A Robust Integrated System for Selecting and Commanding Multiple Mobile Robots

Shokoofeh Pourmehr, Mani Monajjemi, Jens Wawerla, Richard Vaughan and Greg Mori
School of Computing Science, Simon Fraser Univesity, Burnaby, BC, Canada
{spourmeh, mmonajje, jwawerla, vaughan, mori}@sfu.ca

***Abstract*— We describe a system whereby multiple humans and mobile robots interact robustly using a combination of sensing and signalling modalities. Extending our previous work on selecting an individual robot from a population by face-engagement, we show that reaching toward a robot - a specialization of pointing - can be used to designate a particular robot for subsequent one-on-one interaction. To achieve robust operation despite frequent sensing problems, the robots use three phases of human detection and tracking, and emit audio cues to solicit interaction and guide the behaviour of the human. A series of real-world trials demonstrates the practicality of our approach.**

I. Introduction

We have been working on methods for uninstrumented humans to give commands to individual and groups of robots using simple and natural interfaces. By "simple and natural" we mean that the humans interact with the robots in ways familiar from human-human or human-animal interactions, such as pointing gestures, gaze direction, and spoken commands. It has been argued that using these familiar interaction modes for HRI could mean that people require less training or have lower cognitive load compared to using a novel unique-to-robots interface [5]. Certainly a familiar approach has two distinct benefits for teams of multiple humans working with one or more robots. First, humans can interact with robots and human teammates in the same way, which means both that only one method must be learned, and that a single execution of a command could be received by a mixed team of robots and humans. Second, uninstrumented and untrained non-teammate human observers can potentially understand the HRI they are watching.

For example, we have previously shown that an individual robot can be selected from a population for one-on-one interaction by a user simply looking directly at that robot [2]. This also works for humans and many other animals; we are acutely sensitive to a steady gaze. In our artificial system, each robot carries a camera and uses a standard face detection algorithm to evaluate how well it can see the user's face. Our innovation was to use explicit wireless communication between robots to perform a distributed election algorithm to unambiguously decide which robot (if any) was being looked at directly, and was thus the subject of attention. Once elected, the single selected robot would watch the user for motion-based hand gestures that were interpreted as task-allocation commands. Non-selected robots would not attend to this command, and indeed would not waste resources looking for it. Since the election is completed in a few tens of milliseconds and is essentially imperceptible to the user, the user's experience is simply that as you look from robot to robot around you, the selected robot is always "the one I am looking at right now". Below, we show that this method can be generalized by replacing the face-engagement with a pointing-based gesture.

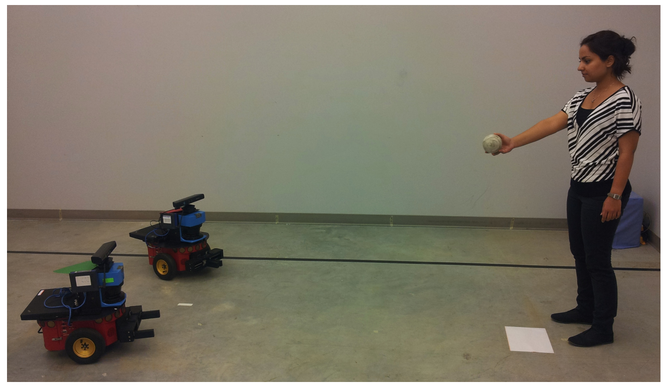


Fig. 1. An uninstrumented person selecting one robot out of a group by offering it a ball - a modified pointing gesture.

