at the same for training (learning). As CNNs and SVMs offer good learning and recognition performance of gestures (see Section 6.7.2), in [Nagi et al., 2012a] we developed a hybrid supervised classifier Convolutional Neural Support Vector Machine (CNSVM) which combines

the properties of both the MPCNN and the SVM. The CNSVM architecture is built by replacing the classification layer (last layer) of the MPCNN (see Figure 5.5) with a multi-class SVM classifier. Training of the CNSVM is performed using a *stochastic gradient descent* (SGD) approach which uses small batches (chunks) of samples acquired by a swarm of robots for incremental learning updates. However, CNSVM training is computationally expensive when using swarm robots that typically have limited on-board processing capabilities. Refer to [Nagi et al., 2012a] for more details of the CNSVM classifier.



(a) (b)

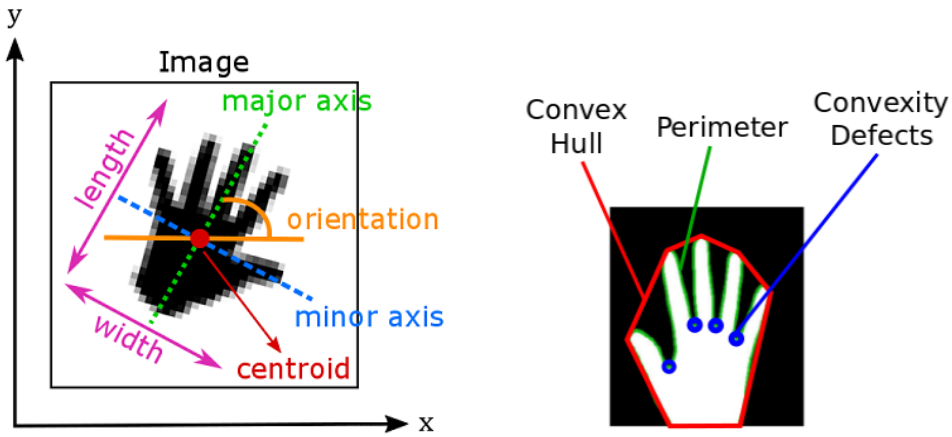
Figure 5.6. Image activations resulting from the MPCNN classifier using 20 maps. (a) Activations from the C1 layer. (b) Activations from the MP1 layer.

As offline learning methods require the prior definition of all commands in the gesture vocabulary (see Section 2.2.1) and require gathering a large training dataset of gesture samples (images), offline methods do not allow learning new information (i.e., information can only be learned once and new knowledge cannot be added or updated into the learned model). In addition, the acquired training dataset needs to be well representative of all the possible conditions that the system might encounter. This is challenging when using robots with low-quality cameras (i.e., illumination conditions can significantly differ between training and normal usage which greatly affects classification performance).

To overcome the limitations of offline methods, we consider *online learning* to be more favourable in real-world scenarios (where samples arrive in increments over the passage of time), as introduced in Section 5.3. The next section presents approaches using which features are computed from the segmented hand masks (see Appendix A). The computed features are used with the distributed cooperative learning strategy presented in Section 5.4.

5.2.1 From Hand-crafted Features to Automatic Features

The computation of image features is an integral part of learning and classifying gestures from the segmented gesture images (see Figure A.1). If swarm robotic systems are to be used for learning gestures in real-time (i.e., learning instantly) efficient strategies for *online feature computation* need to be in place. We investigate techniques that effectively compute image features for gesture recognition and efficiently represent the commands defined in gesture vocabulary: spatial pointing gestures (see Figure 2.4(a)), gestures for performing potential SAR tasks (see Figure 2.4(b)), and finger count gestures (see Figure 2.6).



(a) (b)

Figure 5.7. Hand-crafted properties computed from a segmented hand mask. (a): Properties such as, length, height, centroid, orientation, major and minor axis. (b): Properties such as, convex hull, convexity defects and perimeter.

5.2.1.1 Hand-crafted Features

Although many feature computation methods for vision-based gesture recognition exist, we adopt the most familiar approaches that compute meaningful and discriminative properties from the segmented hand masks (see Figure A.1). As features need to robustly represent all the commands in the gesture vocabulary (see Section 2.2.1), we compute *hand-crafted features* from the segmented hand masks (gestures): shape and blob properties [MATLAB R2014b documentation; Linan, 2007], geometrical characteristics and image moments [OpenCV 3.0.0-dev documentation]. These hand-crafted features have gained importance due to their powerful gesture classification performance [Savaris and Wangenheim, 2010] and for similar recognition tasks [Zhang and Lu, 2004; Daliri and Torre,

2008; Das et al., 2010; Trigo and Pellegrino, 2010; Cox and Budhu, 2008; van der Werff and van der Meer, 2008].

A few hand-crafted properties computed from a segmented hand mask are visualized in Figure 5.7. Image moments $\{u_{00}, u_{01}, \ldots, u_{20}, u_{02}\}$ are used to calculate properties such as the, length, height, area, centroid, orientation, major and minor axis, as shown in Figure 5.7(a). Figure 5.7(b) illustrates properties such as the, convex hull, convexity defects and perimeter. From the above mentioned hand-crafted feature properties (i.e., shape, blob and geometrical characteristics etc.), we select a set of $F = 110$ features that can be efficiently computed from segmented hand masks. However, using a relatively large number of features is redundant and counter-productive, because:

- Classifiers yield better accuracy (and faster predictions) if only a few, highly discriminative features are used. This is true with datasets that have a high-dimensional feature space, namely too many features.
- Computing fewer features is faster as compared to techniques that require training by building a large dataset (e.g., offline methods).
- Using fewer number of features more training samples can be disseminated (spread) throughout the robot swarm for bandwidth-limited scenarios.

Feature selection strategies aim to reduce the dimensionality of the feature space by selecting the best subset of features that have the highest importance in a supervised classification problem. To select an optimal subset of features from the given set of $F = 110$ features, we adopt the Principle Component Analysis (PCA) technique and the Ranker method in WEKA [Hall et al., 2009] (see Section 6.7.3), which provide an assessment of the quality of the features and rank the features: with respect to their individual and mutual discriminative powers and based on their contribution towards the multi-class classification problem. The top 20 hand-crafted feature properties (i.e., the 20 features with the highest ranks) are reported in Table 5.1 together with their measured PCA scores.

After feature selection is performed, the reduced subset of $F < 110$ features are termed as a *feature vector* $\bar{\mathbf{x}}$, also known as feature descriptor. As the gesture vocabulary comprises of one and two-handed gesture commands (see Figure 2.4), for one-handed gestures a one-dimensional feature vector is computed which consists of F numerical elements. In the case of two-handed gestures, a single feature vector of F elements is computed from each of the two segmented hand masks, after which the resulting two feature vectors are concatenated end-to-end to form a feature vector of $F \times 2$ elements.

Table 5.1. The top 20 hand-crafted feature properties selected using the PCA and Ranker methods in WEKA [Hall et al., 2009].

Rank	Feature	PCA Score
1	Solidity	0.594946
2	Extent of minimum enclosing circle	0.482678
3	Extent of minimum area rectangle	0.399841
4	Extent of enclosing bounding box	0.328516
5	Minimum y-axis co-ordinate	0.272821
6	Extent of ellipse	0.236379
7	Formfactor	0.210368
8	Roundness	0.186468
9	Compactness	0.165338
10	Solidity of circle	0.151234
11	Y-axis centroid co-ordinate of minimum enclosing ellipse	0.137960
12	Y-axis centroid co-ordinate of minimum enclosing circle	0.126662
13	Y-axis centroid co-ordinate of minimum area rectangle	0.115440
14	Y-axis centroid co-ordinate	0.105492
15	Y-axis centroid co-ordinate of enclosing bounding box	0.096098
16	Solidity of ellipse	0.086963
17	Minimum y-axis co-ordinate at maximum x-axis co-ordinate	0.078208
18	Minimum y-axis co-ordinate at minimum x-axis co-ordinate	0.070038
19	Sphericity	0.062963
20	Solidity of rectangle	0.056212

5.2.1.2 Automatic Online Feature Computation

The main disadvantage of hand-crafted methods is that, hand-crafted features cannot guarantee that they best represent the segmented hand masks (gestures). To overcome this critical issue, in [Nagi et al., 2014d] we introduce Convolutional Max-Pooling (CMP), a novel approach for *automatic online computation of features* as illustrated in Figure 5.8. Inspired from the alternating convolution and max-pooling layers of the MPCNN [Nagi et al., 2011], the CMP is a two-layer feedforward network which does not make use of any training mechanism (i.e., features are computed independently and irrespective of the gesture class). Refer to [Nagi et al., 2014d] for details of the CMP feature computation method.

Compared to hand-crafted features, the online features computed using the CMP provide a better numerical representation of the segmented hand masks and improve classification performance. This is because, for binary (black and white) images, hand-crafted features use only the segmented mask (i.e., only the

white pixels in the image) for feature computation, while the CMP uses all pixels (i.e., black and white pixels) and performs convolution and pooling operations.

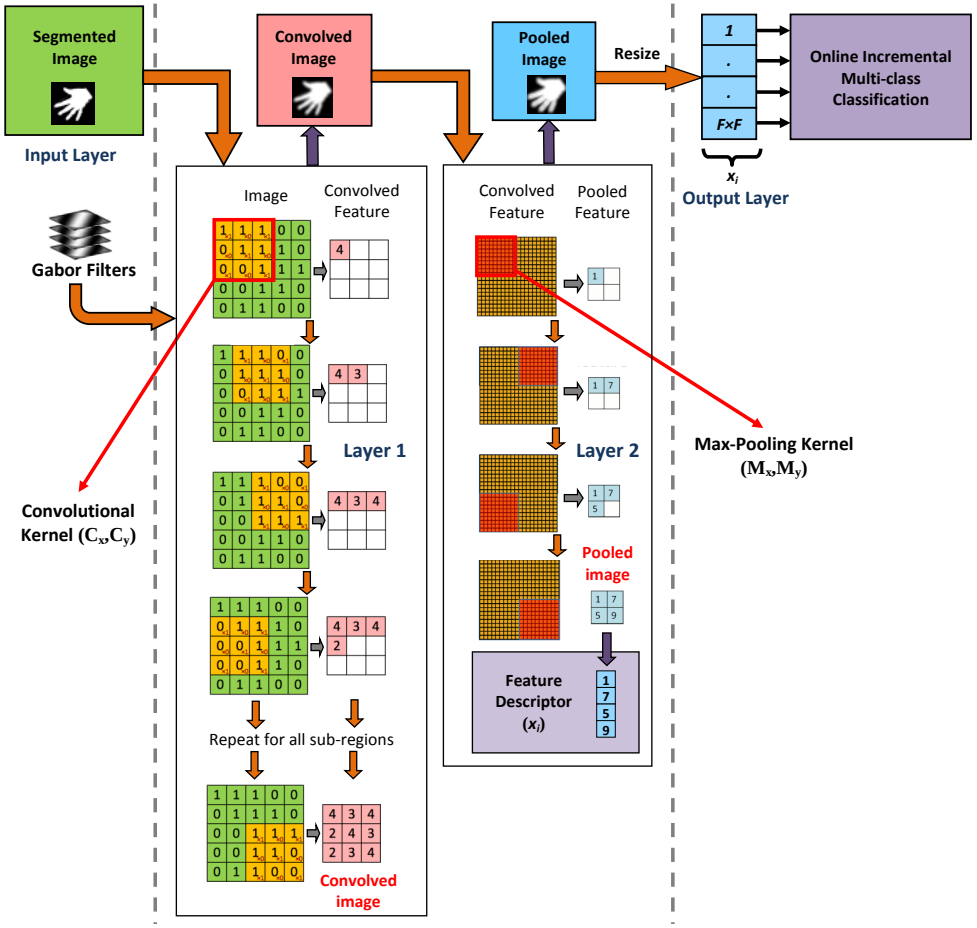


Figure 5.8. Online feature computation using the Convolutional Max-Pooling (CMP) approach, a two-layer feedforward network [Nagi et al., 2014d].

5.3 Online Learning Supervised by Human Feedback

Online incremental learning algorithms compute new hypotheses as soon as new training samples become available (i.e., when gesture samples are given by humans). Since offline learning methods induce a hypothesis from a set of training samples presented at a single point in time, online learning methods allow robot swarms to update their learned knowledge constantly. In online learning, a swarm of robots collects training samples at every point in time and then decides at some specific time-interval to compute (or re-compute) a new updated

hypothesis of the training samples that it has observed (seen) so far. Training samples are not available a priori in online learning, but incrementally arrive over time (usually one at a time) as the learning process needs to go on indefinitely. The time between new samples arrive is not negligible in the way the robots make use of the samples. Therefore, it is not always feasible to wait and gather a large number of samples before learning out of them.

For robot swarms to learn gesture commands in real-time (i.e., learn immediately), human instructors supervise the online learning process. Online learning allows human instructors to teach new gesture commands to swarms and improves what swarms have already learned so far (i.e., previously learned gestures). As swarm robots are equipped with computational capabilities (see Section 1.2.1.1) the general ability of remembering past observations and incrementally updating the related classification models is extremely useful. The next sections present the stages involved in online learning scenarios.

5.3.1 Phases in Online Incremental Learning

An online learning scenario involving a human instructor and a robot swarm is described as follows. A human presents a gesture (from a set of K predefined gestures) to the swarm which needs to be learned by the swarm. Next, each robot acquires an observation (sample) of the presented gesture, segments the image to obtain the hand mask (see Appendix A), computes features from the segmented hand mask (see Section 5.2.1), and the uses computed features with the cooperative recognition and decision-making protocol in Section 3.4 to build a unified swarm-level consensus decision. The swarm-level decision represents the swarm's recognition outcome for the presented gesture which is a gesture class label from a set of known K labels (gesture classes).

Based on the outcome of the swarm-level consensus, the classified gesture sample is used by the individual robots in the swarm to update their local classifiers (see Figure 5.1(c)). Next, the human presents another gesture, and this entire process repeats again for every new gesture issued by the human. In principle, online learning in robot swarms is obtained using a two-phase process: *initial learning* followed by *interaction rounds*, as discussed in the next sections.

5.3.1.1 Initial Learning Phase

The initial learning stage is required by robot swarms to gain preliminary knowledge (i.e., to acquire a small amount of basic information) regarding the gestures to be learned. During the initial learning phase a human instructor explicitly

makes a swarm learn gestures that have been defined in the gesture vocabulary (whose cardinality K is assumed known a priori). In practice, the human presents M samples for each of the K gesture classes and every robot in the swarm acquires $M \times K$ gesture samples. In total, the swarm acquires $M \times K \times N$ samples within a short duration, where N represents the swarm size.

Gesture commands are numbered and presented by humans in a sequential order which does not require humans to provide the class label K (for supervised learning) along with every gesture sample. Since the acquired samples are associated with their respective ground-truth (GT) information (see Section 6.2) every robot individually learns the $M \times K$ samples it acquires by training its local classifier. At the end of the initial learning stage every robot builds an initial classification model (classifier) which is trained using information only acquired from one viewpoint. This implies that, every individual robot in the swarm knows only the appearance of gestures from its own observation point (viewpoint). The initial classifiers trained by every robot are used for learning and classifying gestures during the interaction rounds as discussed in the next section.

5.3.1.2 Interaction Rounds with Human Feedback

Immediately after the initial training phase is complete, every robot in the swarm has a classifier in place and can start to learn and perform classification tasks. As the classifiers resulting from the initial learning stage have only a minimal amount of knowledge regarding the gesture classification problem, these classifiers are not expected to provide good recognition performance. The interaction rounds following the first round begin with a human providing a gesture (that represents a command to be executed by the swarm).

After an interaction round starts, the given gesture is recognized by the swarm using the cooperative recognition protocol introduced in Section 3.4, as shown by steps 1 and 2 in Figure 5.9. With this protocol, every robot uses its local most recently-trained classifier to generate a local classification opinion. After individual robot opinions have been fused and a swarm-level classification decision has been obtained, the human needs to provide feedback to the swarm based on the swarm-level decision of the presented gesture. For instance, if a gesture is not properly recognized by the swarm or it is classified as inappropriate, as shown in Figure 2.13, then the human needs to "correct" the swarm's wrong decision by communicating feedback to the swarm. Feedback from human instructors improves learning and corrects learning mistakes made by swarms. We consider that, during the interaction rounds humans can provide two types of feedback to swarms: *full* and *partial* feedback, as presented below.

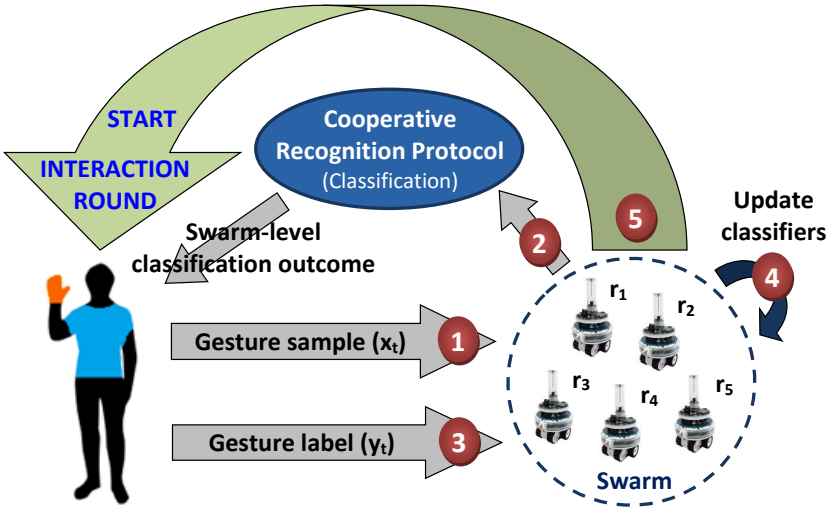


Figure 5.9. Illustration of a single interaction round with a human instructor. The numbers indicate the steps in a sequential order.

Full Feedback: Full feedback implies that humans communicate the actual/true label of the given gesture sample (i.e., the class among a set of K predefined gesture classes) to the swarm. If a gesture has not been not properly recognized by the swarm, the human conveys full feedback to the swarm using a simple visual signal: the rapid waving of hands (which can be robustly detected by a single-robot). After full feedback is given, the swarm proposes the second most-likely decision which represents the gesture class corresponding to the second highest probability in the swarm-level decision. This procedure iterates until the correct gesture class is proposed by the swarm and the human implicitly accepts it by not reacting to it (i.e., by remaining still). The Confidence Weighted Swarm Learning (CWSL) algorithm developed in this research is based on the CW learning method (see Section 5.1.2) and is presented in Appendix D. The CWSL algorithm uses full feedback from humans and has been customized for use with swarms.

Partial Feedback: Online learning with partial feedback (also referred to as *binary feedback*), builds upon results from analogical reasoning. Instead of conveying the actual/true label of the presented gesture which is full feedback, humans provide partial feedback to the swarm represented as a binary variable: *correct* or *incorrect* (yes or no). Humans convey partial feedback to robot swarms using visual signals. For instance, if a gesture has not been not properly recognized by the swarm, the human waves his/her hand indicating to the swarm that the swarm-level classification is incorrect. However, if the gesture is properly rec-

ognized, then the human does not react and remains still. Compared to full feedback, partial feedback is more flexible and comes at a lower cost. The Online Learning Partial Feedback algorithm developed in this research is based on Recursive Least Squares (RLS) methods and supports generality (i.e., applicable to single-robot systems and swarms). Refer to [Nagi et al., 2012b; Ngo et al., 2014] for more details of the RLS-based partial feedback algorithm.

After receiving appropriate feedback (full or partial) from humans, the robot swarm gets informed of the correct label associated with the presented gesture. For supervised learning to take place, an observation (sample) and its corresponding label are required. Since the gesture label is communicated to the swarm using full/partial feedback, the given gesture (which was already classified by the swarm) is now used as training sample and is learned by individual robots in the swarm (i.e., all robots in the swarm update their local classifiers with this new information), as shown by steps 3 and 4 in Figure 5.9. With the use of a mixed instrumented-natural interface, humans can robustly convey the gesture label to robot swarms. For instance, a handheld or wearable device (e.g., smartphone, smartwatch or tablet computer) can be used for communicating the actual/true sample label to a single robot or to the entire swarm.

When human feedback has been provided and the gesture is learned by the swarm, a single interaction round is complete. In summary, an interaction round consists of the following sequence of events: a human provides a gesture, the gesture is classified by the swarm, the swarm-level decision is communicated to the human, based on the swarm-level decision the human conveys full/partial feedback to the swarm, robots in the swarm learn the given gesture.

As the online learning process unfolds (progresses) and gestures are learned by the swarm in real-time (i.e., immediately), eventually the robot swarm will learn the entire vocabulary of gesture commands (see Section 2.2.1). A potential advantage of online learning is that, during the interaction rounds new mission commands can be learned by swarms.

5.4 Distributed Cooperative Learning with Humans

In online learning, selecting the 'right' next samples to learn and update a robot's classifier greatly affects the performance of a robot's learning and recognition capabilities, which is an open problem in *cooperative learning*. Cooperative learning in robot swarms relates to the problem in which every robot independently trains its local classifier, but at the same time it cooperates with the other robots

in the swarm to collectively speed up and improve swarm-level learning. Existing works that deal with cooperative learning are discussed in Section 5.1.3.

Since online learning algorithms abstract away the process of data acquisition, they do not generally incorporate communication which limits a robot's access to the training data. Cooperative learning in robot swarms emerges from the fact that, each robot in the swarm augments its local training set with data acquired from other robots that are located at different viewpoints. Every robot in the swarm needs to exploit both the locally available training information (i.e., the acquired samples from its specific viewpoint) and the training samples received from other robots. The information directly acquired by a robot amounts to its *experience* and the information received from other robots corresponds to the *common knowledge* (or shared knowledge) of the swarm.

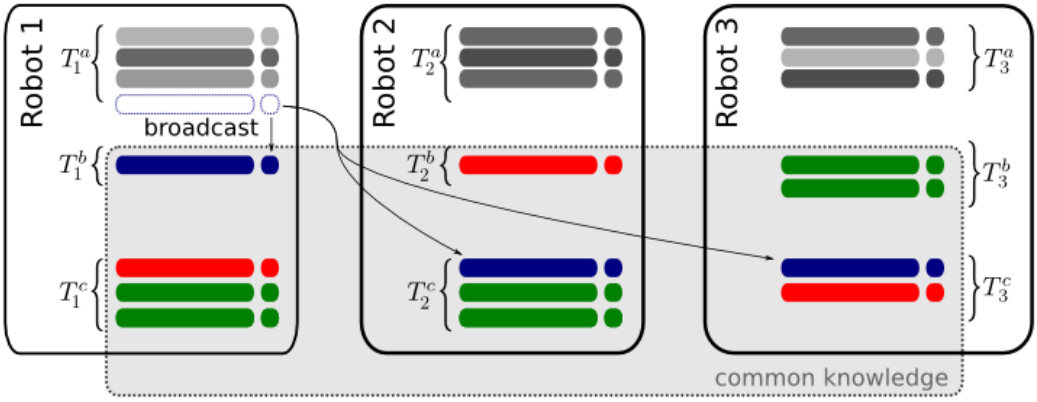


Figure 5.10. Illustration of a cooperative learning scenario with information selection and sharing using a swarm of $N = 3$ robots.

An example of a cooperative learning scenario with information sharing is illustrated in Figure 5.10 using a swarm of $N = 3$ robots. In this scenario every robot possess its own private (local) training set which consists of three subsets of samples: T_r^a are samples acquired by a robot itself, T_r^b represents samples to broadcast (samples to share with the other robots in the swarm), and T_r^c are samples received from the other robots in the swarm. The grey colored samples are the samples acquired by every robot, and the red, blue, and green coloured samples collectively represent the common knowledge of the swarm. Robot 1 broadcasts 1 sample (blue) to all other robots, Robot 2 broadcasts 1 sample (red) to the other robots, and Robot 3 broadcasts 2 samples (green) to the others.

In dynamic environments that allow the online learning of information, distributed cooperative learning is formulated as follows. During an interaction

round a human provides full feedback to the swarm (see Section 5.3.1.2). After human feedback is conveyed, robots in a swarm share and exchange the labelled samples (observations) that they have acquired. When samples have been shared within the swarm, every robot adds the samples it has received from other robots and its own samples into its local training set, and the local classifier of every robot is retrained (updated) using these new samples. In this way, cooperative learning allows individual robots to learn information seen from the viewpoints of other robots, thereby expanding the local training set of individual robots.

The hypothesis we consider is that, distributed cooperative learning can significantly improve swarm-level recognition performance: (i) learning faster and more robustly through information sharing, and (ii) producing more accurate swarm-level classification decisions. Although cooperative learning is potentially very effective, however, it poses intrinsic restrictions in the way it is performed:

- The processing of every training sample (acquired gesture observation) needs to be fast (i.e., within a second).
- The computational capabilities and communication resources available to individual robots in a swarm are very limited.
- The use of cooperation through information sharing can overcome the limitations of individual robots, however, cooperation needs communication bandwidth which is potentially very limited in robot swarms.

These are the precise conditions of the learning scenario taken into consideration: swarm robots are generally not extremely powerful (due to the need to balance the number of robots with the cost), communication bandwidth is limited, and it is necessary to learn fast, and from a few number of samples. The core contribution in cooperative learning consists in accounting for these requirements and restrictions, and developing computationally and communicationally efficient strategies to deal with such situations. Being online and fully distributed (and bandwidth-limited) make the problem settings extremely challenging.

5.4.1 Constraints on Swarm-level Learning

The main goal of cooperative learning in robots swarms is to *maximize* the quality and the speed of learning of information while *minimizing* the use of computational and communication resources. This problem is more generally defined as cooperative learning with computational and communication constraints, which

requires intelligent selection from a stream of data samples (i.e., gesture observations) that are locally acquired and shared among multiple robots in a swarm.

Existing approaches for cooperative learning are either too expensive for computation or communication, or cannot properly scale with large swarms (see Section 5.1.3). Strategies that perform selection of data are based on the distributed form of the popular *bagging* classifier [Breiman, 1996], in which each sensor (robot) individually trains a multi-class classifier based on a bootstrapped replica of the entire training dataset. The replica is implicitly derived from a robot's own observations and from the selected training data it receives from other robots through cooperative information sharing. Bagging is simple, lightweight, and has shown good performance in the case of limited training data [Polikar, 2006; Claesen et al., 2014], as presented in Section 3.1.2.

While it has been demonstrated that sharing learning information in sensor networks is advantageous (i.e., it guarantees a faster convergence in the learning rate; see Section 5.1.3), in many applications, bandwidth, energy, and resource considerations do not allow for constant exchange of complete inter-network information. If communications are delayed as opposed to being instantaneous, they may become obsolete before arriving at their intended destination. To minimize bandwidth consumption and reduce redundancies in shared information, it is important to investigate the "amount" and "quality" of shared information.

As simplicity in communication needs to be emphasized, distributed information sharing strategies that select information based on *intelligent criteria* and share a *selective amount* of information (i.e., robots share compressed statistics) are required. We consider that, every time new training information is available either from local sensing or received from other robots, a robot has to decide whether the data (information) should be kept, shared or discarded based on:

- The data the robot has acquired so far without any knowledge about what training information will be provided in the future (which sets the importance of a training sample).
- The partial and incomplete view of: (i) the data the robot has received so far and (ii) what data other robots have in their training sets.

The most basic strategy for sharing information consists in spreading to the entire swarm every sample acquired by each individual robot. If robots in a swarm make use of the maximal amount of available training information, this has a number of drawbacks. Firstly, the computational complexity of the training process rapidly increases (in the case of large swarms) which has a dramatic consequence on the processing time required to retrain local classifiers. Secondly,

a large communication overhead is generated (consumption of high bandwidth and resources) due to multi-hop message passing algorithms that uniformly disseminate data throughout the swarm network. Lastly, it does not guarantee that the shared samples represent the most novel information known by the swarm. As a consequence, sharing all information (experiences) is not be feasible and a criterion needs to be introduced that can select the appropriate subset of training samples to be shared with the rest of the swarm.

We consider that the criterion for selecting data to be disseminated is based on an original distinction between a robot's personal and shared data, which is derived from previous works [Tangamchit et al., 2003; Navia-Vazquez et al., 2006; Flouri et al., 2009] and has been adapted for optimizing the mutual information gathered by a swarm. One open problem in cooperative learning is the *online selection of training samples*. Without any additional simplifying assumptions, the problem of online training data selection is tackled using a heuristic approach that is compliant with the given restrictions.

5.4.2 Strategies for Sharing and Forgetting Information

The strategies developed for the online selection of training data are heuristic and based on a computational approach. These strategies aim at intelligently selecting training information to share with other robots in the swarm, by performing data selection and keeping the amount of training data for robot learning to be *bounded in size*. We introduce two bounds: (i) bounding the local training set to a restricted size (which induces a bound in time for processing the same dataset) and (ii) bounding the size of the shared training information (which circulates across the swarm network). This cooperative learning behaviour results in simultaneous *sharing* and *forgetting* of information [Di Caro et al., 2013a], as introduced in Sections 5.4.2.1 and 5.4.2.2 respectively.

5.4.2.1 Strategies for Selecting and Sharing Experiences

The problem of selecting which sample to disseminate is directly related to *active learning* [Cohn et al., 1996] in which a robot has a large number of unlabelled samples and can buy labels for some samples. In the case of margin-based classifiers, this problem has been thoroughly investigated: the distance between each sample and the separating hyperplane in the feature space is used as a criterion [Schohn and Cohn, 2000] by privileging samples either close or far from the classification margins. In our case, this problem is different as the focus is to

select training samples to share with other robots, however, only partial knowledge is available of the samples that are known by other robots in the swarm.

Consider R to be a set of robots in a swarm and let T_r denote the set of training samples currently available to robot $r \in R$. The set of training samples T_r can be partitioned into three different subsets:

- *Personal*, T_r^p: Includes the samples acquired by r that have not been shared with the rest of the swarm.
- *Shared* T_r^s: Includes the samples acquired by r that have been shared with the rest of the swarm.
- *External* T_r^e: Includes the samples originally acquired by robot $r' \neq r$, that have been shared by r' and received by r.

As every robot has information from a different set of training samples, this further complicates the task. In the case of full connectivity, r has no interest in broadcasting samples in T_r^s or T_r^e as they are already known to the rest of the swarm. In fact, the set of samples $T_r^s \cup T_r^e$ represent the common knowledge of the swarm. For a robot r to select a sample $x \in T_r^p$ that will be shared first with the rest of the swarm, we consider three sample selection criteria:

1. **Random Selection:** The sample is selected in a purely random way without any intelligent criterion. This strategy serves as a performance baseline.
2. **Novelty-driven Selection:** Robots prefer to share the sample which brings the most novel information compared to the samples already known by the rest of the swarm. Intuitively, this selection strategy mimics the behaviour of a teacher who presents samples that are dissimilar as possible from the samples already known to the students. Robot r selects the training sample to be broadcast as follows. First, a class K is selected at random, then a training sample x for class K is selected using:

$$\underset{x \in T_{r,K}^p}{\arg\max} \; \|\mathbf{f}(x), \mathbf{f}\left(x'\right)\|, \quad \forall x' \in (T_{r,K}^s \cup T_{r,K}^e),$$

where $\mathbf{f}(x)$ represents the feature vector of the training sample x, $\|\mathbf{f}(x), \mathbf{f}(x')\|$ denotes the Euclidean distance in the feature space, and $T_{r,K}^p$, $T_{r,K}^s$, and $T_{r,K}^e$ respectively represent the personal, shared and external training subsets of robot r for class K.

3. **Representativity-driven Selection:** A robot r shares a sample that best represents its current knowledge about a given class. This corresponds to a teacher who privileges samples that are the most typical ones. Sample selection is implemented by firstly sampling class K at random, and then choosing a training sample x such that:

$$\arg\min_{x \in T^p_{r,K}} \|\mathbf{f}(x), \mathbf{f}_K\|,$$

where $\mathbf{f}_K$ represents the centroid in the feature space of all samples in $T^p_{r,K}$.

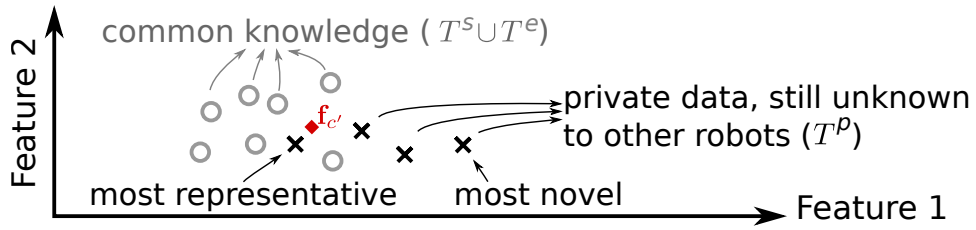


Figure 5.11. Criteria to select the samples to broadcast. For a given class K all training samples known to a robot are represented as points in the feature space. Gray circles: Samples already known to the swarm. Black crosses: Samples only known to a robot itself. Red diamond: Centroid of all personal samples ($\mathbf{f}_K$).

