2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Proximity Human-Robot Interaction Using Pointing Gestures and a Wrist-mounted IMU

Boris Gromov, Gabriele Abbate, Luca M. Gambardella, Alessandro Giusti

Abstract— **We present a system for interaction between co-located humans and mobile robots, which uses pointing gestures sensed by a wrist-mounted IMU. The operator begins by pointing, for a short time, at a moving robot. The system thus simultaneously determines: that the operator wants to interact; the robot they want to interact with; and the relative pose among the two. Then, the system can reconstruct pointed locations in the robot's own reference frame, and provide real-time feedback about them so that the user can adapt to misalignments. We discuss the challenges to be solved to implement such a system and propose practical solutions, including variants for fast flying robots and slow ground robots. We report different experiments with real robots and untrained users, validating the individual components and the system as a whole.**

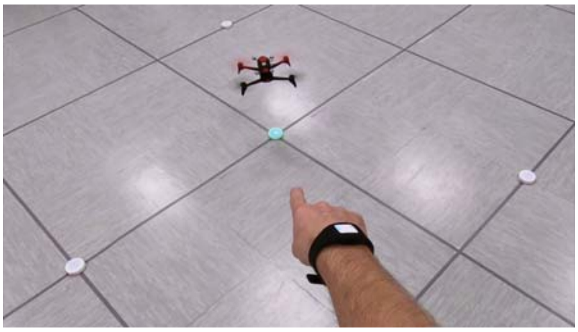

Fig. 1. First person view of the operator equipped with a wrist-mounted IMU interacting with a quadrotor.

Videos, Datasets and Code

Video, datasets and code to reproduce our results are available at: http://people.idsia.ch/~gromov/proximity-hri-pipeline

I. Introduction

We consider the scenario in which an operator wears a bracelet (such as a smartwatch) equipped with an inertial measurement unit (IMU) and needs to occasionally interact with a ground or flying mobile robot that is located in the same indoor or outdoor environment and within line of sight. More specifically, we consider the problem of communicating to the robot a location in the common environment. For example, to command the robot to move to that position, or to perform some specific action there such as: cleaning (for a robotic vacuum cleaner), picking up or placing an object (for a personal assistant robot), or landing (for a drone). We do not require the robot to be capable of detecting the operator with its onboard sensors, nor that the robot and operator are localized on the same map; however, we expect that the robot is equipped with an odometry system, i.e. it knows its recent trajectory or instantaneous position in its own reference frame, and that the robot is in continuous motion whenever it is available for interaction.

This system paper proposes a complete and practical solution, which solves the following sub-problems:

- determining when the operator wants to interact with the robot (*interaction triggering*);
- determining which robot (in case multiple robots are nearby) the operator wants to interact with (*robot identification*);
- determining where the operator is with respect to the robot (*relative localization*);
- *reconstructing* the point that the user is indicating, expressed in the robot's own coordinate frame;
- providing *real-time feedback* to the user about the reconstructed location of such point, so to allow for corrections and to improve accuracy.

The interaction involves the following two steps. *Step 1, follow the robot*: the operator points at the robot with a straight arm and follows its movement; after a few seconds, the robot provides a feedback and the bracelet vibrates; now, the system has linked the operator to the robot; *Step 2, guide the robot*: now the robot follows in real time the position that the operator indicates on the ground. Depending on the scenario, different methods for ceasing the interaction (unlinking) can be implemented, such as keeping the arm still for some time, pressing a button, or pointing at the sky.

Step 1 simultaneously solves triggering, robot identification and relative localization by extending an approach we recently proposed [1]: such approach (briefly described in Section III-B) compares the arm movements in the last few seconds with the trajectory of the robot in the same time window in order to determine the relative pose of an operator and a robot. In step 2, the robot itself provides a real-time visual feedback about the pointed locations that the system is reconstructing, such that the user has the opportunity to compensate for inaccuracies — similarly to how a laser scope improves accuracy of snipers, and seeing the mouse pointer allows one to interact with small elements on a screen.

The system is designed to solve one specific high-level

(*) Boris, Luca, and Alessandro are with the Dalle Molle Institute for Artificial Intelligence (IDSIA USI-SUPSI), Lugano, Switzerland. Email: {boris,luca,alessandrog}@idsia.ch. Gabrielle is with the University of Milano-Bicocca (UNIMIB), Milan, Italy, email: g.abbate1@campus.unimib.it

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

task: allow the user to convey a nearby location to a nearby mobile robot. This is useful in many robotics applications, such as: indicating a precise landing spot to a flying robot, or asking it to explore/map in a given direction; directing robot lawn mowers [2] or vacuum cleaners [3] to work on specific areas, instructing them to avoid some location where they might get stuck, or indicating them a position where to park; indicating to personal assistance robots that carry objects, such as Gita [4], to park close to a loading/unloading area.