However, working with gesture-based interfaces is limited by the quality of sensing. While state of the art techniques provide excellent face tracking, human skeletal pose estimation, etc. in ideal conditions, we often have occlusion, motion-induced blur and false positives from background clutter. This means that building a robust HRI system is challenging. The system described below employs multiple phases of human-detection with timeouts, retries and fallback behaviour to contribute to robustness. However, even with a significant engineering effort, we find that the robot still encounters a tough sensing condition every few minutes, or roughly 10% of interactions. Our approach to this "last 10%" problem is to provide rich feedback to the user about the robot's state, so that the user can choose to make the problem easier; perhaps by adjusting their pose so the robot can see previously occluded limbs or joints. In practice we informally observe that this rich feedback makes interaction feel more responsive even when no problems occur, since no-touch interfaces have no built-in feedback as observed by Adams:

> The machine was rather difficult to operate. For years radios had been operated by means of

> pressing buttons and turning dials; then as the technology became more sophisticated [...] all you had to do was wave your hand in the general direction of the components and hope. [1]

By providing carefully-designed audio feedback about all its interaction state changes, our robots quickly reassure users that their waving is working.

The contributions of this work are (i) the first demonstration of using a pointing-based gesture combined with distributed election to guarantee that at most one robot is selected; (ii) the first demonstration of an uninstrumented human selecting a robot from a population while both robots and humans are moving freely around the workspace; and (iii) a case study of a complete and robust HRI system using several sensing modes, multi-phase robot behaviour and rich audio feedback to guide the user to resolve the "last 10%" of tricky sensing situations.

II. Background

Human-robot interaction (HRI) is an active area of research. Goodrich and Schultz [6] provide a survey of the field. The work by Steifelhagen et al. [17] is a classic example of an integrated system, which includes speech recognition and vision for colour-based hand and face tracking to estimate pointing direction. Recent work includes Wang et al. [19], which describes a fusion approach using scanning LIDAR data for leg detection combined with vision-based human detection. Droeschel et al. [4] also use LIDAR for leg and torso detection, and subsequent vision-based detection. Futher, they provide a study of pointing for human-robot interaction, defining a gaussian process regression model for estimating pointing direction from depth data. In contrast, we consider a multi-robot scenario using pointing for selection, and develop a verification approach for human localization that solicits human interaction to aid the robot in deciding if a candidate detection is valid or not.

A. Robot selection and task delegation

There is little work on human-robot interfaces for multi-robot systems. Examples can be broken up into two general cases:

1) Traditional human-computer interfaces: Rather than interacting directly with robots, a traditional human-computer interface is used to represent the spatial configuration of the robots and allow the user to remotely interact with the robots. Examples of a include McLurkin et al. [11] that uses a overhead-view of the swarm in a traditional point-and-click GUI named "SwarmCraft", and work by Kato that displays an overhead live video feed of the system on an interactive multi-touch computer table, which users can control the robots' paths by drawing a vector field over top of the world [7].

2) Embodied, world-embedded interactions: Embodied, world-embedded interactions occur directly between the human and robot, through mechanical or sensor-mediated interfaces. A useful property of this type of interaction is that since robots observe humans directly using their onboard sensing, they may not need to localize themselves in a shared coordinate frame in constrast to the GUI-based interfaces. Also, human users can walk and work among the robots, and are not tied to an operator station. Examples include work by Payton that uses an omnidirectional IR LED to broadcast messages to all robots, and a narrow, directional IR LED to select and command individual robots [15]. Naghsh et al. present a similar system designed for firefighters, but do not discuss selecting individual robots [14]. Zhao et al. propose the user interacts with the environment by leaving fiducial-based "notes" (for example, "vacuum the floor" or "mop the floor") for the robots at work site locations [21]. Xue et al. introduces a clever fiducial design for imperfect visibility conditions and combines this with user-centric gestures in an underwater scenario. [20].

Audio cues are also often used for human detection, including the recent work of Deleforge and Horaud [3] in which a "cocktail party robot" localizes a sound source with an active robot head with binaural sensors.