Figure 5.11 illustrates the three different sample selection strategies. In practice, once a training sample x is selected by robot r, the feature vector and class label K of sample x are spread to the rest of the swarm as a local broadcast. This message is firstly received by the wireless neighbours of r and is further relayed in a multi-hop fashion to all other robots in the swarm. After the broadcast is complete, the following updates are performed:

- At robot r, x is removed from T^p_r and moved to T^s_r.
- At all other robots $r' \neq r$ that receive the message, x is included in $T^e_{r'}$.

As a result, after this update x will never be broadcast again by any other robot in the swarm. In summary, if robot r has n training samples to disseminate, then the mechanisms mentioned above are repeated n times.

5.4.2.2 Strategies for Forgetting Experiences

The strategies introduced in Section 5.4.2.1 for sharing information and experiences yield a training dataset which monotonically increases in size as more interaction rounds (see Section 5.3.1.2) are performed. Eventually, the size of the training set will increase too large to allow the rapid retraining (update) of the local classifiers. The obvious solution is to limit (bound) the maximum number of training samples stored by every robot. This implies that, when the size limit is reached: a new training sample needs to be inserted into the training set and a previously stored sampled needs to be removed. In other words, it is necessary to forget some samples to make space for newer samples.

A straightforward and computationally light solution consists in forgetting the oldest training samples and implementing a first-in first-out (FIFO) queue in the memory storage (training set). If the environmental conditions slowly changing over time (i.e., yielding a slow drifting classification problem) old training samples may not be relevant any more. Forgetting old training samples can improve the classification performance of robots rather than limiting it. This is because, removing old samples and adding new samples into the training set from time to time will result in a training set that contains an up-to-date representation of the classification problem to be solved.

The three strategies presented in Section 5.4.2.1 that implement an intuitive criteria for selecting samples (by exploiting the topology of training samples in the feature space) are used for forgetting samples. In all three strategies, the first step is to determine class K which is the most representable in the training set. The sample to be forgotten will be selected within $T_{r,K}$, i.e., the set of samples that belong to the training set of robot r with class K. This has an effect to keep the training set balanced. The three sample forgetting strategies are (no distinction is made among samples that are personal, shared, or external):

1. **Random Selection:** The sample is selected in a purely random way within the training set $T_{r,K}$.

2. **Redundancy-driven Selection:** Robots decide to forget a sample which is most similar to another sample that is already known. This strategy mirrors novelty-driven selection (see Section 5.4.2.1) such that, a robot prefers to retain the samples that are the most novel with respect to all the other samples known by the robot. In particular, a robot r selects the training sample x to forget (remove) as:

$$\arg\min_{x \in T_{r,K}} \|\mathbf{f}(x), \mathbf{f}(x')\|, \quad \forall x' \in (T_{r,K} \setminus x).$$

At least two samples always meet the criterion, but only one sample is selected using random selection.

3. **Representativity-driven Selection:** Robots forget the sample which is the least representative from their current knowledge about class K. This mirrors the representativity-driven selection strategy (see Section 5.4.2.1) such that, a robot prefers to keep the samples that are the most representative with respect to what is currently known by the robot. A robot r selects the training sample x to be removed as:

$$\arg\max_{x \in T_{r,K}} \|\mathbf{f}(x), \mathbf{f}_K\|,$$

where $\mathbf{f}_K$ represents the centroid in the feature space of all samples in $T_{r,K}$.

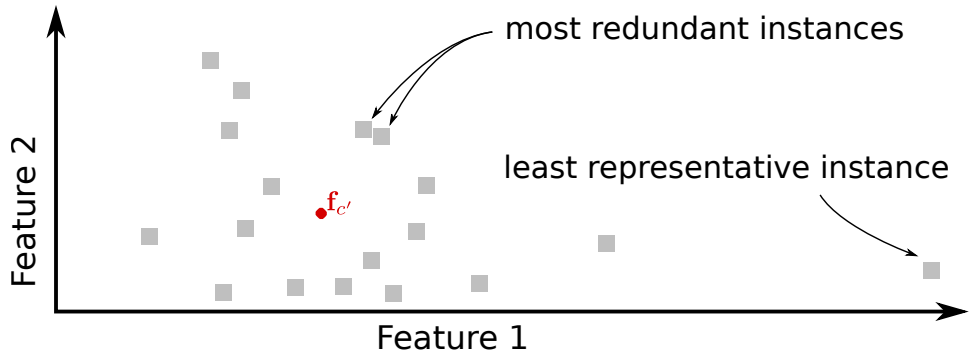


Figure 5.12. Criteria for the selection of samples to forget. For a given class K all training samples known to a robot are represented as points in the feature space (gray squares). Red diamond: Centroid of all samples ($\mathbf{f}_K$). The least representative sample lies farthest away from the centroid of all samples and the most redundant samples lie closer to the centroid.

The three strategies to forget samples are illustrated in Figure 5.12 for a simplified scenario. In practice, at robot r once a training sample x needs to be to forgotten, the feature vector of sample x is removed from the training set of the robot. For every training sample that needs to be forgotten (removed) the mechanisms reported above are repeated once.

5.5 Summary of Experimental Results

The experimental results of this chapter are presented in Section 6.7 along with the discussion. These results investigate the effect of different swarm-level learning strategies such as, offline learning (which requires gathering a large dataset) and real-time learning supervised by human instructors: online learning and cooperative learning. The effect of offline learning is investigated with respect to two different types of supervised classifiers, CNNs (which compute features from images) and SVMs (which require the use of hand-crafted features or automatic feature computation methods). Within the context of hand-crafted features, the impact of different feature selection methods is investigated. Compared to all other methods, the PCA provides the best performance for reducing the dimensionality of the feature space. In terms of cooperative learning, the swarm-level learning performance is investigated with respect to the number of interaction rounds, the amount of information (samples) shared by individual robots in the swarm, the size of the shared feature vector, and the swarm size. Strategies for selecting and sharing information have been compared, and the representativity-driven approach outperforms all the others. Similarly, for forgetting and removing samples, the random selection approach provides the best performance.

5.6 Summary of Contributions

This chapter presented swarm-level learning algorithms and strategies to satisfy the sub-goal outlined in Section 1.4.2.3. In particular, distributed online learning strategies for robot swarms were investigated which provide the capability to include humans in the loop of the swarm learning process.

Offline learning methods justify that robot swarms have the capability to effectively learn instructions and commands defined in the gesture language. The developed online learning strategies support the inclusion of human instructors in swarm-level learning process. Humans supervise the learning of gesture commands by providing full or partial feedback to robot swarms. The strategies developed for cooperative learning allow intelligent online selection of training samples (to disseminate the most novel and representative training information to other robots in the swarm) and provide a trade-off between the *use of communication in the swarm* and the *quality of learning*.

Chapter 6

Experimental Results and Discussions

This chapter presents the experimental results and discussion of the techniques, algorithms and strategies implemented in Chapters 2 to 5. These results have been obtained as a consequence of: (i) emulation experiments (i.e., simulations performed on a computer that uses images acquired by a heterogeneous robotic swarm) and (ii) real-time interaction with robots, as discussed in Section 6.3.

6.1 Implementation on Real Robots

To implement the HSI system on real robots, simulators, middleware and open source libraries have been used. The source code is written entirely in C/C++ and runs on a Linux-based operating system. To control the Foot-bots (see Section 1.2.1.1), ARGoS[1], a large scale multi-robot simulator is adopted, and for the Parrots (see Section 1.2.1.2), ROS[2] is used. ARGoS has been developed to simulate swarms in virtual environments and to control robots in real-world environments. ROS is a middleware that interfaces with many robotic platforms, and at the lowest level it offers a message passing interface that provides inter-process communication. With the use of ARGoS and ROS a heterogeneous swarm of UGVs and UAVs can quickly be assembled.

All computation and processing is done on-board the embedded Foot-bot platform (i.e., the compiled Linux code runs as a controller on the Foot-bots). With the Parrots, all processing is done offline and the acquired images are wirelessly streamed from the Parrots on to a computer that performs all the necessary computation. For image processing the OpenCV library is used and Opencvcblobslib

[1] http://www.argos-sim.info/
[2] http://www.ros.org/

is adopted for feature extraction and computation.[3] The LIBSVM library[4] is used for supervised multi-class learning and classification. The human body motion detector (see Appendix B) is implemented using a C++ circular buffer library.

To fuse information generated from multiple robots, we evaluate the use of *centralized* and *decentralized* communication architectures. A decentralized architecture is used for the Foot-bots as they are equipped with the RAB system and can easily communicate with each other. For the Parrots, a centralized architecture is adopted since the images acquired by the Parrots are streamed to a computer which performs, image processing, feature extraction, and data fusion.

In the decentralized approach, every Foot-bot runs a single Linux process (i.e., the robot controller) and this process executes: the gesture vocabulary thread, the image acquisition thread, the motion detection thread, and the thread to listen to incoming messages from other Foot-bots in the swarm. The centralized approach is realized using a client-server architecture in which clients represent the robots and the server represents the fusion center (FC). A computer is used as a FC which runs a Linux process that performs the following: listens to incoming information from all the Parrots in the swarm, processes and fuses received information, and sends out the swarm-level decision to all the Parrots. The number of threads that listen to incoming data from the Parrots are equal to the number of Parrots in the swarm (i.e., one thread is used for each Parrot). Each Parrot is associated to a single Linux process that runs on the FC and executes threads for, the gesture vocabulary, image acquisition, and motion detection.

The camera properties of the UGVs and UAVs (e.g., resolution, field of view, and frame rate) have significant differences between each other, which greatly effect the performance and robustness of the HSI system. The Foot-bot with its frontal camera acquires images in a native resolution of 384 × 288 pixels (0.1 megapixel) at 2 fps in a 4:3 aspect ratio, as shown in Figure 1.4(b). The frontal camera of the Parrots acquires images in a resolution of 640 × 360 pixels (0.2 megapixel) at 30 fps in a 16:9 aspect ratio, as given in Figure 1.5(b). The image resolution effects gesture recognition performance, and the frame rate has an impact on the performance of human motion detection (see Appendix B).

The image acquisition rate of the Foot-bots is lower than 3 fps, which causes a constant delay in motion detection (see Section 7.2). With the Parrots a frame rate of 30 fps is achieved, and motion detection is very reliable. In terms of camera resolution and image quality, the lower resolution camera of the Foot-bots requires adjusting and correcting the white balance (using the gain and satura-

[3] http://opencv.org/ and https://github.com/opencvblobslib
[4] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

tion settings) based on the illumination conditions. In contrast, the Parrot camera has a higher resolution (i.e., better quality images) and a larger color depth. Experimentally it has been verified that, 2 Parrots can provide gesture recognition performance similar to that of 10 Foot-bots, and 4 Parrots can outperform a swarm of 15 Foot-bots. Overall, the Parrot camera has a better image quality and frame rate as compared to the Foot-bot camera, which is why gesture recognition and motion detection is more robust with the Parrots. Although experiments in this research have been performed in an indoor environment with controlled illumination (lighting), the UGV and UAV cameras are suitable for outdoor use under bright and sunny conditions while taking into consideration shadows.

To assemble a heterogeneous swarm of Foot-bots and Parrots, a platform independent *distributed communication protocol* based on native TCP/IP sockets has been developed which serves as a communication bridge between ARGoS and ROS. Using this communication protocol heterogeneous swarms of up to 20 real robots have been tested, and using an acquired dataset of images (as discussed the next section) up to 100 robots have been simulated.

6.2 Offline Data Acquisition using Real Robots

To allow flexibility in performing different experiments, a large amount of gesture images are acquired using a heterogeneous swarm of robots. This dataset of gesture images which is acquired for training (learning) is used to perform emulation experiments (see Section 6.3). With the use of the Foot-bots and the Parrots (UGVs and UAVs) gesture data is collected from a wider visual perspective: (i) airborne UAVs observe the human from a higher altitude compared to UGVs that are closer to the ground, and (ii) the more in number UGVs gather large amounts of data from different observation points (viewpoints).

Before the dataset can be acquired, a swarm of UGVs and UAVs are positioned in the configuration shown in Figures 6.1(a) and (b) respectively. The swarm of $N = 13$ Foot-bots is positioned to gather images near to the ground and the swarm of $N = 4$ airborne Parrots acquires images hovering at an altitude of 1.5m (which is considered a good height to observe the upper human body and gestures). The Foot-bots are placed at evenly-spaced angles of 15° covering a semi-circle centered around the human and at different distances from the human. The Parrots are made to fly in a semi-circular formation while being 40 to 60° apart from each other (at different distances from the human), and always facing towards the human (i.e., the human is always present within the field of view of every UAV's camera). In this configuration, the central robot (i.e., the

robot directly in front of the human) is at the *optimal viewpoint* to sense and recognize gestures, and the remaining robots see gestures from angled viewpoints.

In the case of a UGV swarm, each Foot-bot acquires and stores 200 unprocessed images while a human for a short time presents a single gesture from the gesture vocabulary (see Section 2.2.1), which is directed towards the robot precisely in front (at $\theta = 0°$) of the human. For the predefined set of $K = 16$ gestures in the vocabulary (see Figures 2.4 and 2.6), the UGV swarm acquires $13 \times 200 \times 16 = 41,600$ images. This process is repeated 5 times, once for a different distance $d = [1, 2, 3, 4, 5]$m between the UGVs and the human, as illustrated in Figure 6.2. This results in a dataset of $41,600 \times 5 = 208,000$ gesture images acquired by the UGV swarm from $13 \times 5 = 65$ different viewpoints. The 65 viewpoints represent the cells of the superimposed grid illustrated in Figure 3.5. Using the same approach with UAVs, $4 \times 600 \times 16 \times 5 = 192,000$ gesture images are acquired by a swarm of airborne Parrots from $4 \times 5 = 20$ viewpoints.

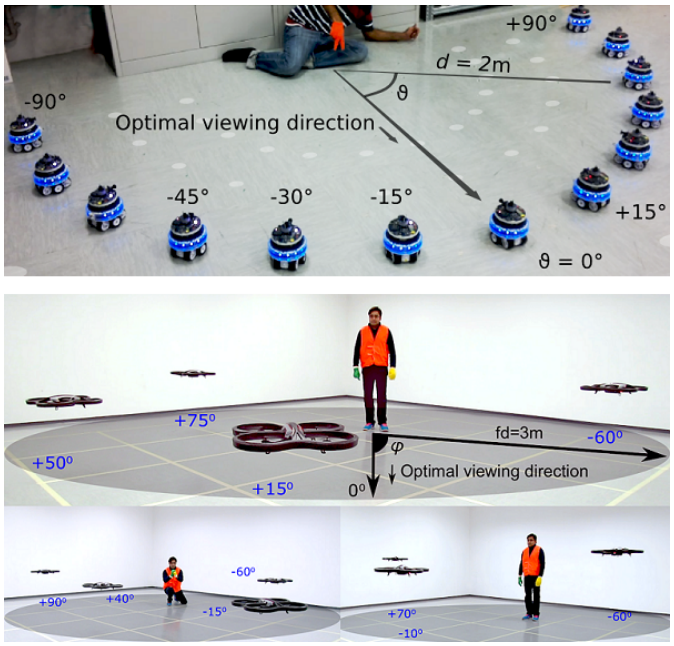


Figure 6.1. Experimental setup for the acquisition of a gesture image dataset. Top: Swarm of $N = 13$ Foot-bots. Bottom: Swarm of $N = [3, 4]$ airborne Parrots.

As gestures are presented naturally: rotational, translational, and scaling invariances are present in the acquired images. Every gesture image acquired by each individual robot in the swarm is tagged (labelled) with *ground truth* (GT) information. The GT information is unique for every acquired gesture image and it contains the following information:

(i) The angular distance (θ,d) between the human and the robot.

(ii) The time when the image was acquired (timestamp), where $t = 0$ represents the starting time of the acquisition process.

(iii) The class/label of the acquired gesture image (used by supervised learning methods for multi-class classification).

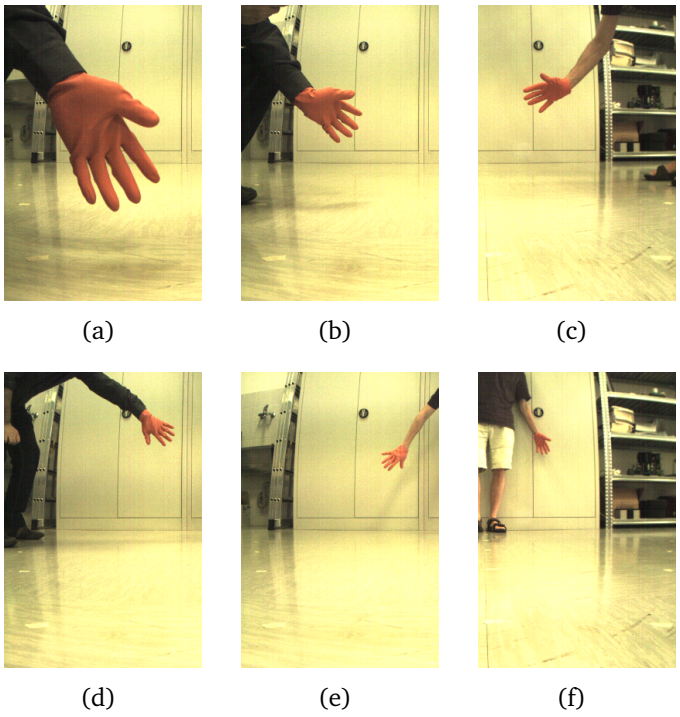


Figure 6.2. Gesture images acquired by a Foot-bot at different human-robot distances (in metres): (a) 0.5m (b) 1m (c) 1.5m (d) 2m (e) 2.5m (f) 3.0m.

In the context of human-relative localization (see Section 4.3.2.3), an airborne UAV is used to acquire images of a human's face from multiple poses (i.e., frontal-views and lateral-views). GT information is obtained using an OptiTrack Motion Tracking System which consists of an overhead Infrared-emitting and -sensing multi-camera system positioned to triangulate and extract location, movement, and orientation information of robots. The OptiTrack represents the location and position information of the airborne UAV in 3D Cartesian coordinate system. To acquire the dataset, the UAV is made to fly in a $[0, 180°]$ semi-circular plane of in front of the human while acquiring images of different face poses and at the same time receiving GT information.

6.3 Types of Experiments

In this research, two types of experiments are performed, *emulation experiments* and *experiments with real robots*. Quantitative emulation experiments are performed as simulations on a computer that make use of an acquired dataset. To challenge emulated swarms with difficult problems, we implement an experimental protocol: using GT information from an observation (image) in the dataset, a simulated robot positioned at (θ,d) in the polar plane 'sees' an observation that is randomly selected from the subset of observations acquired from the viewpoints closest to $(d;\ \theta)$ for the same gesture class. In simpler words, gesture observations are sampled from the acquired image dataset using which realistic simulations are built. To perform a variety of experiments different realizations of random variables (i.e., robot positions, observations in the dataset, and gesture sequences) are considered. Experiments performed with real robots evaluate the real-time performance, robustness and scalability of the HSI system.

The sections below present the results and discussion of Chapters 2 to 5. The results of Chapter 3 are presented first followed by: the bidirectional human-swarm communication system (Chapter 2), swarm-level coordination for robot selection and deployment (Chapter 4), and swarm-level learning (Chapter 5).

6.4 Results for Chapter 3

The experimental results of Chapter 3 are presented first, because the general protocol for the swarm-level classification of gestures provides valuable insights on how swarms cooperatively sense and recognize gestures. Experiments reported in this section are the results of emulation tests that use a dataset of $K = 6$ finger count gestures (see Section 2.6) and a multi-class SVM classifier (see Section 5.2) that is trained using hand-crafted features (see Section 5.2.1.1).

6.4.1 Swarm-level Performance of General Protocol

The experiments described in this section are aimed on studying the swarm-level recognition accuracy and response time depending on different parameters of the cooperative recognition protocol. Experiments are carried out in simulation as emulation tests as classifications are always based on gesture images acquired by a swarm of real robots (see Section 6.2). In all experiments, robots' opinions are produced once per second which is the performance attained on the Foot-bot platform (see Section 1.2.1.1) accounting for image acquisition, processing, and

classification. Robot communications are simulated with parameters matching the characteristic of the Foot-bot's RAB communication system: packets are received with a 0.1s delay and packet loss probability is modelled as a piecewise linear function $\pi(d)$ of the distance d between two communicating robots. That is, $\pi(d \leq d_{min} = 1\text{m}) = 0.2$, $\pi(d \geq d_{max} = 4\text{m}) = 1$, where $\pi(d_{min} < d < d_{max})$ is the line segment between $(d_{min}, \pi(d_{min}))$ and $(d_{max}, \pi(d_{max}))$.

For each tested configuration, a large number of simulation trials are performed using different spatial positions of the robots and different random sampling of observations from the image dataset. Each simulation results in one of three outcomes: *success* (all robots reach a *CollectiveDecision*() state for the correct class), *failure* (all robots reach a *CollectiveDecision*() state for the same, wrong class), or *no consensus* (none of the previous outcomes is true at $t = 2\,T$). For each experiment, the time to decision is recorded which is defined as the earliest time in which all robots are in the *CollectiveDecision*() state for the same class (or $2\,T$ in case of *no consensus*). Two performance measures are computed for each configuration: (i) the *average accuracy* (i.e., the fraction of experiments with *successful* outcome) and (ii) the *average time to decision*.

6.4.1.1 Accuracy vs. Time to Decision and Accuracy vs. Swarm Size

The swarm-level parameter λ_s determines the amount of statistical evidence a robot needs in order to initiate the swarm-level decision phase. Figure 6.3 shows that decisions taken on the basis of less evidence (small λ_s) are less accurate but are issued faster, compared to more prudent decisions (large λ_s) which are taken only when very solid statistical evidence is available. Moreover, larger swarms improve both accuracy and time to decision with respect to swarms composed of fewer robots. The advantages of swarms larger than $N = 10$ robots are clear only when fast response times are needed. A 20-robot swarm has a 7% larger accuracy than a 10-robot swarm when decisions are taken in 0.7s. When λ_s is set to larger values, this gap reduces as the accuracy approaches 100%. It can be observed that a single robot has a comparatively much worse accuracy.

In terms of swarm size, Figure 6.3 illustrates that a larger swarm always results in higher accuracies. Increasing the number N of robots in the swarm has two main positive effects on the cooperative recognition protocol: (i) a larger number of different opinions are available, and (ii) swarm connectivity is improved due to increased robot density. Figure 6.22 reports the impact of swarm size on recognition accuracy in relation to the value of λ and the use of mobility.

The accuracy vs. speed trade-off for each of the $K = 6$ finger count gestures (i.e., finger counts from 0 to 5) is shown in Figure 6.4. It is observed that, some

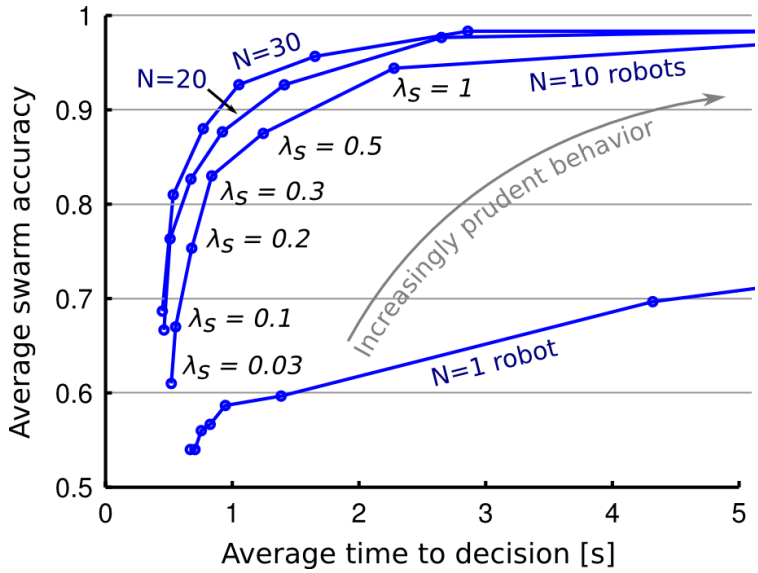


Figure 6.3. Trade-off between swarm-level accuracy and time to decision for different values of λ_s and swarms composed of 1, 10, 20 and 30 robots. Each data point is averaged over 50 trials using the $K = 6$ finger count gestures. Deployment is random, no mobility, no communication losses, $T = 10$s.

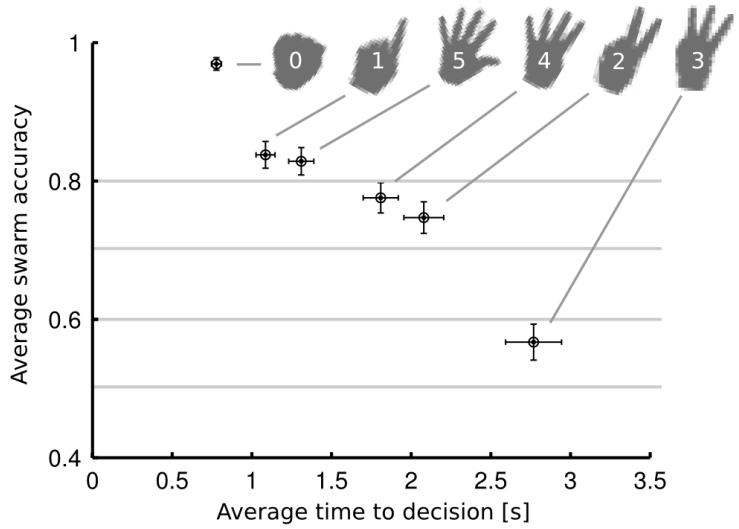


Figure 6.4. Average accuracy and time to decision for the $K = 6$ finger count gestures using the experimental setup in Figure 6.3 (each data point is averaged over 7200 simulations). The accuracy vs. time to decision trade-off is biased towards fast decisions as most simulations use small values of λ_s.

gestures are significantly harder to recognize than others. For instance, this is the case with the gesture that corresponds to finger count 3. The reason of the difficulty lies in the fact that the gesture was represented in the dataset using several different combinations of three fingers. Recognition of difficult classes requires on average a longer time, because the swarm needs to acquire more observations in order to reach the required evidence threshold λ_s. When compared to easier classes, classifications generated from difficult classes tend to be more contradictory with each other. Decision vectors resulting from the fusion of such opinions exhibit less pronounced peaks, consequently leading to lower λ values.

The experiments in Figure 6.4 make use of data resulting from the same set of simulations used in Figure 6.3, the majority of which use a very small λ_s value. As a result, the accuracy vs. speed trade-off is biased towards faster decisions and lower accuracies. When using $N = 10$ robots and a larger $\lambda_s = 2$, finger count 3 is recognized with an average accuracy of 0.86 in an average time of 4.2s. In contrast, with the same parameters finger count 0 (closed hand) is recognized with almost perfect accuracy (0.99) in an average time of 1.4s.

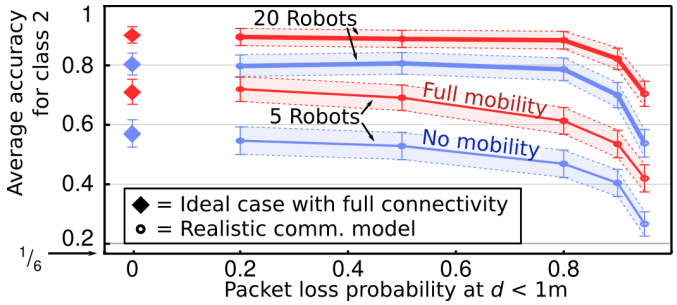


Figure 6.5. Effect of different packet loss probabilities with static and mobile swarms. Only the finger count 2 gesture is considered. Each data point is the average of 500 trials. 95% confidence intervals are reported, $T = 4$s, $\lambda_s = \infty$.

6.4.1.2 Effect of Communication Losses

The average accuracy for swarms of $N = [5, 20]$ robots with respect to their packet loss rates are reported in Figure 6.5. The results illustrate the specific impact of N on communications and show that larger swarms are quite robust to very high packet loss rates. To have approximately the same amount of observations (samples) 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 swarm's resiliency to unreliable communications is also illustrated in Figure 6.5. Large swarms show no significant decrease in performance for up to 80% of the packet loss probability.

6.4.1.3 Effect of Prioritization Strategies for Opinion Propagation

The different prioritization mechanisms for propagating opinions are shown in Figure 6.6 (bright bars). Three simple baseline approaches are compared with the optimizations proposed in Section 3.4.2.2. In all cases, each robot r broadcasts the current opinion of a single robot $s' \in S^r_{\text{new}} \subseteq S^r$ every 0.1s, where S^r denotes the set of all robots in the swarm known to r, and $S^r_{\text{new}} = \{s \mid s \in S^r, \mathbf{o}^s_{\text{cur}} \neq \mathbf{o}^s_{\text{last}}\}$ (i.e., S^r_{new} denotes the set of robots for which r has an updated opinion that has not yet been broadcast). The only difference among different strategies lies in how s' is selected in S^r_{new}. The baseline approaches that we consider are: (i) selecting s' in a purely random way, (ii) selecting s' as the robot whose opinion was most recently updated (LIFO), and (iii) selecting s' as the robot whose opinion was least recently updated (FIFO). These opinion selection approaches consist in selecting s' as the robot maximizing $I_1(s')$ or maximizing $I_2(s')$.

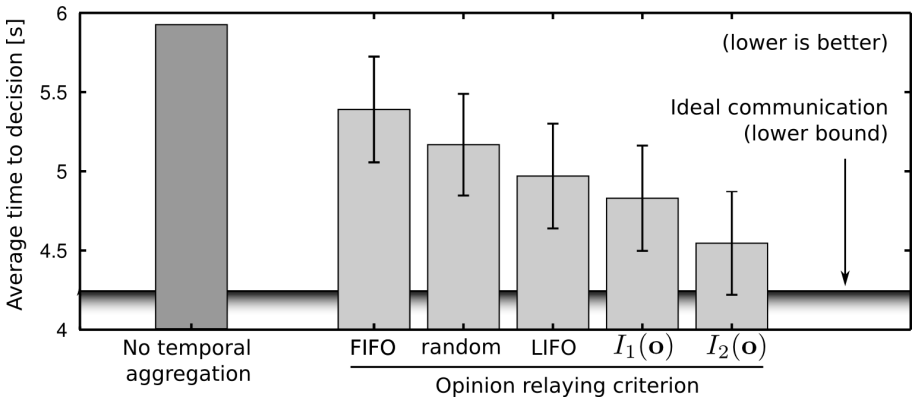


Figure 6.6. Time to decision as a function of message prioritization strategies. $N = 20$, $\lambda_s = 2$ and $T = \infty$ (i.e., all decisions are evidence-triggered). Each data point is the average of 50 trials, 95% confidence intervals are reported. All simulations result in a correct classification due to the large amount of collected evidence (large λ_s). Differences in time to decision are accountable to different prioritization mechanisms. A lower bound is reported which is obtained using an idealized communication model in which all opinions are instantaneously made available to all robots. The leftmost dark bar reports the results in the same scenario when the simplified protocol in [Giusti et al., 2012c] is used.

Among the baseline approaches, FIFO performs the worst and LIFO results in the best performance (shortest time to decision). This is because, in LIFO recent opinions (which normally contain more information) are given priority and tend to spread faster throughout the swarm. Significant performance improvements are observed when the proposed intelligent prioritization criteria are used. In

particular, prioritizing opinions of the robots that maximize $I_2(\cdot)$ yields the best decision speed at around 4.5s, which is close to the performance obtained with an idealized communication model in which all opinions are instantaneously shared among all robots in the swarm immediately after they are generated.

In the cooperative recognition protocol (see Section 3.4), multiple classifications from the same robot are aggregated over time into opinions, which are then propagated. In [Giusti et al., 2012c], we described a simpler approach in which each classification is propagated to the swarm individually: no time aggregation of information is performed before transmission (dark grey bar on the left of Figure 6.6). When the two approaches (with and without temporal aggregation) are compared as shown in Figure 6.6, the local temporal aggregation of information results in an improved swarm-level performance (light grey bars).

6.4.2 Performance of Data Fusion Approaches

The experimental results reported and discussed in this section evaluate the performance of different data fusion approaches for building swarm-level consensus decisions. The goal of this section is to identify the best data fusion method that can be used in conjunction with Algorithm 1. These results are emulation experiments that use the $K = 6$ finger count gesture dataset.

6.4.2.1 Swarm-level Accuracy based on Robot Positions

At first, we investigate the swarm-level classification performance of the different data fusion approaches presented in Section 3.4.3.1. Robot deployment positions are classified into: good, bad, and mixed (good and bad) positions. Good positions refer to locations that provide better quality of sensed information (e.g., facing directly in front of the human, central field of view, at a shorter distance from the human), bad positions refer to locations with worse sensing conditions (e.g., rear field of view, partial occlusions, excessive distance from the human), and mixed positions consist of both good and bad positions.