The system relies on a fast robot (such as a quadrotor) both in step 1 (to generate a trajectory that the user has to follow for a few seconds with their arm) and in step 2 (to provide real-time visual feedback about the reconstructed pointed location). Section IV describes a variant suitable for slow robots, which uses a projected laser dot.

The **main contribution** of this system paper is the design (Section III), implementation (Section IV) and experimental validation (Section V) of a self-contained pointing-based proximity interaction technique. The system relies on an approach [1] we recently proposed for estimating the relative pose of a user and a mobile robot; in this paper, this approach is used as a component and extended to also solve the interaction triggering task. Moreover, our system implements the idea of using the robot position itself to provide visual feedback about the reconstructed pointed location, which we recently demonstrated in a video abstract [5].

II. Related Work

Pointing gestures are an innate [6] and effective device that humans use all the time. They allow a person to intuitively and efficiently communicate locations and other spatial notions (trajectories, directions). Research in robotics has used pointing gestures for disparate tasks: pick-and-place [7, 8, 9], object and area labeling [10], teaching by demonstration [11], point-to-goal [12, 13, 14], selection of a robot within a group [15, 16], and assessment of the joint attention [17]. Other works use non-pointing gestures for interacting with co-located robots, e.g. for quadrotor control [18, 19].

One important issue to be solved is the *perception of the operator's gestures*. Perception can be the responsibility of a robot (or of a group of cooperatively-sensing robots [20, 16]); of the environment [21]; or, as in our case, of a device worn by the user [11, 12, 22]. The first approach is the most popular in human-robot interaction (HRI) research. On one hand, it is natural because it mimics what humans do when communicating with each other—the recipient of the gesture is the one who perceives it. On the other hand, solving the perception problem is challenging and requires the robot to continuously sense the user. Relying on sensors placed in the environment relaxes the requirements on the robots, but limits the applicability of the system to properly instrumented areas; in both cases, the positions being pointed at need to be inferred by external observation, which is typically performed with cameras or RGB-D sensors.

Estimation of the pointed location. Regardless of the specific approach adopted for sensing, assuming a perfect knowledge of the user's posture, one has to solve the problem of interpreting such a posture and mapping it to the point in the environment that the user wants to indicate; this is typically solved in two steps: first, identify a direction (i.e. a ray in 3D space); then, relate such ray with the environment to get a point or object. In human-computer interaction (HCI) and HRI research we can identify two classes of methods for estimating pointing rays: *head-rooted* and *arm-rooted*. The head-rooted techniques consider a pointing ray that originates somewhere within the head: from a dominant eye ([23]), cyclops eye ([24]), or head centroid ([25, 26]). Arm-rooted methods assume the ray originates from a point laying on the pointing arm: at shoulder, elbow ([25, 7, 27, 28]), wrist ([8]) or index-finger ([27]). The pointing direction is then defined by the second point, that lies on the arm. For the head-rooted methods this point is either a centroid of the hand or a tip of the index finger, while for the arm-rooted techniques the second point can be at elbow, wrist / hand, or at the tip of the index finger. In HCI these methods are often referred to as *ray casting* or *ray pointing* techniques. The most popular models in robotics are [7, 9, 25, 27]: *head-finger*, *upper arm*, and *forearm*. In this work, we employ the head-finger model.

One important advantage of using pointing gestures to convey locations (as opposed to using e.g. a joystick or a map on a handheld device) is that the operator interacts in its own reference frame, and does not have to account for the transformation linking the device to the surrounding environment (mental rotation problem). However, to achieve this goal, the *relative localization* of the robot with respect to the user needs to be known. Many solutions to this problem are adopted in robotics, including fiducial markers [29], GPS [30], optical motion capture systems or ultra-wideband (UWB) localization systems [31, 32]. In this paper we use the pointing gesture itself to fix the relative localization, implementing [1].

