Gesturing at Subswarms: Towards Direct Human Control of Robot Swarms

Gaëtan Podevijn[1](✉), Rehan O'Grady[1], Youssef S.G. Nashed[2], and Marco Dorigo[1]

[1] IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
`{gpodevij,rogrady,mdorigo}@ulb.ac.be`
[2] Department of Information Engineering, University of Parma, Parma, Italy
`nashed@ce.unipr.it`

Abstract. The term *human-swarm interaction* (HSI) refers to the interaction between a human operator and a swarm of robots. In this paper, we investigate HSI in the context of a resource allocation and guidance scenario. We present a framework that enables direct communication between human beings and real robot swarms, without relying on a secondary display. We provide the user with a gesture-based interface that allows him to issue commands to the robots. In addition, we develop algorithms that allow robots receiving the commands to display appropriate feedback to the user. We evaluate our framework both in simulation and with real-world experiments. We conduct a summative usability study based on experiments in which participants must guide multiple subswarms to different task locations.

1 Introduction

To date in the field of human-robot interaction, a great deal of effort has been devoted to the study of the interaction between human beings and single agents but little effort has been dedicated to human-swarm interaction (HSI) — the interaction between human beings and robot swarms.

Swarm robotics systems are made up of a large number of relatively simple and cheap robots that carry out complex tasks by interacting and cooperating with each other. The distributed nature of such systems makes them robust (the loss of an agent does not change the collective behavior), scalable (the same control algorithms work with different swarm sizes) and flexible (the system adapts to different types of environments). These characteristics make swarm robotics systems potentially well suited for deployment in dynamic and a priori unknown environments.

However, the large number of robots and the distributed nature of swarm robotics systems also make them much harder to interact with. As the number of robots increases, it becomes increasingly impractical for a human operator to give instructions to or receive feedback from individual robots. Nor is it necessarily easy to broadcast commands to the entire swarm. The distributed control used in

A. Natraj et al. (Eds.): TAROS 2013, LNAI 8069, pp. 390–403, 2014.
DOI: 10.1007/978-3-662-43645-5_41,

swarm robotics systems implies that each robot has a different frame of reference and is therefore liable to interpret the broadcast command differently.

In HSI literature, the interaction systems developed usually rely on a secondary display that provides a human operator with a real-time representation of both the environment and the robot swarms. In such approaches, therefore, the human operator does not interact with the real robots in their real environment, but with a modelled representation of both the robots and the environment. In order to create a modelling layer, it is necessary to collect telemetry data about the robots (i.e., their position and orientation) and data about the environment (i.e., size and obstacles). And to be useful for HSI purposes, such data must be collected and modelled in real-time. Simulated HSI approaches have used the omniscience afforded by robotic simulators to collect all of the relevant data. However, in the real-world, external tracking infrastructure would be required (e.g., GPS or external cameras). Such tracking infrastructure is often infeasible in the dynamic, a priori unknown environments for which swarm robotics systems are best suited.

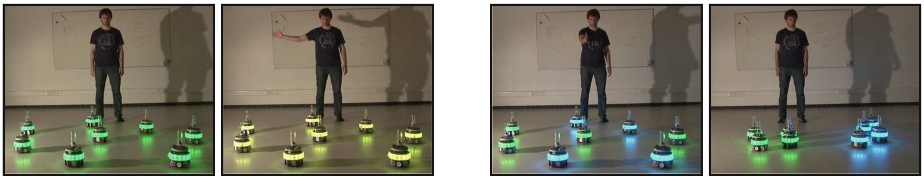
In this paper, we present an approach to HSI that does not involve any modelling layer, and instead allows a human operator to interact directly with real-robots. We design, implement and validate our approach in the context of a resource allocation and guidance scenario. Our scenario involves a human operator selecting particular groups of robots from the swarm (henceforth referred to as *subswarms*) and then guiding them to specific locations in their environment. The key philosophy underlying our approach is that a human operator should be able to interact with a subswarm as if it were a single entity, issuing a single command to and receiving coherent feedback from the subswarm "entity". The challenge is to enable group-level responses in robot swarms that have fully distributed control.