The number of cumulative prediction mistakes made by the swarm and individual robots (in different positions) are reported Figure 6.7, which is the outcome of a single typical run in our experiments. The abbreviations, AA, WdAA, WkAA, Freq, and Avg refer to, the Aggregating Algorithm, the Weighted Average Algorithm, the Weak Aggregating Algorithm, Frequency counting and Averaging respectively, as presented in Section 3.4.3.1. The noticeable repeating pattern in Figure 6.7 is that, the consensus performance with a swarm is better than that of individual robots and the Avg approach (that closely follows the Freq

approach) outperforms all other data fusion approaches. Also as expected, the performance of individual robots deployed in good sensing positions is better than that of robots deployed in bad and mixed positions.

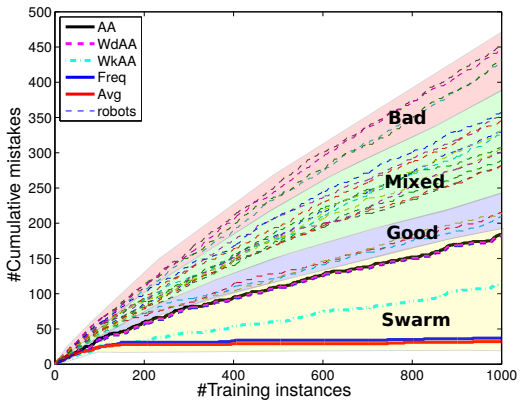


Figure 6.7. Cumulative mistakes vs. number of training (learning) samples for a 20-robot swarm. Individual robots in the swarm are deployed at different (good, bad, and mixed) sensing positions.

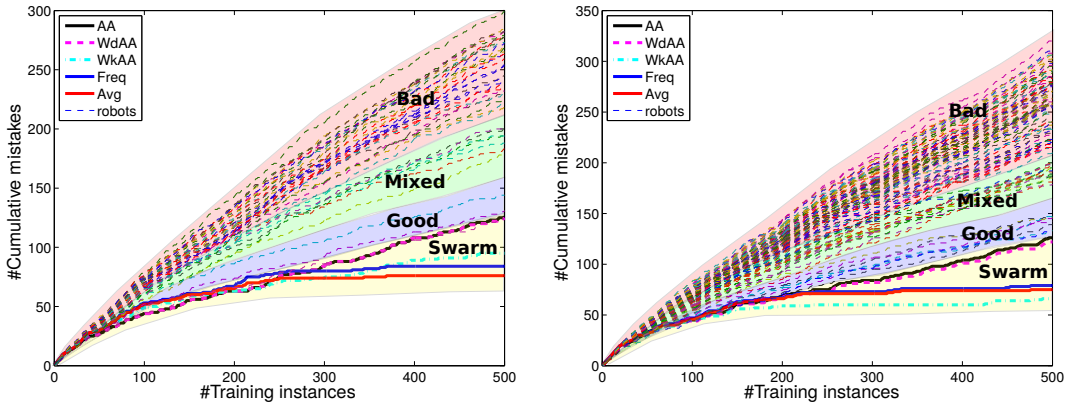


Figure 6.8. (a): A swarm of $N = 50$ robots. (b): A swarm of $N = 100$ robots. In both plots, robots are deployed at different (good, bad, and mixed) sensing positions. The majority of individual robots are located in bad sensing positions due which the number of cumulative mistakes is large.

Experiments with large size swarms are reported in Figure 6.8. More specifically, in Figure 6.8(a) swarm of $N = 50$ robots is emulated and Figure 6.8(b) shows results from a swarm of $N = 100$ robots. For the 50-robot and 100-robot swarms the three approaches, Avg, Freq, and WkAA closely follow each other and

provide the best performance in terms of the smallest number of cumulative mistakes. In the case of the 50-robot swarm the Avg approach clearly outperforms all other data fusion methods, and for the 100-robot swarm the WkAA approach outperforms all other approaches. For both plots in Figure 6.8, on average, there are more individual robots in bad sensing positions as compared to individuals in good positions, which results in a large number of mistakes made by individuals.

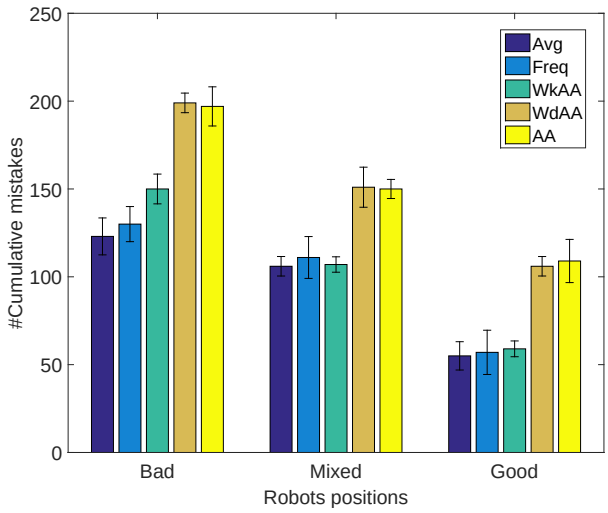


Figure 6.9. Impact of different deployment positions (good, bad, and mixed) on the cumulative mistakes made by a 13-robot swarm after 20 interaction rounds.

The impact of distinctive deployment positions is shown in Figure 6.9 using a swarm of $N = 13$ robots. These results are produced after emulating 20 interaction rounds using full feedback (see Section 5.3.1.2) and on each interaction round 20 gestures are shown to the swarm. These results are computed by averaging the results of 20 trials per experiment. Performance decreases when robots are deployed at bad sensing positions and vice versa for good positions. One interesting observation is that, in bad positions the improvement of averaging methods is the highest as compared to all other methods. As bad sensing positions result in a lower entropy (see Figure 3.5), averaging is most beneficial when the majority of robots produce classifications vectors with high uncertainty.

6.4.2.2 Impact of Swarm Size on Data Fusion Approaches

The impact of swarm size on different fusion methods is reported in Figure 6.10. Experiments have been performed using swarms of N=$[13, 26, 39, 52, 65, 91]$ robots. These results are computed by averaging 20 experimental trials. The

major observable is that, larger swarms yield less mistakes when making swarm-level consensus predictions as compared to smaller size swarms. Also, swarms with $N = [65, 91]$ robots have nearly similar performance. Since the acquired dataset has a maximum number of 65 viewpoints (see Section 6.2), the 91-robot swarm is emulated by assigning one viewpoint to more than one individual robot. As more than one robot acquires information from the same viewpoint, both swarms of $N = [65, 91]$ robots learn the same information with the only difference being that the 91-robot swarm learns highly correlated information.

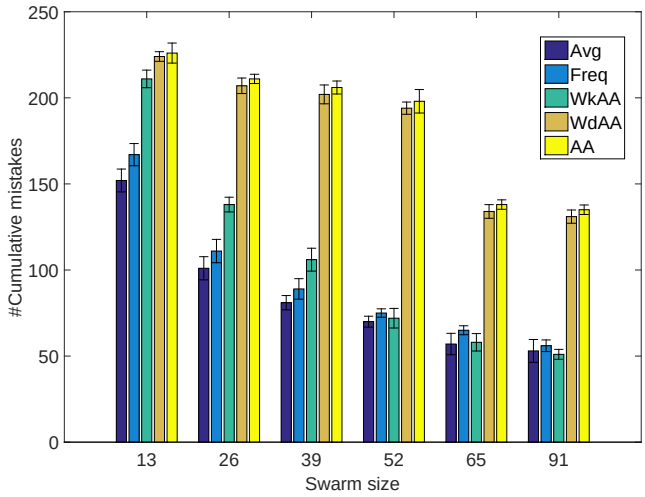


Figure 6.10. Effect of different swarm sizes ($N = [13, 26, 39, 52, 65, 91]$ robots) vs. number of cumulative mistakes after 20 interaction rounds.

6.4.3 Effect of Weighting on Opinion Fusion

This experiment is designed to study the effect of weighting opinion vectors. 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 independent (in the experiments d_m is set to 30°). The weight w_r of a robot r depends on the number its of neighbours and the positions of its neighbours closer than d_m. Let $Q(r) = \{q_1, q_2, \cdots q_{|Q|}\}$ be the set of such neighbours (referred to as *sensing neighbours*) and let $|Q(r)|$ denote its cardinality. We define w_r according to the following equation:

$$w_r = \frac{1}{\left(1 + \sum_{q \in Q(r)} c\left(d\left(r, q\right)\right)^2\right)} \tag{6.1}$$

where $c(\cdot)$ is a linear function of the distance between r and its neighbour is 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 gathered by two robots at the same position (viewpoint) and its value is experimentally set to $\rho = 2/3$. The use of eq. (6.1) to define w has several nice properties:

1. Function $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N$ that maps the position of the robots to weights is continuous.

2. If a robot $q \in Q(r)$ moves farther from robot r, w_r increases such that r's opinion gains importance.

3. It can be proven that the optimization of a robot's own weight consequently equalizes the distances between the robot and its neighbours.

According to eq. (6.1), a robot computes its weight by only knowing the angular distance to each sensing neighbour. With the Foot-bot platform the angular distance with respect to neighbouring robots can be computed by using the measures provided by the RAB system (see Section 1.2.1.1).

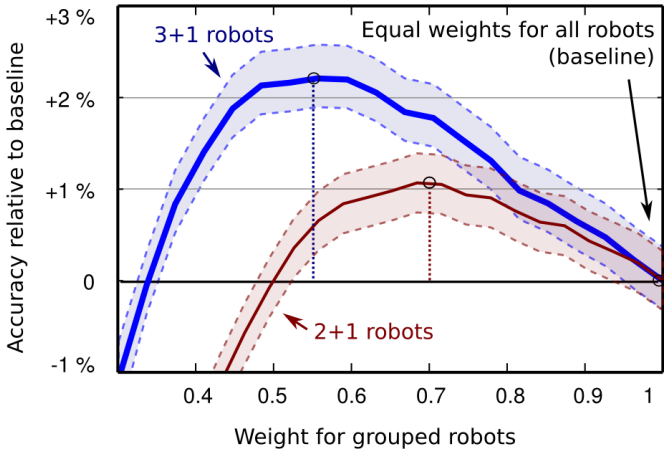


Figure 6.11. Effect of opinion weighting on accuracy. *Thin red line*: group of 2 robots + 1 isolated robot ($w = 1$). *Thick blue line*: group of 3 robots + 1 isolated robot ($w = 1$). 95% confidence intervals are reported.

To verify that the opinion weighting approach is meaningful, we design the following experiment. Three robots a, b, and c are considered all at the same distance d from the gesture. Robots a and b are very close to each other and share the same angular position $\theta_a = \theta_b = \theta'$, whereas robot c is at a much larger angular distance $\theta_c = \theta' \pm 30°$ (i.e., it is isolated from the other two robots). Each

robot outputs one classification vector which is fused with the weight of robot c fixed to the maximal value, while the weight of the grouped robots a and b is varied: $w_c = 1$ and $w_a = w_b = w_{ab} \in [0,1]$. In this way, we study the impact of different weightings for robots a and b which report correlated opinions. For each experiment trial, the swarm-level decision vector $\mathbf{D} = w_{ab}\mathbf{C}_a + w_{ab}\mathbf{C}_b + w_c\mathbf{C}_c$ is computed and the consensus result is considered successful if the largest element of $\mathbf{D}$ corresponds to the true class of the given gesture.

This experiment is performed by using as observations for robot a, b and c, three images belonging to the testing set that are acquired during the same time interval (i.e., timestep). In this way, we approximate the simultaneous acquisition of the same scene from three robots that are located at different viewpoints. This 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}. Experimental results are reported in Figure 6.11. The thin red line shows the average accuracy peaks are $w_{ab} \approx 0.7$. The improvement 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 a peak value $w = 0.7$ and solving for ρ in eq. 6.1 with one single neighbour at distance 0, yields $\rho \approx 2/3$ which is the value of ρ used. 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. 6.1.

6.4.4 Single-robot Performance based on Robot Position

This experiment investigates the classification accuracy of a single robot as a function of the robot's position. To perform this experiment, the $K = 6$ finger count dataset (see Section 6.2) is used and the classification vectors obtained from the entire dataset are scored. The results reported in Figure 6.12 (top) illustrate how effective a trained SVM classifier is in terms of recognition accuracy. 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 accuracy of classification vectors obtained from bad sensing viewpoints is extremely poor and barely larger than $1/K$ (where $K = 6$ represents the number of gesture classes), which is the performance of a random classifier. The performance systematically degrades with the increase of the angle with respect to the gesture and with the increase of the human-robot radial distance. Performance at the radial periphery is extremely poor due to the extremely bad viewpoint.

Robots closer to the gesture generally perform better than robots farther away, although distance has a smaller effect than the angle. From a larger distance the gesture appears smaller (i.e. covering less pixels), which makes the segmented hand masks less accurate due to the limited resolution of the camera, image noise, and segmentation inaccuracies. The dataset includes some viewpoint-dependent disturbances which affect accuracy: strong light sources which create problems to robots at angle $\theta = +60°$ and objects with similar colour as the gloves create segmentation issues.

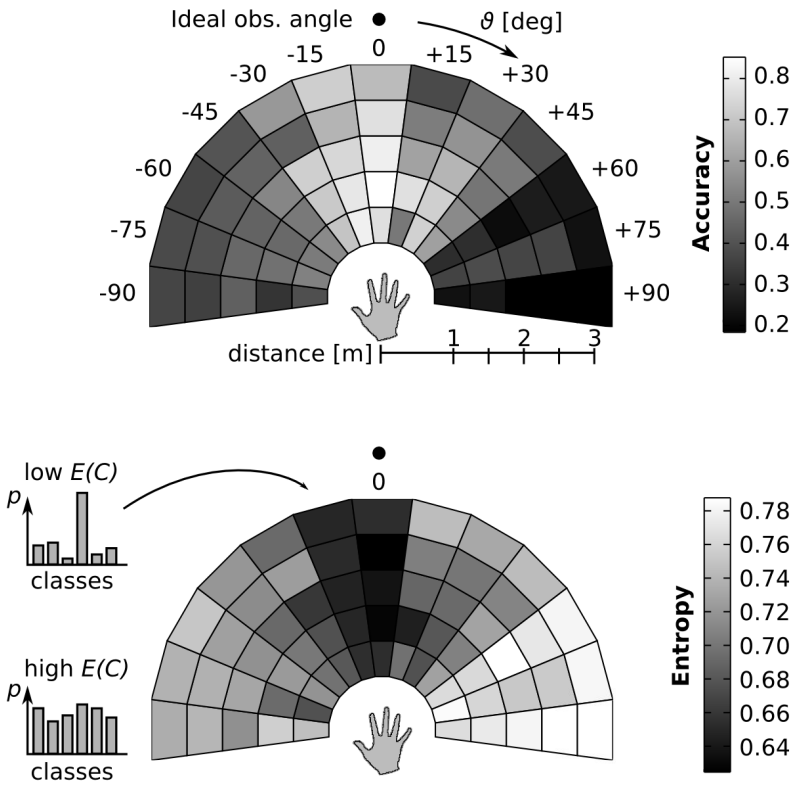


Figure 6.12. Top: Average accuracy as a function of robot position. Bottom: Normalized entropy of classification vectors obtained from different viewpoints.

In addition to classification accuracy, the entropy of the classification vectors is reported in Figure 6.12 (bottom). The impact of an opinion vector **c** in the consensus building process is determined by the opinion's weight and also by the relative differences among **c**'s components (gesture classes). In other words, the entropy of **c** precisely quantifies how much information is carried in the opinion. The values reported in Figure 6.12 (bottom) refer to the nor-

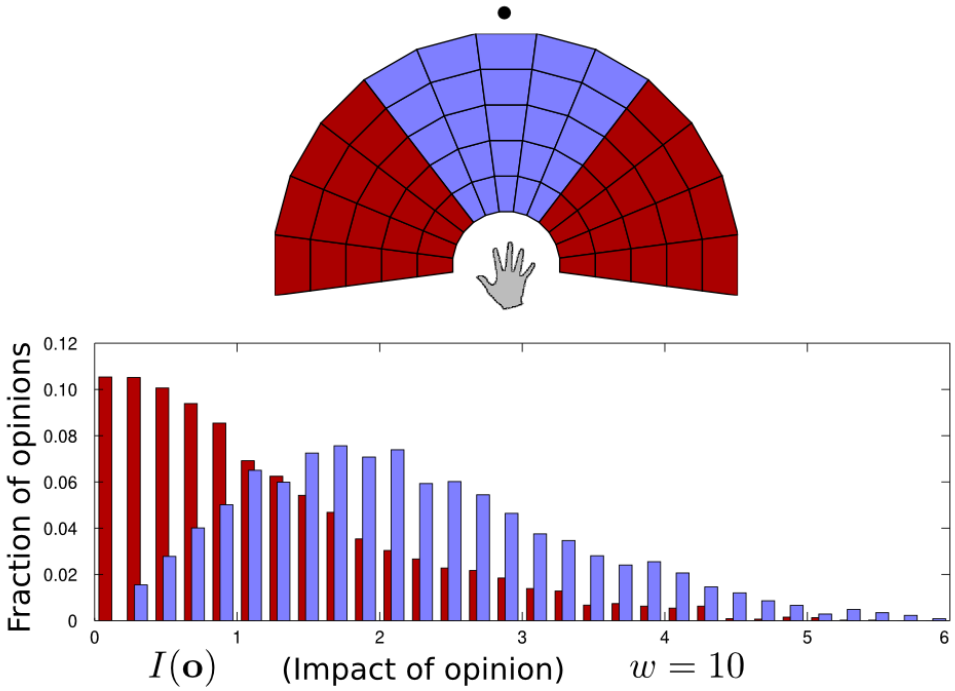


Figure 6.13. Opinions from bad viewpoints (dark red) have, on average, a lower impact $I(\mathbf{o})$ than opinions from good viewpoints (light blue). The histogram is computed by using 4000 opinions for each of the two groups. Each opinion is generated as the sum of $w = 10$ classification vectors that result from 10 consecutive gesture images acquired from a single viewpoint.

malized entropy $H(\mathbf{c})$, which is low only when the opinion strongly favours some gesture classes over others. For instance, $H(\mathbf{c} = \{0,0,1,0,0,0\}) = 0$ and $H(\mathbf{c} = \{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}) = 1$.

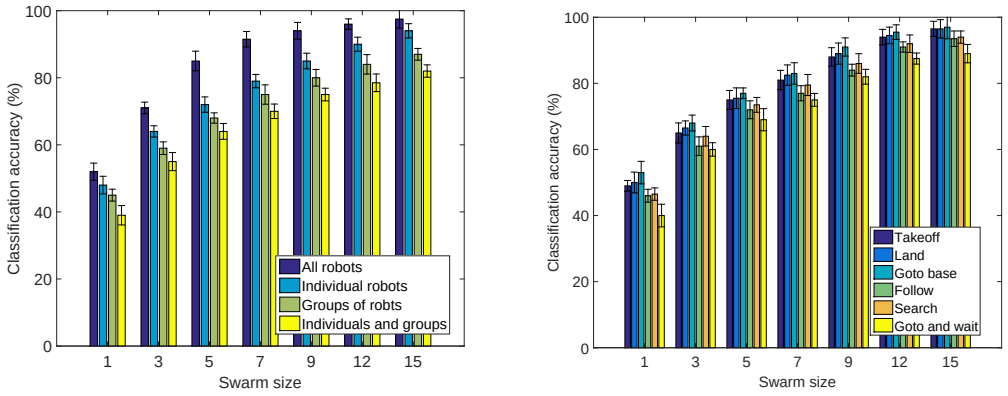
Classification vectors resulting from angled viewpoints exhibit, on average, a larger entropy than classification vectors from good viewpoints. As a result, in the consensus process the opinions of robots in good positions have a larger effect in determining the winning gesture class with respect to the other opinions. Similarly, correct opinions have classification vectors with an average entropy $H = 0.67$, which is significantly lower than the average entropy of wrong classification vectors $H = 0.76$. As classification vectors are aggregated into opinions and spread throughout the swarm network, Figure 6.13 shows that, on average, an opinion $\mathbf{o}$ generated at a good viewpoint has a larger impact $I(\mathbf{o})$ than an opinion from a bad viewpoint. This implies that the opinions from good viewpoints tend to propagate faster in communication-limited scenarios when the importance-based prioritization mechanisms in Section 3.4.2 are adopted.

6.5 Results for Chapter 2

This section presents the experimental results of Chapter 2 and evaluates the techniques and strategies for bidirectional human-swarm interaction and communication. These experiments consider the scalability of multi-robot interfaces [Velagapudi et al., 2008; Humphrey et al., 2007] and the metrics used by HMRI systems [Pourmehr et al., 2015; Olsen and Wood, 2004; Olsen et al., 2004]. Experiments are the results of real-time (online) testing with the Foot-bot robots (see Section 1.2.1.1). The experiments dealing with gesture recognition adopt the SVM classifier and use the entire vocabulary of $K = 16$ gestures.

6.5.1 Swarm-level Classification Performance of Words

At first, we evaluate the recognition performance of individual gestures (words) in the vocabulary using real robots. This experiment is performed with a single robot and with swarms of $N = [3, 5, 7, 9, 12, 15]$ robots. The results reported in Figure 6.14 are averaged over 50 trials (i.e., every bar represents the average of recognizing the same gesture 50 times). As expected, a larger swarm provides a positive effect and significantly improves the swarm-level classification accuracy.



(a) (b)

Figure 6.14. Recognition performance of gestures in Figure 2.4 using real robots. (a): Classification accuracy of robot selection gestures vs. swarm size. (b): Accuracy of application-specific gestures vs. swarm size.

The classification accuracy of the robot selection gestures (see Figure 4.1) is shown in Figure 6.14(a). It is observed that, spatial pointing gestures are more difficult to classify as compared to the non-spatial gesture in Figure 4.1(d) which

used to select all robots in the swarm. Figure 6.14(b) reports the classification accuracy of the $K = 6$ application-specific gestures in the vocabulary (see Figure 2.4(b)). The *takeoff*, *land*, and *go to base* gestures are easier to recognize as compared to the *follow*, *search*, and *go to and wait* gestures. The *go to base* gesture has the highest classification performance and can be easily recognized as compared to all other gestures. This simply is because, the *go to base* gesture represents a one-handed command that requires a correct classification from a single hand, while all other gestures are two-handed are require correct classifications from both hands. The *go to and wait* and *follow* gestures have the lowest recognition accuracy due to their complex shapes.

6.5.2 Swarm-level Accuracy for Classifying Sentences

This section evaluates the performance of robot swarms for classifying full sentences from the gesture language. We setup five different experiments that cover a wide range of scenarios and effectively measure the performance of the gesture language and the swarm-to-human feedback. The five experiments $(E_1, E_2, ..., E_5)$ act as indicators and are presented below.

Experiment E_1: In the first experiment, the classification performance of the gesture language is investigated under normal operating conditions. Performance is evaluated using swarm-to-human feedback that has four typical recognition outcomes: properly recognized, not properly recognized, inappropriate and undefined (see Section 2.3.2). In this experiment, two multi-class SVM classifiers are used to classify an individual gesture (word). The classification performance of a full sentence is evaluated by using 6 classifiers $[C_{F1}, C_{F2}, ..., C_{F6}]$ (see Section 2.3.2.2). For instance, to classify a gesture to select robots, C_{F1} and C_{F5} are used, and to classify an application-specific gesture, C_{F2} and C_{F5} are adopted.

Experiment E_2: In this experiment, the recognition performance of the gesture language is investigated in the case when the grammar is ignored in the sentence. When the grammar is ignored, each individual gesture (word) is recognized by using classifiers that are trained on a combination of gestures from multiple semantic classes (see Section 2.2.2). In simpler words, a single classifier is used to recognize an individual gesture in a sentence. In practice, this experiment uses only 3 classifiers to recognize full sentences, namely, C_{F4}, C_{F5}, and C_{F6} (see Section 2.3.2.2). Each of the three classifiers are trained on gestures that are associated with any two semantic classes.

Experiment E_3: This experiment investigates the scenario in which the swarm does not provide feedback to the human after the recognition of an individual gesture in a full sentence. As no acknowledgement is given to humans after the recognition of gestures, humans present gestures with a natural timing. The swarm does not inform the human when the gesture was recognized and when the next gesture in the sentence is expected. Only at the end of the sentence the human can observe the final classification result and check whether the full sentence was properly recognized or not. In practice, a single classifier is used to recognize an individual gesture in every semantic class. To recognize sentences 4 classifiers are used, one classifier for each semantic class. Every classifier is trained on the set of gestures that are associated only to that semantic class.

Experiment E_4: This experiment evaluates the performance when one random gesture in every semantic class is replaced with a gesture that is not defined in the gesture language, namely an *undefined* gesture. If the full sentence is properly recognized including the undefined gesture, then the classification outcome is considered correct, otherwise in all other cases the recognition outcome is considered incorrect. To recognize an individual gesture 2 classifiers are used in every semantic class, and to classify full sentences all 6 classifiers are used.

Experiment E_5: In the final experiment, one random gesture in every semantic class is replaced with a gesture which is in the defined gesture language but outside the set of expected gestures, namely an *inappropriate* gesture. For instance, when a swarm requests a human to present a gesture to select robots and the human provides a direction for robots to move, the command provided by the human is considered inappropriate based on the semantic class requested by the swarm. If the entire sentence is properly recognized including the inappropriate gesture, then the recognition outcome of the swarm is considered correct, while in all other cases it is considered incorrect. For classifying an individual gesture 2 classifiers are used and to recognize full sentences all 6 classifiers are adopted.

The results of all five experiments with respect to different swarm sizes are reported in Figure 6.15. To perform these experiments, a set of 20 predefined sentences are chosen (see Section 2.2.3.4) which include the $K = 16$ gestures in the vocabulary. Each of the 20 sentences are evaluated with the five experiments, and in this way each experiment evaluates the performance of nearly 50 gestures. The first observable from Figure 6.15 is that, the larger the swarm size, the higher will be the classification performance for all five experiments.

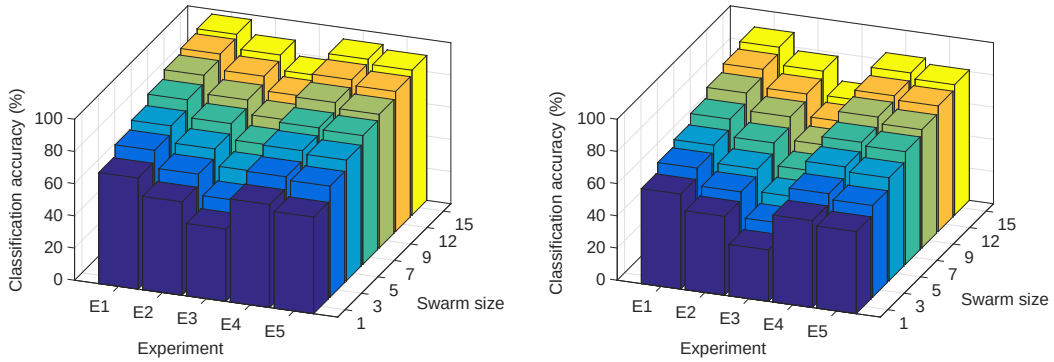


Figure 6.15. Results for the swarm-level recognition of the five experiments ($E_1, E_2, \ldots, E_5$) vs. swarm size, using real robots. (a): Classification accuracy of gestures (words). (b): Accuracy of sentences composed of 2 to 4 words.

The classification performance of words (individual gestures) is shown in Figure 6.15(a) and the classification accuracy of sentences composed of 2 to 4 words is given in Figure 6.15(b). The second noticeable fact from both plots is that, E_1 has the best performance and E_3 has the worst performance. E_3 is the most challenging experiment because the swarm does not provide acknowledgement (feedback) to the human when the human should stop showing the gesture and when the next gesture needs to be presented. The performance of E_2 is lower than E_1, because E_2 does not use classifiers that are trained for each semantic class in the grammar. Instead classifiers in E_2 are trained using gestures associated with multiple semantic classes which makes it more difficult to properly recognize gestures. The performance of E_4 (undefined gestures) and E_5 (inappropriate gestures) is significantly good and similar to E_1. Even in the most difficult settings such as E_2 and E_3, a 70% classification accuracy for individual gestures and a 50% accuracy for full sentences is obtained.

The recognition accuracy of words (individual gestures) and sentences (composed of 2 to 4 words) using E_1 with $N = 15$ robots is reported in Figure 6.16. These results are produced using 20 trials and on each trial the same set of individual gestures and sentences are evaluated. For $N > 10$ robots, the classification accuracy of words and full sentences is greater than 80%. The average accuracy for recognizing words is greater than that for sentences, even though there are 50 words on average and the total number of sentences is less than $1/2$ of the total number of words. As the classification accuracy increases with the increase in swarm size, the variation in the accuracy decreases which is indicated by the

boxes that get smaller in height at larger swarm sizes. With large swarms the classification accuracy of the gesture vocabulary can be reliably determined.

6.5.3 Time Taken for Interaction and Recognition

As interaction time is an essential metric in HRI efficiency [Crandall and Cummings, 2007a,b] and designing interfaces that have smaller interaction times is a strong theme in HRI [Goodrich and Olsen, 2003], we investigate the time taken by a swarm of Foot-bots to recognize individual gestures and full sentences.

The research team at the Autonomy Lab of Vaughan [Pourmehr et al., 2015] identified that, the time taken to interact with multiple robots depends upon the: interface design, communication method and strategy, physical workspace, spatial arrangement of the robots, and time needed by the swarm to sense and recognize inputs given by humans. We consider that, the *total interaction time* t_{int} required for a recognizing an individual gesture or a full sentence can be represented by $t_{int} = t_{human} + t_{swarm}$, where t_{human} represents the time during which the human interacts with the swarm and t_{swarm} is the time during which the swarm conveys multi-modal feedback to the human.

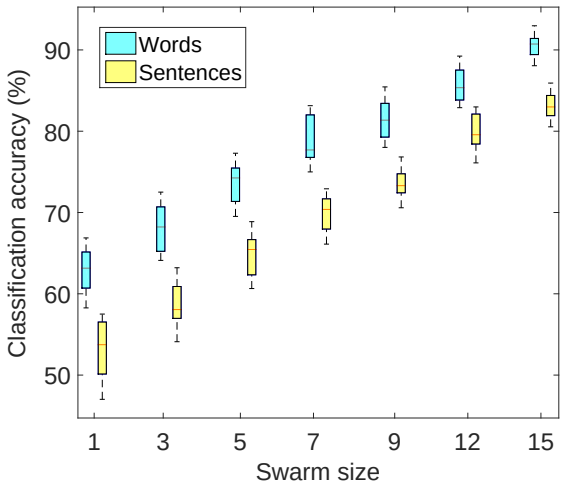
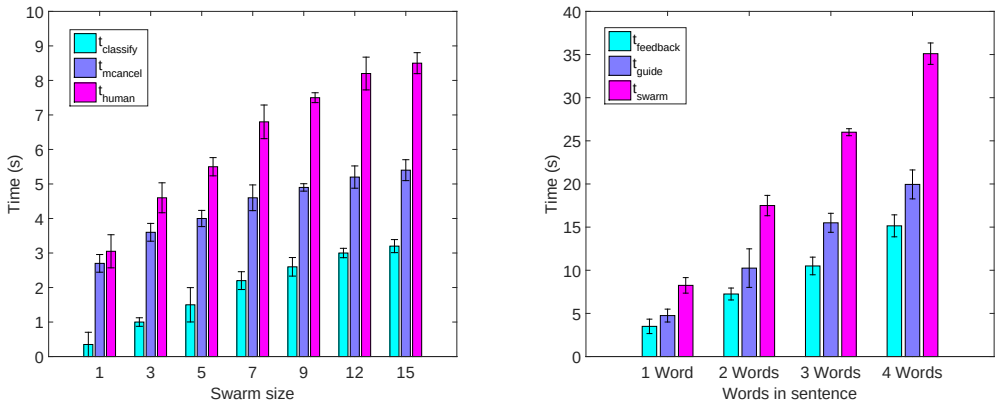


Figure 6.16. The classification accuracy of E_1 for recognizing individual gestures and sentences composed of 2 to 4 words using $N = 15$ robots.

The time during which the human interacts with a swarm is then considered as, $t_{human} = t_{mcancel} + t_{classify}$, where $t_{mcancel}$ represents the time taken by the swarm to identify that human motion is cancelled so the sensing of the gesture can begin, and the time taken by the human to provide a gesture and settle

(finalize) on a gesture. Time $t_{classify}$ represents the time taken by the swarm to: sense a gesture, obtain classification results from individual robots, and fuse individual robot opinions to produce a swarm-level consensus decision.

Similarly, the time taken by the swarm to interact with the human is represented by, $t_{swarm} = t_{feedback} + t_{guide}$, where $t_{feedback}$ represents the time during which the swarm conveys multi-modal (audio/visual) feedback to the human regarding the swarm-level classification outcome of a gesture. Time t_{guide} represents the time taken by the swarm to: guide humans through the interaction process (i.e., request humans to provide a gesture for a specific semantic class), and to convey to humans the classification outcome of fully recognized sentences.