Our system uses *real-time feedback* to convey the current state (e.g. when a quadrotor is linked it performs a small "jump"), and most importantly to visualize in real time the current reconstructed location, to allow the user the opportunity to correct and fine-tune it in a closed-loop way. We convey the location either by using the robot itself (if it is fast enough), or, in case of slow robots, by controlling a laser pointer to shine on the reconstructed location. The toy robot StartWars BB-8 uses the light on its light-emitting diode (LED)-ring as a crucial feedback during the adjustment of the relative heading between the robot and the wrist controller ForceBand worn by the user. Lights are also a popular choice in HRI research. Szafir et al. used a circular strip of LEDs on the circumference of the drone to study how to communicate motion intentions to the user [33]. Monajjemi et al. uses an RGB LED strip on the front side of the drone to communicate its intent to the user [34]. Using the motion of the robot itself as feedback is also explored by Cauchard et al., who studied the ways a drone can communicate its state through emotions that are encoded by changing drone's speed, altitude, and orientation [35].

III. Model

A. Definitions

We define a reference frame of the operator $\{H\}$, which is located at the operator feet with the x-axis pointing forward, y-axis to the left, and z-axis pointing up; we also define a reference frame of the robot $\{R\}$ (to simplify the notation, we initially consider a single robot), as the robot's odometry frame, with the z-axis pointing up; the origin of $\{R\}$ with respect to $\{H\}$ and its rotation around the z-axis are unknown.

We define a *pointing ray* r as a 3D half-line on which the point that the human intends to indicate lies; we adopt a simplified version of the *head-finger* model (eye-finger ray cast (EFRC) method in [24]): we define r as the half-line originating at the operator's dominant eye and passing through the tip of the pointing finger; we further assume that both the eye and the shoulder lie on the z-axis of $\{H\}$. Under these assumptions, r can be reconstructed in frame of reference $\{H\}$ using the data sensed by the wrist-mounted IMU, and some operator body measurements, namely: shoulder height, shoulder-finger length, shoulder-eye distance.

In step 1, in order to take control of a robot the operator points at it and follows its movement with a pointing gesture for a short period of time. Our system needs to detect when this occurs (triggering), which robot the user is pointing at (robot identification) and where the user is with respect to the robot (relative localization).

B. Relative Localization

We first explain how, relying on the approach proposed in our previous work [1], we determine the transformation T between $\{R\}$ and $\{H\}$ assuming that during a time period τ the operator was in fact following a given robot; i.e. we assume triggering and robot identification are known. In Section III-C, we extend the approach to the general case.

Let $\mathcal{R}$ be a set of N pointing rays $r_i^{\{H\}}$ obtained during τ, defined in the frame of reference of the operator $\{H\}$; for each $r_i^{\{H\}}$ we consider the corresponding robot position $P_i^{\{R\}}$ defined in the frame of reference of the robot $\{R\}$, and thus define a set of pairs $\mathcal{C}$:

$$\mathcal{C} = \{\left(r_1^{\{H\}}, P_1^{\{R\}}\right), \dots, \left(r_N^{\{H\}}, P_N^{\{R\}}\right)\},$$

We expect that the points $P_i^{\{R\}}$ lay close to their corresponding rays $r_i^{\{H\}}$. For a given estimate T of the transformation, we can convert the robot positions $P_i^{\{R\}}$ defined in the robot frame into the operator frame, i.e. $P_i^{\{H\}} = TP_i^{\{R\}}$. Using these points we define a new ray $q_i^{\{H\}}$ that shares the origin with the ray $r_i^{\{H\}}$, but passes through the point $P_i^{\{H\}}$.

Now, we can define the error function θ for a set of pairs $\mathcal{C}$:

$$\theta\left(T, \mathcal{C}\right) = \frac{1}{N}\sum_{i=1}^{N} \angle(r_i^{\{H\}}, q_i^{\{H\}}) \tag{1}$$

where $\angle(\cdots) \in [0;\pi]$ represents the unsigned angle between the directions of two rays. The error function $\theta\left(T, \mathcal{C}\right)$ is therefore 0 iff all points lie on the respective ray, and > 0 otherwise.

We search for the coordinate frame transformation T^* between the operator frame $\{H\}$ and the robot frame $\{R\}$ that minimizes the error function, i.e. that minimizes the average unsigned angle between all the pairs of vectors $r_i^{\{H\}}$ and $q_i^{\{H\}}$: $T^* = \arg\min_T \theta(T, \mathcal{C})$.

The residual error $\theta^* = \min \theta(T^*, \mathcal{C})$ indicates how well the transformed robot positions fit the corresponding rays.