We present a gesture-based interface that allows the operator to interact with a swarm. With a gesture-based interface, the operator can devote his full attention to the robots. In contrast, with a secondary display, the operator must divide his attention between both the robots and the display. In Fig. 1(a–c), we show a human operator using our gesture-based interface in order to interact with a swarm of 8 real robots. The robotic platform used in this study is the wheeled *foot-bot* robot (see Fig. 1(d)) [7]. We conduct experiments using both simulated and real foot-bots. The simulations of the foot-bots are provided by the swarm robotics simulator ARGoS [8].

We demonstrate our approach with proof-of-concept experiments on real robots. We also perform an analysis of the usefulness of our gesture-based interface through simulation based experiments with human operators. To evaluate the experiments, we use a summative approach that allows us to quantify the overall usability of the system.

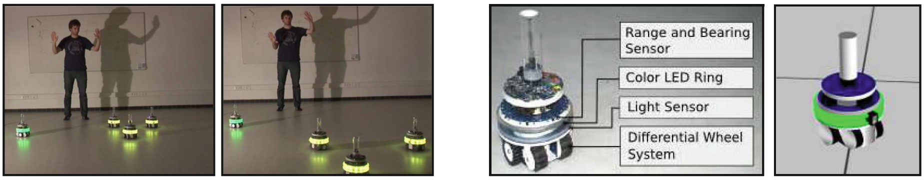
1.1 Related Work

The most common approaches used in the current studies of HSI tend to rely on an intermediate modelling layer. In these studies, an abstract representation



(a) *Left:* The operator faces a group of 8 robots. *Right* The operator gestures initiate the interaction with the group (the *select* command).

(b) *Left:* The operator splits the group into two subgroups (the *split* command). *Right:* The operator now has two groups he can interact with independently.



(c) *Left:* The operator holds a virtual steering wheel in the air to guide the robots. *Right:* To make the robots go right, the operator turns his virtual steering wheel to the right (the *steer* command).

(d) *Left:* Real foot-bot used in our real-world experiments. *Right:* Simulated foot-bot in ARGoS.

Fig. 1. Gesture-based interaction with a swarm of robots.

of both the robots and the environment is shown to the operator in a graphical user interface (GUI). McLurkin *et al.* [1] propose a centralized GUI based on real-time strategy video games where the user controls an army of hundreds of individuals. In addition to displaying modelled robots in a modelled environment, their GUI provides the user with extra debugging information (e.g., waypoints for individual robots, global positioning). The authors note that it can be difficult to display such a large amount of data while ensuring that the user still has a clear understanding of what is going on in the swarm. Kolling *et al.* [2] propose an approach based on so-called *selection* and *beacon* control. To select robots, the user draws a rectangular zone in a GUI. Robots inside this zone are considered selected. Once a subset of robots is selected, the user can send the selected robots different commands. With beacon control, the user can exert an indirect influence on the robots by adding virtual beacons in the GUI. Bashyal *et al.* [3] propose a GUI in which the operator takes the control of a single robot (an *avatar*) in the swarm. Because the avatar is perceived by the other robots as just another robot of the swam, the operator has the same limited influence as any single robot in the swarm. Bruemmer *et al.* [4] present a hierarchical communication architecture. They developed a GUI that allows the operator to send orders to a specific robot called "the sergeant".

Instead of GUIs, some modelling-layer based approaches propose an augmented reality view embedded in a dedicated head-mounted display. In Daily *et al.* [5], users wear an optical see-through head-worn device which receives simple robots' messages. When the device receives these messages, it analyzes them and augments the environment with a visual representation of these messages. A similar system is used in [6] where firefighters are helped in their mission by a robot swarm. The firefighters' helmets are augmented by a visual device, giving them direction information.