(a) (b)

Figure 6.17. (a): Time taken by the human to interact with swarm t_{human} vs. swarm size, to classify individual gestures (words). (b): Time taken by a swarm of $N = 15$ robots to convey feedback to the human after recognizing words and sentences composed of 2 to 4 words.

6.5.3.1 Components of Interaction Time

In this experiment we investigate the time required to recognize individual gestures (words) and sentences from the gesture language. These experiments are performed by randomly evaluating 20 words and sentences, and averaging the classification results over 20 trials. Humans present gestures to swarms with a smooth and natural timing (i.e., gestures issued by humans are not very fast neither too slow). Experimental results are reported in Figure 6.17. As expected, Figure 6.17(a) reports that, as the swarm size increases the time taken to cooperate and reach a swarm-level consensus is longer. Also, Figure 6.17(b) shows that sentences with more words require a larger interaction time.

For the classification of an individual gesture, the time t_{human} taken by a human to interact with swarms of different sizes is shown in Figure 6.17(a). A single robot provides a faster recognition outcome as compared to a swarm. This is because, in the case of a swarm, the cooperative recognition protocol requires a small amount of time to produce a swarm-level recognition outcome, but for a single robot the decision is issued instantly after gesture classification is complete. Using a swarm of $N = 15$ Foot-bots, time $t_{classify}$ is nearly 3s, time $t_{mcancel}$ is roughly 5s, and time t_{human} is less than 9s. These times are reasonable considering the limited computational capabilities of the Foot-bots.

The time t_{swarm} taken by a swarm of $N = 15$ robots to recognize a single gesture and sentences composed of 2 to 4 words is reported in Figure 6.17(b). As observed, time t_{swarm} scales almost linearly with respect to the number of words in a sentence. This is because $t_{feedback}$ and t_{guide} are almost the same for any individual gesture (word) in entire the vocabulary.

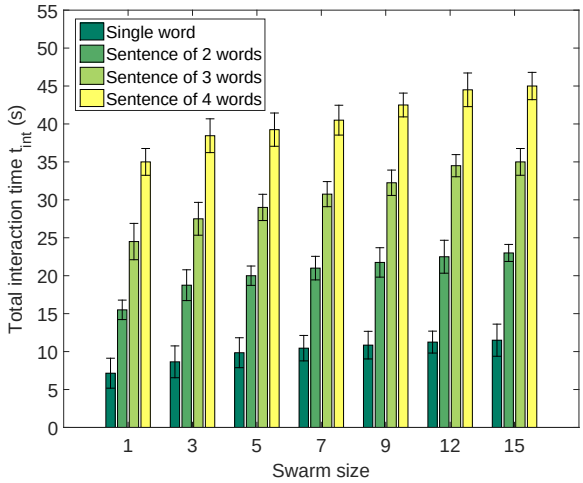


Figure 6.18. The total interaction time t_{int} taken by a swarm to recognize words and sentences composed of 2 to 4 words vs. swarm size.

6.5.3.2 Total Time Taken for Word and Sentence Recognition

This experiment investigates the total interaction time $t_{int} = t_{human} + t_{swarm}$ taken by the different swarm sizes to recognize individual gestures (words) and full sentences. Experimental results are shown in Figure 6.18. Using a swarm of $N = 15$ Foot-bots, an individual gesture (single word) roughly takes 10s and sentences composed of 2, 3 and 4 words require, 25, 36 and 44s respectively. This suggests that sentences composed of 5 words may take up to 1 minute.

On average, with a swarm of robots, time t_{int} for an individual gesture (word) is approximately 10 to 12s. This amount of time is considered rational since gestures are given with a natural timing. Evidently, time t_{int} can be minimized if humans present gestures with a fast timing.

6.5.4 Effect of Uncertain Swarm-level Recognition Decisions

This experiment reports the effect of uncertain (not confident) swarm-level consensus decisions based on the *average probability difference* P_{avg} between two supervised classifiers. Within uncertain decisions, we investigate the effect of *not properly recognized* and *undefined* decisions (see Section 2.3.2.1). By selecting $P_{avg} = 0.5$ as the baseline for *uncertain* decisions (see Section 2.3.2.2), we consider that, not properly recognized decisions lie within the range $P_{avg} > 0.5$ and undefined decisions are within the range $P_{avg} < 0.5$.

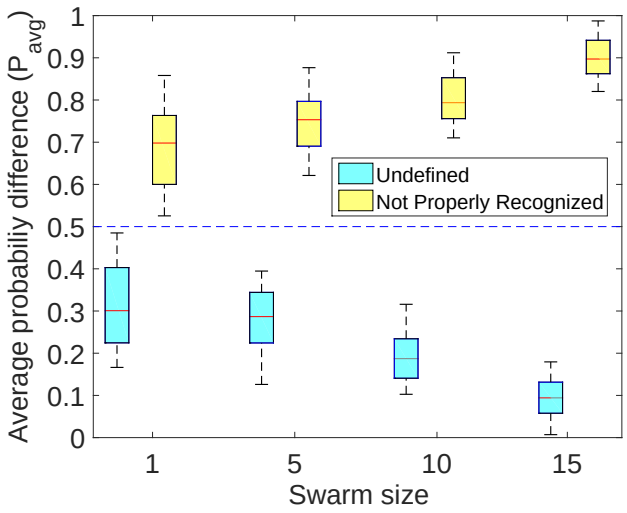


Figure 6.19. The distribution of *average probability difference* P_{avg} values for undefined and not properly recognized decisions vs. swarm size.

In this experiment, we evaluate the distribution of P_{avg} values for different swarm sizes. To perform this experiment, a set of 20 full sentences is evaluated which results in P_{avg} values that correspond to 35 undefined and 38 not properly recognized decisions. The experimental results are reported in Figure 6.19. In the case of a single robot, the majority of not properly recognized decisions lie in the range between $0.5 < P_{avg} \leq 0.85$ (mean at 0.7), undefined decisions are within the range $0.2 \leq P_{avg} < 0.5$ (mean at 0.3), and the distribution of P_{avg} values is large as represented by the tall boxes.

The decisions produced by smaller size swarms (e.g., $N = [1, 5]$ robots) on average have a distribution (spread) of P_{avg} values closer to the baseline (0.5). This is because, swarms with a relatively few number of robots have higher uncertainties in their decisions. As the swarm size increases the distribution of the decisions shifts from $P_{avg} \approx 0.5$ to $P_{avg} \rightarrow 0$ (for undefined) and $P_{avg} \rightarrow 1$ (for not properly recognized) which is indicated by the tall boxes that shrink in size (get smaller in height). For swarms of moderate sizes (e.g., $N = 15$ robots) the majority of not properly recognized decisions have on average $P_{avg} = 0.95$ and undefined decisions have $P_{avg} = 0.1$. This indicates that large swarms yield decisions that are well separated from each other.

6.6 Results for Chapter 4

This section reports the experimental results of Chapter 4 and evaluates the performance of the developed algorithms and techniques. Experiments performed are the results of emulation tests that use the dataset of $K = 3$ robot selection gestures and the $K = 6$ finger count gestures. For the recognition of spatial gestures, one multi-class and one binary-class SVM is adopted, and both classifiers are trained using hand-crafted features.

6.6.1 Swarm Understanding Performance for Robot Selection

To verify the robustness and performance of the algorithms and strategies that select robots from a swarm, several experiments using different spatial configurations of individuals and groups of robots are performed. In every configuration, a human operator attempts to select an individual robot or a groups of robots. Experiments are performed by selecting subsets of spatial gesture images from the dataset and evaluating Algorithms 2 and 3. The results indicate robot selection accuracies as grayscale color maps. All experiments are averaged over 100 trials using images from similar spatial configurations of the robots on each trial.

6.6.1.1 Effect of Individual Selection Scores

In this experiment, we study the effect of the individual selection score $r_{indscr} \in [0, 1]$ on surrounding, non-selected robots. The experimental results are reported in Figure 6.20 (top). For this experiment two configurations of two individual robots are considered. In the first configuration the two individual robots are very close to each other (top right), and in the second configuration the two

individual robots are far apart from each other (top left). All surrounding robots are uniformly spread around in the environment.

The gray colormap in Figure 6.20 (top) illustrates the individual selection score r_{indscr} for all robots that surround the two selected robots. The positions (cells) with dark colors represent surrounding robots that have large values of r_{indscr} (as they are very near to the two selected individuals), while surrounding robots with light coloured cells represent that these robots are far from the two selected individuals. As expected, when robots are positioned within close proximity of each other the success rate of selecting individual robots decreases.

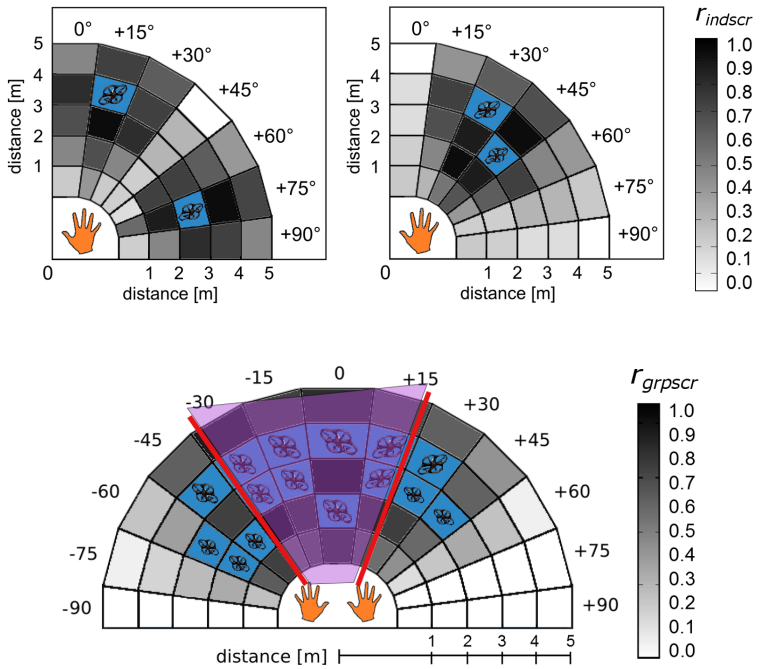


Figure 6.20. Top: Individual robot selection scores r_{indscr} computed using Algorithm 2 for two robots close to each other (left) and the two robots far from each (right). Bottom: Group selection scores r_{grpscr} computed using Algorithm 3.

6.6.1.2 Sensitivity of Group Selection Scores

This experiment investigates the sensitivity of the group selection score r_{grpscr} on surrounding, non-selected robots. Experimental results are shown in Figure 6.20 (bottom). As shown by spatial configurations of robots, we select a group of 8 robots (cells with blue background) from a swarm of $N = 15$ robots while the non-selected robots are placed uniformly in each one of the remaining cells. The gray colormap in Figure 6.20 (bottom) represents the group selection score r_{grpscr}

for all deployed robots. The dark color cells represent surrounding robots with values of r_{grpscr} that are similar to the selected group, while surrounding robots with light coloured cells indicate that these robots are far from the selected group.

Table 6.1. Accuracies for selecting spatially-located individuals and groups of robots from a swarm using an incremental and simultaneous selection approach.

Configuration	Incremental	Simultaneous	Success
I_1I_2	2/2	-	98%
G_1G_2	-	2/2	96%
$I_1I_2G_1$	2/2	1/1	91%
$I_1I_2G_1G_2$	2/2	2/2	87%
$I_1I_2I_3G_1G_2$	2/3	1/2	82%
$I_1I_2I_3G_1G_2G_3$	3/3	2/3	76%
$I_1I_2I_3I_4G_1G_2G_3$	3/4	3/3	68%
$I_1I_2I_3I_4G_1G_2G_3G_4$	3/4	2/4	62%
Total Accuracy	**85%**	**76.5%**	**82.5%**

It is observed that the success rate in selecting a group strongly depends upon: (i) the angular distance between the robots to be selected and (ii) the distance to other surrounding (neighbouring) robots. When a group of robots that needs to be selected is within close proximity of other groups or individuals, there is a high chance that undesired robots may get selected in the group. This is because, when undesired robots are in close proximity r_{grpscr} of undesired robots will be similar to that of the group being selected.

6.6.1.3 Effect of Incremental and Simultaneous Selection

This experiment investigates the success rate for selecting a desired number of individuals and groups of robots using an incremental and simultaneous selection approach. In this experiment, we use 8 different scripted configurations attempted by a human operator to select individuals and groups from a swarm of $N = 15$ robots. The results are summarized in Table 6.1, where I_i and G_i indicate that the human issued a gesture to select the ith individual or group respectively. Individual and groups of robots are indexed from 1 to 4.

It is observed that, on average, the accuracy for selecting individuals (incremental selection) is the highest at 85% and the performance for selecting groups of robots (simultaneous selection) is the lowest at 76.5%. The accuracy for selecting groups is low because after two groups have been selected (i.e., G_1G_2),

there is a high probability that the third and fourth groups to be selected (i.e., G_3 and G_4) will contain undesired robots (i.e., robots that are not meant to be selected). The total success rate for the combined selection of individual and groups over all selection trials is 82.5% which is considered significantly good for combined incremental and simultaneous selection.

6.6.1.4 Effect of Swarm Size on Selection of Individuals and Groups

In this experiment, we investigate the effect of swarm size on the selection accuracy of individuals and groups of robots. This experiment is the average result of the 50 selection trials, where on each trial robots are positioned in different spatial configurations. The effect of the selection accuracy on different swarm sizes is reported in Figure 6.21 for all robots, individuals, groups, and individuals inclusive of groups. As expected, large swarms have a negative impact on the selection accuracy since it decreases with the increase in swarm size.

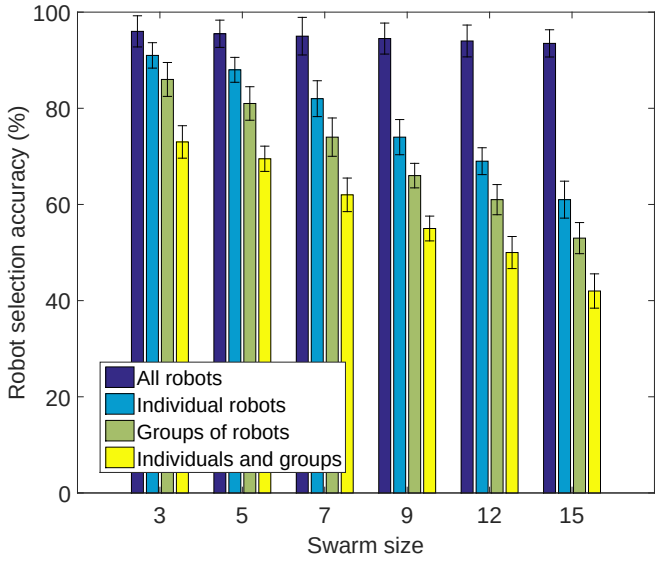


Figure 6.21. The accuracy for selecting, all robots, individual robots, groups of robots, and individuals inclusive of groups vs. swarm size.

The accuracy for selecting of individuals and groups from large swarms (e.g., $N >= 10$ robots) is relatively poor. With a swarm of $N = 15$ robots the selection accuracy of individuals and groups is lower than 60%. On average, the selection accuracy of individual robots is significantly better than the selection accuracy of groups. This is because, individual robots have a wider spatial workspace (i.e.,

the selection of individual robots is robust to a wider set of mutual poses). This indicates that the spatial configurations of individual robots have a reduced effect in relation to the size of the swarm.

6.6.2 Effect of Mobility Rules for Swarm Deployment

This section reports the performance of the spatially-aware deployment techniques and mobility strategies that reshape the spatial distribution of swarms (to obtain better sensing coverage of gestures) and provide human-relative localization. Performance is evaluated both with UGVs and UAVs. Emulation experiments are performed using a dataset of gesture and face images (see Section 6.2).

6.6.2.1 Impact of Mobility on UGVs and UAVs vs. Recognition Accuracy

This section investigates the effect of mobility strategies on the swarm-level classification of gestures. The average swarm accuracy obtained with and without the use of mobility strategies for different swarm sizes is reported in Figures 6.22 and 6.23. These results indicate that the use of swarm mobility provides significant improvements in the swarm-level consensus performance.

The swarm-level accuracy with respect to the time required to issue decisions (which is set using the prudence parameter λ_s) in the case of mobility and without mobility, is shown in Figure 6.22. 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 time to decision is therefore constant in all simulations and accuracy depends on the quality of observations and the efficiency of the opinion propagation mechanism. With the use of mobility strategies the time to reach a swarm-level decision is approximately 20% less as compared to the case with no mobility.

The effect of different mobility strategies and different swarm sizes is reported in Figure 6.23. Positive effects of mobility on swarm-level accuracy are observed: in all settings, the difference between bar a (no mobility) and b (mobility according to the rules in Section 4.3.1) is statistically significant according to the Wilcoxon paired signed-rank test $p < 0.01$. On one hand, mobility generally improves communication for small swarm sizes since it tends to group robots ensuring a more efficient way of multi-hop propagation of messages. On the other hand, some mobility strategies may result in topologies which negatively affect communication ability. For instance, when robots follow mobility Rules 1 and 2 (see Section 1.2.1.1) they tend to break the line-of-sight communication with most of their neighbours except for the two closest neighbours.

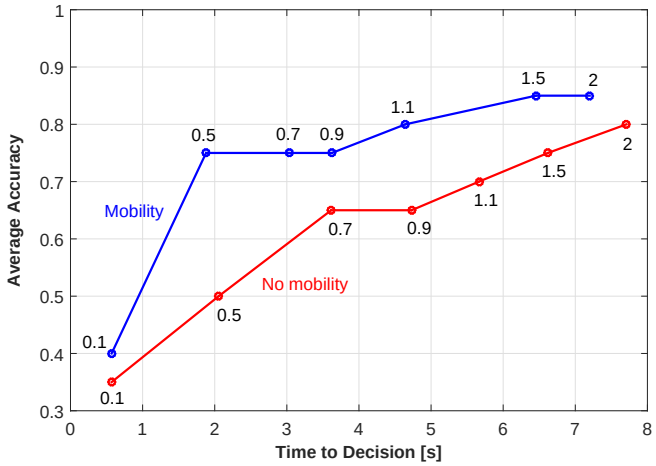


Figure 6.22. Swarm accuracy vs. time to decision as a function of prudency λ_s using $N = 10$ robots with and without mobility and averaged over 20 trials.

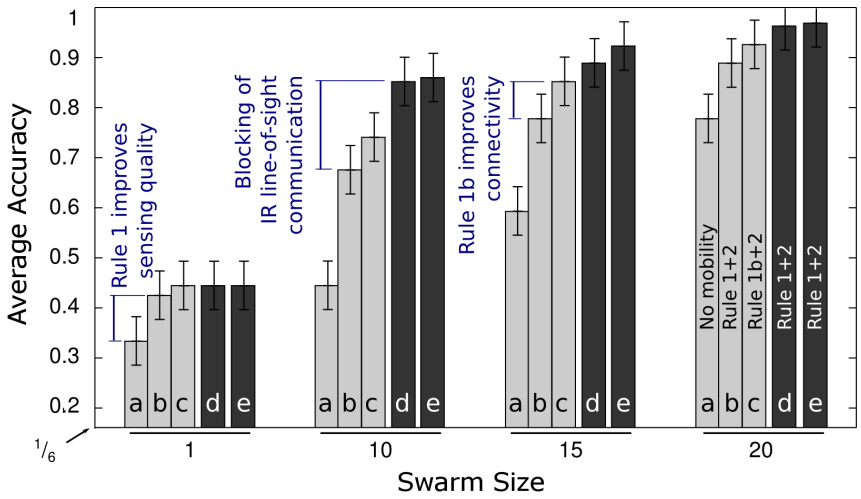


Figure 6.23. Accuracy obtained with swarms of $N = \{1, 10, 15, 20\}$ robots when using different mobility strategies, a fixed time threshold $T = 5$s, and $\lambda_s = \infty$ (all decisions are time-triggered). Each data point is obtained as the average of 100 trials, and the 90% confidence interval is reported. Bar series *a*, *b* and *c*: realistic IR communication model in which two robots communicate only if no other robot blocks their line-of-sight. Bar series *d*: line-of-sight constraint is ignored. Bar series *e*: *XBee* radio communication model (which ensures full connectivity). Bar series *a*: no mobility. Bar series *b*, *d*, *e*: mobility according to Rules 1 + 2. Bar series *c*: mobility according to Rules 1b + 2.

Differences in accuracy are accountable to the quality of observations and the efficiency of opinion propagation. In Figure 6.23 the accuracies obtained with the three mobility scenarios are: no mobility (bar series *a*), mobility according to Rules 1+2 (bar series *b*), and mobility according to Rules 1b+2 (bar series *c*). In the last case, not all robots reach the optimal sensing distance from the gesture as they avoid a perfect semi-circular alignment and avoid occluding each other. Gesture images acquired by the robots are expected to be of low quality but information flows between the robots is improved, which results in an increased accuracy indicated by the gap between bar series *b* and *c*. Bar series *d* shows the accuracy obtained by robots implementing mobility when line-of-sight occlusions do not block communication. The difference of bar series *b* with respect to all other bar series is that, bar series *b* quantifies the loss of accuracy due to broken line-of-sight communication links. Lastly, bar series *e* reports performance obtained with an XBee-based radio communication model which is simulated using the NS-2 simulator. XBee-based communications do grant full connectivity and do not incur bandwidth bottlenecks, however, they do not result in a significant advantage over the less powerful mechanisms in bar series *c* and *d*.

6.6.2.2 Face Pose Estimation Performance using UAVs

This experiment shows the performance of the face pose estimation system for human-relative localization with UAVs (see Section 4.3.2). Observations sampled from the dataset of face images (see Section 6.2) are used for online learning and prediction using the LWPR method (see Section 4.3.2.3). The dataset is partitioned into training and testing sets, where 30% of the samples are used for training and the remaining 70% are used for testing and validation. In regression-based learning $\mathbf{x}_i = \{S_c^i, S_r^i, S_l^i, d^i\}$ of image i represents a set of four *face pose features* and $y_i = \phi_i$ denotes its respective *target label*. Using a non-linear Gaussian kernel, LWPR maps these features into a *face pose* ϕ which is projected onto a $[0, 180^\circ]$ semi-circular plane (with distance d serving as a normalization factor). An ordered pair (ϕ, d) computed from a face image represents the *angular distance* between a human's face and a UAV (see Section 4.3.2.2).

The face pose estimation accuracy of a single robot is investigated as a function of its angular distance (r_ϕ, r_d). Using a subset of images from the dataset, the *average pose accuracy* is computed using GT information which is the difference between the actual and predicted angular distances. Results are reported in Figure 6.24. UAVs located at distances between $d = [1, \ldots, 3]$m in the central sensing positions provide good accuracies up to 92%. With the increase in the human-robot radial distance (e.g., $d \geq 3$m) performance systematically degrades

as the face cannot be reliably detected at larger distances. A distance between $2 \leq d \leq 3$m is considered a safe proximity to interact with airborne UAVs.

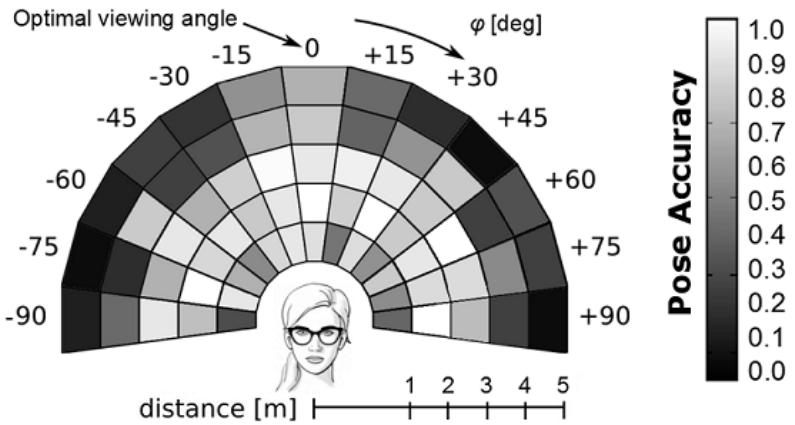


Figure 6.24. Face pose detection accuracy of a single UAV as a function of the angular distance (ϕ, d) with respect to the face pose.

6.7 Results for Chapter 5

This section presents the experimental results of Chapter 5 and evaluates the performance of the developed swarm-level learning algorithms. All experiments reported below are the results of emulations (that use the dataset of $K = 6$ finger count gestures) and real-time testing with the Foot-bots.

6.7.1 Learning Cooperatively with Information Sharing

The performance of the distributed cooperative learning strategy is investigated in this section. For the learning and recognition of gestures the multi-class SVM classifier is adopted which is trained using hand-crafted features.

6.7.1.1 Experimental Scenario

Each simulation run starts at the beginning of the initial learning phase in which every robot is initialized with an *empty training set* and placed at a random position. Initially, $s_{init} = 5$ samples are acquired from each of the $K = 6$ finger count gestures by every robot. In this way, a total of $T_i = 5 \times 6 = 30$ samples are acquired by every robot for the *initial training phase* (see Section 5.3.1.1). After hand-crafted features are computed from the acquired T_i samples, each

robot broadcasts a subset of the acquired samples (i.e., computed feature vectors). The value of parameter $B \in [0,1]$ defines the fraction of newly acquired samples which are disseminated by a robot. As an example, for $B = 0$ robots do not communicate. For $B = 1$ robots exchange all acquired data and share the same training set at any moment in the simulation. For $B = 0.1$ each robot shares $0.1 \times 30 = 3$ training samples. When $B \in (0,1)$ the samples to be shared are selected according to one of the three strategies presented in Section 5.4.2.1.

After information sharing is complete, the training set of each robot contains $KM + BKM(N-1)$ training samples, where N represents the number of robots. As an example, for $B = 0.2$ robot r in a swarm of $N = 10$ robots will have 84 samples in its training set T_r: 24 samples in T_r^p (still unknown to the rest of the swarm), 6 samples in T_r^s (already disseminated to the rest of the swarm), and 54 samples in T_r^e (received from the other robots in the swarm). The 60 samples in $T_r^s \cup T_r^e$ represent the current *common knowledge* of the swarm.

When the initial training phase is complete the *interaction rounds* begin (see Section 5.3.1.2), in which 150 random gesture commands are given by the human to the swarm. Before each gesture command is given, the positions of the robots with respect to the gesture are randomized to simulate a realistic scenario, since in between commands robots perform their own tasks which causes them to be randomly scattered in the environment. By means of the cooperative recognition protocol in Section 3.4, the swarm converges to a decision for the gesture which can be correct or wrong. In both cases, each robot in the swarm acquires the correct label for the given gesture using full feedback from the human and adds the related information to the subset T_r^p of its training set. After every 10 interaction rounds robots exchange $B \times 10$ training samples (selected within the T_r^p subset, which may include samples acquired during previous interaction rounds but not samples which have already been disseminated).

When information sharing is complete the value of parameter R determines if and when robots need to forget some of the training samples in order to limit the size of the training set. Parameter R represents the maximum number of training samples that a robot can retain. If the current size of the training set for a robot exceeds R, one of the sample forgetting strategies in Section 5.4.2.2 is iteratively applied to reduce/shrink the size of the training set (number of training samples) to be exactly R. Finally, all robots retrain their classifiers (i.e., update classifiers with new information) and a new interaction round starts.

The average classification accuracy of the swarm is computed after every 10 interaction rounds (see Section 5.3.1.2). In this way, for a full simulation run of 150 gesture commands, we obtain 15 accuracy values measured at different stages of the cooperative learning process. The first value corresponds to the

first set of 10 commands in the first interaction round, which is obtained using classifiers trained during the initial learning phase (see Section 5.3.1.1). Initial classifiers are trained using $T_i = 30$ samples. Subsequent values correspond to incrementally larger training sets, until the maximum training set size (parameter R) is reached. For each set of simulation parameters, 50 simulation runs are performed using different realizations of random variables in the dataset.

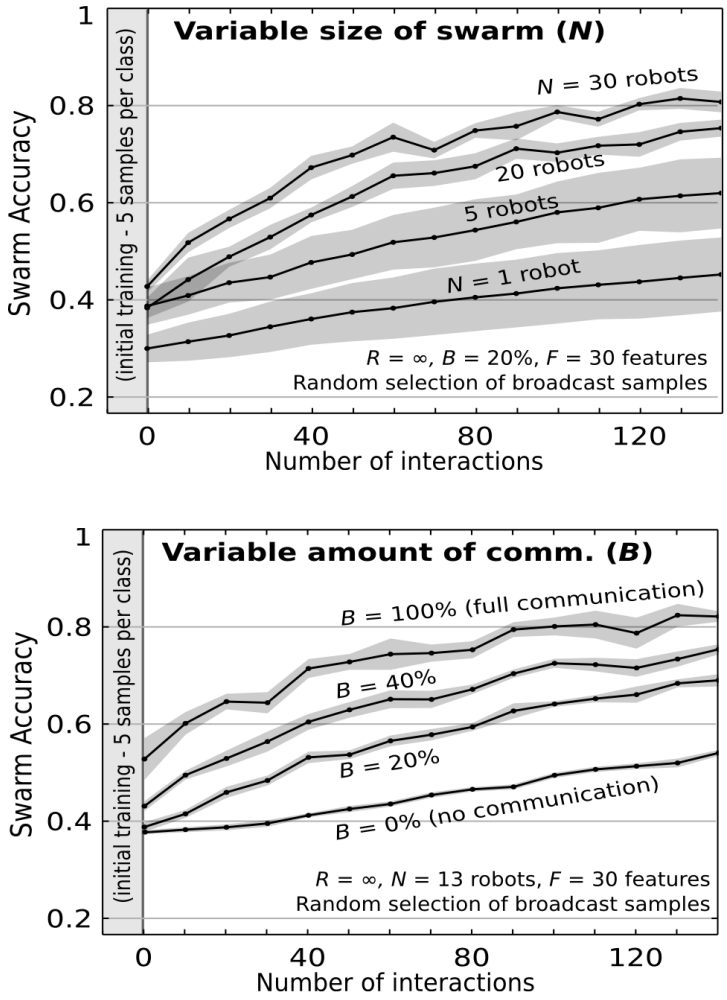


Figure 6.25. Average swarm accuracy vs. number of interaction rounds. Top: Accuracy curves for different swarm sizes with robots sharing $B = 25\%$ of their samples using representativity-driven selection. Bottom: Accuracy curves for $N = 13$ robots corresponding to different communication loads (different percentages of samples shared among robots) $B = \{0\%, 20\%, 40\%, 100\%\}$ with random selection of samples. Grey bands correspond to confidence intervals.

6.7.1.2 Effect of Swarm Size and Amount of Shared Information

The learning curves for different swarm sizes and different amounts of exchanged information (e.g., 20% communication means that every robot only shares 20% of its personal samples) are reported in Figure 6.25. As expected, larger swarms yield a significantly better accuracy in all stages of the learning (training) process. Two factors contribute to this effect:

- When $B > 0$, large size swarms are trained much faster than small size swarms. This is because a large swarm collectively acquires and exchanges a proportionally larger amount of training samples. A single robot or a swarm which does not exchange training data (see the curve with $B = 0\%$ in Figure 6.25 (bottom)) learns very slowly.
- When recognizing a gesture, large swarms enjoy a more powerful consensus ability as more observations are accounted for.

The contribution of the former factor is explored in Figure 6.25 (bottom), which shows how communication improves the learning ability of a swarm of $N = 13$ robots. The latter factor is isolated when comparing the bottom curves of both plots in Figure 6.25. In both cases (bottom curves), no communication is allowed and each robot learns independently from the rest of the swarm. The only difference among the two scenarios is given by the size of the swarm which affects accuracy due to the different amount of data acquired during the consensus phase. As expected, the 13-robot swarm in Figure 6.25 (bottom) with $B = 0\%$ is more accurate than the single robot in Figure 6.25 (top).

The results in Figure 6.25 (bottom) report that, the larger the amount of communication, the better is the swarm-level accuracy. After a very large number of interaction rounds, the training sets of all robots become so large that no further increase in accuracy is possible. At this point, all scenarios in Figure 6.25 (bottom) are expected to yield the same accuracy.

6.7.1.3 Effect of Selection and Sample Sharing Strategies

The effect of the three strategies for selecting training samples to disseminate (see Section 5.4.2.1) is reported in Figure 6.26 (top). Giving priority to novel samples results in a performance which is comparable to purely random selection for almost the entire training process. On the other hand, giving priority to the most representative samples leads to a significantly faster learning rate especially during initial training phase. This is due to the fact that a *representative sample* summarizes multiple samples as it lies near to their centroid. In this

context, representativity-driven selection can be more informative compared to the typical characteristics of a given class. Conversely, *novel samples* appear to be more useful later on in the learning process due to their contribution in refining the decision boundaries of the classifiers.