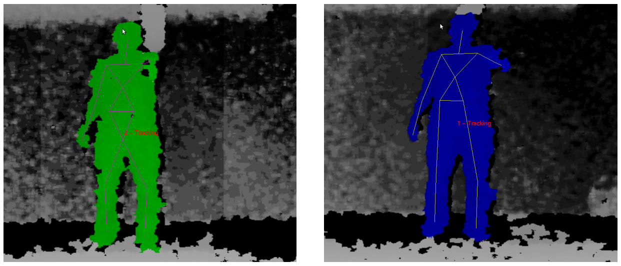
In our previous work, we developed face engagement [2] and circling-gesture [12] techniques for single- and multiple-robot selection. However, these systems had no strategy for human detection other than faces, and the vision system for interpreting circling gestures lacked robustness. In this paper, we provide a novel, robust integrated system that includes human detection strategies, pointing estimation from a depth sensor, and solicits interaction to guide the human's behaviour.

B. Gesture-based robot interaction

There is a vast computer vision literature on gesture recognition: Mitra and Acharya [13] provide a survey. Several gesture-based robot interfaces exist; we do not attempt to provide an exhaustive survey, but rather mention some interesting examples. Systems may use static gestures where the user holds a certain pose or configuration, or dynamic gestures where the user performs a combination of actions.

Waldheer at al. use both static and motion-based gestures to control a trash-collecting robot [18]. Loper et al. demonstrate a indoor/outdoor person-following robot that uses an active depth sensing camera to recognize static gestures [9]. Earlier work by Kortenkamp et al. presents a mobile robot that uses an active vision system to recognize static gestures by building a skeleton model of the human operator; a vector of the human's arm is used to direct the robot to a particular point [8]. Perzanowski et al. present a multimodal speech and gesture-based interface; an active vision system is used to interpret pointing gestures as directional vectors, and to measure distance between the user's two hands [16].

All gesture-based systems discussed so far are designed to work with a single robot, with exception [16]; however, there were no examples of gesture-based interfaces designed for multi-robot systems which rely solely on non-verbal communication. Our previous work [2] was the first to allow for this type of interaction. In this paper, we present a novel variant of this system: a user can select and command an individual robot using a pointing-like gesture.



(a) (b)

Fig. 2. View of a human performing a *reaching gesture* from the intended robot (left) and the unintended robot (right) in a setup similar to that shown in Fig. 1. The colour blob (blue and green) indictates that the user is successfully detected.

III. Method

For the work presented in this paper we use two Pioneer DX3 robots shown in Fig. 1. Both robots are equipped with the well-known Sick LMS200 scanning LIDAR and the popular Kinect[1] active RGB-D sensor. In addition each robot has a 2dof gripper mounted in front. The mobile robot base, laser and the gripper are controlled by the built-in computer running ROS[2]. The Kinect sensor is connected to a laptop mounted on top of the robot. This computer provides the computational power needed for skeleton tracking based on the Kinect data, using the ROS Kinect stack. Using the ROS networking facilities we control the robots from an off-board computer. Note this is done for convenience not for lack of onboard computational resources which are very modest compared to the skeleton extraction process.

The Kinect sensor is designed as a novel human interface for computer games. In normal use the sensor is stationary and human players move around nearby in front of a television. By mounting the sensor on a mobile platform we create two challenges, (i) the range and field of view are smaller than that of sensor traditionally used in robotics such as LIDAR and passive RGB cameras; and (ii) the sensor has difficulties acquiring a skeleton if the sensor itself is in motion. In the following we describe how we address both problems.

A. Coarse Human Detection

The robot's first task is to find a human for interaction. The Kinect sensor's field of view covers only a small part of our 8x10m arena, so we use a 2D laser range finder with a 180 degree X 8m field of view. We look for a sequence of rising and falling edges in the laser range data that corresponds to two human legs close to each other. While this simple method is fast and effective, it is subject to false positives since many objects in the world, such as furniture or a pair of trash cans may appear similar to a pair of legs. Once a candidate leg-pair is detected the robot narrows its laser field of view to the section in the scan where the legs were last detected. This filters out subsequent leg detections which the robot would otherwise have to reject to stay on target, and provides an almost cost-free method of focussing the robot's attention on a single candidate detection. Next the robot servos towards the detected legs to either confirm or reject the presence of a human.