C. Triggering

In case the operator was not following the robot during τ, we can expect that the optimization procedure results in a large residual error; in fact, the user's motion will not be compatible with the robot's trajectory for any value of T. In order to trigger the interaction, we repeat the optimization procedure at regular intervals by considering a fixed-length time window that ends at the current time; each optimization run results in an estimated value for T^* and the corresponding residual error θ^*. We trigger the interaction when θ^* is lower than a threshold.

D. Robot Identification and Multiple Operators

In case multiple robots and/or multiple operators are in the scene, the optimization runs once for each operator-robot pair; we trigger an interaction between a pair as soon as the resulting θ^* is lower than the threshold Θ_{id}. Note that in order to allow for unambiguous robot identification, each robot follows a different trajectory.

E. Pointed location reconstruction and real-time feedback

Once the interaction is triggered between an operator and a robot, the relative pose T^* of the robot frame $\{R\}$ with respect to the human frame $\{H\}$ is known. Then, at each timestep we can transform the pointing ray (which is obtained in the operator's frame $\{H\}$) to the robot's frame $\{R\}$ as $r^{\{R\}} = (T^*)^{-1} r^{\{H\}}$. The robot can use this information to provide real-time feedback about the reconstructed location. In particular, by intersecting $r^{\{R\}}$ with the ground plane the robot identifies a point on the ground, and provides feedback by moving over such point.

IV. Implementation

A. Gesture sensing

We implemented the system using an inexpensive wearable IMU (Mbientlab MetaWearR+ [36]) that has a formfactor of a wrist smartwatch. The device is equipped with a 3-DoF accelerometer, 3-DoF gyroscope, and 3-DoF magnetometer. The onboard firmware runs the necessary sensor fusion algorithms in real time and outputs an accurate estimation of the device's absolute 3D-orientation in an arbitrary fixed reference frame whose z-axis points up; The device also features a micro switch button, an LED light, and a vibration motor. The data is streamed to the host PC with approx. 50 Hz rate via Bluetooth Low-Energy (Bluetooth 4.0) link.

The acquired orientation is then used within the head-finger pointing model (described in Section III) to recover $r^{\{H\}}$, which is then intersected with the ground plane to reconstruct the pointed-to location on the ground, expressed in reference frame $\{H\}$. Once the relative localization T^* between $\{H\}$ and $\{R\}$ is known, such point can be expressed with respect to $\{R\}$ and the robot can provide visual feedback.

The simplified approach used to reconstruct the pointing ray, and inaccuracies in the estimation of T^*, cause errors in the reconstructed point: in practice, these errors are adjusted for by the operator as long as we provide real-time visual feedback.

B. Flying arena and quadrotor control

Experiments take place in a room with a safety net, outfitted with a commercial optical motion capture system (12 Optitrack PRIME17-W cameras). The Optitrack data is streamed to the Robot Operating System (ROS) with 30 Hz rate. We use a lightweight quadrotor (Parrot Bebop 2) tracked through a rigid-body marker to perform closed-loop velocity and position control. We also track the location of the user's head through a rigid-body marker attached to a hat.

Note that knowledge of the absolute locations of the operator and robot is used as ground truth for experiments, but is not required by our system.

C. Variants for slow robots

The approach relies on a fast robot to generate a trajectory for the operator to follow (triggering, relative localization, robot identification) and to provide real-time feedback of the reconstructed position. In order to adapt the approach to slow robots, we also implement a variant in which a laser dot, whose position on the ground plane is precisely controlled in $\{R\}$ by a robot-mounted laser turret (ScorpionX MX-64 Robot Turret, InterbotiX Labs), takes the role of the robot during the whole interaction.

While available for interaction, the robot shines the laser on the floor in its vicinity (Figure 2), continuously tracing some pattern (such as a circle or 8-shape). To take control, the operator should follow the laser dot that the robot is projecting instead of at the robot itself. For triggering, relative localization, and robot identification, the position of the laser dot takes the role of $P^{\{R\}}$ for the specific robot. During real-time feedback, the laser is projected on the reconstructed point that the user is indicating.

V. Experiments

We describe several experiments aimed at validating individual components of the system, as well as the system as a whole.

One key component is relative localization; Gromov et al. [1] report extensive experiments validating this component independently, demonstrating its accuracy and robustness, also for robot identification; therefore, in this paper we focus on validating other functionalities. Previously we also shown that pointing compares favourably with a conventional joystick interface in the precise landing task [5]. In Section V-A, we assume that relative localization is obtained exactly and focus instead on measuring the impact of real-time feedback on the reconstructed pointing location.