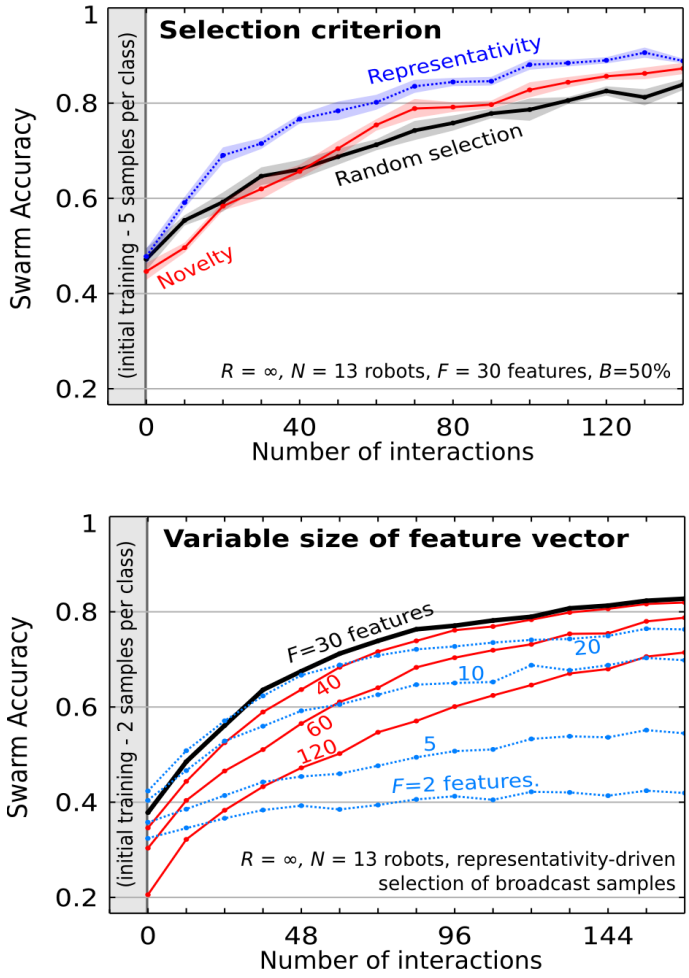


Figure 6.26. Top: Effect of using different selection strategies for sharing samples with $N = 13$ robots, $F = 30$ features, $B = 50\%$ communication, and $R = \infty$ (robots never forget samples). Bottom: Accuracy using representativity-driven selection with $N = 13$ robots and different number of features.

6.7.1.4 Impact of Number of Features in Bandwidth-limited Scenarios

Communication constraints are the main reason for limiting the amount of information in the training samples which are shared among robots in a swarm. As training samples comprise of feature vectors with their respective GT classes, the

dimensionality of the features vectors (i.e., feature space) to be exchanged is an important parameter. Larger feature vectors produce more powerful classifiers, but at the same time require more bandwidth for dissemination. In bandwidth-limited scenarios, a trade-off emerges as using more features implies disseminating less training samples, which has a negative impact on learning rate of the swarm. This trade-off is investigated in Figure 6.26 (bottom) which reports the swarm learning curves when using different feature vector sizes.

Each robot has the opportunity to disseminate a fixed amount of information corresponding to a total of $F = 120$ features after a set of 12 gestures is given (i.e., approximately 500 bytes assuming single-precision floating point representation). In Figure 6.26 (bottom) it is observed that, when small feature vectors are used ($F = \{2, 5, 10\}$) all acquired samples need to be shared among robots in the swarm. However, the individual classifiers are still not powerful enough and yield relatively poor recognition accuracy. If $F = 120$ features are used, each robot can only disseminate $^1/_{12\text{th}}$ of the acquired samples and the size of the local training sets increases at a much slower rate which results in slower learning.

In general, relatively small feature vectors ($F = \{10, 20\}$) lead to better accuracy during the initial training stage as they allow to quickly build moderately-sized training sets. However, in the later training stage classifiers are not powerful enough to exploit the expanding size of the training set. Instead, relatively large feature vectors ($F = \{40, 60\}$) lead to suboptimal results at the beginning but are able to fully exploit the larger training sets accumulated in the later stages of the cooperative learning process. Intermediate feature values ($F = \{30, 40\}$) lead to nearly-optimal swarm accuracy in all learning stages (see Section 6.7.3).

6.7.1.5 Impact of Strategies for Forgetting and Removing Samples

One key requirement for real-time learning is to ensure that the retraining time of a classifier remains manageable. A simple approach is to limit the maximum number of retained training samples (parameter R). In Section 5.4.2.2 three simple strategies are presented that iteratively select the samples to be removed (forgotten) from the training database. The quantitative results of the swarm-level accuracy and the SVM retraining time (computed on the Foot-bots) is reported in Figure 6.27 which evaluates the effects of the three sample forgetting strategies.

In this experiment the same setup used in Figure 6.25 is considered with $N = 13$ robots sharing all acquired samples ($B = 100\%$). As we are interested in the long-term behaviour, we consider a snapshot after 150 interaction rounds are performed. At this point, a robot which did not forget any sample holds around 2000 training samples, and on the Foot-bot platform the SVM retraining

time amounts to nearly 5 minutes for the rightmost data point. With the large amount of training samples, swarm accuracy reaches to 81.5%.

The forgetting strategies indicate that, selecting samples to forget using either representativity-driven or redundancy-driven criteria is detrimental to swarm accuracy when compared to random selection. This result is opposite to that obtained for selecting samples to be shared, where random selection is clearly not the best approach (see Figure 6.26 (top)). This due to the fact that, over time these approaches incrementally bias the training dataset which fails to remain representative of the classification problem to be solved. To forget samples, random selection results in a system which pays a minimal penalty in terms of the swarm accuracy while enjoying a fast retraining time. With $R = 500$, the swarm accuracy decreases marginally to 79.6% but the training time does not exceed 30s compared to the case with $R = 2000 = \infty$ samples.

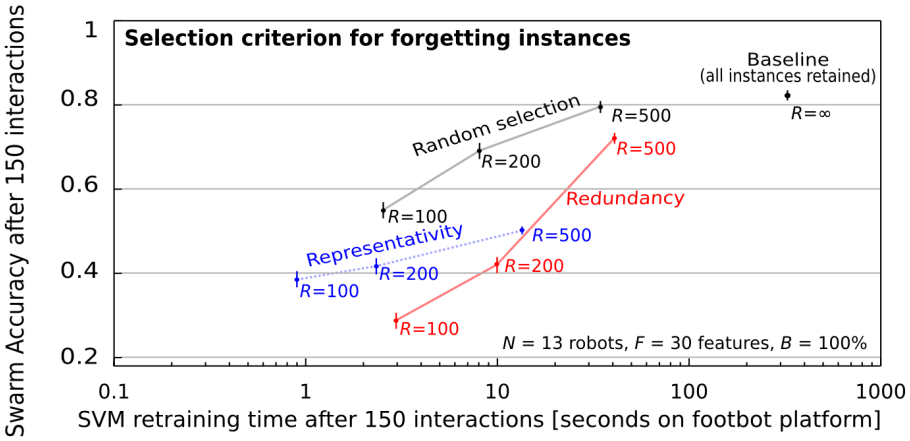
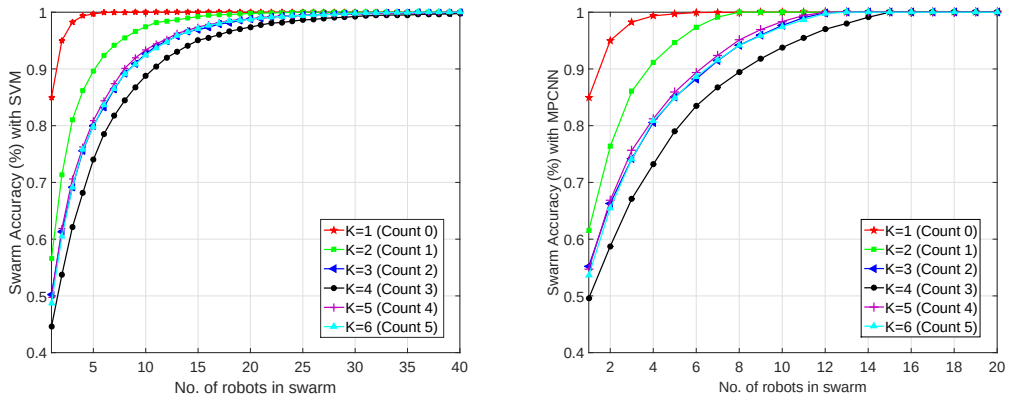


Figure 6.27. Swarm accuracy (y-axis) vs. SVM retraining time on the Footbot platform (x-axis, logarithmic) after 150 interaction rounds, $N = 13$ robots, $F = 30$ features, and $B = 100\%$ communication. Different amounts of samples maintained in the training set ($R = \{100, 200, 500, \infty\}$) along with three different strategies for forgetting samples. For $R = \infty$ samples are never forgotten. Vertical error bars report confidence intervals on the accuracy.

6.7.2 Offline Learning in Robot Swarms

This section presents the swarm-level classification performance using offline learning methods, namely, the SVM and MPCNN classifiers (see Section 5.2). Experiments are reported in Figure 6.28 and make use the dataset of $K = 6$ finger count gestures (finger counts from 0 to 5). The classification accuracy of the SVM (trained using hand-crafted features) for different swarm sizes is shown

in Figure 6.28(a). Figure 6.28(b) illustrates the swarm-level accuracy for the MPCNN classifier using the same setup as for the SVM. The MPCNN architecture used in this experiment consists of 6 hidden layers as shown in Figure 5.5. The output maps of the fully-connected layer are down-sampled to 1 pixel per map which result in a feature vector of $F = 300$ features. The output or classification layer of the MPCNN consists of 6 neurons (i.e., one neuron per gesture class).[5]



(a) (b)

Figure 6.28. Swarm-level classification accuracy vs. swarm size for the $K = 6$ finger count gestures. (a): SVM with hand-crafted features. (b): MPCNN.

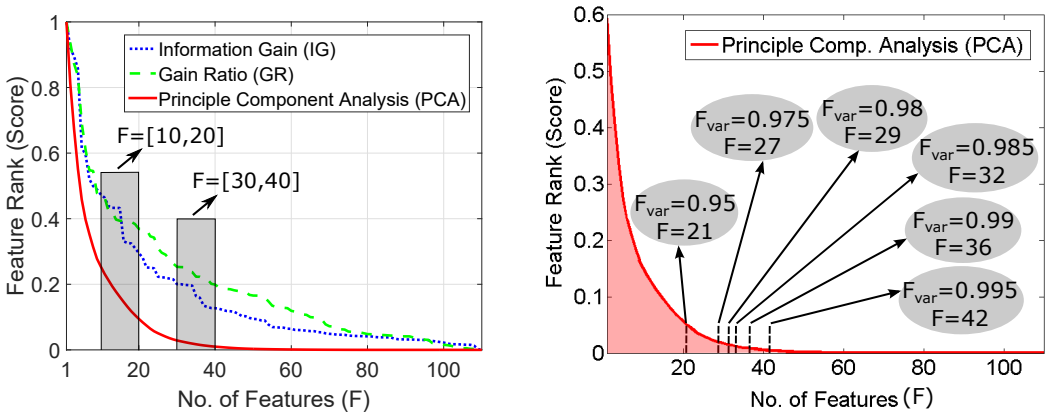
Using the SVM and MPCNN classifiers, simple shapes such as finger counts 0 and 1, are the easiest to classify with smaller swarms. Finger counts 2, 4, and 5 have nearly the same difficulty level in the multi-class classification problem, and finger count 3 is the most difficult gesture to classify in terms of the required number of robots. Finger count 3 is the hardest to recognize even for human observers, due to the fact that, when finger counts 4 and 5 are seen from angled viewpoints they resemble the shape characteristics of finger count 3.

6.7.3 Feature Selection and Ranking

To reduce the curse of dimensionality of a large feature space, this experiment performs *feature selection* to investigate the quality of the hand-crafted features (see Section 5.2.1.1) with respect to their individual and mutual discriminative powers. To make an intelligent selection from the initial set of $F = 110$ hand-crafted features and provide a robust analysis, feature selection techniques com-

[5] See Figure 5.5 for details of the MPCNN parameters used for training.

monly used in existing works for similar tasks have been considered. These include the, Principal Component Analysis (PCA), Information Gain (IG), and Gain Ratio (GR) approaches. In order to rank the $F = 110$ features based on a meaningful score, the Ranking method in WEKA [Hall et al., 2009] is combined with the PCA, IG, and GR methods. Experimental results are shown in Figure 6.29.



(a) (b)

Figure 6.29. (a): Comparison of the PCA, IG, and GR approaches with $F = 110$ features using the SVM and Ranker method in WEKA [Hall et al., 2009]. (b): Effect of using different values of F_{var} (i.e., the amount of principle component attributes retained to account for some proportion of the variance in the data).

The comparison results are shown in Figure 6.29(a), using the SVM with the $F = 110$ hand-crafted features. For comparison purposes, the results of the three feature selection approaches have been normalized (scaled y-axis) in a closed interval $[0, 1]$. Feature scores produced using the GR method do not provide much reliable information. The IG method indicates two significant drops in the discriminative power of the feature scores, one drop between $F = [10, 20]$ and the other drop between $F = [30, 40]$. As a smooth gradient descent is evident in the scores produced by the PCA method, this indicates that the PCA is the most reliable feature indicator among all approaches. Using $F = [30, 40]$ features the PCA provides the most discriminative information for $K = 6$ finger count classification problem. The use of additional features (i.e., $F > 40$ features) will deteriorate the classification performance, as shown in Figure 6.26 (bottom).

The effect of using different values of the PCA parameter F_{var} is reported in Figure 6.29(b). Parameter F_{var} controls the amount (%) of principle component (PC) attributes to retain in order to account for a proportion of variance in the original data. For instance, if $F_{var} = 0.95$ PC attributes accounting to 0.95 pro-

portion of variance (in the original data) are retained. When $F_{var} = 1$, all $F = 110$ features are selected by PCA. Different values of F_{var} result in different numbers of selected features. With $F_{var} = [0.95, 0.975, 0.98, 0.985, 0.99, 0.995]$ the number of features selected are $F = [21, 27, 29, 32, 36, 42]$ respectively. Within a small interval $F_{var} = [0.995, 1] == F = [42, 110]$ (that accounts to a very small proportion of variance) many redundant features are present. The best option is to use $0.98 \leq F_{var} \leq 0.995$ so a large majority of the redundant features can be omitted without sacrificing the quality of the high ranking features.

6.8 Summary of Contributions

This chapter presented and discussed the experimental results and reported the performance of the algorithms, techniques and strategies developed in Chapters 2 to 5. Experiments were performed as emulation tests which made use of a dataset of gesture images acquired by a heterogeneous robotic swarm (UGVs and UAVs), and real-time (online) experiments were performed with real robots to verify the performance, robustness, reliability and scalability of the HSI system.

The insights resulting from this chapter are as follows. The general protocol for the swarm-level classification (cooperative recognition and decision-making) of gestures is capable of building robust consensus decisions for real-time interaction with robot swarms. Individual gesture commands and grammar-based sentences defined the gesture language (vocabulary) can easily be learned and recognized. The bidirectional human-swarm communication systems facilitates proximal interaction within a reasonable amount of interaction time. The distributed algorithms and strategies that select spatially-situated robots from a swarm report a notably good performance, however, with large and densely populated swarms robot selection accuracy significantly decreases. With the use of spatially-aware mobility strategies individual robots in a swarm are able to sense better quality gesture observations, which provides deployment and improves the swarm-level classification accuracy of gestures. The information selection and sharing strategies developed for cooperative learning guarantee a fast convergence rate for the swarm-level learning of gestures while optimizing the use of computational and communication resources.

Chapter 7

Conclusions, Future Work, and Publications

This chapter provides a summary of the research contributions achieved in this work. The achievements obtained with respect to the goals in Section 1.4 and the experimental results in Chapter 6 are highlighted together with the key findings and the limitations encountered. Recommendations and suggestions are provided for conducting future research in this domain. Lastly, the publications resulting from of this research are listed.

7.1 Summary of Research and Main Contributions

This research has addressed several fundamental issues and core challenges in HSI (see Section 1.3) and has dealt with a number of problems commonly arising in swarm robotic systems. At a *systemic* level (the last sub-goal in Section 1.4.2.4), the main contribution lies in the fact that a global HSI system has been coherently built and made to work. It is the synergistic result of combining and integrating the algorithms, techniques, and strategies developed in Chapters 2 to 5. As a result, robot swarms have been made to cooperatively learn and recognize relatively complex non-verbal human instructions and grammar-based sentences given as visual signals (gestures). Multi-modal feedback and coordination schemes have been developed to allow humans to interpret the status, decisions, and intentions of swarms. The HSI system has been evaluated on heterogeneous swarms of up to 20 real robots (see Section 7.4.3) and validated in emulation tests using swarms of up to 100 robots.

In terms of the individual components of the global HSI system, to fulfil the main goal outlined in Section 1.4.1, a bidirectional human-swarm communica-

tion system has been developed in Chapter 2 which allows proximal, instrumented and non-verbal interaction between humans and heterogeneous robot swarms based on the use of coloured passive markers (i.e., inexpensive coloured gloves worn by humans). The introduction of dialogue-based interaction with a grammar-based gesture language and a vocabulary of commands, facilitates human-to-swarm communication and enables human operators to build iconic sentences and convey complex mission instructions to robot swarms using gestures. Moreover, it allows humans to express spatially-addressed commands and select spatially-located *individuals* and *groups* of robots from a swarm [Nagi et al., 2014c]. The developed gesture language supports the concept of generality (see Section 2.2.3.3) and can be adapted for different applications.

The swarm-coordinated multi-modal language uses robot actuation devices for swarm-to-human communication and conveys swarm-level feedback to humans. The language provides three types of multi-modal feedback that robot swarms convey to humans: (i) feedback to convey the swarm's understanding of the recognized gesture commands, (ii) self-assessment feedback that corrects mistakes and errors made by humans and by swarms, and (iii) feedback to guide humans through the interaction process. With the use of these feedback: (a) humans can easily interpret swarm-level status, decisions and intentions during interaction, (b) an intelligent HSI system with a human-friendly interface is built that can identify and minimize mistakes and errors, and (c) swarms can reliably assess consensus decisions without human feedback (external inputs) and provide basic reasoning capabilities. The experimental results in Chapter 6 signify that the *grammar-based gesture language* and the *swarm-coordinated multi-modal feedback language* developed in Chapter 2 allow bidirectional interaction and communication between humans and robot swarms.

The general protocol for the swarm-level classification of gesture commands introduced in Chapter 3 satisfies the sub-goal in Section 1.4.2.1. The developed distributed sensing and cooperative recognition mechanisms enable robot swarms to effectively gather and fuse information (acquired in parallel by individual robots from different viewpoints) and build swarm-level decisions for the collective recognition of commands in the gesture vocabulary. The developed distributed consensus protocol guarantees convergence to a common swarm-level decision. The protocol also introduces a trade-off parameter to balance the accuracy of swarm-level decisions and the time taken to reach swarm-level decisions. For instance, this allows to adapt the response to the urgency of the situation. From the experimental analysis in Chapter 6, the cooperative recognition protocol in Chapter 3 proves to be robust and scalable [Nagi et al., 2014d; Giusti et al., 2012a,b,c; Nagi et al., 2012a, 2015], as shown in Figure 6.3.

In the context of cooperative sensing and recognition, the implemented human body motion detection system (Appendix B) reliably allows robot swarms to identify when humans are presenting gesture commands that need to be classified and when no instructions are given by humans [Nagi et al., 2014c].

The swarm-level coordination mechanisms in Chapter 4 fulfil the sub-goal outlined in Section 1.4.2.2. With spatial pointing gestures humans can select and address spatially-located individuals and groups of robots from a swarm. This is based on coordination strategies internal to the swarm: robots individually classify the given gesture and exchange information to coordinate and cooperatively understand which individual or groups of robots have been selected based on an estimate of their relative position with respect to the gesture. When a gesture is being issued individual robots locally coordinate with each other to move and optimally surround a human operator. Robots position themselves with the goals to: uniformly cover the field of view in front of the human, maximize the mutual information sensed by the swarm regarding the gestures, and support local wireless connectivity in the multi-hop swarm network. The experimental results in Chapter 6 indicate that the coordination algorithms for the swarm's understanding of spatial robot selection and the mobility rules developed in Chapter 4 are robust and efficient for use with both ground and flying robots (UGVs and UAVs) [Giusti et al., 2012c; Nagi et al., 2014c,a,b,d].

The distributed learning strategies developed for robot swarms in Chapter 5 fulfil the sub-goal in Section 1.4.2.3. Supervised online learning approaches have been studied which cover a wide variety of swarm learning scenarios. As offline learning methods limit the capability to include humans in the loop of swarm learning, online incremental learning strategies have been developed for swarms to learn gestures in real-time supervised by humans. The role of human feedback in swarm-level learning has been investigated with the development of online algorithms that support the use of full and partial feedback given by humans. Full feedback comes at a higher cost but provides a faster swarm-level learning rate, while partial feedback comes at a lower cost and provides a reasonable learning performance [Nagi et al., 2014d,e; Ngo et al., 2014; Nagi et al., 2012b].

As robot swarms can collaboratively learn as a team, heuristic approaches for cooperative learning are introduced, which make use of intelligent information selection strategies for optimizing the use of the computational and communication resources that are typical constraints in robot swarms. Compared to methods that do not share learning information between robots, the developed strategies for cooperative learning provide a significant increase in the swarm-level learning rate (i.e., allow swarms to learn gestures within a few interaction rounds) and scale well with swarm size and available resources [Di Caro et al., 2013a].

The novelty and expected impact of this doctoral research lies in the fact that, the core HSI challenges for proximal interaction (see Section 1.3) have been addressed by developing a complete HSI system which has been experimentally evaluated in simulation and with real robots.

7.2 Major Issues Faced

Apart from the notable contributions discussed in the previous section, some major issues were faced during this research and they are presented below.

(i) The swarm robots used in this research, namely the Foot-bots (see Section 1.2.1.1), are equipped with low quality-sensing devices and small-scale computational power. The ARMv6 processor with 533 MHz and cameras on-board the i.MX31 architecture are considered outdated. These early VGA cameras are unable to capture good quality images (compared to today's smartphones and tablet computers) as they highly depend upon illumination conditions which makes them challenging to use. As the HSI system executes multiple threads on a Foot-bot (i.e., the robot controller runs separate processes for, image acquisition, gesture vocabulary, and motion detection), it takes a Foot-bot approximately 0.5s to acquire, process, and classify a single gesture image.

(ii) The acquired dataset of gesture images (see Section 6.2) used to run emulation experiments is challenging because gestures are shown in many different rotations of the hands and some gestures are represented in different variations. Most viewpoints in which robots are located do not clearly allow distinguishing between different gestures even to human observers. Many samples in the dataset present serious segmentation issues (e.g., due to factors such as illumination conditions) which are related to practical problems arising in real-world situations.

(iii) Robot swarms are prone to communication and packet loss. The range-and-bearing (RAB) system that the Foot-bots use for communication (see Section 1.2.1.1) has very low bandwidth and is inherently unreliable.

(iv) With the Foot-bot platform the detection of human body motion (see Appendix B) is not very robust nor reliable. This is mainly due to the limited on-board computational power of the Foot-bots. Images acquired at resolution of 384×288 pixels (0.1 megapixel) provide a frame rate of 2 fps

while images acquired at a resolution of 512 × 384 pixels (0.2 megapixel) achieve only 1 fps. Due to the low fps, a constant delay (lag) is always present when detecting human motion. This means that the Foot-bots detect human motion a bit later in time than it actually occurs.

(v) Individual robot failures are unavoidable problems in swarm robotic systems. These problems are mainly caused due to: dead or low battery, malfunction of sensory-motor systems (e.g., camera or LEDs failure), wireless networking and connectivity problems, color segmentation problems due out of focus camera lens or bad illumination conditions, and storage problems (i.e., read/write errors, memory corruption). Some of these issues can be overcome by rebooting the effected robots, replacing low batteries with charged ones, while other situations require robot maintenance skills.

7.3 Future Directions to Explore

This section presents a spectrum of possibilities for continuing further research in the domain of HSI. Future directions that can be investigated for symbiotic interaction between humans and robot swarms are presented below.

7.3.1 Reduction of Energy Consumption

The swarm-to-human feedback mechanisms, the cooperative recognition protocol, and the coordinated deployment strategies are not energy aware. Energy consumption can be reduced using different types of optimization strategies. As energy minimization was briefly discussed in Section 2.3.1 with the selection of spokes-robot(s), the amount of duplicate robot-to-human communication (i.e., all robots which convey the same feedback to humans) needs to be minimized. For instance, individual robots can coordinate and cooperate with other robots to mutually decide, which robots in the swarm should turn on their LEDs or beacons to provide visual feedback to humans.

One possible optimization strategy is for robots positioned at bad sensing positions. As probability vectors generated from robots in bad positions will tend to have high entropy (see Figure 3.5), these robots can reduce their image acquisition frame rate (fps), resulting in the acquisition and processing of a smaller number of samples as compared to robots in good positions. Robots located at bad viewpoints can be excluded from the consensus building process. However, the effects of this have to be investigated with the case when all robots in the swarm take part in building consensus decisions.

Instead of having the entire swarm deploying around a human operator, a few robots closer to the human can deploy themselves smartly. If gesture assessment (i.e., gesture recognition performance) made by the deployed robots is not robust, then these robots can request assistance from other robots that are located further away from the human. In this way, an adaptive incremental deployment strategy can be developed in which a few robotic informers are deployed first, and if required, additional robots (resources) can be invited to join.

7.3.2 Use of Context for Disambiguation

The grammar-based gesture language introduced in this research is based on the classification of individual words (single gestures). Instead of recognizing each word (gesture) one-by-one in a sentence (as performed in this research), entire sentences can be recognized at once, and context can be used to reduce ambiguity and to interpret the meaning of the words.

To remove ambiguities and to narrow down the meaning of individual words while retaining their semantic meanings, grammar parsing techniques are required. In order to recognize complete sentences using vision-based inputs, all words (gestures) in a sentence need to be correctly interpreted from a stream of continuous images, which is a challenging task. In the context of audio signals (e.g., speech and voice commands), interpreting a sentence of audible commands is simpler, and speech recognition has been demonstrated in the context of multi-robot systems [Pourmehr et al., 2013a; Monajjemi et al., 2014; Pourmehr et al., 2014]. With the use of audio-based inputs, a vocabulary of audible commands needs to be built that has similar functionality as the grammar-based gesture language, and distributed cooperative mechanisms for sensing and recognition of audio signals need to be developed.

7.3.3 Instrumented Interaction with Handhelds and Wearables

As interactions can be *direct* (uninstrumented) or *indirect* (instrumented), Table 7.1 lists potential modalities that can be used for HSI. For indirect interaction, instrumented devices (e.g., smartphones and tablets) can serve as efficient user interfaces: to select and command robot swarms (human-to-swarm communication), and to convey multi-modal feedback from swarms (swarm-to-human communication). For instance, a human rescuer can order robots to search an area using a gesture, but that does not precisely define the search area. A mediated interaction such as an outline on an ad-hoc map (e.g., on a handheld device) provides more finer and precise control.

Table 7.1. Direct and indirect modalities vs. information flow direction.

Modalities vs. Direction	**Human-to-swarm Communication**	**Swarm-to-human Communication**
Direct (Proximal)	Speech, hand or full-body gestures (static or dynamic gestures), touch or direct manipulation of robots, sounds such as whistles	Lights (synchronized among multiple robots), sounds, speech, projected images or laser dots/shapes/rays, specific movement patterns, robot location
Indirect (Distant)	Commands provided through suitable user interfaces (wearable sensors, smartphones and tablets with user interfaces)	Audible and visual notifications on the human's portable device

With the use of wearable devices, humans can robustly provide mission instructions to swarms, and swarms can easily recognize information and signals emitted by wearables without the need of developing sophisticated modality-specific recognition algorithms. A potential example of a wearable sensing device is the Myo Gesture Control Armband[1] which communicates using Bluetooth, and has been adopted by IDSIA [Gromov et al., 2016] in the second phase of the NCCR Robotics project (see Section 1.5). With this armband, gestures are detected through proprietary EMG muscle sensors (i.e., highly-sensitive motion sensors) and haptic feedback is provided using different vibration settings.

7.4 Publications

The publications resulting as a consequence of this doctoral research are listed below. These include journals, conferences and video demonstrations.

7.4.1 Journals

[1] H. Ngo, M. Luciw, N. Vien, J. Nagi, A. Forster, J. Schmidhuber, "Efficient Interactive Multiclass Learning from Binary Feedback", *ACM Transactions on Interactive Intelligent Systems*, vol. 4, no. 3, 2014, pp. 1-25.

[1] https://www.thalmic.com/myo/

7.4.2 Conferences

[1] J. Nagi, H. Ngo, L. M. Gambardella, Gianni A. Di Caro, "Wisdom of the Swarm for Cooperative Decision-Making in Human–Swarm Interaction", in Proc. of the *IEEE International Conference on Robotics and Automation* (ICRA), Seattle, USA, May 26-30, 2015, pp. 1802-1808.

[2] J. Nagi, G. A. Di Caro, A. Giusti, L. M. Gambardella, "Learning Symmetric Face Pose Models Online Using Locally Weighted Projectron Regression", in Proc. of the *IEEE International Conference on Image Processing* (ICIP), Paris, France, Oct. 27-30, 2014, pp. 1400-1404.

[3] J. Nagi, A. Giusti, L. M. Gambardella, G. A. Di Caro, "Human–Swarm Interaction Using Spatial Gestures", in Proc. of the *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), Chicago, USA, Sep. 14-18, 2014, pp. 3834-3841.

[4] J. Nagi, A. Giusti, F. Nagi, L. M. Gambardella, G. A. Di Caro, "Online Feature Extraction for the Incremental Learning of Gestures in Human–Swarm Interaction", in Proc. of the *IEEE International Conference on Robotics and Automation* (ICRA), Hong Kong, May 31-Jun. 5, 2014, pp. 3331-3338.

[5] J. Nagi, A. Giusti, G. A. Di Caro, L. M. Gambardella, "Human Control of UAVs using Face Pose Estimates and Hand Gestures", in Proc. of *ACM/IEEE International Conference on Human-Robot Interaction* (HRI) (Late Breaking Report), Bielefeld, Germany, Mar. 3-6, 2014, pp. 252-253.

[6] J. Nagi, H. Ngo, J. Schmidhuber, L. M. Gambardella, G. A. Di Caro, "Human-Robot Cooperation: Fast, Interactive Learning from Binary Feedback", in Proc. of the *ACM/IEEE International Conference on Human-Robot Interaction* (HRI) (Video Session), Bielefeld, Germany, Mar. 3-6, 2014, pp. 107.

[7] G. A. Di Caro, A. Giusti, J. Nagi, L. M. Gambardella, "A Simple and Efficient Approach for Cooperative Incremental Learning in Robot Swarms", in Proc. of the *International Conference on Advanced Robotics* (ICAR), Montevideo, Uruguay, Nov. 25-29, 2013, pp. 1-8.

[8] J. Nagi, G. A. Di Caro, A. Giusti, F. Nagi, L. M. Gambardella, "Convolutional Neural Support Vector Machines: Hybrid visual Pattern Classifiers for Multi-robot Systems", in Proc. of the *International Conference on Machine Learning and Applications* (ICMLA), Boca Raton, USA, Dec. 12-15, 2012, pp. 27-32.

[9] A. Giusti, J. Nagi, L. M. Gambardella, G. A. Di Caro, "Cooperative Sensing and Recognition by a Swarm of Mobile Robots", in Proc. of the *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), Vilamoura, Portugal, Oct. 7-12, 2012, pp. 551-558.

[10] J. Nagi, H. Ngo, A. Giusti, L. M. Gambardella, J. Schmidhuber, G. A. Di Caro, "Incremental Learning using Partial Feedback for Gesture-based Human–Swarm Interaction", in Proc. of the *IEEE International Symposium on Robot and Human Interactive Communication* (RO-MAN), Paris, France, Sept. 9-13, 2012, pp. 898-905.

[11] A. Giusti, J. Nagi, L. M. Gambardella, G. A. Di Caro, "Distributed Consensus for Interaction between Humans and Mobile Robot Swarms", in Proc. of the *International Conference on Autonomous and Multiagent Systems* (AAMAS) (Demonstration Track), Valencia, Spain, Jun. 4-8, 2012, pp. 1503-1504.

[12] A. Giusti, J. Nagi, L. M. Gambardella, S. Bonardi, G. A. Di Caro, "Human–Swarm Interaction through Distributed Cooperative Gesture Recognition", in Proc. of the *ACM/IEEE International Conference on Human-Robot Interaction* (HRI) (Video Session), Boston, USA, Mar. 5-8, 2012, pp. 401-402.

[13] 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 Proc. of the *IEEE International Conference on Signal and Image Processing and Applications* (ICSIPA), Kuala Lumpur, Malaysia, Nov. 16-18, 2011, pp. 342-347.

Apart from my PhD research that primarily deals with HSI, I had the opportunity to collaborate with other experts at IDSIA, to investigate the learning and prediction of wireless link qualities using the Foot-bot robots.