[1] `http://www.xbox.com/en-US/kinect/`
[2] `http://www.ros.org/`

B. Fine Human Detection

Once the robot is close enough ($< 3m$) to the location of the detected legs to reliably use the Kinect sensor, the robot stops briefly. It now triggers the Kinect's built-in user detection algorithm based on the 3d measurements from the sensor. Fig. 2 shows a successfull user detection, marked by the colour blobs. If a match is found a human is successfully detected. The hypothesis of a human being present is rejected if no user is detected in the Kinect data, or the location of the legs does not match that of any detected user. The latter is important because the Kinect may also report false positives. In case of rejection the robot returns to the laser-based leg detection, turns away from the false positive detection, and wanders around the world.

C. Gesture Recognition

Gestures are recognized by interpreting the skeleton data from the Kinect sensor. After a human is successfully detected the robot triggers the Kinect's skeleton recognition algorithm. Depending on the pose of the human, skeleton matching can fail, e.g. if joints are occluded. In this case the robot plays a "sad" sound as a hint to the user to adjust her pose. If a skeleton cannot be detected after a threshold time the robot gives up and returns to the leg detection behaviour. A successful skeleton match triggers the playback of a "happy" sound. We found (informally) that this basic feedback greatly improves the usability of the system. It makes it easier for the user to assist the robot in situations that would otherwise be difficult for the robot to resolve by itself. For example changing the viewing angle usually does not resolve joint occlusions caused by an unnatural human pose or baggy clothing.

The robots are programmed to distinguish four simple gestures - no gesture, pointing left, pointing right and reaching gesture. Detecting no gesture indicates to the robot that the human is currently not interested in interacting; after some time watching the human with no gesture detected, the robot gives up and turns away to look for a different companion.

The reaching gesture indicates to the robot that the user wants the robot to either fetch or deliver an object, e.g. a ball. Whether the object is to be fetched or delivered is decided based on whether the robot currently holds the object or not. This can be directly measured by reading the touch sensors in the gripper paddles, since our system knows there is exactly one ball in its world. Detecting a reaching gesture requires obtaining the position of the user's head and hand from the Kinect skeleton data. An example skeleton reaching is shown in Fig. 2. A gesture is classified as reaching if a line drawn through the head and hand points intersects with a sphere centred at the origin of the Kinect sensor. This is shown in Fig. 3. The required precision of the gesture can be adjusted

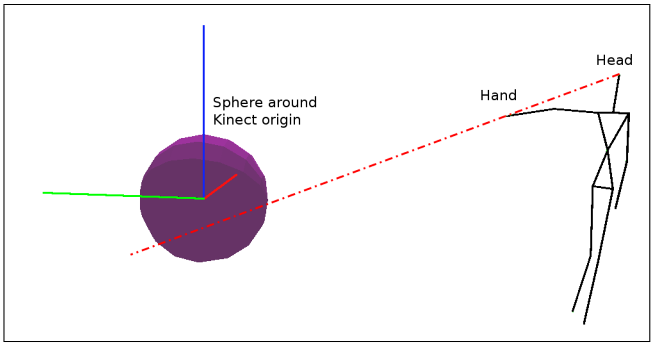


Fig. 3. A reaching gesture is an intersection of a line between head and hand joint of the skeleton with a sphere around the origin of the Kinect sensor.

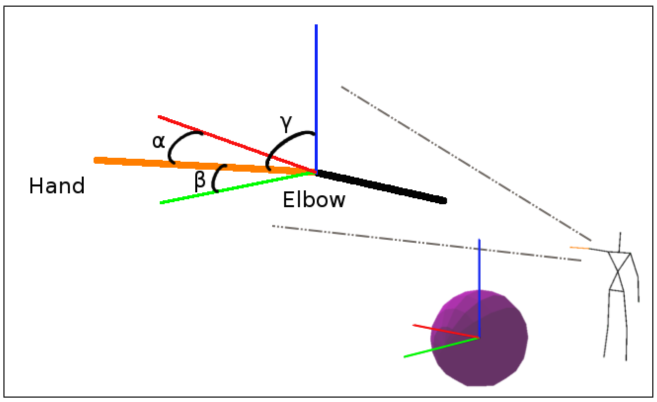


Fig. 4. A pointing gesture is recognized by analyzing the orientation of the line between hand and elbow joints.

via the radius of the sphere. This gesture works with either the left or right arm. Conceptually we consider reaching a modified pointing gesture - the only difference is the shape (and possibly content) of the hand.