To the best of our knowledge, there is only one existing HSI approach that does not rely on an extra modelling layer: Giusti *et al.*'s [9] hand gesture-based interaction system. The goal of their work is to allow robots to decode hand gestures. Their method requires the robots to be placed in a particular spatial arrangement. In real-world situations, this requirement is not practical. Our research has a different focus. In our work, gestures are decoded into their corresponding commands by a central unit, which broadcasts the commands to the robots. The focus of our work is not on the gestures themselves, but on how single commands can be interpreted by decentralised robot swarms, and how decentralised swarms can provide composite feedback.

Usability studies are largely absent from the existing body of research in HSI (with the exception of [2,3]). We believe that this is a major omission, and that both objective and subjective usability results should be an essential part of any HSI research. In the research presented in this paper, we establish objective usability results via time-on-task based statistics and subjective usability results via a usability questionnaire.

2 Resource Allocation and Guidance Scenario

We base our work in this paper around a resource allocation and guidance scenario. In this scenario, a human operator moves selected robot groups of different sizes to specific locations in the environment. These different locations represent sites at which the robots would be required to carry out tasks. In this paper, robots do not actually carry out tasks – we represent task execution by pausing a group of robots for the amount of time that it takes them to "carry out" their hypothetical task. We do, however, let the human operator modify the size of the groups he selects and then guides, corresponding to the scale of different tasks the robot groups are required to "carry out".

This scenario allows us to exemplify, in the context of human-swarm interaction, the two major challenges inherent to any system that deals with bi-directional interaction. Firstly, an operator must be able to give commands to the system concerned. Secondly, the system must be able to provide the operator with appropriate feedback.

Giving commands to robots is challenging in HSI because each robot has its own reference frame. These different reference frames can lead to an operator's commands being interpreted differently by different robots. In the context of our guidance scenario, a command such as "turn left" would be meaningless as robots would interpret this command with respect to their individual reference frame (see Fig. 2). Meanwhile, understanding feedback provided by the robots in a robotic swarm is challenging because it is difficult to avoid overwhelming

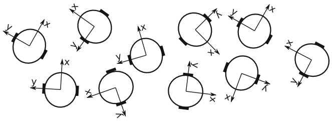


Fig. 2. Each robot in a swarm robotics system has its own local frame of reference. A command like *turn left* would not mean anything for the whole group since each robot would interpret the command differently.

the operator with a flood of data. If each robot provides individual feedback, the operator has too many data-points to process.

These issues could be solved if the operator were able to interact with groups of robots, rather than with individual robots. However, the distributed nature of control in swarm robotics systems makes such group-level interaction difficult to achieve. The challenge is to write distributed control code for a group of robots that lets each robot in the group interpret a single command meaningfully. In addition, this distributed control must provide the operator with group-level feedback, whereby the group of robots together provide a single data-point of feedback for the operator to process.

3 Our Approach: Interacting with Subswarms

We define a subswarm as a distinct group of robots within a swarm, that are identifiable both to a human operator, and to themselves. For a subswarm to be meaningful in the context of human-swarm interaction, a human operator must be able to visually distinguish a subswarm from other nearby robots. And robots in a subswarm must know that they belong to that particular subswarm, while robots outside the subswarm must know that they do not belong to that subswarm.

The first step towards implementing *subswarm-based interaction* is to define technically what the notion of subswarm means to both a human and a robot. In our subswarm-based interaction approach, a human perceives a subswarm as a set of robots that are close to each other and that are lit up in the same color. At the level of robot control code, subswarms are defined using *subswarm identifiers*. Every robot belonging to a subswarm has the same integer identifier and every subswarm identifier is unique.

We implement our subswarm-based interaction approach by first defining what commands and feedback make sense to a human operator in the context of our scenario. We then develop the distributed control code that allows groups of robots to process those commands and provide appropriate feedback.

3.1 Commands Available to Human Operator

In this section we present the list of commands that allow the operator to carry out our resource allocation and guidance scenario.