[1] E. Feo Flushing, M. Kudelski, J. Nagi, L. M. Gambardella G. A. Di Caro, "Link Quality Estimation: A Case Study for On-line Supervised Learning in Wireless Sensor Networks". In K. Langendoen, W. Hu, F. Ferrari, M. Zimmerling, L. Mottola, editors, *Real-World Wireless Sensor Networks* (REALWSN), *Lecture Notes in Electrical Engineering* (LNEE), vol. 281, 2014, pp. 97-101.

[2] G. A. Di Caro, M. Kudelski, E. Feo Flushing, J. Nagi, I. Ahmed, L. M. Gambardella, "On-line Supervised Learning of Link Quality Estimates in Wireless Networks", in Proc. of *IFIP Annual Mediterranean Ad Hoc Networking Workshop* (Med-Hoc-Net), Ajaccio, France, Jun. 24-26, 2013, pp. 69-76.

[3] E. Feo Flushing, J. Nagi, G. A. Di Caro, "A Mobility-Assisted Protocol for Supervised Learning of Link Quality Estimates in Wireless Networks", in Proc. of the *International Workshop on Mobility and Communication for Cooperation and Coordination* (MC^3), in conjunction with the *International Conference on Computing, Networking and Communications* (ICNC), Hawaii, USA, Jan. 30, 2012, pp. 137-143.

7.4.3 Videos

A collection of videos published during this PhD research are available on my website at: `http://videos.jnagi.net/` and on my Youtube channel at: `http://youtube.jnagi.net/`.

7.5 Summary

This section concludes this dissertation. The research goals and objectives outlined in Section 1.4 have been successfully achieved through Chapters 2 to 5 and have been verified by the experimental results in Chapter 6. The outcome of this research provides a complete HSI system that has been implemented and tested on a heterogeneous robotic swarm. The experiment results verify the efficacy and performance of developed HSI schemes and strategies, and conclude that the HSI system is robust, scalable, and works well in real-world environments. A large number of peer-reviewed publications have resulted as a consequence of this research. Although some issues were faced which can be overcome, the contributions made by this research are of much greater significance. It is envisaged that, such HSI systems will enable autonomous and heterogeneous teams of robot swarms to perform SAR missions in cooperation with rescue workers.

Appendix A

Segmenting Gestures from Background using Coloured Gloves

When gesture recognition is performed on natural scenes without relying on simplifying assumptions the task is becomes a challenging pattern recognition problem [ChaLearn Gesture Challenge; Arbelaez et al., 2011]. As the focus of this research is not the vision-based aspect of the gesture recognition problem while at the same time learning and recognition of gestures using robots with limited computational power (see Section 1.2.1.1) is required, to simplify the recognition task we consider that human operators wear passive markers (i.e., gloves) with known characteristic colors, as briefly highlighted in Section 2.1.1.1.

With a swarm of $R = \{r_1, r_2, \ldots, r_N\}$ robots, where N comprises of the number of robots in the swarm, in the *InformationGathering*() state (see Figure 3.4) every robot $r \in R$ acquires an image at time step t. For every image acquired by a robot, the first step requires to separate the gesture from the image background (i.e., to identify which of the image pixels belong to the gesture/glove) using *color-based segmentation*. As gesture projections normally cover a small fraction (portion) of an acquired image [Nagi et al., 2011], gestures are segmented by exploiting the characteristic colors of the gloves (i.e., the green and yellow gloves in Figure 2.4).

A standard per-pixel color-based segmentation approach is adopted in the HSV (Hue, Saturation, Value) color space [Kakumanu et al., 2007] using the OpenCV library. In practice, a simple rectangular area identified by parameters H_{min}, H_{max}, S_{min}, and S_{max} provides satisfactory color segmentation when supplemented by an additional constraint on the minimum value of the V_{min} channel, which is useful to discard dark areas of the image which provide unreliable Hue pixels. These five parameters [H_{min}, H_{max}, S_{min}, S_{max}, V_{min}] can easily be estimated from a single hand gesture image and they remain fixed given the type

of illuminant conditions in the environment. After segmentation, a binary image (i.e., a black and white image with pixel intensities $[0, 255]$) is produced. Image pixels with the value 0 represent the background and pixels with the value 255 represent the gesture. Glove pixels are identified as those pixels whose coordinates match those observed in the Hue-Saturation plane of the sampled glove color. The *largest connected component* resulting from a segmented gesture image is referred as a *segmented hand mask*.

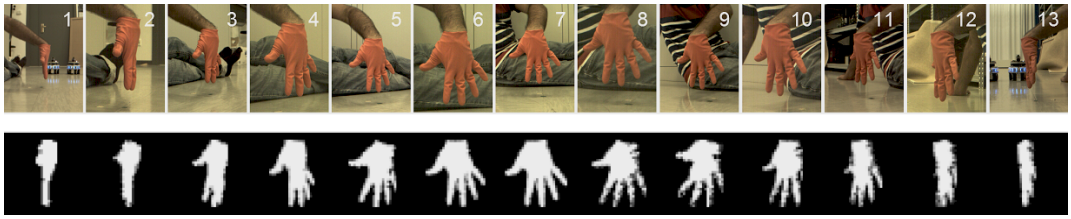


Figure A.1. A hand gesture acquired by a swarm of $N = 13$ robots (from 13 different viewpoints) using a distributed sensing mechanism. Top: Images of a hand gesture acquired by a swarm of Foot-bots. Bottom: Segmented hand masks corresponding to the top image after color-based segmentation.

As robots acquire gesture images at various distances from humans, the size distribution of the segmented hand masks is inspected and a dimension of 28 pixels (similar to the MNIST database [LeCun et al., 1998] of characters) is identified as an appropriate size to represent the segmented hand masks in a square region of interest (ROI). The hand masks are rescaled and resized to a dimension of 28 pixels (while maintaining the same aspect ratio) such that, if the height and width dimensions of the hand mask are not of equal size (which is normally the case), the larger dimension is rescaled to 28 pixels, otherwise if both height and width dimensions are the same, both dimensions are rescaled to 28 pixels. After rescaling, the final operation consists in padding the rescaled images with 4 background pixels (i.e., black color pixels with an intensity value of 0) on each of the four sides which results in a hand mask centered within an image of 32×32 pixels. Figure A.1 illustrates a hand gesture acquired by a swarm of $N = 13$ robots from different viewpoints after color-based segmentation, resizing, and padding. To perform color-based segmentation on an image of 512×384 pixels (acquired by the frontal camera of the Foot-bots; see Section 1.2.1.1) takes roughly 0.2s.

The black and white images which represent the segmented hand masks in Figure A.1 are used for computing a standardized set of meaningful features, as presented in Section 5.2.1. To learn and recognize gesture commands, the feature vectors computed from the segmented hand masks are fed as inputs into supervised classifiers, as discussed in Chapter 5.

Appendix B

Detecting Human Body Motion using Robot Swarms

For a gesture-based interface to be fully acceptable by robot swarms, the interface should allow humans to perform gestures in the same natural way and with the same speed as gestures are performed towards humans. It is necessary to take into account that, human operators can perform various unnecessary additional movements with the arms and hands (i.e., adapters; see Section 2.1.1.1) which are not instructions (gesture commands). We consider that, before a gesture is presented to the swarm, individual robots in the swarm need to collectively identify: if the human is preparing (or is in process) to issue a gesture, or if the human has already settled upon (has issued) a gesture.

A motion detection system can identify the presence of human motion in the upper body. For instance, when human motion is cancelled the swarm gets informed that the human is issuing a gesture. However, if human motion is detected for a continuous interval of time, this indicates that the human is not yet ready (i.e., the gesture has not been issued by the human) as illustrated in Figure 3.1. Using multiple airborne UAVs, the research team at the Autonomy Lab of Vaughan estimated the optical flow of motion in three predefined zones of the upper body [Monajjemi et al., 2013]: the face region and the left and right sides of the body where the arms and hands move freely.

An approach for motion detection and cancellation was presented in [Naseer et al., 2013] which made use of the Kinect sensor. In this approach an autonomous interaction system allowed an airborne UAV to follow humans and respond to gesture commands. We consider that, with the use of coloured gloves as passive markers, robots can estimate the motion of the arms and hands from a stream of images. However, to measure the displacement of the human body

(i.e., if the human torso moves) we adopt another passive marker, namely a jacket, which is worn by humans and has a known characteristic color (i.e., orange) as shown in Figures 2.4 and 2.5. A pair of gloves and a jacket consist of a standard uniform worn by firefighters, air marshallers, and rescue workers.

As color-coded markers can be distinguished based on their individual characteristic colors (see Appendix A), the three *largest connect components* resulting from the segmentation of the three markers (two gloves and the jacket) are retained (see Appendix A). The next step involves in computing the centroid of the jacket $c_{jkt}(\mathbf{x},\mathbf{y})$ and the centroids of both the (green and yellow) gloves $c_{grn}(\mathbf{x},\mathbf{y})$ and $c_{ylw}(\mathbf{x},\mathbf{y})$ with respect to the x-y coordinates of the image plane. The centroids of the three markers $[c_{jkt},c_{grn},c_{ylw}]$, are used to detect motion caused by movements of the arms and hands including the displacement of the body.

B.1 Estimating Magnitude of Optical Flow

To measure the optical flow in the upper body, a multiple-ring *circular buffer* is adopted as a data structure. The circular buffer queues (stores and updates) the magnitude of motion computed from the three passive makers (two gloves and the jacket) as illustrated in Appendix B. The multi-ring buffer consists of 3 buffers, in which each buffer stores the centroids of each of the three passive markers (inputs). Each of the 3 buffers comprise of bN elements and bN is a tunable parameter that controls the *damping of motion*. After the centroids of the 3 inputs $\left[c_{grn}(\mathbf{x},\mathbf{y}),c_{ylw}(\mathbf{x},\mathbf{y}),c_{jkt}(\mathbf{x},\mathbf{y})\right]$ are computed from every acquired image, the next step is to calculate 3 Euclidean distances using every two consecutively (sequentially) acquired images. This includes the distance between:

(a) One hand (i.e., the green glove) and the jacket, D_{grn_jkt}

(b) The other hand (i.e., the yellow glove) and the jacket, D_{ylw_jkt}

(c) Between both hands, D_{grn_ylw}

For instance, the Euclidean distance between the centroid of one hand (i.e., the green glove) and the centroid of the jacket is computed as:

$$D_{grn\_jkt} = abs(\sqrt{(c_{grn}(\mathbf{x})-c_{jkt}(\mathbf{x}))^2+(c_{grn}(\mathbf{y})-c_{jkt}(\mathbf{y}))^2}) \tag{B.1}$$

Similarly, the remaining two Euclidean distances D_{ylw_jkt} and D_{grn_ylw} are computed. The three distance measures $\{D_{grn\_jkt},D_{ylw\_jkt},D_{grn\_ylw}\}$ as illustrated

in Figure B.1, are computed using two consecutively acquired images, and are added/updated into each of the 3 circular buffers. Using a stream of N consecutive images, three distance vectors $\{\vec{D}_{grn\_jkt}, \vec{D}_{ylw\_jkt}, \vec{D}_{grn\_ylw}\}$ are computed, where each vector has a length of bN elements and $bN = N - 1$. The three distance vectors represent a *distance matrix* $D = \{\vec{D}_{grn\_jkt}, \vec{D}_{ylw\_jkt}, \vec{D}_{grn\_ylw}\}$ having a dimension of $3 \times bN$ elements. To determine the magnitude of motion between the two gloves and the jacket (i.e., the motion relative to the movements of the three markers), three magnitudes of optical flow $\{M_{grn\_jkt}, M_{ylw\_jkt}, M_{grn\_ylw}\}$ are computed as shown by the pseudocode in List B.1.

Listing B.1. Magnitude of optical flow relative to the two gloves and jacket

```
for every (acquired image) {
   // Initialization
   M_grn_jkt = 0;
   M_ylw_jkt = 0;
   M_grn_ylw = 0;

      // For every element in distance matrix "D"
      for (int i = 0; i < bN; i++) {
         M_grn_jkt = M_grn_jkt + abs(D_grn_jkt[i] - D_grn_jkt[i+1]);
         M_ylw_jkt = M_ylw_jkt + abs(D_ylw_jkt[i] - D_ylw_jkt[i+1]);
         M_grn_ylw = M_grn_ylw + abs(D_grn_ylw[i] - D_grn_ylw[i+1]);
      }
}
```

The magnitude of optical flow, namely the *motion score* M^t_{score}, is calculated at every time step t (i.e., when every new image is acquired):

$$M^t_{score} = \frac{\left(\frac{M_{grn\_jkt}}{bN}\right) + \left(\frac{M_{ylw\_jkt}}{bN}\right) + \left(\frac{M_{grn\_ylw}}{bN}\right)}{3} \tag{B.2}$$

The M_{score} is a metric that defines if upper body motion is present in a visual scene. Figure 3.1 illustrates the M_{score} for the cooperative sensing and recognition system presented in Chapter 3. To be robust towards the detection of transient motion flows, M^t_{score} which results from a continuous time signal (i.e., from a stream of images) is fed into a *moving average* filter. The higher the value of M^t_{score} the more rapid is the motion detected from the upper body, and the smaller M^t_{score} gets the slower the motion speed becomes.

To determine if upper body motion is present, a threshold parameter M_{TH} is introduced. Using a trial and error approach, a threshold of $M_{TH} = 1$ is identi-

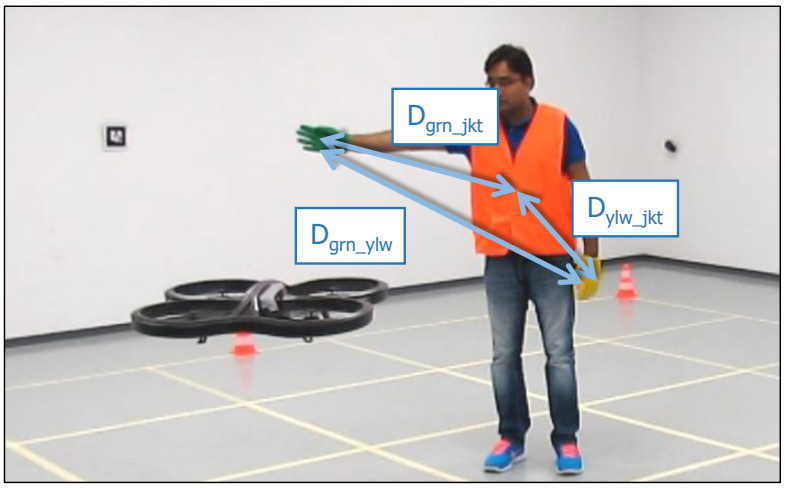


Figure B.1. The three Euclidean distances $\{D_{grn\_jkt}, D_{ylw\_jkt}, D_{grn\_ylw}\}$, computed using the centroids of the coloured passive markers (two gloves and jacket).

fied as the best value to balance the trade-off between small fractions of motion (that are hardly noticeable) and rapid motion. If $M^t_{score} >= M_{TH}$, this represents that motion is detected, else if $M^t_{score} < M_{TH}$, this indicates that motion is cancelled as the human maybe presenting a gesture.[1] Operationally, human motion is detected prior to the sensing and recognition of every gesture command, as illustrated in Figure 3.1. When a robot swarm awaits a command from a human and upper body motion is continuously detected (i.e., $M^t_{score} > M_{TH}$) for a period of $t >= 30$s, the swarm considers that the human is not ready to issue a command (i.e., a maximum time-out occurs) and the interaction process terminates.

B.2 Sensitivity to Human Motion Detection

To ensure that interaction between humans and swarms is natural as possible, we investigate the effect of using different motion damping values of parameter bN on the motion score M_{score} (see Appendix B.1). Experimental results are shown in Figure B.2. In this experiment, we select 4 individual robots over a course of 180 consecutively acquired gesture images in which every selection time corresponds to a vertical stripe, namely the time interval when $M_{score} < M_{TH}$.

In Figure B.2 it is observed that, if bN is too small, M_{score} changes very rapidly, is unstable (i.e., M_{score} fluctuates rapidly), and too sensitive to reliably detect

[1] A video demonstrating the motion detection system using an airborne UAV is available at: http://www.jnagi.net/human_motion_detection

human motion over time. Instead if bN is too large, human motion will not be detected on the spot it occurs but after some delay in time. Small motion damping values (e.g., $bN = 5$) indicate a faster decay in M_{score} (spikes) and large values (e.g., $bN = 20$) provide a slower decay rate (steps).

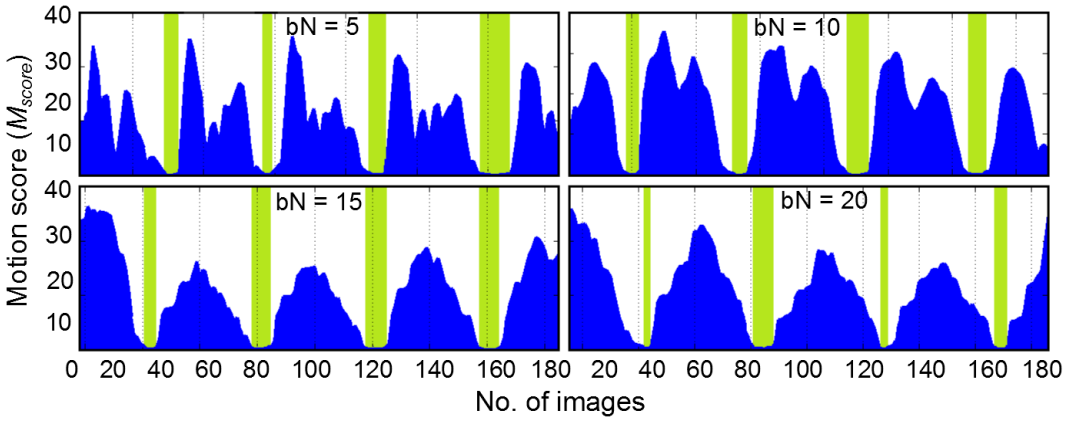


Figure B.2. Average magnitude of human body motion M_{score} for different values of the motion damping parameter bN.

If the decay occurs very rapidly or too slow, the way in which robots in a swarm perceive gestures can cause color-based segmentation errors due to blur in the acquired images, thereby making human motion detection and gesture recognition potentially unreliable. Choosing a good estimate of bN is crucial to support the reliability of motion detection. We determine that, bN in the range of $[9, \ldots, 12]$ provides a near-optimal (i.e., Gaussian like) distribution for detecting human body motion as illustrated by $bN = 10$ in Figure B.2.

Appendix C

Pseudocode for Swarm-to-Human Self-assessment Feedback

Listing C.1. Swarm-to-human assessment feedback for mistake/error correction.

```
// Initialization of trained classifiers
C_F1 → Addressing
C_F2 → Actions
C_F3 → Direction or Numerics
C_F4 → Numerics
C_F5 → Addressing and Actions
C_F6 → Direction and Numerics

// Classifier details
////////////////////////////////////////////////
C_A = No. of classes in classifierA
C_B = No. of classes in classifierB
Classifier outputs = classifierA_out and classifierB_out

// Global variables
double highestprobability;
int swarmfeedback, glove1cnt, glove2cnt, total;


// Find "highest and second highest" predicted classes
int C_A_max = -1;
double probA1 = 0.0, probA2 = 0.0;

for (int i=0; i<C_A; i++) {
```

```
      // Highest score in classification vector //
      if (classifierA_out[i] >= probA2) {
         probA2 = classifierA_out[i];
         C_A_max = i;
      }
      // Second highest score //
      if (classifierA_out[i] >= probA1) {
         probA2 = probA1;
         probA1 = classifierA_out[i];
      }
   }

   // Retrieve index of highest score
   for (int i=0; i<C_A; i++) {
      if (classifierA_out[i] == probA1) {
         C_A_max = i;
         break;
      }
   }

   C_A_max = C_A_max + 1;

   // Repeat method above, for classifierB_out with C_B elements
   // This yields: probB1, probB2 and C_B_max

   // Normalized confidence measures
   double probA = abs((probA2*C_A) - (probA1*C_A)) / C_A;
   double probB = abs((probB2*C_B) - (probB1*C_B)) / C_B;
   // Average prob. difference b/w classifierA_out and classifierB_out
   double P_avg = (probA + probB) / 2;


   // 1. Addressing (Selecting robots)
   ////////////////////////////////////////////////
   if ((C_A == C_F1) and (C_B == C_F5)) {

      // Both classifiers have similarity between results
      if (C_B_max < (C_F1_classes + 1)) {

         if (C_A_max == C_B_max) {
            // Gesture represents ROBOT SELECTION
```

```
            // CONFIDENT --> PROPERLY RECOGNIZED
            swarmfeedback = 1;
            highestprobability = C_{A_max};
        } else {
            // Gesture BELONGS TO SAME SEMANTIC CLASS
            // CONFIDENT --> INAPPROPRIATE
            swarmfeedback = 2;
            highestprobability = C_{B_max};
        }
    }

    // Both classifiers identify different semantic classes
    else {
        if (P_{avg} <= 0.5) {
            // NOT CONFIDENT --> UNDEFINED
            swarmfeedback = 3;
        } else {
            // Gesture represents an ACTION
            // NOT CONFIDENT --> NOT PROPERLY RECOGNIZED
            swarmfeedback = 4;
            highestprobability = C_{B_max};
        }
    }
}

// 2. Action (Commanding robots)
////////////////////////////////////////////////
else if ((C_A == C_{F2}) and (C_B == C_{F5})) {

    if (C_{B_max} > C_{F1_classes}) {
        if (C_{A_max} == (C_{B_max} - C_{F1_classes})) {
            // Gesture represents an ACTION
            // CONFIDENT --> PROPERLY RECOGNIZED
        } else {
            // Gesture BELONGS TO SAME SEMANTIC CLASS
            // CONFIDENT --> INAPPROPRIATE
        }
    }

    else {
        if (P_{avg} <= 0.5) {
```

```
            // NOT CONFIDENT --> UNDEFINED
         }
         else {
            // Gesture represents ROBOT SELECTION
            // NOT CONFIDENT --> NOT PROPERLY RECOGNIZED
         }
      }
   }

   // 3. Hand direction (Action parameters)
   ////////////////////////////////////////////////
   else if ((C_A == C_F3) and (C_B == C_F6) and (C_F3 == 1)) {

      // Both classifiers have similarity between results
      if ((C_A_max == 1) and (C_B_max == 1)) {
         // Gesture is a HAND DIRECTION
         // CONFIDENT --> PROPERLY RECOGNIZED
      }

      // Both classifiers identify different semantic classes
      else {
         if (P_avg <= 0.5) {
            // NOT CONFIDENT --> UNDEFINED
         }
         else {
            // NOT CONFIDENT --> NOT PROPERLY RECOGNIZED
         }
      }
   }

   // 4. Numeric quantity (Action parameters)
   ////////////////////////////////////////////////
   else if ((C_A == C_F3) and (C_B == C_F6) and (C_F3 == 2)) {

      if ((C_A_max == 2) and (C_B_max > 1)) {
         // Gesture is a FINGER COUNT
         // CONFIDENT --> PROPERLY RECOGNIZED

         // Use C_F4 to predict counts in both hands
         // Compute total number of finger counts
         hand1cnt = C_A_max;
```

```
        hand2cnt = C_{B_max};

        // Sum of counts from both hands
        total = hand1cnt + hand2cnt;
    }

    else {
        if (P_{avg} <= 0.5) {
            // NOT CONFIDENT --> UNDEFINED
        }
        else {
            // NOT CONFIDENT --> NOT PROPERLY RECOGNIZED
        }
    }
```

Appendix D

Confidence-Weighted Swarm Learning (CWSL)

Confidence-Weighted Swarm Learning (CWSL) is inspired from 2nd-order online learning methods which includes the *Confidence Weighted* (CW) large-margin learning scheme [Dredze et al., 2008; Crammer et al., 2012] and its successor *Soft Confidence Weighting* (SCW) [Wang et al., 2012b] learning. The SCW learning method [Wang et al., 2012b] was introduced to address the problems of CW learning, by applying the soft-margin idea in SVMs [Cortes and Vapnik, 1995] (see Passive-Aggressive algorithms [Crammer et al., 2006]) to CW learning. SCW learning is the first online algorithm that has the properties of: (i) large-margin learning, (ii) confidence weighting, (iii) capability to handle noisy and non-separable data, and (iv) adaptive margin constraints. Previous analysis have shown these properties to be very effective in improving performance.

I greatly thank Hung Ngo who collaborated with me for developing the CWSL algorithm. In CWSL, as given in Algorithm 4, the consensus weights of a swarm are updated on every interaction round using full feedback from humans (see Section 5.3.1.2). For more details on CW and SCW learning refer to [Crammer et al., 2008, 2009, 2012, 2013; Wang et al., 2012b].

D.1 The CWSL Algorithm

Inspired from the *game-theoretic competitive online learning* [Littlestone, 1987; Vovk, 2001; Azoury and Warmuth, 2001; Crammer et al., 2006; Cesa-Bianchi and Lugosi, 2006] framework and *multi-class prediction with expert advice* [Vovk and Zhdanov, 2009], CWSL makes no statistical assumptions on the underlying data generating process and does not rely on any pre-acquired training or testing

Algorithm 4: Confidence-Weighted Swarm Learning (CWSL)

1 `//Initialization`
2 $\boldsymbol{\mu}^i = \mathbf{0}$; $\Sigma^i = \mathbf{I}$, $i = \{1:K\}$ `// Learning parameters for` K `classes`
3 $w_0 = 1$ `//Consensus weight of every robot in swarm`
`//Main learning loop`
4 **for** $t = \{1, 2, \ldots, T_{rounds}\}$ **do**
5 **Receive new observation** $\mathbf{x}_t \in \mathbb{R}^d$
6 **Output normalized classification vector c:** $\mathbf{c}^i = \frac{\boldsymbol{\mu}^i \cdot \mathbf{x}_t}{\sum_{j=1}^{K} \boldsymbol{\mu}^j \cdot \mathbf{x}_t}$, $i = \{1:K\}$
`// BEGIN swarm-level consensus`
7 **Compute loss vector** $\boldsymbol{\lambda}$: $\lambda^i = \sum_{j=1}^{K} (\mathbf{c}^j - \delta_{i,j})^2$, $i = \{1:K\}$
8 **Compute surrogate consensus weights** $\boldsymbol{\omega}_1 = w_{t-1} e^{-\boldsymbol{\lambda}}$
9 **Exchange** $\boldsymbol{\omega}_1$ **to all** $N-1$ **robots in the swarm**
`// On receiving all` $\boldsymbol{\omega}_r$ `from` $N-1$ `robots`
10 **Compute generalized prediction vector** $\mathbf{g} = \ln\left(\boldsymbol{\omega}_1 + \sum_{r=2}^{N} \boldsymbol{\omega}_r\right)$
11 **Solve for** $s \in \mathbb{R}$: $\sum_{i=1}^{K} (s + g^i)^+ = 2$
12 **Set consensus prediction margin vector** $\mathbf{c}_w = (s + \mathbf{g})^+ / 2$
`// END swarm-level consensus`
13 **Output predicted label** $\hat{y}_t = \underset{i=1,\ldots,K}{\arg\max}\left(\mathbf{c}_w^i\right)$
14 **Observe full feedback from human** $y_t \in \{1, \ldots, K\}$
`// UPDATE weights and communicate`
15 **Update consensus weight** $w_t = \omega_1^{y_t}$
16 **Broadcast** $\mathbf{x}_t$ if $w_t \in L$, largest consensus weights $\{\omega_r^{y_t}\}$, $r = \{1:N\}$
17 **Update weight distribution** $\{(\boldsymbol{\mu}^i, \Sigma^i)\}_{i=1}^{K}$ using y_t and L observations
18 **end**

datasets. CWSL uses the CW learning framework [Crammer et al., 2008; Dredze et al., 2008; Crammer et al., 2009; Wang et al., 2012b] which makes use of the weight distribution $(\boldsymbol{\mu}_t, \Sigma_t)$ for online learning. The weights of a linear classifier in CW learning are associated with the *confidence information* via a multivariate Gaussian distribution, with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, where $d \in \mathbb{Z}^+$ (see Section 5.1.2). For each training sample the CW learning model is updated aggressively while maintaining the knowledge learned so far by not changing too much the Kullback-Leibler (KL; [Kullback, 1959]) divergence

from the previously trained model. The CWSL method presented in Algorithm 4 uses confidence information from CW learning.

Considering a swarm of $r = \{1, 2, \ldots, N\}$ robots trained using the initial learning phase (see Section 5.3.1.1), Algorithm 4 runs distributively on every robot r in the swarm. Figure 5.9 illustrates the flow of information in Algorithm 4. The multi-class prediction output from CW learning consists of a classification vector $\mathbf{c}$ over the K possible gesture classes (see Section 3.3). We adopt a square-loss function for multi-class prediction [Vovk and Zhdanov, 2009]: for an output of class $i \in \{1, \ldots, K\}$ the learner with classification vector $\mathbf{c}$ suffers a loss $\lambda^i = \sum_{j=1}^{K} (\mathbf{c}^j - \delta_{i,j})^2$ on every interaction round in $\{1, \ldots, Trounds\}$, where $\delta_{i,j}$ is a Dirac delta function $\delta_{i,j} = 1$, if $i = j$ and $\delta_{i,j} = 0$ otherwise.

In the beginning, every robot in the swarm is assigned a consensus weight with the same value $w_0 = 1$. As the online learning process unfolds (see Section 5.3.1.2), robots with more accurate classifications obtain higher weights while robots with more mistakes receive diminishing weights. As a swarm-level consensus needs to be built before full feedback is provided, each robot r in the swarm computes and exchanges *surrogate weights* for all possible outputs: $\omega_r^i = w_{t-1} e^{-\lambda^i}$, $\forall i = 1 : K$. Based on the exchanged surrogate weight vectors $\{\boldsymbol{\omega}_r\}_{r=1}^{K}$ every robot in the swarm makes the same prediction by calculating the consensus prediction margins $\mathbf{c}_w$ (line 12 in Algorithm 4). Using the exchanged surrogate weights, every robot decides which robots in the swarm are among the "top L experts" (i.e., the best L robots based on prediction performance).

With this strategy individual robots initiate (or inhibit) the broadcast of their local observations to the rest of the swarm for bootstrap learning. After individual robots have updated their classifiers with the gesture label (feedback from the human) as given in step 4 of Figure 5.9, a 5th step is performed which requires updating the consensus weight of the swarm. At this aim, after an output of class i is revealed and the human provides the actual/true label as full feedback to the swarm (see Section 5.3.1.2), the swarm's consensus weight is updated exponentially based on the loss of the learner: $w_t = w_{t-1} e^{-\lambda^i}$. The theoretical analysis of the CWSL approach in Algorithm 4 is provided next in Appendix D.2.

D.2 Theoretical Analysis of CWSL

The confidence interval in the prediction margin estimate of the CWSL method (see Appendix D) is bounded using the Azuma-Hoeffding bound [Azuma, 1967; Alon and Spencer, 2004] and the representer theorem [Crammer et al., 2008]. These are restated in Lemma 1 and Theorem 1 as given below.

Lemma 1 (Azuma-Hoeffding bound): *Let $X_1, \ldots, X_m$ be random variables with $|X_j| \leq \alpha_j$ for some $\alpha_1, \ldots, \alpha_m > 0$. Then for all positive integers m and positive reals b:*

$$\mathbf{P}\left\{\left|\sum_{j=1}^{m} X_j - \sum_{j=1}^{m} \mathrm{E}[X_j | X_1, \ldots, X_{j-1}]\right| \geq b\right\} \leq 2\exp\left\{-\frac{b^2}{2\sum_{j=1}^{m}\alpha_j^2}\right\}$$

The Azuma-Hoeffding theorem bounds the probability of the deviation of a sum of bounded random variables from their mean. The deviation probability decays exponentially with the distance b from the mean.

Theorem 1 (Representer Theorem): *The mean μ_t^i and covariance $\mathbf{\Sigma}_t^i$ can be represented as linear combinations of the input vectors*:

$$\mathbf{\Sigma}_t^i = \sum_{p,q=1}^{t-1} \pi_{p,q}^{(t)} \mathbf{x}_p \mathbf{x}_q^\top + aI, \quad \mu_t^i = \sum_{p}^{t-1} \nu_p^{(t)} \mathbf{x}_p,$$

where the variables $\pi_{p,q}^{(t)}$ and $\nu_p^{(t)}$ only depend up on the inner product of inputs [Crammer et al., 2008]:

$$\begin{aligned}
\nu_t^{(t+1)} &= 1, \\
\nu_j^{(t+1)} &= \nu_j^{(t)} + \alpha_t z_t \sum_{i}^{t-1} \pi_{i,j}^{(t)} \mathbf{x}_i \cdot \mathbf{x}_t, \text{ for } \forall j < t, \\
\pi_{i,j}^{(t+1)} &= -\beta_t \sum_{r,s} \pi_{i,r}^{(t)} \pi_{s,j}^{(t)} \mathbf{x}_r \cdot \mathbf{x}_t + \pi_{i,j}^{(t)}, \\
\pi_{j,t}^{(t)} &= \pi_{t,j}^{(t)} = -\beta_t \sum_{j,s}^{t-1} \pi_{j,s}^{(t)} \mathbf{x}_s \cdot \mathbf{x}_t \\
\pi_{t,t}^{(t+1)} &= -\beta_t
\end{aligned}$$

The representer theorem is similar to the one in kernel methods, in which the estimated function defined over a Reproducing Kernel Hilbert Space (RKHS) can be represented as a finite linear combination of kernel products evaluated at the input points in the training set. Therefore, with the representer theorem, the mean and covariance can be re-written as linear combinations of input vectors whose weights can be computed with Mercer kernels. This helps to intuitively see the dependence of the mean and covariance on the inputs. Note that the kernel

at time t can be built recursively as linear combination of kernels at time $t-1$. Next, the confidence interval in prediction margin estimate can be bounded, as stated in the following lemma.