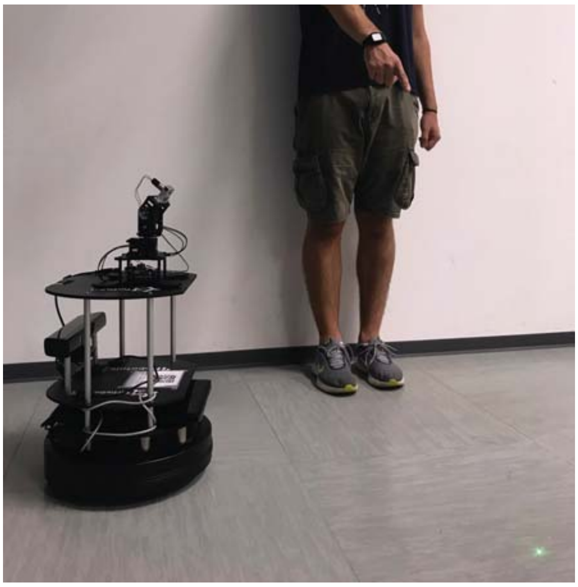

Fig. 2. Interaction using pointing gestures with a slow-moving robot equipped with a laser turret for a real-time visual feedback.

Finally, in Section V-B we test the system as a whole, including its triggering, relative localization, and real-time feedback components, in a task where a quadrotor is flown through several waypoints; in this task, we compare the performance of the proposed pointing-based interface on operators that never used it, to their performance as the skilled pilots using a joystick.

In all the experiments we collect the data with a standard ROS tool `rosbag` and analyze it offline.

A. Impact of real-time feedback on pointing accuracy

1) Setup: In order to evaluate the impact of visual feedback on the accuracy of pointed locations, we implemented an experiment using the laser turret mounted on a fixed platform, that would provide the feedback accurately. Ten people volunteered to participate in this study. Each participant wears the IMU-equipped bracelet, and has their body measurements taken to set up the parameters of the system.

Three targets are laid out at known positions as shown in Figure 3: with the user standing at a known relative position to the turret. The user's heading is fixed with respect to the turret.

The experiment proceeds using the following sequence, which is advanced by audio prompts played at known times: 1) the user is asked to relax their pointing arm and wait for an audio signal; 2) after 5 s the system plays a beep and the user is asked to point at the first target and hold their arm still; during this time, no feedback is given; 3) after another 5 s and a beep, the user can relax the arm and wait for another

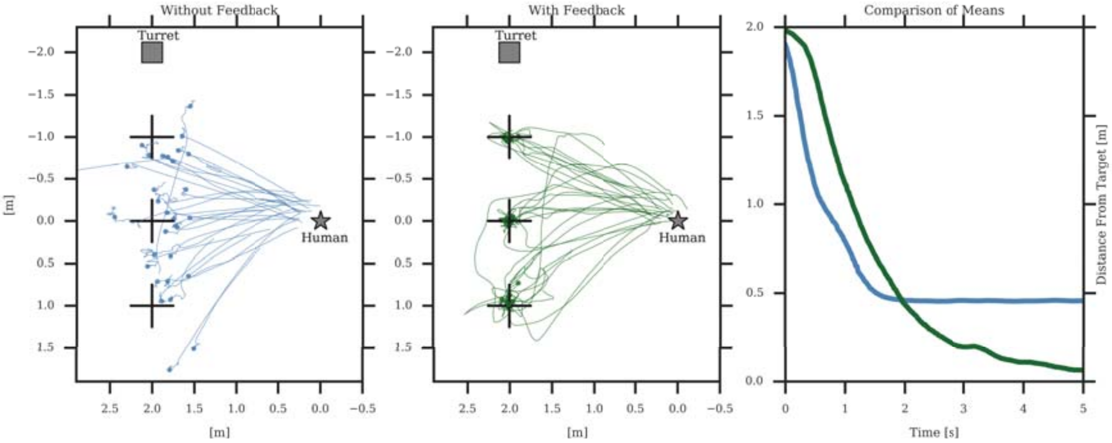

Fig. 3. Effects of real-time visual feedback on pointing accuracy.

command; 4) after next 5 s, the user is asked to point to the same target again; now, however, the laser provides real-time feedback by shining at the pointed location. The procedure is then repeated for another two targets.