Steer The steer command is issued to guide the selected subswarm in the environment. When the selected subswarm receives the command, it starts moving straight. Subsequently, the operator can turn the subswarm left or right. In our gesture-based interface (see Sect. 4), the operator moves his hands just as he would turn the steering wheel of a car to change the subswarm's direction (see Fig. 1(c)).

Stop The stop command is issued to bring the selected subswarm to a halt.

Split The split command is issued to create new subswarms. When the selected subswarm receives the command, it splits into two independent subswarms of approximately the same size.

Merge The merge command is issued to reassemble two subswarms. When two selected subswarms receive the command, they move towards each other and unify into a single subswarm. The user can arrive at groups of required sizes by repeatedly splitting and merging subswarms.

Select The select command is issued by the operator in order to choose which subswarm to interact with. Once the select command is sent, one subswarm at random gets selected. All robots of the selected subswarm illuminate their yellow LEDs. If the selected subswarm is not the one the operator wants to interact with, he re-issues the command, and another subswarm becomes selected. He continues to issue the select command until the desired subswarm is highlighted. Note that before issuing the merge command, the operator must select two subswarms to merge. The gesture-based interface allows the operator to select a second subswarm. Robots belonging to the second selected subswarm illuminate their LEDs in red.

3.2 Distributed Robotic Control

In this section, we present the distributed behavioural control algorithms that implement the above commands. As is standard in swarm robotics systems, the same control code is run independently on each of the robots. The gesture-based interface broadcasts commands to the robots. Once the robots receive a command, they execute it only if they belong to the currently selected subswarm (see the Select algorithm below to know how a robot can determine if it belongs to the selected subswarm).

Our implementation assumes the existence of a fixed light source that defines a common point of reference in the environment. It also assumes the existence of a means for the robots to exchange short messages, calculate the distance and bearing between themselves and sense their own orientation. The short messages the robots have to exchange are their subswarm identifier. It is important for the subsequent algorithms that the robots all know in which subswarm they belong to. On the foot-bot platform, the range and bearing (R&B) module is used by the robots to communicate their subswarm ID and to know the position and bearing of their neighbors. The light sensor is used by a robot to measure its orientation with respect to a fixed light source.

For a subswarm to be clearly identifiable to a human operator, it is important that the robots of a subswarm remain close to each other and do not disperse.

To give subswarms this cohesive quality, we use a mechanism known as *flocking*. This flocking mechanism is implemented by having each robot use the distance and bearing information given by the R&B. The distance information allows each robot to adjust its position by placing itself at a constant distance from its neighbors. The bearing information allows each robot to adjust its orientation according to the average orientation of its neighbors. We based this flocking mechanism on [10], where robots are considered as particles that can exert virtual attractive and repulsive forces on one another. These forces are said virtual because the robots calculate them. At each time unit, each robot calculates a vector $\mathbf{f} = \mathbf{p} + \mathbf{h}$, which incorporates position information (encapsulated in vector $\mathbf{p}$) and orientation information (encapsulated in vector $\mathbf{h}$) of its neighbors. This vector $\mathbf{f}$ must then be converted into wheel actuation values.

In order for the robots to convert their vector $\mathbf{f}$ into wheel actuation values, each of them calculates its forward speed u and its angular speed ω. Robots set their forward speed to a constant value U, and their angular speed to a value proportional to the angle of vector $\mathbf{f}$ ($\theta = \angle \mathbf{f}$):

$$\omega = K\theta, \tag{1}$$

where K is a proportionality constant. Finally, robots convert their forward and angular speed into linear speed of their left (v_l) and right (v_r) wheel to $v_l = (u + \frac{\omega}{2}d)$ and to $v_r = (u - \frac{\omega}{2}d)$, where d is the distance between the wheels. The resulting behaviour of the robots is to place themselves at a constant distance of each other, to take the same orientation, then to move coherently in the same direction.