Lemma 2 (Bayes optimal): *Assume there exists a Bayes optimal solution* $\boldsymbol{u}^i, \boldsymbol{\Sigma}^{*i}$ *for* $i = 1,\ldots,K$ *and* $\|\boldsymbol{u}^i - \boldsymbol{u}^j\|^2 \leq c$. *With probability* $1-\delta$, *for all labels* $i \in \{1,\ldots,K\}$:

$$\left|\boldsymbol{\mu}_t^i \cdot \mathbf{x}_t - \boldsymbol{u}^i \cdot \mathbf{x}_t\right| \leq \sqrt{2\ln(2M/\delta)\sum_{j=1}^{t-1}|\nu_j^{(t)}|^2}$$

In this lemma, the first assumption $\|\boldsymbol{u}^i - \boldsymbol{u}^j\|^2 \leq c$ means that the distance between bounded random variables is also bounded. The intuitive goal of this lemma is to bound the difference between the prediction margins of the adaptive learning classifier and its Bayes optimal classifier. The bound consists of two parts: the first part depends on δ, and the second part, composed of $\nu_j^{(t)}$'s, is considered as constant given a particular input sequence.

Proof 1: For each $i \in \{1,\ldots,K\}$, $X_j = \nu_j^{(t)}\mathbf{x}_j \cdot \mathbf{x}_t$ Lemma 1 is used, thus $|X_j| \leq |\nu_j^{(t)}|$, assuming that $\|\mathbf{x}_t\| \leq 1$. Then,

$$\sum_{j=1}^{t-1} X_j = \sum_{j=1}^{t-1} \nu_j^{(t)}\mathbf{x}_j \cdot \mathbf{x}_t = \boldsymbol{\mu}_t^i \cdot \mathbf{x}_t$$

and the Bayes optimal margin is:

$$\sum_{j=1}^{t-1} \mathrm{E}[X_j|X_1,\ldots,X_{j-1}] = \boldsymbol{u}^i \cdot \mathbf{x}_t$$

The following is the result of applying Lemma 1:

$$Pr\left[\left|\boldsymbol{\mu}_t^i \cdot \mathbf{x}_t - \boldsymbol{u}^i \cdot \mathbf{x}_t\right| \geq \sqrt{2\ln(2M/\delta)\sum_{j=1}^{t-1}|\nu_j^{(t)}|^2}\right] \leq \frac{\delta}{M}$$

By applying a union bound, then there is a guarantee that with a probability of $1-\delta$, for all labels $i \in \{1,\ldots,M\}$:

$$\left|\boldsymbol{\mu}_t^i \cdot \mathbf{x}_t - \boldsymbol{u}^i \cdot \mathbf{x}_t\right| \leq \sqrt{2\ln(2M/\delta)\sum_{j=1}^{t-1}|\nu_j^{(t)}|^2}$$

Now the difference between the distributions of the Bayes optimal classifier and CWSL is bounded. The difference is measured by the KL-divergence between

the two weight distributions, where the distance at time T is:

$$\chi_t = D_{\mathrm{KL}}\Big(\mathcal{N}(\boldsymbol{u}_t \cdot \mathbf{x}_t, \mathbf{x}_t^\top \Sigma_t^* \mathbf{x}_t) || \mathcal{N}(\boldsymbol{\mu}_t^{\hat{y}_t} \cdot \mathbf{x}_t, \mathbf{x}_t^\top \Sigma_t^{\hat{y}_t} \mathbf{x}_t)\Big)$$

where $\boldsymbol{u}_t, \Sigma_t^*$ represent one of K pairs $\boldsymbol{u}^i, \Sigma^{*i}$.

Theorem 2: *With probability $1-\delta$, the difference χ_t can be bounded by the term $\mathcal{O}(C_1 + \ln(2M/\delta)C_2)$, where C_1 and C_2 depend on the data only through inner products.*

Again, similar to Lemma 2, this difference is probabilistically quantified, to depend on the inner products of the inputs, which can be considered as constants given a particular input sequence. Similar to the representer theorem, the components depending on the inner products of the inputs can be quantified recursively, therefore given the inputs, the upper bound can be computed exactly. Then, given δ, it is possible to quantify the confidence regarding the divergence between the learned CWSL classifier and the Bayes optimal classifier. The proof makes use of Lemma 2.

Proof 2: Let $\dagger$ denote $\hat{y}_t$. By the definition of the Bayes optimal solution:

$$v_t^\dagger = \mathbf{x}_t^\top \Sigma_t^\dagger \mathbf{x}_t \geq \mathbf{x}_t^\top \Sigma_t^* \mathbf{x}_t = v_t^*$$

Furthermore,

$$\begin{aligned}
\chi_t &= \ln\Big(\frac{\mathbf{x}_t^\top \Sigma_t^\dagger \mathbf{x}_t}{\mathbf{x}_t^\top \Sigma_t^* \mathbf{x}_t}\Big) + \frac{\mathbf{x}_t^\top \Sigma_t^* \mathbf{x}_t}{\mathbf{x}_t^\top \Sigma_t^\dagger \mathbf{x}_t} + \frac{\big(\boldsymbol{\mu}_t^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}_t \cdot \mathbf{x}_t\big)^2}{\mathbf{x}_t^\top \Sigma_t^\dagger \mathbf{x}_t} - 1 \\
&= \ln\Big(\frac{v_t^\dagger}{v_t^*}\Big) + \frac{v_t^*}{v_t^\dagger} + \frac{\big(\boldsymbol{\mu}_t^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}_t \cdot \mathbf{x}_t\big)^2}{v_t^\dagger} - 1 \\
&= \ln\Big(\frac{v_t^\dagger}{v_t^*}\Big) + \frac{v_t^*}{v_t^\dagger} + \frac{\big(\boldsymbol{\mu}_t^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}^\dagger \cdot \mathbf{x}_t + \boldsymbol{u}^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}_t \cdot \mathbf{x}_t\big)^2}{v_t^\dagger} - 1 \\
&\leq \ln\Big(\frac{v_t^\dagger}{v_t^*}\Big) + \frac{v_t^*}{v_t^\dagger} + \frac{\big(\boldsymbol{\mu}_t^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}^\dagger \cdot \mathbf{x}_t\big)^2 + \big(\boldsymbol{u}^\dagger \cdot \mathbf{x}_t - \boldsymbol{u}_t \cdot \mathbf{x}_t\big)^2}{2v_t^\dagger} - 1 \\
&\overset{\text{(Lemma 2)}}{\leq} \ln\Big(\frac{v_t^\dagger}{v_t^*}\Big) + \frac{v_t^*}{v_t^\dagger} + \frac{2\ln(2M/\delta)\sum_{j=1}^{T-1} |\nu_j^{(T)}|^2 + \|\boldsymbol{u}^\dagger - \boldsymbol{u}_t\|^2 \|\mathbf{x}_t\|^2}{2v_t^\dagger} - 1 \\
&\leq \ln\Big(\frac{v_t^\dagger}{v_t^*}\Big) + \frac{v_t^*}{v_t^\dagger} + \frac{2\ln(2M/\delta)\sum_{j=1}^{T-1} |\nu_j^{(T)}|^2 + c}{2v_t^\dagger} - 1
\end{aligned}$$

with probability $1-\delta$. As $v_t^\dagger \geq v_t^*$, then:

$$\chi_t \leq \ln\left(\frac{v_t^\dagger}{v_t^*}\right) + \frac{2\ln(2M/\delta)\sum_{j=1}^{T-1}|\nu_j^{(T)}|^2 + c}{2v_t^\dagger}$$

According to the representer theorem [Crammer et al., 2008]:

$$\frac{\sum_{j=1}^{T-1}|\nu_j^{(T)}|^2}{2v_t^\dagger} \leq \frac{\phi^2\sum_{j=1}^{T-1}|\nu_j^{(T)}|^2}{2z_t^2\left(\boldsymbol{\mu}_t^\dagger\cdot\mathbf{x}_t\right)^2} = \frac{\phi^2\sum_{j=1}^{T-1}|\nu_j^{(T)}|^2}{2\left(\sum_p^{T-1}\nu_j^{(T)}\mathbf{x}_j\cdot\mathbf{x}_t\right)^2}$$

The final term also depends on inner products of inputs, so does $v_t^\dagger$. Therefore, $\chi_t \leq O\left(C_1 + \ln(2M/\delta)C_2\right)$ with:

$$C_1 = \ln\left(\frac{v_t^\dagger}{v_t^*}\right);\;\; C_2 = \frac{\phi^2\sum_{j=1}^{T-1}|\nu_j^{(T)}|^2}{2\left(\sum_p^{T-1}\nu_j^{(T)}\mathbf{x}_j\cdot\mathbf{x}_t\right)^2}$$

It is noteworthy here to mention that, in passing that the computed bound with components C_1 and C_2, depending on the data through inner products is similar to the quantity $\boldsymbol{\mu}^{*\top}\Sigma_{T+1}^{-1}\boldsymbol{\mu}^*$ in the exact convex bound [Crammer et al., 2008], is because the quantity Σ_{T+1}^{-1} is also represented in the same manner, as a direct result of the representer theorem.

Appendix E

Online Fusion of Classifiers on-board Individual Robots

Since each robot in the swarm is equipped with an individual (local) classifier, multiple robots in the swarm learn the same classification task in parallel from different viewpoints. When multiple classifiers (or *ensembles of classifiers*; see Section 3.1.2) on-board multiple robots are trained on different portions of the sensed data, online model fusion is fundamental for combining multiple classifiers, in order to produce a *better single classifier* and reduce duplicated learning efforts. As CW learning (see Appendix D) uses the mean $\boldsymbol{\mu}_t$ and the covariance Σ_t to update the distribution $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ of the weight vector $\mathbf{w}_t$, it offers an informed and effective way to fuse the information learned by multiple classifiers/robots.

Considering a set of $\{r_k\}_{k=1}^N$ robots in a swarm, where r_k corresponds to the kth robot, the combined model of all robots, also a Gaussian, can be computed as the one that is closest to all other k distributions in the sense of a chosen divergence. The distribution over the learning parameters in the CW algorithm (Appendix D) can be exploited, to provide a weighted combination of the learning parameters from individually trained classifiers. In this analysis, we consider that every robot in the swarm is equipped with a binary (two-class) classifier. In the case of KL divergence [Kullback, 1959]), the combined model parameters $(\bar{\mu}_i, \bar{\Sigma}_i)$ for each ith binary classifier can be represented by:

$$\bar{\Sigma}_i = \left(\sum_{k=1}^{N} (\Sigma_k^i)^{-1} \right)^{-1} \tag{E.1}$$

$$\bar{\mu}_i = \bar{\Sigma}_i \sum_{k=1}^{N} (\Sigma_k^i)^{-1} \mu_k^i \tag{E.2}$$

where (μ_k^i, Σ_k^i) denotes the ith binary CW classifier of the kth robot in the swarm. This approach can easily be extended for multi-class classifiers. With this strategy, classifier fusion can be performed for selective robots within a specific communication range (i.e., neighbouring robots located within a certain no. of hops), and the schedule for classifier fusion can be set periodically or when the difference (change) in the classifiers of the robots exceeds a threshold.

Bibliography

Abidi, S., Williams, M., and Johnston, B. Human pointing as a robot directive. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–68, 2013.

Aghajan, H. and Cavallaro, A. *Multi-camera networks: principles and applications*. Academic press, 2009.

Aghajan, H. and Wu, C. Layered and collaborative gesture analysis in multi-camera networks. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1377–1380, 2007.

Aghajan, H., Wu, C., and Kleihorst, R. Distributed vision networks for human pose analysis. In Mandic, D., Golz, M., Kuh, A., Obradovic, D., and Tanaka, T., editors, *Signal processing techniques for knowledge extraction and information fusion*, pages 181–200. Springer, 2008.

Alboul, L., Saez-Pons, J., and Penders, J. Mixed human-robot team navigation in the guardians project. In *Proc. of International Workshop on Safety, Security and Rescue Robotics*, pages 95–101, 2008.

Alissandrakis, A. and Miyake, Y. Human to robot demonstrations of routine home tasks: Acknowledgment and response to the robot's feedback. In Dautenhahn, K., editor, *Proc. of the Symposium New Frontiers in Human-Robot Interaction, Adaptive & Emergent Behaviour & Complex Systems (AISB)*, pages 9–15. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, Edinburgh, 2009.

Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. Toward conversational human-computer interaction. *AI magazine*, 22(4):27, 2001.

Alon, N. and Spencer, J. H. *The probabilistic method*. John Wiley & Sons, 2004.

Alonso-Mora, J., Lohaus, S. H., Leemann, P., Siegwart, R., and Beardsley, P. Gesture based human – multi-robot swarm interaction and its application to an interactive display. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5948–5953, 2015.

Aragues, R., Cortes, J., and Sagues, C. Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4):840–854, 2012.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

Argyle, M., Salter, V., Nicholson, H., Williams, M., and Burgess, P. The communication of inferior and superior attitudes by verbal and non-verbal signals. *British journal of social and clinical psychology*, 9:222–231, 1970.

Arkin, R. C. *Behavior-based Robotics*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

Arygle, M. *Bodily Communication*. Madison: International Universities Press, 2nd edition, 1988.

Austermann, A. and Yamada, S. "good robot", "bad robot"–analyzing users' feedback in a human-robot teaching task. In *Proc. of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 41–46, 2008.

Azoury, K. S. and Warmuth, M. K. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Journal of Maching Learning Research (JMLR)*, 43(3):211–246, 2001.

Azuma, K. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., Naur, P., Perlis, A. J., Rutishauser, H., Samelson, K., Vauquois, B., et al. Report on the algorithmic language algol 60. *Numerische Mathematik*, 2(1):106–136, 1960.

Bailey, T., Julier, S., and Agamennoni, G. On conservative fusion of information with unknown non-gaussian dependence. In *Proc. of International Conference on Information Fusion*, pages 1876–1883, 2012.

Batalin, M. A. and Sukhatme, G. S. Sensor coverage using mobile robots and stationary nodes. *SPIE Proceedings*, 4868:269–276, 2002.

Benediktsson, J. A. and Swain, P. H. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):688–704, 1992.

Beni, G. From swarm intelligence to swarm robotics. In Sahin, E. and Spears, W. M., editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science (LNCS)*, pages 1–9. Springer Berlin Heidelberg, 2005.

Berman, P., Garay, J. A., and Perry, K. J. Towards optimal distributed consensus. In *Proc. of Annual Symposium on Foundations of Computer Science*, pages 410–415, 1989.

Bonabeau, E., Dorigo, M., and Theraulaz, G. *From Natural to Artificial Swarm Intelligence*. Oxford University Press, 1999.

Bonani, M., Longchamp, V., Magnenat, S., Retornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4187–4193, 2010.

Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168. The MIT Press, 2007.

Bottou, L. and LeCun, Y. Large scale online learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16. The MIT Press, 2004.

Bramberger, M., Doblander, A., Maier, A., Rinner, B., and Schwabach, H. Distributed embedded smart cameras for surveillance applications. *Computer*, 39 (2):68–75, 2006.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

Breazeal, C. Social interactions in HRI: the robot view. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(2):181–186, 2004.

Breiman, L. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Bruemmer, D. J., Few, D. A., Boring, R. L., Marble, J. L., Walton, M. C., and Nielsen, C. W. Shared understanding for collaborative control. *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, 35(4): 494–504, 2005.

Camazine, S. *Self-organization in biological systems*. Princeton University Press, Princeton, NJ, USA, 2003.

Canedo-Rodriguez, A., Regueiro, C. V., Iglesias, R., Alvarez-Santos, V., and Pardo, X. M. Self-organized multi-camera network for ubiquitous robot deployment in unknown environments. *Robotics and Autonomous Systems*, 61(7):667–675, 2013.

Capkun, S. and Hubaux, J.-P. Secure positioning of wireless devices with application to sensor networks. In *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1917–1928, 2005.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge University Press, 2006.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

Cesa-Bianchi, N., Gentile, C., and Orabona, F. Robust bounds for classification via selective sampling. In *Proc. of the ICML*, pages 121–128, 2009.

ChaLearn Gesture Challenge. Challenge and workshop on pose recovery, action and gesture recognition, 2014. URL http://gesture.chalearn.org/2014-looking-at-people-challenge.

Chambers, N., Allen, J., Galescu, L., and Jung, H. A dialogue-based approach to multi-robot team control. In Parker, L. E., Schneider, F. E., and Schultz, A. C., editors, *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 257–262. Springer Netherlands, 2005.

Chellappa, R., Heinzelman, W., Konrad, J., Schonfeld, D., and Wolf, M. Special section on distributed camera networks: Sensing, processing, communication, and implementation. *IEEE Transactions on Image Processing*, 19(10):2513–2515, 2010.

Chen, G., Chen, G., Zhang, J., Chen, S., and Zhang, C. Beyond banditron: A conservative and efficient reduction for online multiclass prediction with bandit setting model. In *Proc. of IEEE International Conference on Data Mining*, pages 71–80, 2009.

Chen, J., Zhang, C., Xue, X., and Liu, C.-L. Fast instance selection for speeding up support vector machines. *Knowledge-Based Systems*, 45:1–7, 2013.

Chen, J. Y. C., Barnes, M. J., and Harper-Sciarini, M. Supervisory control of multiple robots: Human-performance issues and user-interface design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(4):435–454, 2011.

Chen, T. Novel machine learning for hand gesture recognition using multiple view. In *Proc. of IEEE International Conference on Control, Automation and Systems Engineering*, pages 575–579, 2009.

Chiu, H.-P., Kaelbling, L. P., and Lozano-Pérez, T. Virtual training for multi-view object class recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

Choras, R. S. Hand shape and hand gesture recognition. In *Proc. of IEEE Symposium on Industrial Electronics and Applications*, pages 145–149, 2009.

Cireşan, D., Meier, U., Gambardella, L. M., and Schmidhuber, J. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12): 3207–3220, 2010.

Cireşan, D., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In *Proc. of 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1237–1242, 2011.

Claesen, M., Smet, F. D., Suykens, J. A. K., and Moor, B. D. EnsembleSVM: A library for ensemble learning using support vector machines. *Journal of Machine Learning Research (JMLR)*, 15:141–145, 2014.

Clare, A. S. and Cummings, M. L. Task-based interfaces for decentralized multiple unmanned vehicle control. In *Proc. of AUVSI: Unmanned Systems North America*, 2011.

Cohn, D., Ghahramani, Z., and Jordan, M. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Coppin, G. and Legras, F. Controlling swarms of unmanned vehicles through user-centered commands. In *Proc. of AAAI Fall Symposium Series: Human-Control of Bioinspired Swarms*, pages 21–25, 2012.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Couture-Beil, A., Vaughan, R. T., and Mori, G. Selecting and commanding individual robots in a multi-robot system. In *Proc. of Canadian Conference on Computer and Robot Vision (CRV)*, pages 159–166, 2010a.

Couture-Beil, A., Vaughan, R. T., and Mori, G. Selecting and commanding individual robots in a vision-based multi-robot system. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, pages 355–356, 2010b.

Cox, M. R. and Budhu, M. A practical approach to grain shape quantification. *Engineering Geology*, 96(1-2):1–16, 2008.

Crammer, K. and Gentile, C. Multiclass classification with bandit feedback using adaptive regularization. In *Proc. of IEEE International Conerence on Machine Learning (ICML)*, pages 273–280, 2011.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, 7:551–585, 2006.

Crammer, K., Fern, M. D., and Pereira, O. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, 2008.

Crammer, K., Kulesza, A., and Dredze, M. Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems (NIPS)*, 22:414–422, 2009.

Crammer, K., Dredze, M., and Pereira, F. Confidence-weighted linear classification for text categorization. *Journal of Machine Learning Research (JMLR)*, 13 (1):1891–1926, 2012.

Crammer, K., Kulesza, A., and Dredze, M. Adaptive regularization of weight vectors. *Machine Learning*, 91(2):155–187, 2013.

Crandall, J. W. and Cummings, M. L. Developing performance metrics for the supervisory control of multiple robots. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 33–40, 2007a.

Crandall, J. W. and Cummings, M. L. Identifying predictive metrics for supervisory control of multiple robots. *IEEE Transactions on Robotics*, 23(5):942–951, 2007b.

Cruz, C., Sucar, L. E., and Morales, E. F. Real-time face recognition for human-robot interaction. In *Proc. of IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6, 2008.

Cui, S., Xiao, J.-J., Goldsmith, A. J., Luo, Z.-Q., and Poor, H. V. Estimation diversity and energy efficiency in distributed sensing. *IEEE Transactions on Signal Processing*, 55(9):4683–4695, 2007.

Cummings, M. L. Human supervisory control of swarming networks. In *Proc. of the 2nd Annual Swarming: Autonomous Intelligent Networked Systems Conference*, 2004.

Daily, M., Cho, Y., Martin, K., and Payton, D. World embedded interfaces for human-robot interaction. In *Proc. of Annual Hawaii International Conference on System Sciences*, 2003.

Daliri, M. R. and Torre, V. Robust symbolic representation for shape recognition and retrieval. *Pattern Recognition*, 41(5):1782–1798, 2008.

Das, D., Ghosh, M., Chakraborty, C., Pal, M., and Maity, A. K. Invariant moment based feature analysis for abnormal erythrocyte recognition. In *Proc. of International Conference on Systems in Medicine and Biology*, pages 242–247, 2010.

DeGroot, M. H. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

Deits, R., Tellex, S., Thaker, P., Simeonov, D., Kollar, T., and Roy, N. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2(2):58–79, 2013.

Di Caro, G. A., Giusti, A., Nagi, J., and Gambardella, Luca M. A simple and efficient approach for cooperative incremental learning in robot swarms. In *Proc. of International Conference on Advanced Robotics (ICAR)*, pages 1–8, 2013a.

Di Caro, G. A., Kudelski, M., Flushing, E.F., Nagi, J., Ahmed, I., and Gambardella, L. M. On-line supervised learning of link quality estimates in wireless networks. In *Proc. of the 12th IEEE/IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 69–76, 2013b.

Dietl, M., Gutmann, J. S., and Nebel, B. Cooperative sensing in dynamic environments. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1706–1713, 2001.

Dillenbourg, P. What do you mean by collaborative learning. In *Collaborative-learning: Cognitive and computational approaches*, pages 1–19. Elsevier, 1999.

Dorigo, M., Maniezzo, V., and Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 26(1):29–41, Feb 1996.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., and Gambardella, L. M. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17 (2-3):223–245, 2004.

Dorigo, M., Floreano, D., Gambardella, L.M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., Burnier, D., Campo, A., Christensen, A.L., Decugniere, A., Di Caro, G. A., Ducatelle, F., Ferrante, E., Forster, A., Martinez Gonzales, J., Guzzi, J., Longchamp, V., Magnenat, S., Mathews, N., Montes de Oca, M., O'Grady, R., Pinciroli, C., Pini, G., Retornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stutzle, T., Trianni, V., Tuci, E., Turgut, A.E., and Vaussard, F. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics Automation Magazine*, 20(4): 60–71, 2013.

Dorigo, M. et al. Swamanoid: Towards humanoid robotic swarms, 2006. URL http://www.swarmanoid.org/. FET-OPEN project funded by the European Commission.

Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J. D., and Gandy, M. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, 2005.

Dredze, M., Crammer, K., and Pereira, F. Confidence-weighted linear classification. In *Proc. of International Conference on Machine Learning (ICML)*, pages 264–271, 2008.

Duan, P., Tian, G., and Zhang, W. Human localization based on distributed laser range finders. *International Journal of Hybrid Information Technology*, 7(3): 311–324, 2014.

Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. Self-organized cooperation between robotic swarms. *Swarm Intelligence Journal*, 5(2):73–96, 2011.

Duta, N. A survey of biometric technology based on hand shape. *Pattern Recognition*, 42(11):2797–2806, 2009.

Erdem, Z., Polikar, R., Gurgen, F., and Yumusak, N. Ensemble of svms for incremental learning. In *Proc. of International Conference on Multiple Classifier Systems*, pages 246–256, 2005.

Espès, D., Pistea, A.M., Canaff, C., Iordache, I., Le Parc, P., and Radoi, E. New method for localization and human being detection using uwb technology: Helpful solution for rescue robots. *Computing Research Repository (CoRR)*, 2013. URL http://arxiv.org/abs/1312.4162.

Fagiolini, A., Pellinacci, M., Valenti, G., Dini, G., and Bicchi, A. Consensus-based distributed intrusion detection for multi-robot systems. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA) 2008*, pages 120–127, 2008.

Fang, Y., Cheng, J., Wang, K., and Lu, H. Hand gesture recognition using fast multi-scale analysis. In *Proc. of International Conference on Image and Graphics*, pages 694–698, 2007a.

Fang, Y., Wang, K., Cheng, J., and Lu, H. A real-time hand gesture recognition method. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 995–998, 2007b.

Fleck, S. and Straqer, W. Smart camera based monitoring system and its application to assisted living. *Proceedings of the IEEE*, 96(10):1698–1714, Oct. 2008.

Flouri, K., Beferull-Lozano, B., and Tsakalides, P. Optimal gossip algorithm for distributed consensus svm training in wireless sensor networks. In *Proc. of IEEE International Conference on Digital Signal Processing*, pages 1–6, 2009.

Folmer, E. and Morelli, T. Spatial gestures using a tactile-proprioceptive display. In *Proc. of International Conference on Tangible, Embedded and Embodied Interaction*, pages 139–142, 2012.

Fong, T., Thorpe, C., and Baur, C. Collaboration, dialogue, human-robot interaction. In Jarvis, R. A. and Zelinsky, A., editors, *Robotics Research*, volume 6 of *Springer Tracts in Advanced Robotics*, pages 255–266. Springer Berlin Heidelberg, 2003.

Fong, T. W., Nourbakhsh, I., Kunz, C., Flueckiger, L., Schreiner, J., Ambrose, R., Burridge, R., Simmons, R., Hiatt, L. M., Schultz, A., Trafton, J. G., and Bugajska, M. The peer-to-peer human-robot interaction project. In *Proc. of AIAA Space*, pages 2005–6750, 2005.

Fong, T. W., Bualat, M., Edwards, L., Flueckiger, L., Kunz, C., Lee, S. Y., Park, E., To, V., Utz, H., Ackner, N., Armstrong-Crews, N., and Gannon, J. Human-robot site survey and sampling for space exploration. In *Proc. of AIAA Space*, September 2006.

Fu, Z., Robles-Kelly, A., and Zhou, J. Milis: Multiple instance learning with instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):958–977, 2011.

Garcia-Pedrajas, N. Constructing ensembles of classifiers by means of weighted instance selection. *IEEE Transactions on Neural Networks*, 20(2):258–277, 2009.

Gasparri, A., Priolo, A., and Ulivi, G. A swarm algorithm for human supervisory control based on local communication. Technical Report RT-DIA-193-2012, Universitá Degli Studi Roma Tre, Rome, Italy, 2012.

Gay-Bellile, V., Tamaazousti, M., Dupont, R., and Collette, S. N. A vision-based hybrid system for real-time accurate localization in an indoor environment. In *Proc. of the International Conference on Computer Vision Theory and Applications*, pages 216–222, 2010.

Genest, C. and Zidek, J. V. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Gerkey, B. P. and Matarić, M. J. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9): 939–954, 2004.

Gerkey, B. P., Vaughan, R. T., and Howard, A. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proc. of the International Conference on Advanced Robotics (ICAR)*, pages 317–323, 2003.

Giovannangeli, C. and Gaussier, P. Interactive teaching for vision-based mobile robots: A sensory-motor approach. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(1):13–28, 2010.

Giraud-Carrier, C. A note on the utility of incremental learning. *AI Commununi-cations*, 13(4):215–223, 2000.

Giusti, A., Nagi, J., Gambardella, L. M., Bonardi, S., and Di Caro, G. A. Human-swarm interaction through distributed cooperative gesture recognition. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, pages 401–402, 2012a.

Giusti, A., Nagi, J., Gambardella, L. M., and Di Caro, G. A. Distributed consensus for interaction between humans and mobile robot swarms. In *Proc. of International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (Demonstration Track)*, pages 1503–1504, 2012b.

Giusti, A., Nagi, J., Gambardella, L. M., and Di Caro, G. A. Cooperative sensing and recognition by a swarm of mobile robots. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 551–558, 2012c.

Glaude, H., Akrimi, F., Geist, M., and Pietquin, O. A non-parametric approach to approximate dynamic programming. In *Proc. of International Conference on Machine Learning and Applications (ICMLA)*, pages 317–322, 2011.

Goodrich, M. A. and Olsen, D. R. Seven principles of efficient human robot interaction. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pages 3942–3948, 2003.

Goodrich, M. A. and Schultz, A. C. Human-robot interaction: a survey. *Foundation and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

Grieder, R., Alonso-Mora, J., Bloechlinger, C., Siegwart, R., and Beardsley, P. Multi-robot control and interaction with a hand-held tablet. In *Workshop at IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

Gromov, B., Gambardella, L. M., and Di Caro, G. A. Wearable multi-modal interfaces for mixed-initiative interaction in human multi-robot teams. In *Proc. of ICRA Workshop on Fielded Multi-robot Systems Operating on Land, Sea, and Air (Poster with extended abstract)*, Stockholm, Sweden, 2016.

Guestrin, C., Krause, A., and Singh, A. P. Near-optimal sensor placements in gaussian processes. In *Proc. of International Conference on Machine Learning (ICML)*, pages 265–272, 2005.

Guinaldo, M., Fábregas, E., Farias, G., Dormido-Canto, S., Chaos, D., Sánchez, J., and Dormido, S. A mobile robots experimental environment with event-based wireless communication. *Sensors*, 13(7):9396–9413, 2013.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

Harriott, C. E., Seiffert, A. E., Hayes, S. T., and Adams, J. A. Biologically-inspired human-swarm interaction metrics. In *Proc. of Human Factors and Ergonomics Society Annual Meeting*, pages 1471–1475, 2014.

Harris, T. K., Banerjee, S., and Rudnicky, A. I. Heterogeneous multi-robot dialogues for search tasks, 2005.

Hayes, S. T. and Adams, J. A. Human-swarm interaction: Sources of uncertainty. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 170–171, 2014.

Hearst, M. A. Trends & controversies: Mixed-initiative interaction. *IEEE Intelligent Systems*, 14(5):14–23, 1999.

Hu, R.-X., Jia, W., Zhang, D., Gui, J., and Song, L.-T. Hand shape recognition based on coherent distance shape contexts. *Pattern Recognition*, 45(9):3348–3359, 2012.

Huang, Y., Huang, K., Tao, D., Tan, T., and Li, X. Enhanced biologically inspired model for object recognition. *IEEE Transactions on Systerms, Man, and Cybernetics–Part B: Cybernetics*, 41(6):1668–1680, 2011a.

Huang, Y., Monekosso, D., Wang, H., and Augusto, J. C. A concept grounding approach for glove-based gesture recognition. In *Proc. of the International Conference on Intelligent Environments*, pages 358–361, 2011b.

Humphrey, C. M., Henk, C., Sewell, G., Williams, B. W., and Adams, J. A. Assessing the scalability of a multiple robot interface. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 239–246, 2007.

Iyengar, S. S. and Brooks, R. R. *Distributed Sensor Networks: Sensor Networking and Applications*. Chapman & Hall, 2012.

Jones, G., Berthouze, N., Bielski, R., and Julier, S. Towards a situated, multimodal interface for multiple uav control. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1739–1744, 2010.

Jones, H. and Rock, S. Dialogue-based human-robot interaction for space construction teams. In *Proc. of IEEE Aerospace Conference*, pages 3645–3653, 2002.

Jones, M. and Viola, P. Fast multi-view face detection. Technical Report TR-20003-96, Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts, USA, 2003.

Jorstad, A., DeMenthon, D., Wang, I. J., and Burlina, P. Distributed consensus on camera pose. *IEEE Transactions on Image Processing*, 19(9):2396–2407, 2010.

Joshi, A. J., Porikli, F., and Papanikolopoulos, N. Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2995–3002, 2010.

Joshi, A. J., Porikli, F., and Papanikolopoulos, N. Scalable active learning for multi-class image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11):2259–2273, 2012.

Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. Efficient bandit algorithms for online multiclass prediction. In *Proc. of IEEE International Conference on Machine Learning (ICML)*, pages 440–447, 2008.

Kakumanu, P., Makrogiannis, S., and Bourbakis, N. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007.

Kalnishkan, Y. and Vyugin, M. V. The weak aggregating algorithm and weak mixability. In *Learning theory*, pages 188–203. Springer, 2005.