2) Results: We recorded and analyzed the pointing locations and the timings of the audio prompts for each participant. The collated trajectories of the pointed locations and the evolution of the distance to target are visualized in Figure 3. The data shows that, without feedback, users quickly reach an average distance from the target of 0.5 m but do not improve any further; this is expected as the system has intrinsic inaccuracies (for example in the reconstruction of pointing rays $r^{\{H\}}$) which the user is unable to see and correct. When the feedback is provided, distance decreases to almost 0 within 5 seconds.

This demonstrates that real-time feedback (provided with a laser or with the robot's own position) is a key component in our approach.

B. System validation

1) Setup: It this experiment (Figure 4) we validate the entire pipeline, considering one quadrotor and one operator. We recruited 5 participants whose goal was to fly the quadrotor (Parrot Bebop 2) over a set of five flat stationary wireless LED beacons (in-house hardware based on Adafruit nRF52 Feather Bluetooth LE board with a ring of 24 RGB NeoPixel LEDs) placed on the floor at known locations. All operators are proficient or expert joystick users, and had little or no previous experience with the proposed pointing interface. No ”dry run” was performed for familiarizing the user with either interface, or the task. The pointing model is initialized with the rough measurements of each user (height of shoulder, height of eyes, shoulder-finger length).

For each operator we recorded two runs: the first using the pointing interface (wearing the IMU-equipped bracelet on the wrist of their dominant arm); the second using joystick control (Logitech F710).

The procedure with the pointing interface proceeds as follows. 1) The operator enters the flying arena, where the quadrotor is performing an abstract autonomous task (flying on a closed trajectory), and stands at a location at their discretion. 2) When they decide to, the operator starts pointing and follows the moving quadrotor. 3) After several seconds, the system triggers the interaction and determines the relative localization: the robot stops and makes a small “jump”; at the same time the bracelet on subject's wrist vibrates to indicate that the robot is now linked to them. 4) One of the targets on the floor lights up (blue). 5) The operator directs the robot to the target; once within the confirmation zone (20 cm from target's center), the target turns yellow and a timer is triggered. 6) To clear the target the subject must keep the robot within the confirmation zone for 2 seconds. Then, the target shortly turns green and switches off; if the robot leaves the confirmation zone before the two seconds expire, the target turns blue and the timer is reset.

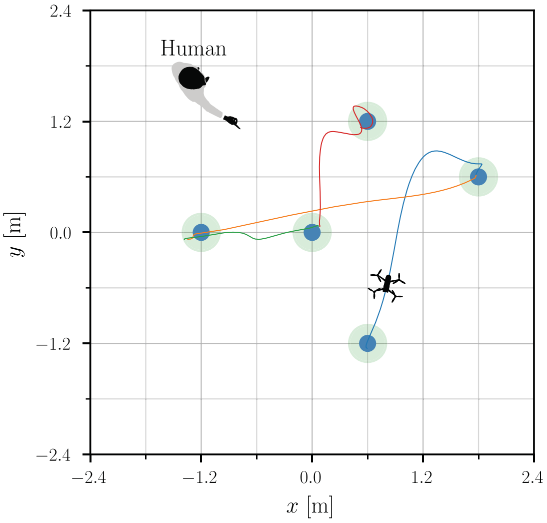

Fig. 4. The map of the experimental environment with targets (*blue circles*), their confirmations zones (*green circles*), the actual quadrotor trajectory in one of the pointing experiments (*colored curves*), and human ground truth position. The silhouettes of the quadrotor and user are depicted in the appropriate scale.