Steer. When the steer command is issued by the operator, robots of the current selected subswarm all compute vector $\mathbf{f}$. Then, they transform this vector into the relevant wheel actuation values. As a result, the robots start moving in the same (initially random) direction. When the operator decides to change the selected subswarm direction, each robot of the selected subswarm receives an angle of turn β from the gesture-based interface, corresponding to the angle at which the human operator has made his steer gesture. To turn β radians, each robot computes Eq. 1 by replacing the angle of vector $\mathbf{f}$, θ, by the angle of turn β.

Stop. When the stop command is issued, the robots of a selected subswarm stop moving by setting their linear wheel speeds to zero ($v_l = v_r = 0$).

Split. When the split command is issued to the robots of a selected subswarm, robots from this subswarm choose a new subswarm ID A or B with probability 0.5. Immediately after the robots chose their new subswarm ID, no robots from subswarm A and B have moved yet — they are still not spatially separated into two clearly distinct subswarms. To separate the newly distinct subswarms, we modify the cohesion behaviour of the constituent robots, so that robots with the same subswarm ID are attracted to each other, while robots with different subswarm IDs repel each other.

Merge. To reassemble two subswarms A and B, we assume that each robot r_a of subswarm A is able to compute its average distance to subswarm B. Every robot r_a calculates this average distance by averaging its distance (given by the R&B) to every robot r_b. Robots from subswarm B perform the same calculation with respect to the robots of subswarm A. Robots of the two subswarms then calculate the number t of time units necessary to travel half the average distance (assuming constant velocity) between the two subswarms. After moving for t time units in the direction of the subswarm they merge with, the two subswarms consider themselves joined. Robots from subswarms A and B then all adopt whichever of the two existing identifiers of A and B is smaller.

Select. In order for the robots to know if they belong to the subswarm that is currently selected by the operator, each robot of each subswarm maintains a variable `subswarm_selected` that contains either the selected subswarm ID or a sentinel value ϵ (if no subswarm is selected). Our distributed algorithms ensure that at any given moment, every robot across all the different subswarms has the `subswarm_selected` variable set to the same value. By comparing their own subswarm ID to the variable `subswarm_selected`, robots know if they belong to the selected subswarm.

Each time the select command is issued by the operator, every robot of every subswarm updates its `subswarm_selected` variable. To update the variable, every robot maintains a list of all subswarm IDs in the swarm. The update rules for this variable change based on context. There are three possible situations. In the first situation, the select command is issued while no subswarm is selected (`subswarm_selected` $= \epsilon$). In this case, the variable takes the lowest subswarm ID in the ID list. In the second situation, one subswarm is already selected. The selection must move to another subswarm. The variable is updated by taking the lowest ID in the list greater than `subswarm_selected`. In the third situation, every subswarm already has been selected once (there are no subswarm IDs greater than `subswarm_selected`). The variable is set to ϵ and no subswarm is selected anymore. Note that in case of a merge, the user must select two subswarms. The algorithm of the second selection command available to the user works as explained above with a minor modification: if the subswarm ID that is supposed to be selected (a second variable containing the second selected subswarm is maintained by the robots) is already selected with the first select command, then this ID is skipped and the next ID in the ID list is taken.

4 Gesture-Based Interface

Our interface allows a human operator to give commands to the robots by performing gestures. We use the Kinect system from Microsoft using the OpenNI library.[1] In our gesture-based interface, each command is associated with a specific gesture (see Fig. 3). When the interface recognizes[2] a gesture, it sends the corresponding command to every robot via a client-server mechanism.

---

[1] http://www.openni.org

[2] As it goes beyond the scope of this paper, we do not discuss the gesture recognition algorithm.