Kankuekul, P., Kawewong, A., Tangruamsub, S., and Hasegawa, O. Online incremental attribute-based zero-shot learning. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3657–3664, 2012.

Kawewong, A., Tangruamsub, S., Kankuekul, P., and Hasegawa, O. Fast online incremental transfer learning for unseen object classification using self-organizing incremental neural networks. In *Proc. of International Joint Conference on Neural Networks (IJCNN)*, pages 749–756, 2011.

Kelly, D., McDonald, J., and Markham, C. A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11):1359–1368, 2010.

Kim, K. and Medioni, G. G. Distributed visual processing for a home visual sensor network. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 1–6, 2008.

Kim, S.-W., Lee, J.-Y., Kim, D., You, B.-J., and Doh, N. L. Human localization based on the fusion of vision and sound system. In *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 495–498, 2011.

Kira, Z. and Potter, M. A. Exerting human control over decentralized robot swarms. In *Proc. of International Conference on Autonomous Robots and Agents*, pages 566–571, 2009.

Kirishima, T., Manabe, Y., Sato, K., and Chihara, K. Real-time multiview recognition of human gestures by distributed image processing. *Journal on Image and Video Processing*, pages 1–13, 2010.

Kivinen, J. and Warmuth, M. K. Averaging expert predictions. In *Computational Learning Theory*, pages 153–167. Springer, 1999.

Klanke, S., Vijayakumar, S., and Schaal, S. A library for locally weighted projection regression. *Journal of Machine Learning Research (JMLR)*, 9:623–626, 2008. URL http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr.

Kokiopoulou, E. and Frossard, P. Distributed svm applied to image classification. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 1753–1756, 2006.

Kokiopoulou, E. and Frossard, P. Distributed classification of multiple observations by consensus. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 2697–2700, 2010.

Kokiopoulou, E. and Frossard, P. Distributed classification of multiple observation sets by consensus. *IEEE Transactions on Signal Processing*, 59(1):104–114, 2011.

Kolling, A., Nunnally, S., and Lewis, M. Towards human control of robot swarms. In *Proc. of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 89–96, 2012.

Kolling, A., Sycara, K., Nunnally, S., and Lewis, M. Human swarm interaction: An experimental study of two types of interaction with foraging swarms. *Journal of Human-Robot Interaction*, 2(2):103–128, 2013.

Kolling, A., Walker, P., Chakraborty, N., Sycara, K., and Lewis, M. Human interaction with robot swarms: A survey. *IEEE Transactions on Human-Machine Systems*, 46(1):9–26, 2016.

Konda, K., Königs, A., Schulz, H., and Schulz, D. Real time interaction with mobile robots using hand gestures. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 177–178, 2012.

Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proc. of International Conference on Information Processing in Sensor Networks*, pages 2–10, 2006.

Kulkarni, P., Ganesan, D., Shenoy, P., and Lu, Q. Senseye: a multi-tier camera sensor network. In *Proc. of Annual ACM International Conference on Multimedia*, pages 229–238, 2005.

Kullback, S. *Statistics and information theory*. J. Wiley and Sons, New York, 1959.

Kumar, V., Bullo, F., Koditschek, D., Jadbabaie, A., Morse, A. S., Pappas, G., Rus, D., Sastry, S. S., and Skelly, D. Swarms: Scalable swarms of autonomous robots and mobile sensors. Technical Report W911NF-05-1-0219, University of Pennsylvania, Philadelphia, USA, 2013.

Kurdyukova, E., Redlin, M., and André, E. Studying user-defined ipad gestures for interaction in multi-display environment. In *Proc. of ACM International Conference on Intelligent User Interfaces*, pages 93–96, 2012.

Kwon, D. Y. and Gross, M. A framework for 3d spatial gesture design and modeling using a wearable input device. In *Proc. of IEEE International Symposium on Wearable Computers*, pages 1–4, 2007.

Lang, C., Wachsmuth, S., Wersing, H., and Hanheide, M. Facial expressions as feedback cue in human-robot interaction - a comparison between human and automatic recognition performances. In *Proc. of IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, pages 79–85, 2010.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Leistner, C., Roth, P. M., Grabner, H., Bischof, H., Starzacher, A., and Rinner, B. Visual on-line learning in distributed camera networks. In *Proc. of ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–10, 2008.

Lewis, B. and Sukthankar, G. Configurable human-robot interaction for multi-robot manipulation tasks. In *Proc. of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1219–1220, 2012.

Li, J., Li, K., and Wei, Z. Improving sensing coverage of wireless sensor networks by employing mobile robots. In *Proc. of IEEE International Conference on Robotics and Biomimetics*, pages 899–903, 2007.

Li, Y., Gong, S., Sherrah, J., and Liddell, H. Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5):413–427, 2004.

Lichtenstern, M., Frassl, M., Perun, B., and Angermann, M. A prototyping environment for interaction between a human and a robotic multi-agent system. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 185–186, 2012.

Lichtenstern, M., Angermann, M., Frassl, M., Berthold, G., Julian, B. J., and Rus, D. Pose and paste – an intuitive interface for remote navigation of a multi-robot system. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1632–1639, 2013.

Linan, C. C. cvBlob. http://cvblob.googlecode.com, 2007. URL http://cvblob.googlecode.com.

Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

Liu, B., Brass, P., Dousse, O., Nain, P., and Towsley, D. Mobility improves coverage of sensor networks. In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 300–308, 2005.

Liu, H. and Motoda, H. *Instance selection and construction for data mining*, volume 608. Springer Science & Business Media, 2013.

Loper, M. M., Koenig, N. P., Chernova, S. H., Jones, C. V., and Jenkins, O. C. Mobile human-robot teaming with environmental tolerance. In *Proc. of ACM/IEEE International Conference on Human Robot Interaction (HRI)*, pages 157–164, 2009.

Lopes, M., Gouyon, F., Koerich, A. L., and Oliveira, L. E. S. Selection of training instances for music genre classification. In *Proc. of International Conference on Pattern Recognition (ICPR)*, pages 4569–4572, 2010.

Mai, N. T., Hai, T. T., and Son, N. V. Wizard of oz for designing hand gesture vocabulary in human-robot interaction. In *Proc. of the 3rd International Conference on Knowledge and Systems Engineering*, pages 232–238, 2011.

Malima, A., Ozgur, E., and Cetin, M. A fast algorithm for vision-based hand gesture recognition for robot control. In *Proc. of IEEE Conference on Signal Processing and Communications Applications*, pages 1–4, 2006.

Martin, C., Steege, F.-F., and Gross, H.-M. Estimation of pointing poses for visually instructing mobile robots under real world conditions. *Robotics and Autonomous Systems*, 58(2):174–185, 2010.

Martinoli, A. and Easton, K. Modeling swarm robotic systems. In *Experimental Robotics VIII*, pages 297–306. Springer, 2003.

Martinoli, A., Easton, K., and Agassounon, W. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4-5):415–436, 2004.

MATLAB R2014b documentation. Regionprops–Measure properties of image regions, 2014. URL http://www.mathworks.com/help/images/ref/regionprops.html.

Matsumoto, D. Culture and nonverbal behavior. *Handbook of nonverbal communication*, pages 219–235, 2006.

McLurkin, J. and Smith, J. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Proc. of International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2004.

McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., and Schmidt, B. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *Proc. of AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 72–75, 2006.

Micire, M., Desai, M., Courtemanche, A., Tsui, K. M., and Yanco, H. A. Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces. In *Proc. of ACM International Conference on Interactive Tabletops and Surfaces*, pages 41–48, 2009.

Mihaylov, M., Tuyls, K., and Nowé, A. Decentralized learning in wireless sensor networks. In Taylor, M. E. and Tuyls, K., editors, *Adaptive and Learning Agents*, volume 5924 of *Lecture Notes in Computer Science (LNCS)*, pages 60–73. Springer Berlin Heidelberg, 2010.

Milligan, B., Mori, G., and Vaughan, R. T. Selecting and commanding groups in a multi-robot vision based system. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, pages 415–415, 2011.

Mitra, S. and Acharya, T. Gesture recognition: A survey. *IEEE Transactions on Systerms, Man, and Cybernetics–Part C: Applications and Reviews*, 43(3):311–324, 2007.

Mohammad, Y. and Nishida, T. Talkback: Feedback from a miniature robot. In Orgun, M. A. and Thornton, J., editors, *Advances in Artificial Intelligence*, volume 4830 of *Lecture Notes in Computer Science (LNCS)*, pages 357–366. Springer Berlin Heidelberg, 2007.

Mohammad, Y. and Nishida, T. Getting feedback from a miniature robot. In *Proc. of International Conference on Information and Automation*, pages 941–947, 2008.

Monajjemi, V. M., Wawerla, J., Vaughan, R. T., and Mori, G. Hri in the sky: Creating and commanding teams of uavs with a vision-mediated gestural interface. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 617–623, 2013.

Monajjemi, V. M., Pourmehr, S., Sadat, S. A., Zhan, F., Wawerla, J., Mori, G., and Vaughan, R. T. Integrating multi-modal interfaces to command uavs. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, page 106, 2014.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. The e-puck, a robot designed for education in engineering. In *Proc. of Conference on Autonomous Robot Systems and Competitions*, pages 59–65, 2009.

Murphy, R. R. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, 34(2):138–153, 2004.

Muslea, I., Minton, S., and Knoblock, C. A. Active + semi-supervised learning = robust multi-view learning. In *Proc. of International Conference on Machine Learning (ICML)*, pages 435–442, 2002.

Naghsh, A. M., Gancet, J., Tanoto, A., and Roast, C. Analysis and design of human-robot swarm interaction in firefighting. In *Proc. of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 255–260, 2008.

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., and Gambardella, L. M. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Proc. of IEEE International Conference on Signal and Image Processing Applications*, pages 342–347, 2011.

Nagi, J., Di Caro, G. A., Giusti, A., Nagi, F., and Gambardella, L. M. Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems. In *Proc. of International Conference on Machine Learning and Applications (ICMLA)*, pages 27–32, 2012a.

Nagi, J., Ngo, H., Giusti, A., Gambardella, L. M., Schmidhuber, J., and Di Caro, G. A. Incremental learning using partial feedback for gesture-based human-swarm interaction. In *Proc. of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 898–905, 2012b.

Nagi, J., Di Caro, G. A., Giusti, A., and Gambardella, L. M. Learning symmetric face pose models online using locally weighted projectron regression. In *Proc.*

of IEEE International Conference on Image Processing (ICIP), pages 1400–1404, 2014a.

Nagi, J., Giusti, A., Di Caro, G. A., and Gambardella, L. M. Human control of uavs using face pose estimates and hand gestures. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Late Breaking Report)*, pages 252–253, 2014b.

Nagi, J., Giusti, A., Gambardella, L. M., and Di Caro, G. A. Human-swarm interaction using spatial gestures. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3841, 2014c.

Nagi, J., Giusti, A., Nagi, F., Gambardella, L. M., and Di Caro, G. A. Online feature extraction for the incremental learning of gestures in human-swarm interaction. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3331–3338, 2014d.

Nagi, J., Ngo, H., Schmidhuber, J., Gambardella, L. M., and Di Caro, G. A. Human-robot cooperation: Fast, interactive learning from binary feedback. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, page 107, 2014e.

Nagi, J., Ngo, H., Gambardella, L. M., and Di Caro, G. A. Wisdom of the swarm for cooperative-decision making in human-swarm interaction. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1802–1808, 2015.

Naikal, N., Yang, A. Y., and Sastry, S. S. Towards an efficient distributed object recognition system in wireless smart camera networks. In *Proc. of Conference on Information Fusion*, pages 1–8, 2010.

Naseer, T., Sturm, J., and Cremers, D. Followme: Person following and gesture recognition with a quadrocopter. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 624–630, 2013.

Naur, P., Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., Perlis, A. J., Rutishauser, H., Samelson, K., Vauquois, B., et al. Revised report on the algorithmic language algol 60. *Communications of the ACM*, 6(1):1–17, 1963.

Navarro, I. and Matía, F. An introduction to swarm robotics. *ISRN Robotics*, 2013, 2012.

Navarro-Serment, L. E., Dolan, J. M., and Khosla, P. K. Optimal sensor placement for cooperative distributed vision. In *Proc. of IEEE Interantional Conference on Robotics and Automation (ICRA)*, pages 939–944, 2004.

Navia-Vazquez, A., Gutierrez-Gonzalez, D., Parrado-Hernandez, E., and Navarro-Abellan, J. J. Distributed support vector machines. *IEEE Transactions on Neural Networks*, 17(4):1091–1097, 2006.

Ngo, H., Luciw, M., Vien, N., Nagi, J., Forster, A., and Schmidhuber, J. Efficient interactive multiclass learning from binary feedback. *ACM Transactions on Interactive Intelligent Systems*, 4(3):1–25, 2014.

Nickel, K. and Stiefelhagen, R. Visual recognition of pointing gestures for human-robot interaction. *Image Vision Computing*, 25(12):1875–1884, 2007.

Olfati-Saber, R. Distributed kalman filtering for sensor networks. In *Proc. of IEEE Conference on Decision and Control*, pages 5492–5498, 2007.

Olfati-Saber, R. and Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

Olfati-Saber, R. and Sandell, N. F. Distributed tracking in sensor networks with limited sensing range. In *Proc. of American Control Conference*, pages 3157–3162, 2008.

Olfati-Saber, R. and Shamma, J. S. Consensus filters for sensor networks and distributed sensor fusion. In *Proc. of IEEE Conference on Decision and Control*, pages 6698–6703, 2005.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

Olsen, D. R. and Wood, S. B. Fan-out: measuring human control of multiple robots. In *Proc. of ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 231–238, 2004.

Olsen, D. R., Wood, S. B., and Turner, J. Metrics for human driving of multiple robots. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2315–2320, 2004.

Olvera-Löpez, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., and Kittler, J. A review of instance selection methods. *Artificial Intelligence Review*, 34(2): 133–143, 2010.

OpenCV 3.0.0-dev documentation. Contour properties, 2014. URL http://goo.gl/TBGfUt.

Oza, N. Online bagging and boosting. In *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pages 2340–2345, 2005.

Panait, L. and Luke, S. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

Parikh, D. and Polikar, R. An ensemble-based incremental learning approach to data fusion. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 37(2):437–450, 2007.

Parvini, F., Mcleod, D., Shahabi, C., Navai, B., Zali, B., and Ghandeharizadeh, S. An approach to glove-based gesture recognition. In *Human-Computer Interaction. Novel Interaction Methods and Techniques*, pages 236–245. Springer, 2009.

Payne, J., Keir, P., Elgoyhen, J., McLundie, M., Naef, M., Horner, M., and Anderson, P. Gameplay issues in the design of spatial 3d gestures for video games. In *Proc. of Conference on Human Factors in Computing Systems (CHI) (Extended Abstract)*, pages 1217–1222, 2006.

Payton, D., Daily, M. J., Hoff, B., Howard, M. D., and Lee, C. L. Pheromone robotics. In *Intelligent Systems and Smart Manufacturing*, pages 67–75, 2001.

Payton, D., Estkowski, R., and Howard, M. Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems*, 44(3):229–240, 2003.

Payton, D., Estkowski, R., and Howard, M. *International Workshop on Swarm Robotics*, chapter Pheromone Robotics and the Logic of Virtual Pheromones, pages 45–57. Springer Berlin Heidelberg, 2005.

Peissig, J. J., Wasserman, E. A., Young, M. E., and Biederman, I. Learning an object from multiple views enhances its recognition in an orthogonal rotational axis in pigeons. *Vision Research*, 42(17):2051–2062, 2002.

Perez, D., Maza, I., Caballero, F., Scarlatti, D., Casado, E., and Ollero, A. A ground control station for a multi-uav surveillance system. *Journal of Intelligent and Robotic Systems*, 69(1-4):119–130, 2013.

Perkins, C. E. *Ad Hoc Networking*. Addison-Wesley Professional, 1 edition, 2008.

Pfeil, K., Koh, S. L., and LaViola, J. Exploring 3d gesture metaphors for interaction with unmanned aerial vehicles. In *Proc. of the International Conference on Intelligent User Interfaces (IUI)*, pages 257–266, 2013.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G. A., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

Platt, J. C. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, USA, 1998.

Plaza, A. and Plaza, J. Automatic selection of informative samples for svm-based classification of hyperspectral data using limited training sets. In *Proc. of Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2010.

Podevijn, G. Gesture control for swarms of robots. Master's thesis, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, Belgium, 2012.

Podevijn, G., O'Grady, R., and Dorigo, M. Self-organised feedback in human swarm interaction. In *Proc. of Workshop on IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2012.

Podevijn, G., O'Grady, R., Nashed, Y. G., and Dorigo, M. Gesturing at subswarms: Towards direct human control of robot swarms. In *Towards Autonomous Robotic Systems*, pages 390–403. Springer, 2013.

Polikar, R. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

Polikar, R. Bootstrap - inspired techniques in computation intelligence. *IEEE Signal Processing Magazine*, 24(4):59–72, Jul. 2007.

Polikar, R., Upda, L., Upda, S. S., and Honavar, V. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systerms, Man, and Cybernetics—Part C: Applications and Reviews*, 31(4):497–508, 2001.

Pourmehr, S., Monajjemi, V. M., Vaughan, R. T., and Mori, G. You two! take off!: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–142, 2013a.

Pourmehr, S., Monajjemi, V. M., Wawerla, J., Vaughan, R. T., and Mori, G. A robust integrated system for selecting and commanding multiple mobile robots. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2874–2879, 2013b.

Pourmehr, S., Monajjemi, V. M., Sadat, S. A., Zhan, F., Wawerla, J., Mori, G., and Vaughan, R. T. "You are green": A touch-to-name interaction in an integrated multi-modal multi-robot HRI system. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Video Session)*, pages 266–267, 2014.

Pourmehr, S., Wawerla, J., Vaughan, R. T., and Mori, G. On the scalability of spatially embedded human multi-robot interfaces. In *Proc. of Workshop on ACM/IEEE International Conference on Human-Robot Interaction (HRI): Human-Robot Teaming*, 2015.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. Distributed learning in wireless sensor networks. *Computing Research Repository (CoRR)*, 2005. URL http://arxiv.org/abs/cs/0503072.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, 2006a.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. Consistency in models for distributed learning under communication constraints. *IEEE Transactions on Information Theory*, 52(1):52–63, 2006b.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. Distributed kernel regression: An algorithm for training collaboratively. In *Proc. of IEEE Information Theory Workshop*, pages 332–336, 2006c.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. A collaborative training algorithm for distributed learning. *IEEE Transactions on Information Theory*, 55(4):1856–1871, 2009.

Purohit, A., Zhang, P., Sadler, B. M., and Carpin, S. Deployment of swarms of micro-aerial vehicles: From theory to practice. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5408–5413, 2014.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Ros: an open-source robot operating system. In *Proc. of Open Source Software Workshop of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

Quinlan, J. R. Bagging, boosting, and C4.5. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 725–730, 1996.

Raducanu, B. and Vitria, J. Online learning for human-robot interaction. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007.

Remagnino, P., Shihab, A. I., and Jones, G. A. Distributed intelligence for multi-camera visual surveillance. *Pattern recognition*, 37(4):675–689, 2004.

Rinner, B. and Wolf, W. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10):1565–1575, 2008.

Rouanet, P., Danieau, F., and Oudeyer, P.-Y. A robotic game to evaluate interfaces used to show and teach visual objects to a robot in real world condition. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 313–320, 2011.

Rouanet, P., Oudeyer, P.-Y., Danieau, F., and Filliat, D. The impact of human–robot interfaces on the learning of visual objects. *IEEE Transactions on Robotics*, 29 (2):525–541, 2013.

Royer, E. M. and Toh, C.-K. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.

Rule, A. and Forlizzi, J. Designing interfaces for multi-user, multi-robot systems. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 97–104, 2012.

Sadat, S. A., Chutskoff, K., Jungic, D., Wawerla, J., and Vaughan, R. T. Feature-rich path planning for robust navigation of mavs with mono-slam. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3870–3875, 2014.

Sahin, E. Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, pages 10–20. Springer, 2005.

Saïdi, F. and Pradel, G. Contribution to human multi-robot system interaction application to a multi-robot mission editor. *Journal of Intelligent Robotic Systems*, 45(4):343–368, 2006.

Sankaranarayanan, A. C., Veeraraghavan, A., and Chellappa, R. Object detection, tracking and recognition for multiple smart cameras. *Proceedings of the IEEE*, 96(10):1606–1624, 2008.

Saska, M., Chudoba, J., Precil, L., Thomas, J., Loianno, G., Tresnak, A., Vonasek, V., and Kumar, V. Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. In *Proc. of IEEE International Conference on Unmanned Aircraft Systems*, pages 584–595, 2014.

Sato, E., Yamaguchi, T., and Harashima, F. Natural interface using pointing behavior for human–robot gestural interaction. *IEEE Transactions on Industrial Electronics*, 54(2):1105–1112, 2007.

Sauppé, A. and Mutlu, B. Robot deictics: How gesture and context shape referential communication. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 342–349, 2014.

Savaris, A. and Wangenheim, A. Comparative evaluation of static gesture recognition techniques based on nearest neighbor, neural networks and support vector machines. *Journal of the Brazilian Computer Society*, 16(2):147–162, 2010.

Schohn, G. and Cohn, D. Less is more: Active learning with support vector machines. In *Proc. of International Conference on Machine Learning (ICML)*, pages 839–846, 2000.

Schriebl, W., Winkler, T., Starzacher, A., and Rinner, B. A pervasive smart camera network architecture applied for multi-camera object classification. In *Proc. of ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–8, 2009.

Schwager, M., Julian, B. J., Angermann, M., and Rus, D. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.

Shan, Y., Han, F., Sawhney, H. S., and Kumar, R. Learning exemplar-based categorization for the detection of multi-view multi-pose objects. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1431–1438, 2006.

Shen, W.-M. Efficient incremental induction of decision lists - can incremental learning outperform non-incremental learning? Technical Report USC-ISI-96-012, University of Southern California, USA, 1996.

Sklar, E., Parsons, S., Ozgelen, A. T., Schneider, E., Costantino, M., and Epstein, S. L. HRTeam: a framework to support research on human/multi-robot interaction. In *Proc. of International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1409–1410, 2013.

Skubic, M., Perzanowski, D., Blisard, S., Schultz, A. C., Adams, W., Bugajska, M., and Brock, D. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, 34(2):154–167, 2004.

Skubic, M., Anderson, D., Blisard, S., Perzanowski, D., and Schultz, A. C. Using a hand-drawn sketch to control a team of robots. *Autonomous Robots*, 22(4): 399–410, 2007.

Song, B., Ding, C., Kamal, A. T., Farrell, J. A., and Roy-Chowdhury, A. K. Distributed camera networks. *IEEE Signal Processing Magazine*, 28(3):20–31, 2011.

Soro, S. and Heinzelman, W. A survey of visual sensor networks. *Advances in Multimedia*, pages 1–21, 2009.

Soto, C., Song, B., and Roy-Chowdhury, A. K. Distributed multi-target tracking in a self-configuring camera network. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1493, 2009.

Stefan, A., Athitsos, V., Alon, J., and Sclaroff, S. Translation and scale invariant gesture recognition in complex scenes. In *Proc. of ACM International Conference on Pervasive Technologies Related to Assistive Environments*, pages 1–8, 2008.

Steil, J. J. and Wersing, H. Recent trends in online learning for cognitive robotics. In *Proc. of the European Symposium on Artificial Neural Networks*, pages 77–87, 2006.

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A. C., and Goodrich, M. A. Common metrics for human-robot interaction. In *Proc. of ACM SIGCHI/SIGART Conference on Human-robot Interaction (HRI)*, pages 33–40, 2006.

Stern, H., Wachs, J. P., and Edan, Y. Optimal hand gesture vocabulary design using psycho-physiological and technical factors. In *International Conference on Automatic Face and Gesture Recognition*, pages 257–262, 2006.

Stern, H., Wachs, J. P., and Edan, Y. Optimal consensus intuitive hand gesture vocabulary design. In *IEEE International Conference on Semantic Computing*, pages 96–103, 2008a.

Stern, H., Wachs, J. P., and Edan, Y. Designing hand gesture vocabularies for natural interaction by combining psycho-physiological and recognition factors. *International Journal of Semantic Computing*, 2(1):137–160, 2008b.

Stern, H., Wachs, J. P., and Edan, Y. Gesture-based human-computer interaction and simulation. In Dias, M. S., Gibet, S., Wanderley, M. M., and Bastos, R., editors, *A Method for Selection of Optimal Hand Gesture Vocabularies*, pages 57–68. Springer, 2009.

Stiefelhagen, R., Fugen, C., Gieselmann, R., Holzapfel, H., Nickel, K., and Waibel, A. Natural human-robot interaction using speech, head pose and gestures. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2422–2427, 2004.

Stoica, A., Theodoridis, T., H., Huosheng, McDonald-Maier, K., and Barrero, D. F. Towards human-friendly efficient control of multi-robot teams. In *Proc. of International Conference on Collaboration Technologies and Systems*, pages 226–231, 2013.

Stoica, A., Salvioli, F., and Flowers, C. Remote control of quadrotor teams using hand gestures. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Late Breaking Report)*, pages 296–297, 2014.

Stroupe, A. W., Martin, M. C., and Balch, T. Distributed sensor fusion for object position estimation by multi-robot systems. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1092–1098, 2001.

Sun, A., Lim, E.-P., Benatallah, B., and Hassan, M. Fisa: feature-based instance selection for imbalanced text classification. In *Advances in Knowledge Discovery and Data Mining*, pages 250–254. Springer, 2006.

Sun, M., Su, H., Savarese, S., and Fei-Fei, L. A multi-view probabilistic model for 3d object classes. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1247–1254, 2009.

Sun, S. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

Sycara, K. and Lewis, M. Human control strategies for multi-robot teams. In *Proc. of WSEAS International Conference on Computers*. World Scientific and Engineering Academy and Society, 2012.

Tabar, A. M., Keshavarz, A., and Aghajan, H. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proc. of ACM International Workshop on Video Surveillance and Sensor Networks*, pages 145–154, 2006.

Tan, Y. and Zheng, Z.-Y. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.

Tangamchit, P., Dolan, J. M., and Khosla, P. Crucial factors affecting cooperative multirobot learning. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2023–2028, 2003.

Teixeira, T., Dublon, G., and Savvides, A. A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, 5:1–77, 2010.

Thomas, A., Ferrar, V., Leibe, B., Tuytelaars, T., Schiel, B., and Van Gool, L. Towards multi-view object class detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1589–1596, 2006.

Timofte, R., Zimmermann, K., and Van Gool, L. Multi-view traffic sign detection, recognition, and 3d localisation. In *Proc. of Workshop on Applications of Computer Vision*, pages 1–8, 2009.

Topkis, D. M. Performance analysis of information dissemination by flooding. *IEEE Journal on Selected Areas in Communications*, 7(3):335–340, 1989.

Torralba, A., Murphy, K. P., and Freeman, W. T. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 762–769, 2004.

Torralba, A., Murphy, K. P., and Freeman, W. T. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(5):854–869, 2007.

Trigo, T. R. and Pellegrino, S. R. M. An analisys of features for hand-gesture classification. In *Proc. of International Conference on Systems, Signals and Image Processing*, pages 412–415, 2010.

Tron, R. and Vidal, R. Distributed computer vision algorithms. *IEEE Signal Processing Magazine*, 28(3):32–45, 2011.

Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2-3):153–167, 2002.

van der Werff, H. M. A. and van der Meer, F. D. Shape-based classification of spectrally identical objects. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):251–258, 2008.

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

Vapnik, V. N. and Vapnik, V. *Statistical Learning Theory*, volume 1. Wiley New York, 1998.

Vasile, C., Pavel, A., and Buiu, C. Integrating human swarm interaction in a distributed robotic control system. In *Proc. of IEEE Conference on Automation Science and Engineering*, pages 743–748, 2011.

Velagapudi, P., Scerri, P., Sycara, K., Wang, H., Lewis, M., and Wang, J. Scaling effects in multi-robot control. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2121–2126, 2008.

Vijayakumar, S., D'souza, A., Shibata, T., Conradt, J., and Schaal, S. Statistical learning for humanoid robots. *Autonomous Robots*, 12(1):55–69, 2002.

Vijayakumar, S., D'souza, A., and Schaal, S. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.

Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

Viola, P. and Jones, M. J. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

Vovk, V. Competitive on-line statistics. *International Statistical Review*, 69(2): 213–248, 2001.

Vovk, V. and Zhdanov, F. Prediction with expert advice for the Brier game. *Journal of Machine Learning Research (JMLR)*, 10:2445–2471, 2009.

Wachs, J. P., Stern, H., and Edan, Y. A holistic framework for hand gestures design. In *Proc. of 2nd Annual Visual and Iconic Language Conference*, pages 24–34, 2008.

Wachs, J. P., Kölsch, M., Stern, H., and Edan, Y. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

Wahlster, W. and Kobsa, A. Dialogue-based user models. *Proceedings of the IEEE*, 74(7):948–960, 1986.

Wang, G., Tao, L., Di, H., Ye, X., and Shi, Y. A scalable distributed architecture for intelligent vision system. *IEEE Transactions on Industrial Informatics*, 8(1): 91–99, 2012a.

Wang, J., Neskovic, P., and Cooper, L. N. Selecting data for fast support vector machines training. In *Trends in Neural Computation*, pages 61–84. Springer, 2007.

Wang, J., Zhao, P., and Hoi, S. C. H. Exact soft confidence-weighted learning. In *Proc. of International Conference on Machine Learning (ICML)*, pages 121–128, 2012b.

Wang, R. Y. and Popović, J. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3), 2009.

Wang, S., Jin, R., and Valizadegan, H. A potential-based framework for online multi-class learning with partial feedback. *Journal of Machine Learning Research (JMLR)*, 9:900–907, 2010.

Wang, X., Clady, X., and Granata, C. A human detection system for proxemics interaction. In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 285–286, 2011.

Weizenbaum, J. ELIZA–a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9 (1):36–45, 1966.

Westell, J. and Saeedi, P. 3D object recognition via multi-view inspection in unknown environments. In *Proc. of International Conference on Control Automation Robotics Vision*, pages 2088–2095, 2010.

Wieselthier, J. E., Nguyen, G. D., and Ephremides, A. Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting. *Computer Networks*, 39(2):113–131, 2002.

Winfield, A. F. T. Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In *Distributed Autonomous Robotic Systems*, pages 273–282. Springer Japan, 2000.

Wolf, M. T., Assad, C., Vernacchia, M. T., Fromm, J., and Jethani, H. L. Gesture-based robot control with variable autonomy from the JPL biosleeve. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1160–1165, 2013.

Wu, C. and Aghajan, H. Collaborative gesture analysis in multi-camera networks. In *Proc. of ACM SenSys Workshop on Distributed Smart Cameras*, 2006.

Wu, C. and Aghajan, H. Real-time human pose estimation: A case study in algorithm design for smart camera networks. *Proceedings of the IEEE*, 96(10): 1715–1732, 2008.

Xavier, J. and Nunes, U. Interfacing with multiple robots using environmental awareness from a multi-modal hri. In *Proc. of Conference on Mobile Robots and Competitions*, 2007.

Xiaohui, H. and Eberhart, R. Human vs. swarm: An NK landscape game. In *IEEE Swarm Intelligence Symposium*, pages 1–5, 2008.

Xu, C., Tao, D., and Xu, C. A survey on multi-view learning. *Computing Research Repository (CoRR)*, 2013. URL http://arxiv.org/abs/1304.5634.

Yang, P., Freeman, R. A., and Lynch, K. M. Distributed cooperative active sensing using consensus filters. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 405–410, 2007.

Yin, X. and Xie, M. Finger identification and hand posture recognition for human-robot interaction. *Image and Vision Computing*, 25(8):1291–1300, 2007.

Yin, X. and Zhu, X. Hand posture recognition in gesture-based human-robot interaction. In *Proc. of IEEE Conference on Industrial Electronics and Applications*, pages 1–6, 2006.

Yu, C.-H. and Nagpal, R. Self-adapting modular robotics: A generalized distributed consensus framework. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1881–1888, 2009.

Zechner, M. and Granitzer, M. A competitive learning approach to instance selection for support vector machines. In *Proc. of International Conference on Knowledge Science, Engineering and Management*, pages 146–157, 2009.

Zhang, D. and Lu, G. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.

Zhao, X., Evans, N., and Dugelay, J. Multi-view semi-supervised discriminant analysis: A new approach to audio-visual person recognition. In *Proc. of European Signal Processing Conference*, pages 31–35, 2012.

Zheng, S., Xie, B., Huang, K., and Tao, D. Multi-view pedestrian recognition using shared dictionary learning with group sparsity. In Lu, B.-L., Zhang, L., and Kwok, J., editors, *Neural Information Processing*, volume 7064 of *Lecture Notes in Computer Science (LNCS)*, pages 629–638. Springer, 2011.

Zhu, Y., Yang, Z., and Yuan, B. Vision based hand gesture recognition. In *Proc. of International Conference on Service Sciences*, pages 260–265, 2013.