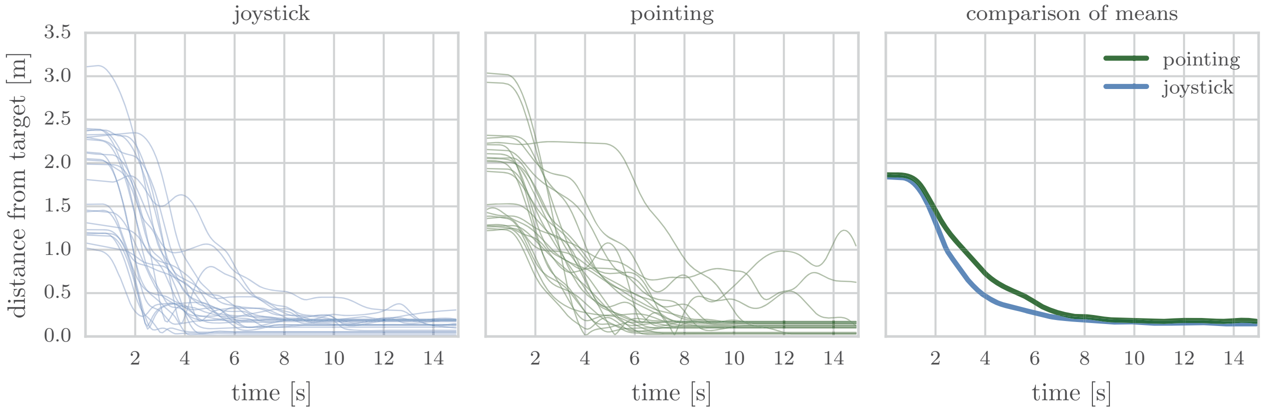

Fig. 5. Waypoints experiment: Analysis of the evolution in time of the distance to the target, for each trajectory flown; each trajectory is represented as a line; $t = 0$ on the plot corresponds to the t_0 time of each trajectory. Left: joystick interface ($N = 20$). Center: pointing interface ($N = 20$). Right: average over all trajectories for each interface.

7) Once a target is cleared, the next target lights up in blue: steps 5, 6 are repeated. 8) Once all targets are cleared, the robot is unlinked and the interaction ends. The procedure for experiments with the joypad is similar, but omits steps 3 and 4 of the above sequence (relative localization), which are specific to the pointing interface.

To assess the performance of the system, we collected the ground truth positions of the participants and the quadrotor, and the times of state transition events of the targets. In our analysis we ignore the first segment of the robot movements, i.e. from the moment the operator takes control till the moment the *first* target is cleared. This yields four segments per run per operator, i.e. a total of 20 segments for each modality (pointing, joystick).

2) Results: We report the average time-to-target metric for both interfaces. It can be seen from the plots (Fig. 5), that the performance of the pointing interface used by inexperienced users is very similar to their performance with the joypad controller.

VI. Conclusions

We designed and implemented a system for interaction between co-located humans and mobile robots, based on pointing gestures sensed by a wrist-mounted IMU. The approach relies on the results in [1] for solving relative localization and robot identification, and extends the approach to triggering; then, the system implements real-time feedback. We report extensive experimental results with untrained users, demonstrating the viability of the approach.

Acknowledgments

This work was partially supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics.

References

[1] B. Gromov, L. Gambardella, and A. Giusti, “Robot identification and localization with pointing gestures,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, to appear.

[2] Robotic lawn mowers. Husqvarna official webpage. https://www.husqvarna.com/us/products/robotic-lawn-mowers/. [Online; accessed: 2018-09-13].

[3] Roomba robot vacuum. I-Robot official web-page. https://www.irobot.com/for-the-home/vacuuming/roomba.aspx. [Online; accessed: 2018-09-13].

[4] IEEE Spectrum, “Piaggio’s Cargo Robot Uses Visual SLAM to Follow You Anywhere,” http://spectrum.ieee.org/automaton/robotics/home-robots/piaggio-cargo-robot, Feb 2017, [Online; accessed: 2018-09-13].

[5] B. Gromov, L. Gambardella, and A. Giusti, “Video: Landing a drone with pointing gestures,” in *HRI ’18 Companion: 2018 ACM/IEEE International Conference on Human-Robot Interaction Companion, March 5–8, 2018, Chicago, IL, USA*. ACM, Mar. 2018.

[6] G. Butterworth, *Pointing: Where language, culture, and cognition meet*. Manwah, NJ: Lawrence Erlbaum Associates, 2003, ch. Pointing is the royal road to language for babies, pp. 9–33.

[7] D. Droeschel, J. Stückler, and S. Behnke, “Learning to interpret pointing gestures with a time-of-flight camera,” *Proceedings of the 6th international conference on Human-robot interaction - HRI ’11*, pp. 481–488, 2011.

[8] B. Großmann, M. R. Pedersen, J. Klonovs, D. Herzog, L. Nalpantidis, and V. Krüger, “Communicating Unknown Objects to Robots through Pointing Gestures,” in *Advances in Autonomous Robotic Systems 15th Annual Conference, TAROS 2014*. Birmingham: Springer, 2014, pp. 209–220.

[9] A. Cosgun, A. J. B. Trevor, and H. I. Christensen, “Did you Mean this Object?: Detecting Ambiguity in Pointing Gesture Targets,” in *HRI’15 Towards a Framework for Joint Action Workshop*, 2015.