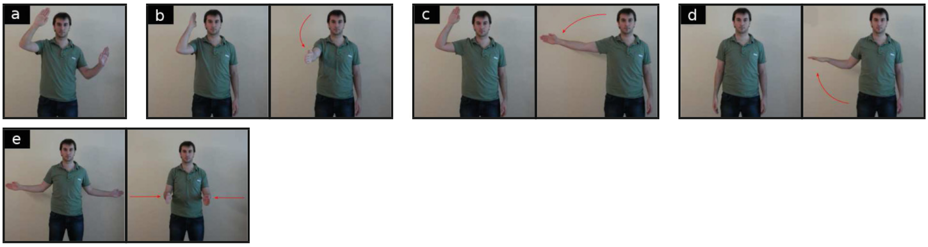


Fig. 3. Gestures associated with the commands (a) Steer (b) Split (c) Select (right arm) – The operator uses his left hand to select a second subswarm. (d) Stop (e) Merge

We discovered that the recognition was more accurate if we recorded gestures separately for each user, since different body shapes of the different users (e.g. short or tall) reduced the efficiency of our recognition algorithms. We recorded user gestures with a dedicated tool that we developed. Further work will focus on removing this constraint in order to recognize gestures with different types of body shapes without having the users to record the gestures [11].

5 Usability Study

The objective of our usability study is twofold. Firstly, we want to test if our participants can understand the concept of issuing commands to a robot swarm. Secondly, we want to study if they are able to carry out the test scenario with our gesture-based interface.

5.1 Experimental Test Scenario

We designed a specific instance of our resource allocation and guidance scenario, that would allow us to measure the performance of our interface. In this scenario, the participant has to use a swarm of 30 robots to carry out three tasks. The participant has to create three separate subswarms of robots by splitting the swarm, then guide each subswarm to one of the three task locations. Afterwards, the participant has to re-merge all of the robots back into a single swarm.

The simulated environment used in our scenario is depicted in Fig. 4. In Fig. 4 (Left), we show an initial swarm of 30 robots in the environment at the beginning of the experiment. In Fig. 4 (Right), the participant has split the swarm into three subswarms and has moved two of these subswarms to task locations.

Each participant had to perform the experiment with two interfaces: our gesture-based interface and a graphical user interface (GUI). We developed this simple button-based GUI as an alternative way for human operators to issue commands to the swarm. The GUI functionality is similar to that of our gesture-based interface, with one GUI button corresponding to each recognized gesture.

Fig. 4. (Left) A swarm waiting for orders. (Right) The initial swarm has been split into three subswarms. Two of them have been moved to task locations (the green/light gray areas) (Color figure online).

The one exception is the *steer* command. In the gesture-based interface, the operator can change the subswarm's direction while the subswarm is moving (the interface provides the operator with a steering wheel mechanism). In the GUI however, the operator cannot turn the subswarm while it is moving. The operator must first stop the subswarm moving, rotate it (right or left), then move the subswarm straight.[3]

5.2 Experimental Setup

We recruited 18 participants for this study: 6 of them were PhD students in robotics, 6 were Master students in Computer Science and 6 were non-technical people recruited from the entourage of the authors. Our participants were between 23 and 33 years old with an average age of 27.7 years ($SD = 2.9$). We started each experiment with a five minute presentation. The purpose of this presentation was to explain the resource allocation and guidance scenario and to present the simulated environment (see Fig. 4), the available commands and the two interfaces (the gesture-based interface and the GUI).

By having each participant perform experiments with both types of interfaces (the gesture-based interface and the GUI), we introduce the risk of a carryover effect. That is, the order in which the interfaces are tested by a participant might affect that participant's results. We prevent any possible carryover effect by alternating which interface the participants encountered first. We randomly divided our participants into two groups. The first group started the experiment with the gesture-based interface and finished with the GUI while the second group started with the GUI and finished with the gesture-based interface. Immediately before starting the experiment with the gesture-based interface, the participants underwent a brief preparation session in which they recorded the gestures and performed them several times in order to memorize them.

During the experiments with both the gesture-based interface and the GUI, the simulated environment was displayed on a large projection screen. By using a projection screen, we tried to approximate as closely as possible the experience

---

[3] Ideally, the GUI should also have had steering functionality. However, GUI design was not the purpose of this research, and for time constraints we kept the GUI as simple as possible and restricted to button-based functionality.

that a participant would have had with real robots — e.g. percentage of visual field occupied by the arena and by individual robots. For the experiment involving the gesture-based interface, our participants stood 1.5 m from the projection screen. For the experiments with the GUI, our participants were seated in front of a computer. The GUI was displayed on the computer monitor. The participants had, therefore, to look at both the projection screen and the computer monitor.