By pointing either right or left the user can instruct the robot to turn in the respective direction. This allows a user uninterested in giving or receiving a ball to send the robot in the direction of someone who is. This gesture is detected by drawing a line through the points of the hand and elbow joints. If the orientation of this line in the Kinect frame of reference is within a given range the gesture is classified as pointing. The angle α between the line and the x-axis has to be $-40° < \alpha < 40°$ for the left arm and $-40° < |\alpha| - 180° < 40°$ for the right arm. At the same time the angle β to the y-axis and γ to the z-axis both have to be between $50°$ and $130°$. Fig. 4 illustrates the concept.

D. Sounds

The robots emit sounds to provide feedback to users at the moments described in the next section. All sounds for this demonstration are from the *Willow Garage Robot Sounds Library*[3]. We believe the sounds make an important contribution to system robustness by informing the user about the robots' internal state. However, we do not provide

[3]`http://hri.willowgarage.com/sounds/`

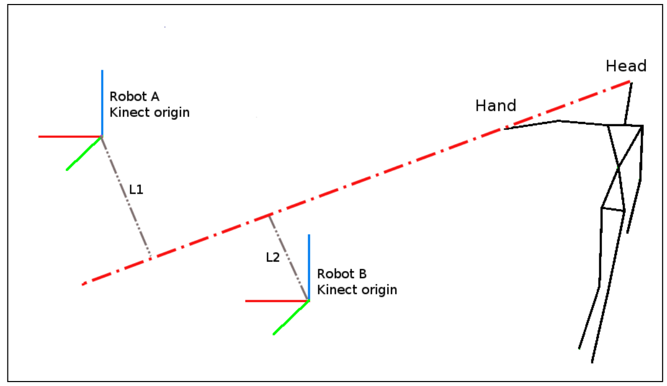


Fig. 5. Disambiguating the reaching gesture for multiple robots (details in the text).

evidence for this here: this is left for future work, along with the interesting topic of how to design effective sounds.

E. Multi-Robot System

The method described above works well (see experiments below) in a single robot setting with one or more humans present. It also works well if two robots approach a human from very different directions, that is the angle between the robot's trajectory is $45°$ or larger. If the robots approach more or less in parallel, as in Fig 1, the gesture classification algorithm has difficulties determining if the human intended to reach for robot A or B, see Fig. 2. The problem is that the reach-detection spheres can overlap in these situations and a reaching line (line between head and hand joint) can intersect both spheres and hence both robots will positively identify the reaching gesture.

We address this problem with an election based method developed earlier by our group [2]. In the original version robots compared the quality of their human face detections to determine which robot (if any) the user was looking at. Generalizing that idea, we seek to obtain a scalar value that varies from different robot view points, broadcast that value and elect the robot with the best score. To disambiguate the reaching gesture we use the length of a normal to the reaching line through the origin of the Kinect sensor (shown in Fig. 5). The robot with the smallest length value is the one the human intends to engage. Note this method does not require the robots to be localized or share a common reference frame, since each robot uses only the local appearance to score a gesture.

IV. Demonstration and Discussion

We performed two different robot navigation scenarios. In the first scenario, one robot and two human operators, one of whom starts holding a ball, are located in a 8x10m room clear of obstacles. Each robot: 1) first finds the users in the room, who are located at arbitrary locations; 2) attends to one user and approaches her, emitting a happy sound indicating readiness to interact; 3) receives a command, and executes it. The commands are either fetching or delivering the ball, or turning right or left to search for other users in the room.