[10] A. J. B. Trevor, J. G. Rogers, A. Cosgun, and H. I. Christensen, "Interactive object modeling & labeling for service robots," *ACM/IEEE International Conference on Human-Robot Interaction*, p. 421, 2013.

[11] J. Sugiyama and J. Miura, "A wearable visuo-inertial interface for humanoid robot control," in *ACM/IEEE International Conference on Human-Robot Interaction*. IEEE, mar 2013, pp. 235–236.

[12] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the JPL BioSleeve," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1160–1165, 2013.

[13] A. Jevtić, G. Doisy, Y. Parmet, and Y. Edan, "Comparison of Interaction Modalities for Mobile Indoor Robot Guidance: Direct Physical Interaction, Person Following, and Pointing Control," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 6, pp. 653–663, 2015.

[14] M. Tölgyessy, M. Dekan, F. Duchoň, J. Rodina, P. Hubinský, and L. Chovanec, "Foundations of Visual Linear Human-Robot Interaction via Pointing Gesture Navigation," *International Journal of Social Robotics*, vol. 9, no. 4, pp. 509–523, 2017.

[15] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 3834–3841.

[16] S. Pourmehr, V. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "A robust integrated system for selecting and commanding multiple mobile robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2874–2879, 2013.

[17] A. G. Brooks and C. Breazeal, "Working with Robots and Objects: Revisiting Deictic Reference for Achieving Spatial Common Ground," *Gesture*, pp. 297–304, 2006.

[18] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pp. 143–149, 2011.

[19] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A Kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, no. 3, pp. 179–186, 2013.

[20] A. Giusti, J. Nagi, L. Gambardella, and G. A. Di Caro, "Cooperative sensing and recognition by a swarm of mobile robots," *IEEE International Conference on Intelligent Robots and Systems*, pp. 551–558, 2012.

[21] Z. Zivkovic, V. Kliger, R. Kleihorst, A. Danilin, B. Schueler, G. Arturi, C.-C. Chang, and H. Aghajan, "Toward low latency gesture control using smart camera network," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.

[22] B. Gromov, L. M. Gambardella, and G. A. Di Caro, "Wearable multi-modal interface for human multi-robot interaction," *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 240–245, 2016.

[23] K. Plaumann, M. Weing, C. Winkler, M. Müller, and E. Rukzio, "Towards accurate cursorless pointing: the effects of ocular dominance and handedness," *Personal and Ubiquitous Computing*, pp. 1–14, 2017.

[24] S. Mayer, V. Schwind, R. Schweigert, and N. Henze, "The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[25] K. Nickel and R. Stiefelhagen, "Pointing Gesture Recognition based on 3D-Tracking of Face , Hands and Head Orientation Categories and Subject Descriptors," *Proceedings of the 5th international conference on Multimodal interfaces*, pp. 140–146, 2003.

[26] S. Ueno, S. Naito, and T. Chen, "An efficient method for human pointing estimation for robot interaction," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, oct 2014, pp. 1545–1549.

[27] S. Mayer, K. Wolf, S. Schneegass, and N. Henze, "Modeling Distant Pointing for Compensating Systematic Displacements," in *Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems*, vol. 1, 2015, pp. 4165–4168.

[28] K. Kondo, G. Mizuno, and Y. Nakamura, "Analysis of Human Pointing Behavior in Vision-based Pointing Interface System - difference of two typical pointing styles -," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 367–372, 2016.

[29] M. Fiala, "ARTag, a fiducial marker system using digital techniques," *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. II, pp. 590–596, 2005.

[30] G. Dudek and M. Jenkin, *Inertial Sensors, GPS, and Odometry*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 477–490.

[31] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-Level Localization with a Single WiFi Access Point," in *NSDI 2016*, 2016, pp. 165–178.

[32] B. Hepp and N. Tobias, "Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 189–194, oct 2016.

[33] D. Szafir, B. Mutlu, and T. Fong, "Communicating Directionality in Flying Robots," *Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2, no. 1, pp. 19–26, 2015.

[34] M. Monajjemi, S. Mohaimenianpour, and R. Vaughan, "UAV, come to me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. Figure 1. IEEE, oct 2016, pp. 4410–4417.

[35] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A.

Landay, “Emotion encoding in human-drone interaction,” *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2016-April, pp. 263–270, 2016.

[36] Wearable technology for healthcare. Mbientlab official web-page. https://mbientlab.com/. [Online; accessed: 2018-09-13].