5.3 Usability Metrics

We evaluated the usability of both the gesture-based interface and the GUI with two kinds of metrics. The first metric is an objective metric: a time-on-task based statistic. We kept track of the amount of time taken by each participant to carry out their task. The counter was launched as soon as the first subswarm was selected and stopped exactly when the two last subswarms finished merging. The second metric is a subjective metric that allows us to measure the participants' evaluation of both the gesture-based interface and the GUI. After the experiment, each participant was asked to fill out two System Usability Scale questionnaires (SUS) [12] (one for each of the two interfaces). SUS is a reliable (i.e., consistent in its measure) and validated (i.e., it measures what it intends to measure) questionnaire with a scale of five options ranging from *strongly agree* to *strongly disagree*. The resulting score takes into account the efficiency (i.e., amount of effort needed to finish the task), effectiveness (i.e., ability to successfully complete the task) and satisfaction of participants. The score is a number that varies from 0 (low usability) to 100 (high usability) giving a global evaluation of the interface's usability.

5.4 Experimental Results

In Table 1 we present the time-on-task (ToT) results of both the gesture-based interface and the GUI. Results show that on average, our participants were slightly slower (+10.4 %) with the gesture-based interface. In Table 2 we present results regarding participants' subjective evaluation of the interfaces' usability. We can see that our participants evaluated the gesture-based interface with a mean score of 75.8 while they evaluated the GUI with a mean score of 78.5 (+3.4 %).

Results reveal that our participants managed to use both the gesture-based interface and the GUI effectively. Although our participants seemed to achieve marginally better results with the simpler GUI, we were quite satisfied by the overall usability of our gesture-based interface, especially given that all of our

Table 1. ToT statistics in minutes

$N = 18$	Mean	SD	C.I. (95 %)
Gesture-based	15.4	3.4	(13.7, 17.3)
GUI	13.8	3.3	(12.2, 15.5)

Table 2. SUS questionnaire results

$N = 18$	Mean	SD	C.I. (95 %)
Gesture-based	75.8	13.7	(69, 82.7)
GUI	78.5	12.4	(72.2, 84.7)

participants had prior experience in using GUI-based systems, and few had prior experience in gesture control systems.[4] We believe that the minor ToT superiority of the GUI is anyway counterbalanced by the numerous advantages of a gesture-based interface. With the emergence of wireless video cameras such as the Mobile Kinect Project[5], video camera deployment will get easier and easier. Furthermore, a single camera can be used to recognize the gestures of multiple users. On the other hand, a GUI requires that each user has his own device (e.g., personal computer or tablet). Moreover, with a gesture-based interface the operator can keep his attention wholly focused on the robots and their task, while with a GUI, the operator must concentrate on both the robots and the device displaying the GUI.

6 Real Robot Validation

We validated our approach with proof-of-concept experiments on real robots. We conducted experiments with groups of up to 8 real foot-bots. We do not report quantitative data as we did not run any usability experiments on the real robots. However, with only minor modifications to our distributed algorithms (e.g., parameters in the flocking algorithm), robots were able to receive and perform all the commands issued by the operator. In Fig. 1, we show an operator selecting a subswarm, sending the *split* command to the selected subswarm and then guiding a selected subswarm with the *steer* command.

7 Conclusions and Future Work

In this paper, we presented a gesture-based human-swarm interaction framework. We designed a resource allocation and guidance scenario in which a human operator is asked to move different robot subswarms to different locations in a physically detailed simulation environment with the help of five commands (steer, split, merge, stop and select). Instead of interacting with a representation of the robots, the operator interacts directly with the robots. We conducted a summative usability study and we collected both objective and subjective results. The results show that our participants (i) successfully managed to interact with a swarm of robots and (ii) were satisfied with using the gesture-based interface to carry out our scenario. Finally, we ran experiments on real robots in order to validate the technical feasibility of our approach in the real world.