After getting or giving the ball, robot starts searching for other users in the room. In our trials, the two users are instructed to execute the following interaction script:

1) $user_1$ sends the robot right by pointing to the right.
2) $user_2$ has the ball, and gives it to the robot by offering it with the reaching gesture.
3) $user_1$ issues no commands.
4) $user_2$ sends the robot left by pointing to the left.
5) $user_1$ requests the ball with the reaching gesture.

The users are instructed to attempt to recover from any failures, which happened on two occasions as described below.

An example trial where the script is executed perfectly is shown in Figure 6, based on data recorded from trial 1. The robot trajectory is recorded using an overhead vision system not used in the robot control loop.

The results of 10 trials are presented in Table I. The robot correctly detected users' legs on 48/50 opportunities (96% success). In trial 5 the robot did not immediately detect $user_1$'s legs, so it targeted $user_2$ and approached him. Inspecting the video of the trial we see that $user_1$ was standing with legs tight together, causing the leg detector to fail. $User_2$ pointed to the right to help the robot find $user_1$, and the system subsequently executed the script without errors. RGB-D user detection worked in all 50 cases and point right/left gestures were detected in all 10 cases. The success rate of reaching gestures (offering and requesting the ball) was 18/20 (90%). All of the failures in gesture detection were occurred when skeleton could not be detected within the threshold time of 15 seconds, despite encouraging the user with sounds, so the robot began searching for other users.

In the second scenario, two robots interact with one user who starts holding a ball. The robots:

1) first find the user in the room, who is located at arbitrary location. 2) approach the user, emitting a happy sound indicating readiness to interact. 3) wait to be selected by the user. Once selected, a robot drives forward to collect the ball.

When both robots arrive at the user and indicate their attention by sound, the user choses one robot by offering it the ball. An example trial (#1) is shown in Fig 7, showing the robot trajectories and user behaviour.

Ten trials, labeled 1 to 10, are performed with the robots initially located 3m apart. When they arrive at the user, waiting for a command, they are roughly 2m apart. Thus their selection spheres (see section III.D) barely overlap. The results of the trials are recorded in Table II, showing the success of the interaction step. Leg and body detections are omitted, since they worked in every case. In every trial except #7 the human skeleton was correctly observed and the correct robot selected to collect the ball. In no case did both robots detect a reaching gesture intersecting their ego-sphere.

To test the ambiguity resolution mechanism, ten futher trials, labeled 11* to 20*, are performed with robots intially placed as close together as possible. Their ego-spheres overlapped by around 50%. In nine of these trials, both robots observed the reaching gesture vector intersecting with their sphere, so the election algorithm was used to determine which robot was selected; in each case the robot intended by the human was selected. In trial 20*, robot 1 could not aquire a skeleton track, so the other robot could not complete the election. In such cases the system could decide to elect any robot that observes the gesture instead of failing, the option we chose.

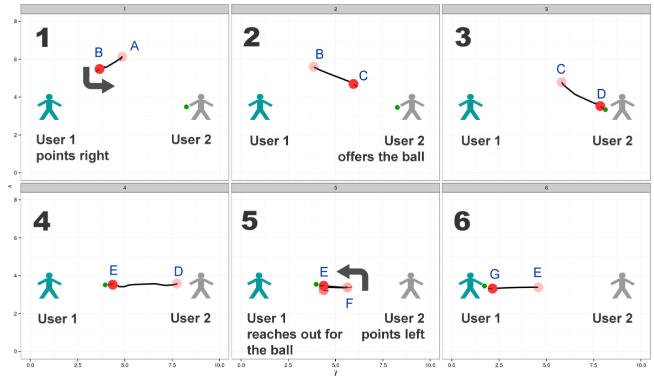


Fig. 6. Robot and human behaviour during scenario 1, trial 1, showing the interaction script performed perfectly: 1) Robot finds and approaches $user_1$. $User_1$ makes point to right gesture. 2) Robot turns right, finds and approaches $user_2$. $User_2$ makes reaching gesture. 3) Robot drives close to $user_2$ to receive the ball. 4) Robot finds and approaches $user_1$. 5) $User_1$ makes no gesture. Robot turns to find other users. Robot finds and approaches $user_2$. $User_2$ makes point to left gesture. Robot turns left, finds and approaches $user_1$. 6) $User_1$ makes reaching gesture. Robot goes closer to $user_1$ to deliver the ball.

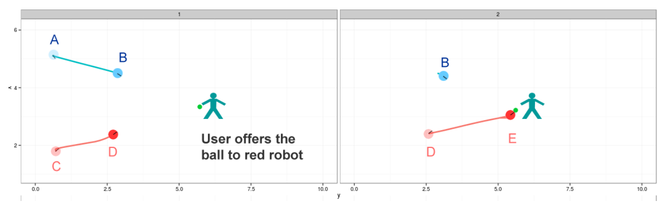


Fig. 7. Robot and human behaviour during scenario 2, trial 1: 1) Two robots find and approach user. 2) User selects red (lower) robot to receive the ball. Red robot goes closer to fetch the ball.

V. Conclusion and Future Work

We described a system whereby multiple humans and mobile robots interact robustly using a combination of sensing and signalling modalities. Extending our previous work on selecting an individual robot from a population by face-

TABLE II
RESULTS OF EXPERIMENT WITH TWO ROBOTS

Trial No.	Robot	Skeleton Detection	Line-Sphere intersection	Gesture Detection	Selection
1-6,8-10	R1	Success	Success	Success	Correct
	R2	Success	Failure	-	
7	R1	Success	Failure	-	-
	R2	Failure	-	-	
11*-19*	R1	Success	Success	Success	Correct
	R2	Success	Success	Success	
20*	R1	Failure	-	-	-
	R2	Success	Success	Success	

TABLE I

RESULTS OF EXPERIMENT WITH ONE ROBOT

Trial No.	Leg Detection	User Detection	Point to Right Gesture	Point to Left Gesture	Reaching Gesture	No Gesture
1,2,4,7,8,10	5/5	5/5	Correct	Correct	2/2	Correct
3	5/5	5/5	Correct	Correct	1/2	Correct
5	6/7	5/5	Correct	Correct	2/2	Correct
6	5/5	5/5	Correct	Correct	1/2	Correct
9	5/6	5/5	Correct	Correct	2/2	Correct
Sucess Rate	96%	100%	100%	100%	90%	100%

engagement, we showed that reaching toward a robot - a specialization of pointing - can be used to designate a particular robot for subsequent one-on-one interaction. To achieve robust operation despite frequent sensing problems, the robots use three phases of human detection and tracking, and emit audio cues to solicit interaction and guide the behaviour of the human. A series of real-world trials demonstrates the practicality of our approach.

A proper user-study with a naive participants would be required to justify a formal claim that this system is "intuitive" or better than any other method. We do not make this claim, but note informally that selecting and commanding a robot to take a ball by simply holding out the ball to the chosen robot feels fun and right, as does holding out your empty hand to the robot with the ball and having the robot come and drop it at your feet. The first few times you try it, you have to smile.

We used a very small set of discrete gestures. The gesture set could be extended to allow a user to point to *any* arbitrary place in the environment. This has been done for a single robot system (e.g. [8], [10]); however, an interesting extension would be to exploit multiple robots to jointly estimate the vector given the system's ability to simultaneously capture images of the user from multiple angles.

Finally, the audio feedback from the robot is compelling in practice. We aim to extend this from indicating only discrete robot states to continuous internal states, possibly with multiple dimensions. We would like to hear the robots whistle while they work.

ACKNOWLEDGEMENTS

This work was supported by NSERC in Canada.

REFERENCES

[1] Douglas Adams. *The Hitchhiker's Guide to the Galaxy*. Pan Macmillan, 1979.