---

[4] We did not succeed in establishing statistically significant difference between the gesture-based interface and the GUI ($p_{ToT} = 0.07$, $p_{SUS} = 0.55$). However, as the goal of this paper was not to design a GUI, such precise comparison would be fairly meaningless.

[5] http://big.cs.bris.ac.uk/projects/mobile-kinect

Future research will focus on improving feedback provided by the subswarms. Our goal is to leverage the same self-organised mechanisms that govern the robots' behaviour to generate the feedback to send to a human operator [13].

Acknowledgements. This work was partially supported by the European Research Council through the ERC Advanced Grant "E-SWARM: Engineering Swarm Intelligence Systems" (contract 246939). Rehan O'Grady and Marco Dorigo acknowledge support from the Belgian F.R.S.-FNRS.

References

1. McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., Schmidt, B.: Speaking swarmish: human-robot interface design for large swarms of autonomous mobile robots. In: Proceedings of the AAAI Spring Symposium, pp. 72–75. AAAI Press, Menlo Park (2006)
2. Kolling, A., Nunnally, S., Lewis, L.: Towards human control of robot swarms. In: Proceedings of the 7th Annual International Conference on H, pp. 89–96. ACM, New York (2012)
3. Bashyal, S., Venayagamoorthy, G.K.: Human swarm interaction for radiation source search and localization. In: Proceedings of Swarm Intelligence Symposium, pp. 1–8. IEEE Press (2008)
4. Bruemmer, D.J., Dudenhoeffer, D.D., Marble, J.L.: Mixed-initiative remote characterization using a distributed team of small robots. In: AAAI Mobile Robot Workshop. AAAI Press, Menlo Park (2001)
5. Daily, M., Cho, Y., Martin, K., Payton, D.: World embedded interfaces for human-robot interaction. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, Big Island, pp. 125–130. IEEE Computer Society (2003)
6. Naghsh, A., Gancet, J., Tanoto, A., Roast, C.: Analysis and design of human-robot swarm interaction in firefighting. In: Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, pp. 255–260. IEEE Press (2008)
7. 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., Decugnière, A., Di Caro, G., Ducatelle, F., Ferrante, E., Förster, A., Martinez Gonzalez, J., Guzzi, J., Longchamp, V., Magnenat, S., Mathews, N., Montes de Oca, M., O'Grady, R., Pinciroli, C., Pini, G., Rétornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stützle, T., Trianni, V., Tuci, E., Turgut, A.E., Vaussard, F.: Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. IEEE Rob. Autom. Mag. **20**(4), 60–71 (2013)
8. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Gambardella, L.M., Birattari, M., Dorigo, M.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intell. **6**(4), 271–295 (2012)
9. Giusti, A., Nagi, J., Gambardella, L., Bonardi, S., Di Caro, G.A.: Human-swarm interaction through distributed cooperative gesture recognition. In: Proceedings of the 7th International Conference on HRI, pp. 401–402. ACM, New York (2012)
10. Ferrante, E., Turgut, A.E., Huepe, C., Stranieri, A., Pinciroli, C., Dorigo, M.: Self-organized flocking with a mobile robot swarm: a novel motion control method. Adapt. Behav. **20**(6), 460–477 (2012)

11. Nashed, Y.S.G.: GPU hierarchical quilted self organizing maps for multimedia understanding. In: Proceedings of International Symposium on Multimedia, pp. 491–492. IEEE (2012)
12. Brooke, J.: SUS: a 'quick and dirty' usability scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, A.L. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, London (1996)
13. Podevijn, G., O'Grady, R., Dorigo, M.: Self-organised Feedback in human swarm interaction. In: Workshop on Robot Feedback in Human-Robot Interaction: How to Make a Robot Readable for a Human Interaction Partner (RO-MAN'12), France, Paris (2012)