[2] Alex Couture-Beil, Richard T. Vaughan, and Greg Mori. Selecting and commanding individual robots in a vision-based multi-robot system. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 159–166, May 2010.

[3] Antoine Deleforge and Radu Horaud. The cocktail party robot: sound source separation and localisation with an active binaural head. In *Proceedings of the Int. Conf. on Human-Robot Interaction (HRI)*, pages 431–438, March 2012.

[4] David Droeschel, Jörg Stückler, and Sven Behnke. Learning to interpret pointing gestures with a time-of-flight camera. In *Proceedings of the Int. Conf. on Human-Robot Interaction (HRI)*, pages 481–488, March 2011.

[5] Brian R. Duffy. Anthropomorphism and the social robot. *Robotics and Autonomous Systems*, 42(3–4):177–190, 2003.

[6] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[7] Jun Kato, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. Multi-touch interface for controlling multiple mobile robots. In *Proc. of the 27th Int. Conf. on Human factors in Computing Systems (Extended Abstracts)*, pages 3443–3448. ACM, 2009.

[8] David Kortenkamp, Eric Huber, R. Peter Bonasso, and Metrica Inc. Recognizing and interpreting gestures on a mobile robot. In *Proc. of the 13th National Conf. on Artificial Intelligence*, pages 915–921. AAAI Press/The MIT Press, 1996.

[9] Matthew M. Loper, Nathan P. Koenig, Sonia H. Chernova, Chris V. Jones, and Odest C. Jenkins. Mobile human-robot teaming with environmental tolerance. In *Proc. of the 4th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 157–164, March 2009.

[10] Christian Martin, Frank-Florian Steege, and Horst-Michael Gross. Estimation of pointing poses for visual instructing mobile robots under real-world conditions. *Robotics and Autonomous Systems*, 58(2):174–185, February 2010.

[11] James McLurkin, Jennifer Smith, James Frankel, David Sotkowitz, David Blau, and Brian Schmidt. Speaking swarmish: Human-Robot interface design for large swarms of autonomous mobile robots. In *Proc. of the AAAI Spring Symposium*, pages 72–75, 2006.

[12] Brian Milligan, Greg Mori, and Richard T. Vaughan. Selecting and commanding groups in a multi-robot vision based system. In *6th ACM/IEEE International Conference on Human-Robot Interaction (Video Session)*, 2011.

[13] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews*, 37(3):311–324, May 2007.

[14] Amir M. Naghsh, Jeremi Gancet, Andry Tanoto, and Chris Roast. Analysis and design of human-robot swarm interaction in firefighting. In *IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 255–260, Munich, Germany, August 2008.

[15] David Payton. Pheromone robotics. `http://www.swarm-robotics.org/SAB04/presentations/payton-review.pdf`, 2004. Slides from a presentation at the Swarm Robotics Workshop, SAB04. Retrieved September 28, 2009.

[16] Dennis Perzanowski, Alan C. Schultz, William Adams, Elaine Marsh, and Magda Bugajska. Building a multimodal human-robot interface. *Intelligent Systems, IEEE*, 16(1):16–21, Jan-Feb 2001.

[17] R. Stiefelhagen, C. Fugen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Proeedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2422–2427, Sendai, Japan, September 2004.

[18] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9:151–173, 2000.

[19] Xiao Wang, Xavier Clady, and Consuelo Granata. A human detection system for proxemics interaction. In *Proceedings of the 6th Int. Conf. on Human-Robot Interaction (HRI)*, pages 285–286, Lausanne, Switzerland, March 2011.

[20] Anqi Xu, Gregory Dudek, and Junaed Sattar. A natural gesture interface for operating robotic systems. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA '08)*, pages 3557–3563, Pasadena, California, USA, May 2008.

[21] Shengdong Zhao, Koichi Nakamura, Kentaro Ishii, and Takeo Igarashi. Magic cards: a paper tag interface for implicit robot control. In *Proc. of the 27th Int. Conf. on Human Factors in Computing Systems (CHI)*, pages 173–182. ACM, April 2009.